SIMULATION OF HYDROPOWER GENERATION FOR THE CITARUM MULTI-RESERVOIR SYSTEM USING SYNTHETIC FLOWS

AGUNG W. H. SOEHARNO

SIMULATION OF HYDROPOWER GENERATION FOR THE CITARUM MULTI-RESERVOIR SYSTEM USING SYNTHETIC FLOWS

by

©Agung W.H. SOEHARNO B. Eng., M.S.

A thesis submitted to the School of Graduate
Studies in partial fulfilment of the
requirements for the degree of
Master of Engineering

Faculty of Engineering and Applied Science
Memorial University of Newfoundland
December, 1992

St. John's Newfoundland Canada

ISBN 0-315-82620-7

To

My children:

Retno Utami,

Fajar Budi Prasetyo,

Arif Wicaksono,

and

My wife:

Achadiyani Agung, M.D.,

Bandung, Indonesia.

Abstract

The demand for electricity in Java and Bali, Indonesia increases an average of 14% per year. This condition forces the electrical authority in Indonesia to consider various generation alternatives, including hydroelectric power, to meet the growing demand. One alternative is improving reservoir operations of existing hydro power systems so that improved production can be achieved.

One such hydropower system considered in this thesis is the series of three hydroelectric stations, Saguling, Cirata and Jatiluhur, on the Citarum river, West Java. At present these three hydropower plants are operated by different authorities. They developed the current reservoir operation guideline (CROG) based on consensus. An average of assumed "normal" years of historical flows are used to obtain the current rule curve. The estimation of annual energy generated is then obtained based on the series of twelve monthly flows of the average of the assumed "normal" year flows.

In this study a simulation of reservoir operations was performed using synthetic inflows into the reservoirs. The synthetic flows were generated by a stochastic model that on average preserved the statistical characteristics of both historical annual and monthly flows. An auto-regressive moving average model was used to model the annual flows, and the Method of Fragments and the Two-Tier method were examined to model the monthly flows.

Two rule curves were applied in the simulation of reservoir operations. These were the CROG rule curve and the water resources development guideline (WRDG) rule curve proposed by the Water Resources Development project. Two other limiting rule curves were also examined, one assuming that the reservoirs are kept empty, and the other assuming that the reservoirs are kept full. The operation of the system was simulated using both historical and synthetic inflow sequences, as well as the different rule curves, to evaluate the performance of the system.

The results showed that the preferred stochastic model for the annual flows was an ARMA(2,0) model, and for the monthly flows, the Two-Tier model. The CROG rule curve was more suitable for the system than that based on the WRDG rule curve. From the reservoir operation simulation it was found that on average the available water in the Citarum River was smaller than the installed capacity of the Saguling, Cirata and Jatiluhur plants. The results of the simulation using synthetic flows could give more comprehensive analysis than using only historical flows.

Acknowledgements

I gratefully acknowledge the Government of Canada through CIDA and the Government of Indonesia through Ministry of Public Work (DPU) for providing financial assistance for this Master of Engineering Program. I also wish to thank Prof. Caspar Booy of the University of Manitoba who has provided his efforts for the success of the program.

I also express gratitude to Memorial University of Newfoundland, all members and staff of the Faculty of Engineering and Applied Science and especially to my supervisors. I express my deep sense of gratitude to Dr. Leonard M. Lye for providing valuable guidance, assistance, encouragement and willingness to allow me to pursue my own ideas, and for showing me the useful technique of stochastic hydrology. My special thanks also go to Ms. Susan Richter for providing guidance and for showing me the techniques of reservoir operations. I also wish to thank Dr. David Hansen and Dr. Gary Sabin for providing guidance in computing work.

I am grateful to all friends, colleagues, relatives, and whoever have provided support and assistance for the success of this program. To my wife Achadiyani Agung, and my children Tami, Fajar and Arif, I express my warm appreciation for their patience, understanding and support. Last but not least, I thank God who gave me time and space for life, independency and freedom of thinking, love and peace of life.

Contents

List of Tables

List of Figures

List of Symbols

Symbols	Descriptions
a_i, b_i	- constants for equation 2-1
$A_{i(t)}$	- reservoir surface area at the beginning of month t
$A_{i(t+1)}$	- reservoir surface area at the end of month t
ACC	- Area Control Centre
ACF	- auto-correlation function
AIC	- Akaike Information Criterion
ARMA	- auto-regressive moving average model
$b_{(j)}$	- least square regression coefficient for estimating the $(j+1)^{th}$ flow from the j^{th} flow
c_i, d_i, e_i	- constants for equation 2-2
$c_{ei(t)}$	- coefficient of evaporation of month t
CRCG	- current reservoir operating guideline
C_s	- coefficient of skewness
CTR-SIMOD	- Citarum Simulation Model
CTR-STOCHMOD	- Citarum Stochastic Model
D	- dry
η	- plant efficiency
$e_{i(t)}$	- evaporation coefficients of month t
$E_{i(t)}$	- evaporation from reservoir i during month t
g	- acceleration due to gravity (9.81 m/s^2)
GWh	- Giga Watt hour
$H_{i(t)}$	- head at the beginning of month t

FSL	- full supply level
i	- node
$I_{i(t)}$	- inflow to reservoir i during month t
IRIS	- Interactive River System Simulation
ISL	- intermediate supply level
JCC	- Java Control Centre
K	- Hurst's coefficient
LSL	- low supply level
MLE	- maximum likelihood method
MS	- mean square error
N	- normal
PACF	- partial auto-correlation function
$P_{i(t)}$	- energy generations during month t
PLN	- Perusahaan Listrik Negara (the state electric corporation)
$P_{Max,i}$	- plant installed capacity
PPCC	- Probability Plot Correlation Coefficient
$Q_{Spi(t)}$	- spill flow from reservoir i during month t
$Q_{Tbi(t)}$	- turbine flow of hydropower plant during month t
$Q_{TbMax,i(t)}$	- maximum turbine flow
$Q_{WAi(t)}$	- available water of month t
$r_{(j)}$	- serial correlation coefficient of month j
$r_{k(x)}$	- auto-correlation coefficient of lag-k
$s_{(j+1)}$	- standard deviation of historical flows of month j+1
VD	- very dry
$V_{i(t)}$	- volume of reservoir i at the beginning of month t

$V_{i(t+1)}$	- volume of reservoir i at the end of month t
$V_{Ti(t+1)}$	- target volume for reservoir i at the end of month t
VW	- very wet
W	- wet
WRD	- Water Resources Development
WRDG	- Water Resources Development Guideline
$x_{(j)}$	- historical monthly flow
$x_{t(j)}$	- transformed monthly flow
X	- historical annual flow
$y_{(i)}$	- generated monthly flow
$Y_{(t)}$	- generated annual flow in year t
z(5%)	- 5% significant level of z
$z_{(t)}$	- normal random variate with mean zero and variance unity
$\overline{x}_{(j)}$	- mean of historical flows of month j
ϵ_t	- independent stochastic component
λ	- transformation parameter
μ	- mean
ρ	- water density (1000 kg/m^3)
σ	- standard deviation

Chapter 1

Introduction

Water: too much, too little, too dirty. These are some of the problems of management and planning in water resources development (Loucks *et al.*, 1981). The Indonesian government has realized that good water resources planning and management is one of the key factors in obtaining optimal benefits from the available water resources in the country. Indonesia consists of five large islands and hundreds of small islands. The large islands are Java, Sumatra, Kalimantan, Sulawesi and Irian Jaya (see Figure 1-1). One of the water resources projects in West Java is the Cisadane-Cimanuk Integrated Water Resources Development (WRD) Project (DGWRD, 1988), described below.

1.1 General Description

The development of the hydro electric power system on the Citarum River is part of the WRD project, under the Directorate General Water Resources Development (DGWRD) of the Indonesian Government in West Java, Indonesia. The program is divided into six areas (see Figure 1-2), namely:

Area 1: Northern Banten,

Area 2: CJB (Cianjur-Jakarta-Bogor),

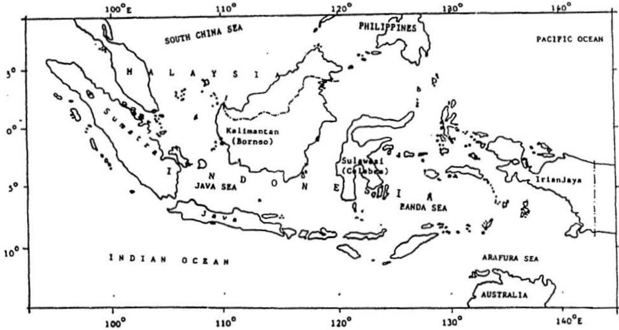


Figure 1-1: Map of Indonesia

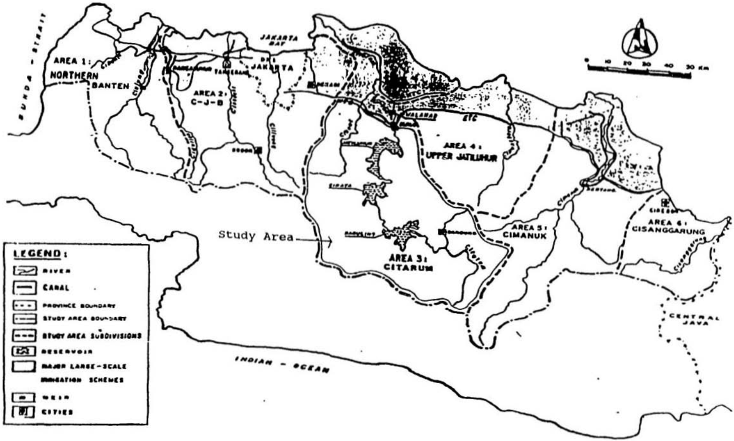


Figure 1-2: Location of study area.

Area 3: Citarum,

Area 4: Upper Jatiluhur,

Area 5: Cimanuk,

Area 6: Cisanggarung.

This study concentrates on the Citarum Area. The main river in this area is the Citarum River which is located about 50 km from Bandung and 100 km from Jakarta. There are three dams on the Citarum River:

the Saguling Hydroelectric Power Dam,

the Cirata Hydroelectric Power Dam, and

the Jatiluhur Multipurpose Dam.

These three hydropower installations are connected to the Java and Bali Electric Power system.

The demand for electricity in Java and Bali from 1983/1984 to 1988/1989 has increased, on average, at the rate of 14% per year (Sihombing *et al.*, 1990). This demand has increased rapidly because electricity is one of the essentials for the rapid industrial development that has also occurred in this decade. If this rate of increased demand remains constant, the power demand 7 years from now will be more than twice of the demand this year. The energy production, the number of generators and the installed capacity in the Java and Bali System is shown in Tables 1-1, 1-2, and 1-3, respectively. These tables show that the power system is predominantly thermal. The hydroelectric energy generation might be always firm under the load curve. Therefore the comparisons of rule

Table 1-1: Energy production of the Java and Bali Power System (Sihombing *et al.*, 1990)

Power Plants	1988/89 Energy (GWh)	1989/90 Energy (Gwh)	1990/91 Energy (Gwh)
Hydro	3958	5304	4786
Thermal (Oil)	7372	6181	8170
Thermal (Coal)	6283	9215	10024
Diesel	85	81	165
Gas Turbine	370	158	373
Geothermal	1013	1007	980
Sub-total	19081	21946	24498
Supplied (bought) from:			
Jatiluhur	654	795	860
Brantas	605	748	717
Krakatau Diesel	15	15	0
T o t a l	20354	23504	26075

Table 1-2: Number of units of generator in the Java and Bali Power System (Sihombing *et al.*, 1990)

Power Plant	Area 1	Area 2	Area 3	Area 4	Total
Hydro	6	32	19	23	80
Thermal (Coal)	4	-	-	-	4
Thermal (Oil)	9	-	3	8	20
Geothermal	-	3	-	-	3
Gas Turbine	13	4	4	5	26
Diesel	4	-	-	11	15
Total	36	39	26	47	148

Table 1-3: Installed capacity of generating plants (MW) in the Java and Bali Power System (Sihombing *et al.*, 1990)

Power Plant	Area 1	Area 2	Area 3	Area 4	Total
Hydro	33	1412	276	238	1960
Thermal (Coal)	1600	-	-	-	1600
Thermal (Oil)	850	-	300	750	1900
Geothermal	140	-	-	-	140
Gas Turbine	400	80	75	112	667
Diesel	10	-	-	76	86
Total	2893	1632	652	1177	6353

curves to get maximum energy generation is an appropriate consideration.

Table 1-2 shows that there are 80 hydropower plants in the Java and Bali Power System. The installed capacity of eight of these plants ranges from 125 to 175 MW. The other 72 plants are smaller than 125 MW. Included in the eight larger units are the Saguling, Cirata and Jatiluhur hydropower plants. The Hydroelectric stations in the Citarum River provide 15% of the total energy and 21% of the capacity of the Java and Bali system. The Jatiluhur plant has been producing electricity since 1965. The Saguling plant has been in operation since 1986, and the Cirata plant has been fully operational since 1989. The Saguling and Cirata plants are operated under the authority of the State Electricity Corporation of the Department of Mines and Energy. The Jatiluhur plant is under the authority of the Jatiluhur Project of the Department of Public Works of the Indonesian government.

To operate the power plants on the Citarum River, the two institutions meet regularly. Every month the record of reservoir operations for the previous month is examined so as to plan the reservoir operations for the coming month based on operating guidelines. The authorities determine the operating guidelines of the reservoirs by consensus, assuming an average of "normal" year of historical flows. Since this procedure estimates only one value of annual power generation, the reliability of estimated annual power generation in the long term cannot be predicted.

1.2 Objective of the Study

The objective of this research was to study the operational behaviour of the hydro power system in the Citarum River. Included in this objective were the following tasks:

1. to develop a stochastic model of the monthly flows of Citarum River at Nanjung,
2. to obtain a reliability curve for the annual energy generated by the system,
3. to examine the performance of the current operating guideline that is used by the authorities and that is proposed by the WRD project,
4. to study the characteristics of the Saguling, Cirata and Jatiluhur hydropower plants in general.

To meet with the above objectives, a model that simulates the operation of the series of three reservoirs is needed. This thesis describes such model and its use in achieving the objectives.

1.3 Method of Research

Simulation is widely used for the planning, management and operation of water resources systems. Simulation can be used to estimate the parameters of interest of a system for many given scenarios.

In this study the operations of the Citarum hydropower system were simulated mathematically. Synthetic inflows to the simulation model were generated by a stochastic model that preserved the statistical characteristics of the historical flows. By using series of synthetic inflows the performance of the hydropower system can thus be studied, in particular, the reliability of annual energy generation for a given rule curve can be observed. Two rule curves were examined in the simulation: the rule curve that is currently used by the authorities, and the rule curve that is proposed by the WRD Project of the Water Resources Development in the Citarum Area. In addition two other rule curves, the empty reservoir rule curve, assuming all reservoirs are empty, and the full reservoir rule curve, assuming all reservoir are full, were also simulated as limiting cases. Hence, rule curves with their corresponding reliability curves could be obtained. Finally, the results were analyzed and comparisons were made (see Figure 1-3).

1.4 Thesis Outline

The general description of the problem and development of the hydro electric power system in the Citarum river have been presented in this chapter. Next, the climate and hydrology of the study area are given in Chapter 2. In Chapter 3, stochastic modelling of monthly and annual flows is discussed. In the Chapter 4, reservoir operations are discussed. This chapter contains the description of the model used, reservoir operation software and the applications of the simulation method. The discussion of the results is presented in Chapter 5. Conclusions and recommendations are presented in Chapter 6. Historical data and listing of programs are presented in appendices.

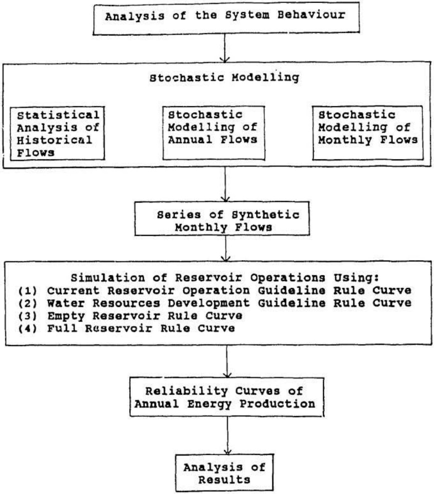


Figure 1-3: Method of research

Chapter 2

Description of Study Area

2.1 Introduction

This chapter presents a description of the multi-reservoir hydropower system on the Citarum River in West Java, Indonesia. In addition, this chapter describes the existing reservoirs and hydropower plant characteristics, the climate and hydrology, and the present system of reservoir operations.

2.2 Reservoir and Hydropower Plant Characteristics

In this section the system configuration, reservoir data and hydropower characteristics are described.

2.2.1 System Configuration

The system configuration of the Citarum multi-reservoir system is shown in Figure 2-1. The Saguling is the most upstream reservoir, the middle reservoir is the Cirata, and the most downstream reservoir of the system is the Jatiluhur. Figure 2-2 shows the profile of the Citarum River from the Saguling to the Jatiluhur Reservoirs. There are no channels flowing out of the system. Therefore, the water that leaves the Saguling reservoir

Figure 2-1: Citarum Multi-Reservoir System

Figure 2-2: Profile of the Citarum River from the Saguling Reservoir to Jatiluhur Reservoir

through the turbines (or over the spillway) flows directly into the Cirata Reservoir, and the water that leaves the Cirata Reservoir through the turbines (or over the spillway) flows directly into the Jatiluhur Reservoir. There is an irrigation area downstream of the Jatiluhur Reservoir. The water for irrigation is taken from the Citarum River just downstream of the Jatiluhur Hydropower Plant.

In this study the parameter of interest is the annual energy generation of the hydropower system of the Citarum River. All water that is used for the hydropower plants flows back to the river and the annual water demand for irrigation is not significantly affected by the operation of the hydropower plants. Therefore, it was not necessary to take into account the water demand for irrigation in this study.

2.2.2 Reservoir Characteristics

Reservoir capacities are presented in Table 2-1. For convenience the surface area of a reservoir can be expressed as reservoir storage in the form of a power function:

$$A_i = a_i \; V_i^{b_i} \tag{2-1}$$

in which:

A_i = reservoir surface area of the reservoir i [10^6 m^2],

V_i = reservoir storage of the reservoir i [10^6 m^3],

a_i, b_i = constants for the reservoir i.

The constants a_i and b_i for each reservoir are presented in Table 2-2 as determined by NEWJEC (1988).

Table 2-1: Water level and the storage capacities of the Saguling, Cirata and Jatiluhur Reservoirs (NEWJEC, 1988)

Descriptions	Saguling	Cirata	Jatiluhur
Full supply levels (m)	643	220	107
Low supply levels (m)	623	205	90
FSL volumes (10^6 m^3)	881	1973	3000
LSL volumes (10^6 m^3)	272	1177	1200

2.2.3 Hydropower plant characteristics

Hydropower plants may be characterized by various parameters, including maximum plant capacity, plant efficiency, minimum and maximum heads, and the minimum and maximum turbine flows. Table 2-3 shows the maximum plant capacities, the plant efficiencies, and the minimum turbine flows of the Saguling, Cirata and Jatiluhur Hydropower Plants. In general, the water head may be expressed as a function of the reservoir volume. The variation in tail water level is small relative to the total head. Therefore, in this study, it was assumed that the water head is not a function of tail water level. This is given by:

$$H_i = c_i \; V_i^{d_i} + e_i \qquad \textbf{(2-2)}$$

in which:

H_i = head of the reservoir i [m],

V_i = reservoir storage of the reservoir i [10^6 m^3],

c_i, d_i, e_i = constants for the reservoir i.

The constants c_i, d_i and e_i for each reservoir are given in Table 2-2 (NEWJEC, 1988).

Table 2-2: Constants of a_i & b_i in the Equation 2-1, and c_i, d_i & e_i in the Equation 2-2 for the Saguling, Cirata and Jatiluhur Reservoirs (NEWJEC, 1988)

Coefficients	Saguling	Cirata	Jatiluhur
a_i	289.7000	17.4600	2.1500
b_i	0.0441	0.2500	0.4523
c_i	0.1130	0.3963	0.9605
d_i	0.8939	0.6667	0.5546
e_i	0.2664	0.6340	0.6214

Table 2-3: Characteristics of the Saguling, Cirata and Jatiluhur Hydropower Plants (NEWJEC, 1988)

Descriptions	Saguling	Cirata	Jatiluhur
Max. plant capacities (MW)	715	518	175
Plant efficiencies (%)	0.835	0.816	0.870
Min. heads (m)	371	102	64
Max. heads (m)	391	117	81
Min. turbine flow (10^6 m^3)	147	315	315
Max. turbine flow (10^6 m^3)	586	1453	665
Tail water level (m)	3	2	1

2.3 Climate and Hydrology

The climate of the Citarum river basin is dominated by two seasons, the dry season and the wet season. The seasonal rainfall distribution, which relates to the seasonal flows, is affected by the monsoons. The dry season is from May to October and is caused by the movement of the dry southeast monsoon. The wet season occurs during the remaining months, which is caused by the movement of the wet northeast monsoon. Evaporation increases during the dry season and decreases during the wet season.

In this study, the historical inflow sequence, and the coefficients of evaporation were taken from the report on the operation guidelines for the Saguling, Cirata and Jatiluhur Reservoirs for the years 1990-1991 [Suladjiono, 1990]. The Citarum River flows at Nanjung for the period 1928 to 1987 are presented in Appendix A. Inflows to the Saguling Reservoir and local inflows to the Cirata and Jatiluhur Reservoirs were calculated by multiplying the flows of the Citarum River at Nanjung by constants. These flow constants were calculated based on a comparison of drainage areas, as shown in Figure 2-3. The flow constants for the Saguling inflows, the local Cirata inflows and the local Jatiluhur inflows were taken as 1.3, 0.68 and 0.33, respectively.

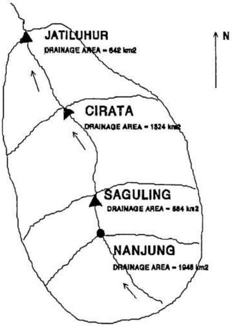


Figure 2-3: Drainage area of the Citarum River

The coefficients of evaporation were determined by simulating inflows (in volume units) to a reservoir and comparing with the reservoir volume for each month. The difference in the reservoir volume between inflow and actual additional volume into the reservoir in the related month was defined as water that was lost to evaporation. The evaporation was then compared with the related reservoir surface area (in square metres) to obtain the evaporation coefficient (in metres) (Kananto, 1992). The evaporations were calculated using the following equation.

$$E_i = c_{e,i} A_i \tag{2-3}$$

in which:

E_i = evaporation of the reservoir i [10^6 m^3],

A_i = surface area of the reservoir i [10^6 m^2],

c_{ei} = evaporation coefficients of the reservoir i (see Table 2-4).

Table 2-4: Evaporation coefficients for the Saguling, Cirata and Jatiluhur Reservoirs (Suladjiono, 1990).

Months	c_{ei} of Saguling (m)	c_{ei} of Cirata (m)	c_{ei} of Jatiluhur (m)
May	0.12	0.12	0.12
June	0.12	0.12	0.12
July	0.12	0.12	0.12
August	0.12	0.12	0.12
September	0.12	0.12	0.12
October	0.09	0.09	0.09
November	0.06	0.06	0.06
December	0.06	0.06	0.06
January	0.06	0.06	0.06
February	0.06	0.06	0.06
March	0.06	0.06	0.06
April	0.09	0.09	0.09

2.4 Current Reservoir Operations

The reservoir operation guidelines used in the Citarum project at present are determined by the requirements of the Java and Bali power systems. The electric power system in Indonesia is authorized by the state electric corporation, known as the Perusahaan Listrik Negara (PLN). The PLN is responsible for the operation, transmission and distribution of electricity. However, multipurpose dams such as the Jatiluhur and the Brantas dams, which have hydropower plants, are exceptions. These reservoirs are operated by the Department of Public Works.

The Java and Bali power systems are divided into four operational areas: West Java, western West Java, Central Java, and East Java (including the island of Bali). Each area has an Area Control Centre (ACC), and these ACC's are controlled by the Java Control Centre (JCC). The ACC's are responsible for the coordination of power system operations in their respective areas. The Saguling, Cirata and Jatiluhur hydropower plants are coordinated by the ACC of West Java, which is located in Jakarta.

The main job of the ACC is to control system operations. Daily, weekly and monthly scheduling of reservoir operations are done by the ACC, based on load forecasting that is done by the JCC. The main function of the JCC is daily load forecasting, hydropower scheduling, and thermal unit commitment, based on the daily data from the system. The data for the JCC is sent one day ahead to the JCC by each ACC, using facsimile and computer networks, between 08.00 am and 10.00 am. Instructions are sent back on the same day, as operation statements, between 16.00 pm to 17.00 pm to all ACC's. Therefore, the daily scheduling decided upon by the JCC is received by the ACC's about seven hours before real-time operations. These real-time operations are finalized by matching load-forecasting with the forecasting of monthly power capacities. The forecasting of monthly power capacities is developed using the existing reservoir operation guidelines, which are ostensibly based on optimal forecasting of the annual power capacity.

At present the reservoir operation guidelines are developed as follows. The historical flows are divided into five classifications, as shown in Table 2-5. Twelve monthly flows of the average of the "normal flows" of the historical data were obtained. These monthly flows were then used to obtain the guidelines for reservoir operations of the Saguling, Cirata and Jatiluhur Reservoirs (Suladjiono, 1991). The current reservoir operating guidelines (CROG) for the Saguling, Cirata and Jatiluhur Hydropower Plant are presented in Figure 2-4 (a), (b) and (c). The water resources development guidelines, the guidelines proposed by the Water Resources Development Project, are also presented in the same figures. These guidelines are the two of the rule curves examined in Chapter 4.

Table 2-5: Five classifications of historical annual flows (Adji, 1989)

No.	Classifications	Probability
1	Very dry	(0 - 20)%
2	Dry	(20 - 45)%
3	Normal	(45 - 55)%
4	Wet	(55 - 80)%
5	Very Wet	(80 - 100)%

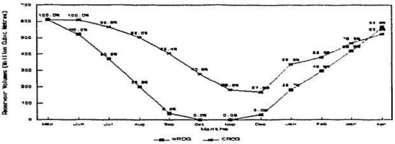


Figure 2-4 (a): CROG and WRDG for the Saguling Hydropower Plant

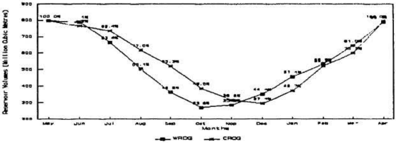


Figure 2-4 (b): CROG and WRDG for the Cirata Hydropower Plant

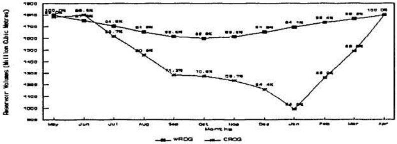


Figure 2-4 (c): CROG and WRDG for the Jatiluhur Hydropower Plant

Chapter 3

Stochastic Modelling of Flows

3.1 Introduction

This chapter discusses the three stochastic models that were used to simulate the annual and monthly flows of Citarum River at Nanjung. First, the statistical analysis of historical flows is presented. Second, the stochastic modelling of the annual flows is examined. Third, the stochastic modelling of monthly flows is discussed. Finally, the rationale for the selection of the stochastic models used is described.

3.2 Statistical Analysis of Historical Flows

This sub-section discusses the statistical analysis of the historical data of annual and monthly flows of the Citarum River at Nanjung. In the first stage, moving average plots of both annual and monthly flows are presented. In the second stage, the annual flows are analyzed and then the monthly flows are examined in the next stage.

3.2.1 Time Series and Moving Average Plots

A scatter diagram of the historical annual flows versus time

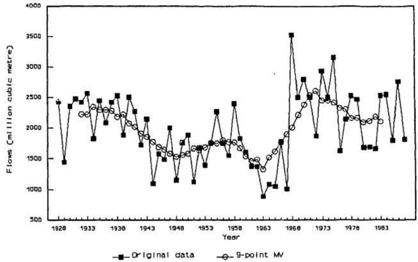


Figure 3-1: Scatter and a 9-point moving average diagram of historical annual flows of the Citarum River at Nanjung

was plotted (see Figure 3-1). To have more understanding of the flows, a 9-point moving average diagram was plotted in the same scatter diagram. From these plots it was evident that points above the mean were likely to be followed by other points above the mean, and that points below the mean were likely to be followed by other points below the mean. It was therefore expected that the historical annual flows had a high coefficient of correlation. The auto-correlation function (ACF) and partial auto-correlation function (PACF) of the flows are presented in Figures 3-2 and 3-3 respectively. The ACF and PACF are dimensionless.

The historical monthly flows and the average of the 60 year monthly flows versus time are presented in Figures 3-4 and 3-5.

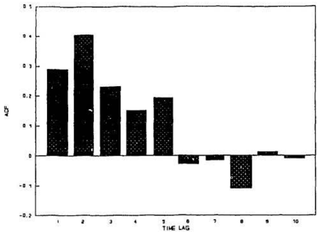


Figure 3-2: Auto-correlation function of the historical annual flows of the Citarum River at Nanjung

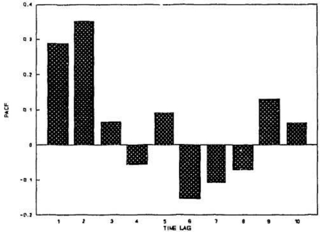


Figure 3-3: Partial auto-correlation function of the historical annual flows of the Citarum River at Nanjung

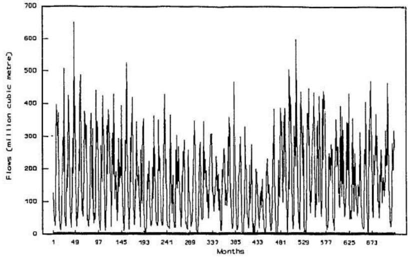


Figure 3-4: Historical monthly flows of the Citarum River at Nanjung for the periods of 1928-1987

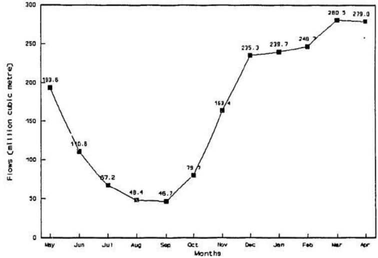


Figure 3-5: Average of the 60 years of the historical monthly flows of the Citarum River at Nanjung

It was evident from these plots that the low flows (the dry season) are from May to October and the high flows (the wet season) are from November to April.

3.2.2 Statistical Characteristics of Historical Annual and Monthly Flows

The statistical characteristics of the historical annual and monthly flows of Citarum River at Nanjung are presented in Appendix B. The minimum, mean, maximum and standard deviation of the annual flows are 886, 1990, 3525 and 568 million cubic metre respectively. The coefficient of skewness, the lag-1 auto-correlation coefficient and the Hurst coefficient are 0.16, 0.28 and 0.86 respectively. It is important to note here that the means of monthly flows in July, August and September are low (67, 48 and 46 million cubic metre) the standard deviations are relatively high (59, 52 and 48 million cubic metre), the coefficient of skewness are relatively high (1.9, 3.2, and 1.7).

3.3 Stochastic Modelling of Annual Flows

In this study, the Box-Jenkins approach was used (Salas, *et al.*, 1988). In general, stochastic modelling of annual flows using the Box-Jenkins approach consists of three stages. These are:

(1) preliminary analysis and model identification,

(2) parameter estimation, and

(3) model testing.

3.3.1 Preliminary Analysis and Model Identification

There are five steps in the preliminary analyses and model identification. First, the normality of the annual flows is examined. Second, if the data is not normal, transformation of the non-normal data into normally distributed data is necessary. Third, the sample correlogram of the annual flows ($r_k(x_a)$) versus time-lag k is plotted. Fourth, partial auto-correlation versus time lag-k is also plotted. Finally the identification of the best stochastic model is performed.

Herein, the normality test was based on the Probability Plot of Correlation Coefficient (PPCC) test (Lye, 1992). It was found that the probability plot correlation coefficient of the annual flows (r_{xa}) is 0.986, and the critical value of the probability plot correlation coefficient for a sample size of 60 and at significance level of 5% (r(05;60)) is 0.985. Therefore it could be concluded that the annual flows of Citarum River at Nanjung could be fitted by a normal distribution.

The transformation to the normal distribution was therefore not necessary. The correlogram and the partial correlogram of annual flows were determined and plotted (see Figures 3-2 and 3-3 respectively). Two kinds of independence tests were used in this study: non parametric tests and parametric tests. The non-parametric test used was the Ranked Von-Neumann Ratio Test. It was found from this test that the absolute value of Ranked Von-Neumann parameter (11.307) was greater than 1.96 (z(5%)). From this test, it was concluded that the flows were dependent. The parametric test were Barlett's Test and Anderson's Test. A summary of the results of applying Barlett's and Anderson's tests are presented in Table 3-1.

The graphs of Barlett's and of Anderson's tests are shown in Figures 3-6 and 3-7. It was concluded from the non-parametric and the parametric tests that the annual flows of Citarum River at Nanjung were dependent. Figures 3-6 and 3-7 also indicate that the values of ACF and PACF are significantly greater than Barlett's and Anderson's lines in time lag-1 and 2. Therefore it was concluded that a Box-Jenkins model; such as an ARMA(2,0), ARMA(1,1), ARMA(1,2), ARMA (2,2), ARMA(2,1) might be appropriate for modelling the flows.

Table 3-1: Summary of Barlett's and Anderson's tests for the historical annual flows of the Citarum River at Nanjung

No	Auto-correlation of:	r	Barlett	Anderson
1	lag-1 (r1)	0.288	0.250	0.240
2	lag-2 (r2)	0.406	0.250	0.242
3	lag-3 (r3)	0.229	0.250	0.244
4	lag-4 (r4)	0.149	0.250	0.246

3.3.2 Estimation of Parameters

The estimation of parameters was performed using the method of maximum likelihood (MLE), computed using Minitab (Minitab Reference Manual, 1989). The possible models are presented in Table 3-2. The Minitab program used to estimate parameters by maximum-likelihood method is listed in Appendix D.

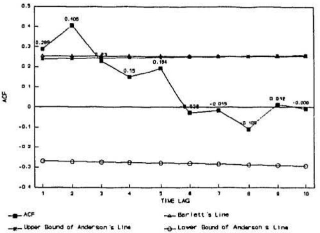


Figure 3-6: Barlett's and Anderson's tests for the ACF of the annual flows of the Citarum River at Nanjung

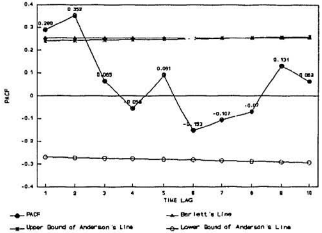


Figure 3-7: Barlett's and Anderson's tests for the PACF of the annual flows of the Citarum River at Nanjung

3.3.3 Model Tests

There were several tests available to choose the optimal model. These were mean square errors (MS) and residual analyses, parsimony and optional tests based on Monte Carlo methods. The MS test shows how well the model fits the data. Residual analysis is used to test the normality of the residuals. The parsimony test is used to choose the best model based on the number of variables and the value of the mean square error.

In this study the MS was obtained from the parameters estimated using the maximum likelihood method. The normality of residuals was analyzed using the probability plot correlation coefficient test and the parsimony test was examined using the AIC (Akaike Information Criterion). These results are presented in Table 3-2. This table shows that the smallest values of the MS and the AIC(p) are associated with the ARMA(2,0) model. This model also has normal residuals. Hence the ARMA(2,0) was concluded to be the preferable model. The ARMA (2,0) model of the annual flows of Citarum River at Nanjung is given by the following equation:

$$Y_{a(t)} = a_1 Y_{a(t-1)} + a_2 Y_{a(t-2)} + \epsilon_t \qquad \textbf{(3-1)}$$

where:

$Y_{a(t)}$, $Y_{a(t-1)}$, $Y_{a(t-2)}$ = generated annual flows in year t, t-1 and t-2 respectively,

$a_1 = 0.1758$,

$a_2 = 0.3737$,

$\epsilon_t \sim N\ (900.37,\ 520.422)$

Table 3-2: Summary of the stochastic annual flow models and the respective values of r of residuals, AIC(p) and MS

No	Stochastic Models	AIC(p)	r of residuals	MS
1	ARMA(2,0)	754.55	0.972	270839
2	ARMA(1,1)	757.72	0.969	285499
3	ARMA(1,2)	758.01	0.970	277494
4	ARMA(2,2)	760.57	0.971	280078
5	ARMA(2,1)	757.54	0.970	275303

Monte carlo simulation was used to check whether the ARMA(2,0) model on average could reproduce the sample mean, sample standard deviation, coefficient of skewness, first and second order auto correlations, and the maximum and minimum of the historical flows. Since the number of the years of historical flows was 60, the number of the generated flows was also 60. The number of replications used in this simulation was 1000. It was found on average that the mean from the model was a little too low and the standard deviation was a little too high, hence adjustment to the mean and standard deviation were made for a better match. The corrected equation of ARMA(2,0) for the annual flows of the Citarum River at Nanjung is shown in Equation 3-2.

$$Y_{a(t)} = 0.1758\,Y_{a(t-1)} + 0.3737\,Y_{a(t-2)} + 943.87 + 478.056\,z_t \qquad (3-2)$$

where:

$z_t \sim N(0,1)$.

The result summary of the simulations is shown in Table 3-3.

Table 3-3: Summary of results of simulation of ARMA(2,0) stochastic model the annual flows of the Citarum River at Nanjung (number replications 1000)

No	Moments	Historical flows (10^6 m^3)	Generated flows (10^6 m^3)	Lower Bound	Upper Bound
1	μ	1990.775	2013.878	1705.388	2322.368
2	σ	573.597	556.153	437.097	675.2087
3	C_s	0.169	−0.079	−0.746	0.588
4	r1	0.289	0.195	−0.166	0.556
5	r2	0.406	0.405	0.162	0.647
6	K	0.868	0.754	0.620	0.887
7	Minimum	886.700	685.222	158.778	1211.665
8	Maximum	3525.100	3280.819	2764.829	3796.809

Based on the 95% confidence limit tests it was concluded that an ARMA (2,0) stochastic model could reproduce the statistical characteristics of the historical flows. It is important to note here that the Hurst coefficient (K) for the historical flows was very close to the upper bound value. The high Hurst coefficient suggest that long-term persistence may be in the time series, and this required a long-memory model (Mc Mahon and Mein, 1985). However, since the Citarum multi-reservoir system (Saguling, Cirata and Jatiluhur) is filled every year, a short-memory stochastic model was chosen.

3.4 Stochastic Modelling of Monthly Flows

The important criterion in modelling monthly flows is that the model should preserve both the characteristics of the monthly and the annual flows. Two methods were applied in this study: the Method of Fragments and the Two-Tier model (Mc Mahon and Mein, 1985). Theoretically the fragments method would preserve characteristics of the annual flows; the characteristics of monthly flows would then need to be checked. To apply the method of fragments, the ARMA(2,0) stochastic model was first used to generate synthetic annual flows. The annual generated flows were then disaggregated to get monthly synthetic flows.

The Two-Tier model, which is a modification of the Thomas-Fiering model (Thomas and Fiering, 1966), is based on making adjustments to the monthly generated flows produced by the Thomas-Fiering model such that the sum of the monthly generated flows matched the annual synthetic flows produced by the stochastic annual model (Harms *et al.*, 1967).

3.4.1 Method of Fragments

The Method of Fragments is one of the so-called disaggregation models. The steps used to generate monthly flows using the Method of Fragments are as follows (Mc Mahon and Mein, 1985).

First, the observed monthly flows were standardized year by year using the corresponding annual flow volumes. The standardization of monthly flows for each year was then considered to be a fragment. Second, the annual flows from the observed data were ranked in ascending order. Third, classes were formed, where the number of classes was equal to the number of observed data (60). The value of class 1 (the lowest class) was zero and the value of class 60 had no upper limit. Third, annual flows were generated using the appropriate model (ARMA(2,0) for annual flows). Fourth, the generated annual flows were checked individually to determine the appropriate class for each flow. Finally the generated annual flows were disaggregated using the corresponding fragments.

3.4.2 Two-Tier Model

Before discussing the Two-Tier model, the Thomas-Fiering seasonal model is examined. The equation of the Thomas-Fiering (Thomas and Fiering, 1966) seasonal model is as follows:

$$y_{tm(i+1)} = \bar{x}_{tm(j+1)} + b_j (y_{tm(i)} - \bar{x}_{tm(j)} + z_{(i)} \sigma_{xtm(j+1)} \sqrt{(1 - r_{xtm}^2)} \qquad \textbf{(3-3)}$$

where:

$y_{tm(i+1)}$, $y_{tm(i)}$ = generated flows during the $(i+1)^{th}$ and i^{th} seasons reckoned from the start of the synthesized sequences,

$\bar{x}_{tm(j+1)}$, $\bar{x}_{tm(j)}$ = mean flows during $(j+1)^{th}$ and j^{th} seasons within a respective annual cycle of seasons. Since the "seasons" which were modelled here were months, $1 \le j \le 12$,

b_j = least squares regression coefficient for estimating the $(j+1)^{th}$ flow from the j^{th} flow,

$$b_{(j)} = r_{xtm(j)} \frac{\sigma_{xtm(j+1)}}{\sigma_{xtm(j)}}$$

$z_{(i)}$ = random normal variate with mean zero and variance unity,

$\sigma_{(j+1)}$, $\sigma_{(j)}$ = standard deviations of flows during the $(j+1)^{th}$, j^{th} seasons,

$r_{(j)}$ = serial correlation coefficient between flows in the j^{th} and $(j+1)^{th}$ seasons.

To generate monthly flows using the Thomas-Fiering model, 36 parameters are required. These are: monthly means, standard deviations, and lag-one serial correlations for the 12 months. The Thomas-Fiering model is restricted to normally distributed flows. Therefore the 36 parameters that would be used in the equation had to be obtained from normally distributed data.

The steps for generating monthly flows using this model were as follows. The first step was the normality tests for the monthly historical flows. These tests were performed using the Probability Plot Correlation Coefficient (PPCC) Method. If the flows are not normal, then they are transformed to be normal using the Box-Cox transformation (Box and Cox, 1964).

$$x_{tm\lambda(j)} = \frac{(x_{m(j)}^{\lambda} - 1)}{\lambda} \qquad \lambda \neq 0$$
$$x_{tm\lambda(j)} = \log x_{m(j)} \qquad \lambda = 0 \tag{3-4}$$

where:

$x_{m(j)}$ = historical monthly flows,

λ = transformation parameter,

$x_{tm\lambda(j)}$ = transformed monthly flows.

The transformation parameter λ is determined using the PPCC method (Lye, 1992), so that the transformed monthly flows are approximately normally distributed at the 95% confidence level. The example of the PPCC test for monthly flows of May is presented in Appendix C. From these tests the transformations for each monthly flows were chosen.

In the second step, the data were transformed and the transformed data were again tested to check whether the data had been successfully transformed to a normal distribution. The appropriate transformations were then chosen. A summary of the transformations for the monthly flows is presented in Table 3-4. Finally, from the transformed data, the necessary 36 parameters were obtained (see Table 3-5) and were used in the generation of monthly flows.

The output of the monthly generated flows from the Thomas-Fiering model does theoretically preserve the statistical characteristics of the monthly historical flows. However, it does not preserve the statistical characteristics of the *annual* historical flows. To solve this problem the monthly generated flows from the Thomas-Fiering model were summed up for each year and were then divided by the annual generated flows from the stochastic annual flow model (ARMA(2,0)) so as to obtain adjustment factors. Finally, the monthly generated flows from the Thomas-Fiering model

Table 3-4: Summary of the transformations of the historical monthly flows

No	Months	λ	$r(\lambda)$
1	May	0.234	0.995
2	June	0.317	0.996
3	July	0.297	0.987
4	August	0.304	0.983
5	September	0.293	0.989
6	October	0.405	0.992
7	November	0.545	0.992
8	December	0.860	0.993
9	January	0.073	0.994
10	February	0.592	0.994
11	March	0.360	0.993
12	April	1.183	0.995

Table 3-5: Means, standard deviations, and coefficients of correlation of transformed flows

No	Months	μ (10^6 m^3)	σ (10^6 m^3)	r_1
1	May	10.001	1.864	0.591
2	June	10.292	2.766	0.732
3	July	7.514	2.948	0.825
4	August	6.362	3.122	0.654
5	September	5.979	3.194	0.673
6	October	10.719	5.176	0.695
7	November	26.220	10.050	0.492
8	December	124.770	46.840	0.158
9	January	6.654	0.523	0.392
10	February	41.560	10.030	0.377
11	March	17.854	3.479	0.461
12	April	669.000	269.300	0.975

were multiplied by these adjustment factors to give monthly generated flows in which the annual statistics were preserved. This procedure is a modified Thomas-Fiering model and is called the Two-Tier model.

3.4.3 Analysis of Results

The Method of Fragments and the Two-Tier Model were evaluated using Monte Carlo Simulation based on 500 replications. Both methods used an ARMA(2,0) model to generate annual flows that had been previously validated as the stochastic model of the annual flows (as discussed in sub-section 3.2). The Citarum stochastic model, so-called as the CTR-STOCHMOD was written using Quickbasic language (Hergert, 1989). The listing of the CTR-STOCHMOD is presented in Appendix F.

Comparisons of the statistical characteristics of the generated monthly flows between both models are shown in Table 3-6. The comparisons are based on range values between the lower and upper bounds of 95% confidence limit tests. If the statistical characteristics of the observed flows lie between the lower and upper bounds, this means the stochastic model preserved the statistical characteristics of historical data, and vice versa. Table 3-6 shows that both method could not preserve the standard deviation and coefficient of skewness in the month of August because the mean of the historical monthly flows is low (48 million cubic metre), the standard deviation is high (52 million cubic metre) and the coefficient of skewness is unusually high (3.2).

Table 3-6 shows that the Two-Tier model preserved more parameters of the historical monthly flows than did the Method of Fragments. Therefore, the Two-Tier model, while not completely adequate, was considered as a more suitable model than the Method of Fragments to generate monthly flows of the Citarum River at Nanjung.

Table 3-6: Comparison of the method of Fragments and the Two-Tier method for generating synthetic monthly flows

No	Descriptions	Statistics not Preserved	
		Method of Fragments	Two-Tier Model
1	May	r1	-
2	June	r1, r2	-
3	July	r1, r2	σ, Max
4	August	σ, C_s, r1, r2, Max	σ, C_s, Min, Max
5	September	r1, r2	-
6	October	-	-
7	November	r1	-
8	December	-	-
9	January	σ, r1	-
10	February	r1, r2	-
11	March	r1	-
12	April	r1	r1, r2

3.5 Final Stochastic Models

From sub-sections 3.2 and 3.3 it was concluded that the most appropriate stochastic model for the annual flows was the ARMA(2,0) model, and the more appropriate stochastic model for the monthly flows was the Two-Tier model. In this study the simulations of the hydropower generation of the Citarum multi-reservoir system was based on monthly flow data. Therefore the Two-Tier model was used to generate synthetic monthly flows that would be used in the simulation.

Chapter 4

Reservoir Operations

4.1 Introduction

This chapter discusses the operation of the Saguling, Cirata and Jatiluhur reservoirs. As discussed in Chapter 2, these three reservoirs are operated by different authorities, but because they are in series, the operation of the upstream reservoirs affects those downstream. The consequences of applying two rule curves were examined. The rule curves used were the current reservoir operation guideline (CROG) and the water resources development guideline (WRDG) proposed by the Water Resources Development Project. The comparisons were based on the energy generation that resulted from applying these rule curves. In order to better understand the *nature* of these differences in generation and the *reasons* for them, some parameters of interest that were produced by the simulation were plotted. They include inflows, heads, turbine flows, spill flows and energy generated.

When seeking to understand the behaviour of a non-deterministic system output, it is often of interest to consider limiting cases. In this regard, two other rule curves were used in the simulations: (1) the empty reservoir (ER) rule curve, assuming that the reservoirs are kept as empty as possible to minimize spill, and (2) the full reservoir (FR) rule curve, to keep the reservoirs as full as possible, to maximize head.

The operation of the reservoirs was simulated using various synthetic inflow sequences (generated using methods in Chapter 3) using the above four rule curves. Since the synthetic flow sequences were probabilistic, the energy generations from the simulations resulted in reliability curves which are curves showing the probability that a given energy level is not exceeded. The estimation of annual energy generation was compared for three cases:

(1) those estimated by the authorities using 12 monthly flows of the average of assumed "normal" year flows,

(2) those estimated using reliability curves that obtained using one 60 year historical flow sequence,

(3) those estimated using reliability curves that obtained using 100 synthetic flow sequences of 60 years each.

4.2 Operating Policy

An operating policy is a set of rules defining system operation. Included in the operating policy of a hydropower plant system are rule curves and release rules for the reservoir. Rule curves are defined as reservoir storage volume *targets*, which should be maintained, and are made based on hydrologic characteristics of the system and the desired output. Release rules are identified as the quantity of water to be released based on power or other demands (Loucks, *et al.*, 1981). A rule curve may be made for a one year time period, or for "over-year" time periods. The one year rule curve only requires a carry-over of water from the wet season to the dry season within a given year. An over-year period rule curve requires a carry-over of water for more than one year (Kuiper, 1971).

In practice, for real time operations, both types of rules are needed to define the operating policy. Release rules provide guidance for the operation of reservoirs from day to day. For long-term planning, however, utilities develop rule curves which guide them in the allocation of water. In the dry season, the water releases should generally be arranged so that there would be sufficient water to operate the plants up to the end of the dry season. The reservoir storage is normally drawn down to a minimum before the onset of the wet season. The policy to operate reservoirs to have a minimum storage before wet season flooding serves two functions. It increases energy production by minimizing spill flows and it provides storage for flood control during the wet season.

An illustration of the application of a rule curve is described below. Figure 4-1 shows a typical reservoir that is used for hydropower. The reservoir storage between the low supply level (LSL) and the full supply level (FSL) is an intermediate supply level (ISL). Reservoir storage in the ISL is used for hydropower.

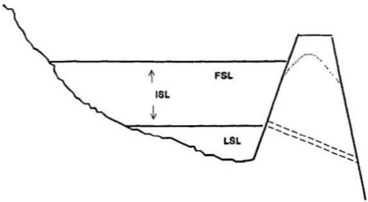


Figure 4-1: Schematic of hydropower dam.

For example, consider a rule curve based on the average of the historical monthly flows, as shown in Figure 4-2(a). The flows in the first, second, seventh and eight months might be classified as "normal" (N) flows. In the third and sixth month, they might be designated as "dry" (D), and in the fourth and fifth month as "very dry" (VD). In the ninth and tenth months, they might be designated as "wet" (W), and in the eleventh and twelfth months as "very wet" (VW).

Since the river flows are "normal" in the first and second months, the reservoir should be maintained at FSL with all water releases passing through the turbine (assuming that the turbine have been suitably sized). High flows usually occur in months eleven and twelve. Hence, the reservoir should be at LSL at the start of wet season (in month eight) so that the water in the very wet months can be stored in the reservoir without spillage (if possible) up to the end of month twelve. The reservoir levels in other months are then determined so that high energy generation might be achieved. The limiting cases of determining reservoir levels are empty reservoir and full reservoir levels. The empty level will minimize spill but at the cost of reduced head, whereas the full level will maximize head at the cost of lost water. Figure 4-2(b) shows an alternative of monthly reservoir levels for hydropower dam.

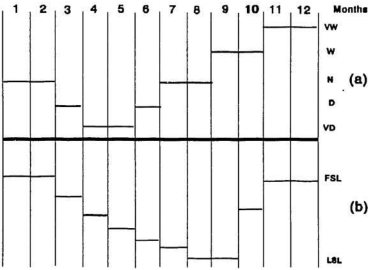


Figure 4-2: Illustration of the application of a rule curve.

In this study the rule curve associated with the current reservoir operation guideline (CROG) and the water resources development guideline (WRDG) were examined, as well as the two limiting cases, the empty and full reservoir rule curves. The following points were considered in evaluating the rule curves:

1. the influence of a given rule curve on the long term average of reservoir storage, heads on the plants, turbine flows, spills, and most importantly energy generations,
2. the effect of a given rule curve associated with upstream reservoir on energy output at a downstream reservoir,
3. the long term average of the annual energy generation,
4. the monthly variation in energy generations, and
5. the differences between the results of simulation using *generated* flows and *historical* flows.

Dealing with reservoir operations, there are two methods available: optimization and simulation methods. Optimization methods are used if an optimal solution is sought. Simulation methods are used to study the behaviour of a system given a number of differing scenarios (Loucks *et al.*, 1981). For this study the simulation method was used because long time series is used for evaluating different rule curves.

4.3 Simulation

The original intent of this study was to test IRIS (Interactive River System Simulation) Program (IRIS, 1989; 1990).

IRIS is an a readily available, user friendly, inexpensive software package to study reservoir operation guideline. However some problems were found with the IRIS software, particularly in its application tc hydroelectric systems. They are listed as follows (Louck, 1992):

1. there was no useful summary of results or statistical analysis of the output data,
2. the maximum number of replications was only 12,
3. there was no calculation of spill.

The problems of a lack of statistical analysis and an output summary were handled by writing a post-processing program. Since the calculation of spill was considered *essential* to the evaluation of energy output in this study, the IRIS software was finally found to be inappropriate for hydroelectric applications. The author therefore developed a program called the Citarum Simulation Model (CTR-SIMOD) for the purposes of this study. The advantages of developing his own program are:

(1) the output from the stochastic flow model could be incoporated,

(2) the statistical output of the program could be made,

(3) the program can be easily updated.

4.4 Citarum Simulation Model (CTR-SIMOD)

There were six stages in the simulation of the hydropower plant system on the Citarum river using CTR-SIMOD. These were:

a) estimate inflows,

b) define a rule curve,

c) calculate available water, obtain turbine flows and spill flows,

d) calculate energy generation, and

e) analyze the statistical characteristics of the parameters of interest.

The program listing of the CTR-SIMOD is presented in Appendix G. A brief description of the program is given below.

Referring to the schematic of the hydropower plant system on the Citarum River and its components (Figure 4-3) the sequence used for the simulations was as follows.

a) Estimate inflows.

The inflows to Node 1 (the Nanjung gauge site) were read from a file and were converted to inflows at Node 2 (the Saguling reservoir) by multiplying by flow coefficients based on the relative drainage areas.

b) Define a rule curve.

The rule curve for Node 2 was read from a file. This rule curve was then used in the simulation as targets for reservoir storage. There were twelve values in a rule curve, corresponding to end-of-month reservoir storage targets in a year. Hence, in one simulation of a rule curve, all reservoir storage targets in all years and for all replications had the same values.

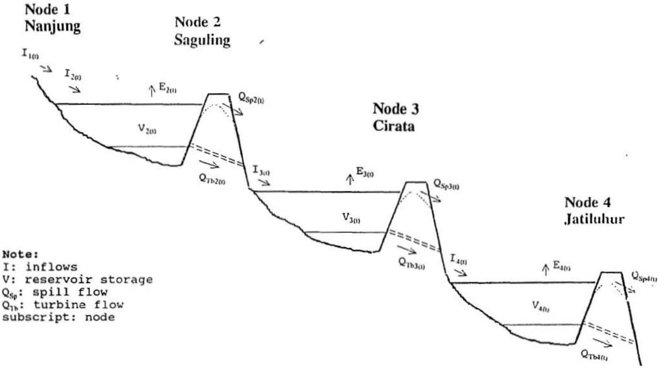


Figure 4-3: Schematic of the Citarum Hydropower Plant System, showing definitions of variables as used in equations 4-1, 4-2, 4-3 and 4-4

beginning of month t, = $a_i\ V_{i(t)}^{bi}$,

$A_{Ti(t+1)}$ = water reservoir surface area target for reservoir i at the end of month t, = $a_i\ V_{Ti(t+1)}^{bi}$,

a_i = reservoir surface coefficient for reservoir i,

b_i = reservoir surface exponent for reservoir i,

I_{it} = inflows during month t for reservoir i.

There were five possible outcomes of this calculation. The amount of available water could be:

1. negative,
2. equal to zero,
3. greater than zero but smaller than the minimum turbine flow,
4. greater than the minimum turbine flow but smaller than the maximum turbine flow, and
5. greater than the maximum turbine flow.

In this simulation program the operation of the reservoir is driven by the rule curves. This policy is reflected in the handling of these five possible cases, as described below and shown in Figure 4-4.

Case one (Figure 4-4 (a)): if the available water in month t ($Q_{Wai(t)}$) was negative, the reservoir storage at the end of month t ($V_{i(t+1)}$) was set equal to the reservoir storage target at the end of month t ($V_{Ti(t+1)}$) minus the amount of water available. The turbine flow and spill of month t were then set equal to zero.

Case two (Figure 4-4 (b)): if the available water in month t

c) Calculate water available, and obtain turbine and spill flows.

The water available for energy generation at Node 2 was calculated using the mass balance equation. The water available in month t was equal to the difference between the reservoir storage at the beginning of month and the reservoir storage target at the end of month, plus the inflows during the month, minus evaporation during the month. The reservoir evaporation during the month was equal to the average of surface area during the month (assuming the reservoir was at its target at the end of the month), multiplied by an evaporation coefficient for the month. Equation 4-1 summarizes the mass balance equation for month t.

$$Q_{WAi(t)} = V_{i(t)} - V_{Ti(t+1)} - E_{i(t)} \left(V_{i(t)}, V_{Ti(t+1)}\right) + I_{i(t)} \qquad (4.1)$$

in which:

$Q_{WAi(t)}$ = water releases from reservoir i during month t,

$V_{i(t)}$ = reservoir storage in reservoir i at the beginning of month t,

$V_{Ti(t+1)}$ = reservoir storage target for reservoir i at the end of month t,

$E_{i(t)}$ = evaporation from reservoir i during month t,

= $c_{ei(t)}$ ($A_{i(t)}$ + $A_{Ti(t+1)}$) 0.5,

$c_{ei(t)}$ = coefficient of evaporation for reservoir i for month t,

$A_{i(t)}$ = water reservoir surface area of reservoir i at the

($Q_{Wa(t)}$) was equal to zero, the reservoir storage at the end of month t ($V_{i(t+1)}$) was set at the reservoir storage target at the end of month t ($V_{Ti(t+1)}$). The turbine flow and spill flow were again set equal to zero.

Case three (Figure 4-4 (c)): if the available water in month t ($Q_{Wai(t)}$) was greater than zero but smaller than the minimum turbine flow, the reservoir storage at the end of month t ($V_{i(t+1)}$) was set at the reservoir storage target at the end of month t ($V_{Ti(t+1)}$), plus the available water. The turbine flow was again equal to zero. If the calculated reservoir storage at the end of month t ($V_{i(t+1)}$) was greater than the maximum reservoir storage, the excess was spilled. That is, the spill flow in month t was equal to the calculated reservoir storage at the end of month t ($V_{i(t+1)}$) minus the maximum reservoir storage. In this study the maximum reservoir storage is the FSL volume that was used as a constraint in the simulation. The full supply level was not used as a constraint. (This clearly does not reflect what would happen in practice, but since it is a very unlikely occurrence, no special account was taken of it in this study.)

Case four (Figure 4-4 (d)): if the calculated available water ($Q_{Wa(t)}$) was greater than the minimum turbine flow but smaller than the maximum turbine flow, all the available water was assumed to pass through the turbine. The reservoir storage at the end of month t ($V_{i(t+1)}$) was set equal to the reservoir storage target at the end of month t ($V_{Ti(t+1)}$). The maximum turbine flow was calculated for the average head during the month.

$$Q_{TbMax,i(t)} = \frac{P_{Max,i}}{\eta_i \, \rho \, g \, \overline{H}_{i(t)}} \tag{4-2}$$

in which:

$Q_{TbMax,i(t)}$ = maximum turbine flow of the reservoir i, for month t [10^6 m^3],

$P_{Max,i(t)}$ = electric power generations produced by hydropower plant i during month t, [GWh],

η_i = plant efficiency of hydropower plant i,

ρ = water density = 1000 kg/m^3,

g = acceleration due to gravity, (9.81 m/s_2),

$\overline{H}_{i(t)}$ = average head reservoir i for month t

Case five (Figure 4-4 (e)): if the calculated water available in month t ($Q_{W_{i}(t)}$) was greater than the maximum turbine flow, the turbine flow in month t was set equal to the maximum turbine flow. The reservoir storage at the end of month t ($V_{i(t+1)}$) was equal to the reservoir storage target at the end of month t ($V_{T_i(t+1)}$) plus the difference between the calculated water available and the maximum turbine flow. If this reservoir storage at the end of month t was greater than the maximum reservoir storage, this excess was spilled; that is, the spill was equal to the calculated reservoir storage at the end of month t minus the maximum reservoir storage (equation 4-3):

$$Q_{Spi(t)} = V_{i(t+1)} - V_{Max,i} \tag{4-3}$$

where:

$Q_{Sp,i(t)}$ = spill flow of the reservoir i during month t,

$V_{Max,i}$ = maximum reservoir storage of the reservoir i.

d) Calculate energy generation.

After the turbine flow for month t and the average head for month t were calculated, the energy generated was obtained from the turbine flows for the average head during the month and multiplied by the plant efficiency as shown in the following equation.

$$P_{i(t)} = \eta_i \ \rho \ g \ Q_{Tbi(t)} \ \overline{H}_{i(t)} \tag{4-4}$$

in which:

$P_{i(t)}$ = electric power generations produced by hydropower plant i during month t, [Gwh],

$\overline{H}_{i(t)}$ = average head reservoir i for month t.

The same procedure was used for all reservoirs. The only difference was in defining inflows. Inflows to downstream reservoirs were taken as the sum of the outflows of the upstream reservoir plus local inflows.

e) Analyze the statistical characteristics of the parameters of interest.

The output of the simulation included:

(1) inflows,

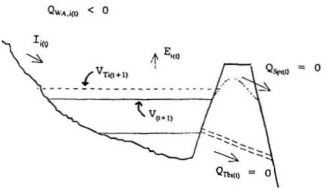


Figure 4-4 (a): Calculated water available negative.

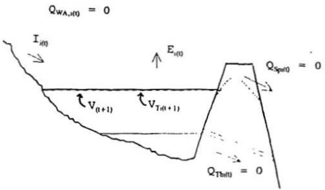


Figure 4-4 (b): Calculated water available equal to zero.

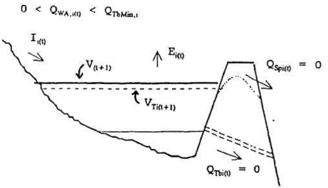


Figure 4-4 (c): Calculated water available greater than zero but smaller than minimum turbine flows

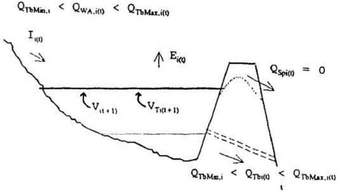
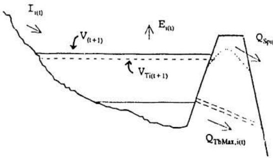


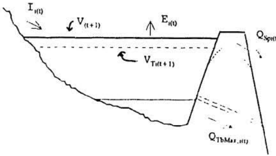
Figure 4-4 (d): Calculated water available greater than minimum turbine flows but smaller than maximum turbine flows.

$Q_{TbMax,i(t)} < Q_{WA,i(t)}$ and $V_{(t+1)} < V_{Max,i}$



(e-1)

$Q_{TbMax,i(t)} < Q_{WA,i(t)}$ and $V_{(t+1)} > V_{Max,i}$



(e-2)

Figure 4-4 (e): Calculated water available greater than maximum turbine flows and: (1) Reservoir storage smaller than maximum reservoir storage, (2) Reservoir storage greater than maximum reservoir storage.

(2) reservoir storage volumes,

(3) turbine flows,

(4) reservoir evaporation,

(5) spill flows, and

(6) energy generation,

for each reservoir.

Statistical analyses were then carried out for each of the variables above. Hence, for 60 repetitions (years), the average of 60 years of each variable of 12 months were produced. For 100 replications the average of 100 replications of the average of 60 years of each variable for 12 months were produced. For each year simulated, the monthly energy generation of twelve months were added to the annual energy generation (see Table 4-1). Hence, for one simulation of a rule curve, 6000 points of annual energy generated were produced.

Table 4-1: Estimated parameters produced by simulation using the CTR-SIMOD

No	Description	Saguling	Cirata	Jatiluhur
1	Inflows	$\mu_{I2(t)}$	$\mu_{I3(t)}$	$\mu_{I4(t)}$
2	Reservoir storage	$\mu_{V2(t)}$	$\mu_{V3(t)}$	$\mu_{V4(t)}$
3	Turbine flows	$\mu_{QTb2(t)}$	$\mu_{QTb3(t)}$	$\mu_{QTb4(t)}$
4	Spill flows	$\mu_{QSp2(t)}$	$\mu_{QSp3(t)}$	$\mu_{QSp4(t)}$
5	Energy generations	$\mu_{P2(t)}$ $P_{a2(t)}$	$\mu_{P3(t)}$ $P_{a3(t)}$	$\mu_{P4(t)}$ $P_{a4(t)}$

Chapter 5

Results and Discussion

5.1 Introduction

This chapter presents results and discussions of the stochastic modelling and the application of the simulation methods used in examining reservoir operation guidelines of the Citarum Hydropower system. Two guidelines were applied in the simulation, they were the current operating reservoir guideline (CROG) and the water resources development guideline (WRDG) rule curves. Two other rule curves were also simulated as limiting cases, representing extremes of operations. They were the empty reservoir (ER) rule curve, assuming all reservoirs are kept empty, and the full reservoir (FR) rule curve, assuming all reservoirs are full.

In this chapter, the results and discussions of the stochastic modelling are presented in the first section. The results of annual energy generation are presented and discussed in the second section. Variations in monthly energy generation are then discussed. Next, load factors of the Saguling, Cirata and Jatiluhur Plants are examined. Then, reliability curves of annual energy generation applying the CROG rule curve are presented. Finally, comparisons of energy estimated applying the CROG rule curve simulated using historical and synthetic inflows are made.

5.2 Stochastic Flow Modelling

Two stochastic models of the Citarum River flows at Nanjung were developed. These were the stochastic annual flow and the monthly flow models. The annual model was first developed. This was then used for modelling monthly flows. The proposed stochastic annual flow model was an ARMA(2,0) model, as described by Equation 3-2. The proposed stochastic monthly flow model was the Two-Tier model, which is a modification of the Thomas-Fiering model. Synthetic monthly flows were then used in the simulation of hydropower generation for the Citarum Hydropower system.

In this study the parameter of interest was annual energy generation, the statistical characteristics of synthetic annual flows were therefore of prime importance. A comparison between the statistical characteristics of synthetic annual flows and the historical annual flows are presented in Table 3-3. It was found that the ARMA(2,0) model preserved eight parameters of the historical annual flows at 95% confidence limits (see Table 3-3). However the mean of the synthetic annual flows was a little higher than the mean of the historical flows, whereas the minimum and the maximum were a little lower. The mean, minimum and maximum of the historical and synthetic annual flows may affect the annual energy generation produced by the simulation.

5.3 Annual Energy Generation

Four rule curves (the WRDG, CROG, ER and FR) were applied in the reservoir operation simulation. The results of annual energy generation are presented and discussed below.

Results

A summary of the results of the annual parameters and of the related annual energy generations is presented in Table 5-1. These annual parameters were obtained by summing the average of the monthly parameters that were produced by the simulation.

Table 5-1: Summary result of the annual parameters and the related annual energy generation (flows in 10^6 m^3, energy in GWh)

No	Descriptions	WRDG rule curve	CROG rule curve	ER rule curve	FR rule curve
1	Saguling				
	- Inflows	2618	2618	2618	2618
	- Turbine Flows	2544	2480	2608	2172
	- Evaporations	34	43	19	52
	- Spills	39	93	0	393
	- Annual Energy	2225	2183	2249	1932
2	Cirata				
	- Inflows	3953	3943	3977	3935
	- Turbine Flows	3708	3589	3942	2964
	- Evaporations	61	62	48	67
	- Spills	183	291	0	903
	- Annual Energy	941	908	902	771
3	Jatiluhur				
	- Inflows	4555	4545	4606	4532
	- Turbine Flows	4024	4136	4554	3170
	- Evaporations	86	82	55	87
	- Spills	445	326	18	1273
	- Annual Energy	764	758	636	608
4	Total Output Energy	3931	3850	3789	3312

For the Saguling hydropower plant, the highest energy generation was obtained from the simulation using the ER rule curve. The second highest was from the WRDG rule curve, the third and the fourth were from the CROG and the FR rule curves respectively. This order can be explained by the fact that the turbine flows for the ER rule curve were the highest, followed by the WRDG, CROG and FR rule curves as the second, third, and fourth highest energy productions, respectively. The water lost to evaporation and spill had the opposite order. The lowest evaporation and spill was from the ER rule curve, followed by the WRDG, CROG and FR rule curves as the second, third and fourth lowest spills, respectively.

For the Cirata hydropower plant the highest energy generation arose from the simulation using the WRDG rule curve. The second, third and the fourth highest generations were from the CROG, ER, and FR rule curves. The ascending order of the water lost from the Cirata reservoir to evaporation and spill were the ER, WRDG, CROG, and FR rule curves. The descending order for the turbine flows were the ER, WRDG, CROG and FR rule curves. Although the turbine flows of the ER rule curve were greater than the turbine flows for the WRDG rule curve, however, the energy associated with the ER rule curve was lower than that of the WRDG rule curve. This was because the ER rule curve lost more head than did the WRDG rule curve.

For the Jatiluhur hydropower plant the highest energy generation was associated with the WRDG rule curve. The second, third and fourth highest generations were for the CROG, ER and FR rule curves, respectively. The ascending order of the water lost to evaporation and spill were the ER, CROG, WRDG and FR rule curves, respectively.

Discussion

As stated in Equation 4-4, the energy produced by a turbine is a function of head and turbine flow. Turbine flow is the power flow, for example, the volume of water that is used to generate energy. It is calculated as available water minus water that is lost to evaporation and spill, and head as a function of reservoir storage. For the Saguling Reservoir, the maximum head is 391 m (full reservoir) and the minimum head is 371 m (empty reservoir). The maximum head lost when the reservoir empty is only 5%, hence the available head is 95% of the maximum head. Therefore, for the Saguling Hydropower Plant, the higher the turbine flow would produce the higher energy. The results showed that the simulation applying the empty reservoir rule curve gave the highest energy because the turbine flow was the highest.

The Cirata Plant, however, has different characteristics from the Saguling Plant. The maximum head is 117 m and the minimum head is 102 m. If the reservoir is empty, the available head is only 87% of the maximum head. Even though the empty reservoir rule curve gave the highest turbine flow but lost too much head. The results showed that the simulation applying the empty reservoir rule curve gave the highest turbine flows but did not produce the highest energy. For the Jatiluhur Hydropower Plant, the maximum head is 81 m and the minimum head is 64 m. If the reservoir is empty, the available head is only 79% of the maximum head. Hence the loss of head in the Jatiluhur Plant is even more significant than at the Cirata Plant.

As expected, the results showed that for the Cirata and Jatiluhur Plants, the simulation applying the empty reservoir rule curve did not produce the highest energy mainly because of the loss in head. Whereas applying the full reservoir rule curve gave the lowest energy mainly because of losing water to spill. The value of spill saved by keeping the reservoir low must be carefully balanced against maintaining head. Table 5-1 shows that the spill from the Jatiluhur reservoir is higher than the spills from the Saguling and Cirata reservoirs. To increase the energy production of the Citarum Hydropower System might be achieved by saving spill at the Jatiluhur reservoir by storing water at Saguling or Cirata reservoirs.

Based on the annual energy generation, applying the empty reservoir rule curve to the Saguling Plant and applying the WRDG rule curve to the Cirata and Jatiluhur Plants produced the highest energy. However, the first and the second hiहhest of the total annual energy produced by the Citarum Hydropower system is the energy that obtained by applying the WRDG and CROG rule curves respectively (see Table 5-1). The variation in monthly energy generation applying the WRDG and CROG rule curves would be discussed in the following section.

5.4 Variation in Monthly Energy Generation Applying the WRDG and CROG Rule Curves

In practice the monthly energy demands do not fluctuate too much. Hence, monthly energy production generally tends to also be less variable. The variation in monthly energy generation by applying the WRDG and CROG rule curves to the Saguling Plant is discussed in the first sub-section, and to the Cirata and Jatiluhur Plants are then examined.

Saguling Reservoir

Figure 5-1 shows the average of monthly energy generation for the Saguling Plant simulated using the WRDG and CROG rule curves. To study the variation in monthly energy generation, the average of inflows, heads, turbine flows and spill flows of the Saguling Plant are discussed. The inflows to the Saguling Reservoir are shown in Figure 5-2. The simulation used the same inflows, so that the lines for these graphs are coincident. Figure 5-3 shows the heads of the Saguling hydropower plant for both rule curves. The heads associated with the CROG rule curve are higher than the heads for the WRDG rule curve for eleven points. Only in the month of April, the head for the CROG rule curve was lower. In the month of May the heads for both rule curves had their maximum heads.

For the WRDG rule curve, the lowest head was in the month of September, the month which showed the lowest reservoir storage. This resulted in the difference between the reservoir storage

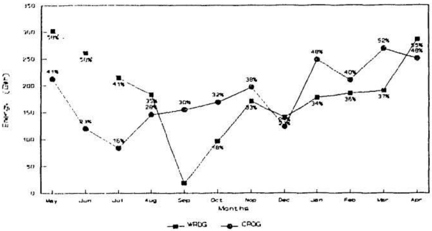


Figure 5-1: Energy generation of the Saguling Hydropower Plant

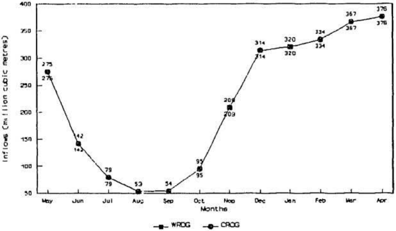


Figure 5-2: Inflows to the Saguling Reservoir

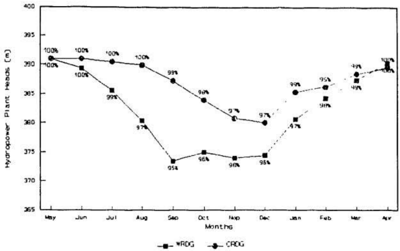


Figure 5-3: Heads on the Saguling Hydropower Plant

targets of the WRDG rule curve for the months of September and October being *negative*. The lowest inflows occurred in the months of August and September. In order to try and meet the target elevations, all the inflow was stored, hence the simulated turbine flow for the month of September was extremely low for the WRDG rule curve (only 3.7% of the maximum turbine flow of the Saguling hydropower plant, see Figure 5-4). The energy output was consequently very low as well (see Figure 5-1).

For the WRDG rule curve the highest turbine flow was found to be 57.9% of the maximum turbine flow (in May), and the lowest was 3.7% (in September). The turbine flows for the CROG rule curve were less variable than the turbine flow associated with the WRDG rule curve, because the target elevations were higher. The energy output for the CROG rule curve was consequently less variable as well. For the CROG rule curve, the highest turbine flow was 51.7% of the maximum turbine flow (in March) and the lowest was 16.0% (in July).

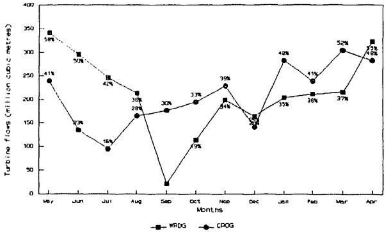


Figure 5-4: Turbine flows for the Saguling Hydropower Plant

As expected, the spill flows are inversely proportional to the turbine flow. The WRDG rule curve shows less spill and higher turbine flows because the reservoir is kept lower. Figure 5-5 shows the spill flows from the Saguling reservoir. The spill flows for the CROG rule curve were generally higher than the spill flows for the WRDG rule curve, except in the month of April. The highest spill flow for the CROG rule curve was found to occur in May (29.0 x 10^6 m^3). This was 10.8% of the t tal outflow from the Saguling reservoir (29.0×10^6 m^3 of spill flow plus 239.2×10^6 m^3 of turbine flow). The highest spill flow for the WRDG rule curve occurred in April (16.2×10^6 m^3). This was 4.8% of the total outflow from the Saguling node (16.2×10^6 m^3 of spill flow plus 321.8×10^6 m^3 of turbine flow).

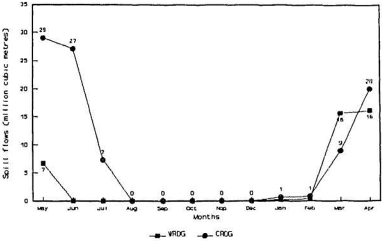


Figure 5-5: Spill flows from the Saguling Reservoir

The graphs of the average monthly energy generation, Figure 5-1, shows that the energy production for the WRDG rule curve in September was the lowest, at 3.5% of the maximum monthly energy production, where as for the CROG rule curve, the lowest monthly energy production occurred in July, that is, 16%. The maximum monthly energy production was equal to the maximum plant capacity multiplied by the average number of hours in a month.

As discussed previously in sub-section 5.3.1, that the Saguling Plant on an annual basis by applying the WRDG rule curve gave lower spills and produced higher energy. On a monthly basis, however, by applying the CROG rule curve the plant produced less variability in monthly energy.

Cirata and Jatiluhur Reservoirs

The analysis of the parameters of inflow, head, turbine flow, spill flow, and energy generation for the Cirata and Jatiluhur Hydropower Plants followed ·he same pattern as result of the analysis of the Saguling plant. It is important to note that inflows to the Cirata reservoir were set equal to the total outflow from Saguling plus local inflows from the Cirata drainage area. Inflows to the Jatiluhur Reservoir were set equal to the total outflows from Cirata plus local inflows from the Jatiluhur drainage area. The total outflow from the Saguling and Cirata Reservoirs were set equal to the turbine flow plus the spill flows governed by the rule curve used.

From the graphs of the average monthly energy generated for the Saguling Plant (Figure 5-1), Cirata Plant (Figure E-1 (5)) and Jatiluhur Plant (Figure E-1 (10)), and Citarum Hydropower system (Figure 5-6), it can be seen that the fluctuations of the monthly energy using the CROG rule curve were smaller than using the WRDG rule curve. Even though the plants generated a little higher annual energy by applying the WRDG rule curve, they produced less variability of monthly energy by applying the CROG rule curve.

Therefore the CROG rule curve might be preferred.

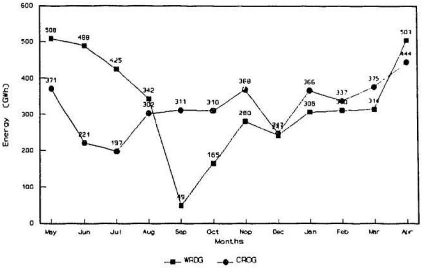


Figure 5-6: Energy generation of the Citarum Hydropower System

5.5 Load Factors

Load factor is defined as the ratio of the annually produced kilowatt-hours and of the energy theoretically producible at installed capacity during the whole year. The low load factor of a hydropower plant indicates that the installed capacity of the plant is larger than the available water. The installed capacity for the Saguling, Cirata and Jatiluhur Plants is 715 MW, 518 MW and 175 MW, respectively. If the current guideline was applied, the lcad factor for the Saguling Plant would be 2183 divide by (715 x 365.25 x 24

/ 1000), that is 35%. Load factors of the Saguling, Cirata and Jatiluhur Hydropower Plants applying the WRDG, CROG, ER and FR rule curves are presented in Table 5-2.

Table 5-2: Load factors of the Saguling, Cirata and Jatiluhur Plants

No	Rule Curves	Saguling Plant (%)	Cirata Plant (%)	Jatiluhur Plant (%)	Citarum Hydropower System (%)
1	WRDG	36	21	50	32
2	CROG	35	20	49	31
3	ER	36	20	42	31
4	FR	31	17	40	27

Table 5-2 shows that the load factor of the Citarum hydropower system by applying the four rule curves are not too much different. By applying the CROG rule curve, the load factors for the Saguling, Cirata and Jatiluhur Plants, are the 35%, 20% and 49%, respectively. These numbers indicate that the available water in the Citarum River might be smaller than the installed capacity of the Saguling, Cirata and Jatiluhur Plants.

5.6 Reliability Curves

The reliability curves of annual energy generation for the Saguling, Cirata and Jatiluhur Plants by applying the CROG and WRDG rule curves are presented in Appendix E. The 5% and 95% confidence limit reliability curves for the results of the simulation using the CROG rule curve are also presented. Figures 5-7 to 5-9 show the

5% and 95% confidence limit reliability curves of annual energy generation for the Saguling, Cirata and Jatiluhur Plants respectively. Figure 5-10 shows the 5% and 95% confidence limit reliability curves for the total annual energy generation of the Citarum Hydropower System.

The graphs include:

(1) reliability curves of annual energy generated using 60 year historical flows,

(2) reliability curves of annual energy using 100 replications of 60 year generated flows, and

(3) 5% and 95% confidence limit reliability curves of annual energy generation produced using synthetic flows.

The graphs indicate that the reliability curve simulated using historical flows fall within the 5% and 95% confidence limit curves. The comparisons of the estimated energy generated using assumed "normal year" flows, historical flows and synthetic flows are discussed in the following section.

5.7 Comparisons of Energy Estimates

The present procedure that is used to estimate energy generation uses 12 monthly flows of the average assumed "normal year". Hence, they estimate a *single point* of annual energy generation. For 1991 - 1992 these estimations were 2234 GWh for the Saguling plant, 1312.4 GWh for Cirata and 847.2 GWh for the

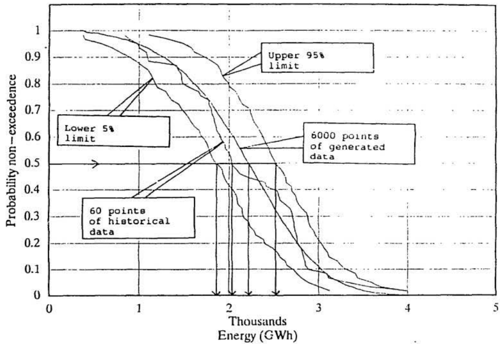


Figure 5-7: Energy generation of the Saguling Plant using the CROG rule curve

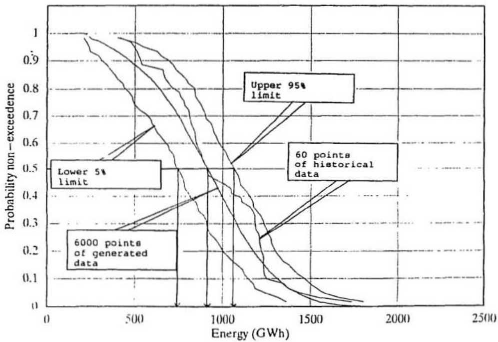


Figure 5-8: Energy generation of the Cirata Plant using the CROG rule curve

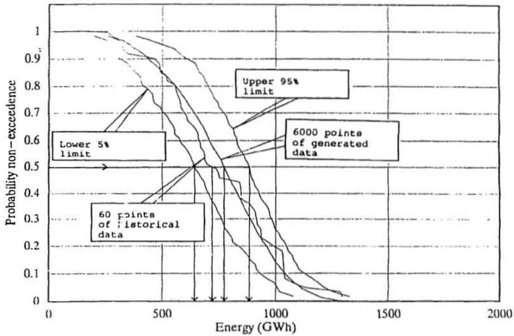


Figure 5-9: Energy generation of the Jatiluhur Plant using the CROG rule curve

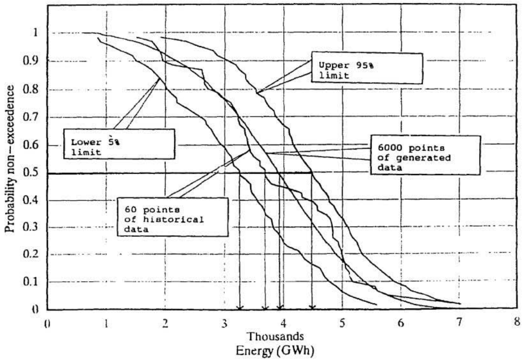


Figure 5-10: Total energy generation of the Citarum Hydropower Plant System using the CROG rule curve

Table 5-4: Comparisons of energy estimates using synthetic flows (energy in GWh)

Prob.non-exceedence	Confidence level	Saguling Plant	Cirata Plant	Jatiluhur Plant	Citarum Plant System
20%	5%	2400	1050	850	4300
	50%	2750	1200	950	4900
	95%	3000	1350	1100	5450
50%	5%	1900	750	650	3300
	50%	2200	900	750	3850
	95%	2500	1150	900	4550
80%	5%	1200	450	400	2050
	50%	1600	650	600	2850
	95%	2000	850	700	3550

Jatiluhur plant. Further, this procedure could not answer the question of the *probability* of getting these quantities of energy.

The estimates of the annual energy generated using the reliability curve from historical flows for the plants in the Citarum Hydropower System are presented in Table 5-3. The estimates of the annual energy generated using the reliability curve from synthetic flows are presented in Table 5-4. (See Figures 5-7 to 5-10).

The Figures 5-7 to 5-10 show that the reliability curve using synthetic flows is actually more informative than that using historical flows. A series of historical flows represents only one "event" in the past. Series of synthetic flows generated using the Monte Carlo method produces a large number of "events", from which probabilities of occurrence may be obtained. Therefore the reliability curve of energy generation using the graph based on the synthetic flows may indeed be expected to be more comprehensive than the reliability curve of energy generation based only on historical flows.

Table 5-3: Comparisons of energy estimates using historical flows (energy in GWh)

Prob.non-exceedence	Saguling Plant	Cirata Plant	Jatiluhur Plant	Citarum Plant System
20%	2750	1250	1000	5000
50%	2100	900	700	3700
80%	1500	750	600	2850

Chapter 6

Conclusions and Recommendations

This research has demonstrated the usefulness of reservoir simulation using synthetic flows. Two main sub-studies were carried out in this research. The first study involved stochastic flow modelling to generate synthetic monthly flows, that on average, preserved the statistical characteristics of both the historical annual and the monthly flows. The second study was a simulation of reservoir operation using the synthetic flows in conjunction with four rule curves. The four rule curves were: the current reservoir operation guideline (CROG), the water resources development guideline (WRDG) proposed by the Water Resources Development project, the empty reservoir (ER), and the full reservoir (FR) rule curves.

6.1 Conclusions

1. The most appropriate stochastic model for the annual flows of the Citarum River at Nanjung was found to be an Auto Regressive Moving Average (ARMA) 2,0 model.

2. The more appropriate stochastic model for the monthly flows

of the Citarum River at Nanjung was found to be the Two-Tier model. This model is a modification of the Thomas-Fiering model and preserved most of the statistical characteristics of both the historical annual and monthly flows.

3. The annual energy generated by the simulation applying the WRDG rule curve for the Saguling, Cirata and Jatiluhur Plants was greater than the annual energy generated using the CROG rule curve. Applying the WRDG rule curve, the energy estimated were 2225 GWh, 941 GWh and 764 GWh for the Saguling, Cirata and Jatiluhur Plants respectively. Applying the CROG rule curve, the energy estimates were 2183 GWh, 908 GWh and 758 GWh for the Saguling, Cirata and Jatiluhur Plants. However since it was found that the monthly energy generated by each plant applying the CROG rule curve was less variable than the monthly energy generated applying the WRDG rule curve, therefore the CROG rule curve might be preferred.

4. It was found that the load factors of the Citarum plants applying the four rule curves were not too much different. The load factors for the Saguling, Cirata and Jatiluhur plants by applying the CROG rule curve were 35%, 20% and 49%, respectively. These results indicate that the available water in the Citarum River was appeared to be

smaller than the installed capacity of the Saguling, Cirata and Jatiluhur Plants.

5. By applying the full reservoir rule curve, the energy estimates were 1932 GWh, 771 GWh and 608 GWh for the Saguling, Cirata and Jatiluhur Plants respectively. These results indicate that on average the Citarum Hydropower System still could produce 3312 GWh if the three reservoirs were kept full for all years.

6. The simulated reservoir operation, using both the synthetic and historical flows, together with the CROG rule curve showed that the results of the simulation using synthetic flows provided more information than that using only historical flows.

7. There are opportunities to improve energy production if the reservoirs were operated as a single system, for example, saving spills at the Jatiluhur Reservoir by storing water at the Cirata or the Saguling Reservoirs.

6.2 Recommendations

The following suggestions are made for future research on this reservoir system:

1. Further modification to the stochastic monthly flow model

should be made, so that all statistical characteristics of the historical flows can be reproduced.

2. The simulation of reservoir operations was only performed using two guidelines: the CROG and the WRDG rule curves. It is recommended that other variations of rule curves be examined by modifying the CROG and WRDG rule curves so that greater annual energy generation and smaller variation of monthly energy generation might be achieved.

3. The reservoir operation simulation performed herein only considered the energy generation, whereas in practice the Jatiluhur reservoir is used as a multipurpose dam. Further research is recommended to determine the best rule curves by considering the Jatiluhur reservoir as a *multipurpose* dam.

References

1. Adji, M. (1989). "Prospek Pengusahaan Waduk Kaskade Citarum Sampai 1990", *Rencana Operasi Sistem Terpadu Se-Jawa dan Bali 1989/90*, Perusahaan Umum Listrik Negara, Jakarta, pp. M-5.1-M-5.11

2. Box, G.E.P. and D.R. Cox (1964). "An Analysis of Transformation". *J.R. Statistical Society, Series B*. Vol. 26, pp. 211-252.

3. Directorate General of Water Resources Development (DGWRD), Ministry of Public Works, Republic of Indonesia, (1988). *Cisadane-Cimanuk Integrated Water Resources Development (BTA-155), Volume XIV: WRD Analysis Citarum Area*, Bandung, Indonesia, 160 p.

4. Harms, A.G. and Campbell, T. H., (1967). "An Extension to the Thomas-Fiering Model for the Sequential Generation of Streamflow". *Water Resources Research*. Vol. 3, pp. 653-660.

5. Hergert, D. (1989). *Microsoft Quickbasic Updated for Version 4.5: Developing Structured Program in the Microsoft QuickBASIC Environment*, Microsoft Press, Redmond, Washington, 449 p.

6. IRIS an Interactive River System Simulation Program: General Introduction and Description, *School of Civil and Environmental Engineering, Cornell University, Ithaca, New York, U.S.A. and International Institute for Applied System Analysis A-2361 Laxenburg, Austria, November 1989*, 101 p.

7. IRIS an Interactive River System Simulation Program: User's Manual Version 1.1, *School of Civil and Environmental Engineering, Cornell University, Ithaca, New York, U.S.A. and International Institute for Applied System Analysis A-2361 Laxenburg, Austria, January 1990*, 75 p.

8. Kananto, K., (1992). *Personal Communications*, St. John's, Canada.

9. Kuiper, E. (1971). *Water Resources Development, Planning, Engineering and Economics*, London, Butterworths, 483 p.

10. Loucks, D. P., Steidinger, J. R. and Haith, D. A. (1981). *Water Resource Systems Planning and Analysis*, Cornell

University, Pretice-Hall, Inc., Englewood Cliffs, New Jersey, 559 p.

11. Loucks, D. P. (1992). *Personal Communications*, Ithaca, New York.

12. Lye, L. M. (1992). "Selecting an Appropriate Box-Cox Transformation Using the Probability Plot Correlation Coefficient", *International Conference of Statistics in Public Resources and Utilities, and Care of the Environment*, Lisbon, Portugal, pp. 1-18.

13. Mc Mahon, T. A. and Mein, R. G. (1985). *Reservoir Capacity and Yield*, Department of Civil Engineering, Monash University, Clayton, Vic., Australia, Elsevier Scientific Publishing Company, Amsterdam - Oxford, New York, 213 p.

14. Minitab: Reference Manual Release 7, *Minitab Inc., Staten Island, NY, April, 1989*, pp. 12.1-12.8.

15. New Japan Engineering Consultants, (NEWJEC) Inc., (1988). *Cirata Hydroelectric Power Project, Optimum Operation of Three Reservoirs at Saguling, Cirata and Jatiluhur*, Bandung, 18 p.

16. Salas, J. D., Delleur, J. W., YevJevich, V. and Lane, W. L. (1988). *Applied Modelling of Hydrologic Times Series*, Water Resources Publications, Colorado, U.S.A., 484 p.

17. Sihombing, S., Marsudi, D. and Sakya, I. M. R. (1990). "Java Control Center Present State and Development", *Energi & Listrik*, Vol. 1-1990, pp. 52-85.

18. Suladjiono, S. (1990). "Pola Operasi Waduk Kaskade Citarum 1990", *Rapat Pelaksana Koordinasi Tata Pengaturan Sungai Citarum*, Sekretariat Pelaksana, Jakarta, Vol. 1-1990, pp. 1-7.

19. Suladjiono, S. (1991). "Pola Operasi Waduk Kaskade Citarum 1991", *Rapat Pelaksanaan Koordinasi Tata Pengaturan Sungai Citarum*, Sekretariat Pelaksana, Jakarta, Vol. 1-1991, pp. 1-4.

20. Thomas, H. H. and Fiering, M. B., (1966). "Mathematical Synthesis of Streamflow Sequences for the Analysis of River Basin by Simulation," *Design of Water-Resource Systems*, ed. Maass, A., Harvard University Press, Cambrige, Massachusetts, pp. 459-493.

Appendix - A

Historical Annual and Monthly Flows of the Citarum River at Nanjung

The historical annual and monthly flows of the Citarum River at Nanjung are presented in this appendix. There are 60 years of 12 monthly flows and 60 years of annual flows. The unit of flows is in million cubic metre.

Table A1 Historical annual and monthly flows of the Citarum River at Nanjung for periods of 1928–1987

Year	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	Annual Flows
1928	125.0	85.4	37.8	52.5	26.3	77.7	280.8	396.1	361.9	301.8	315.4	371.7	2432
1929	85.6	54.4	25.5	15.5	12.3	46.2	69.3	165.2	198.0	280.0	296.8	190.7	1440
1930	507.2	93.0	62.5	25.5	25.0	102.7	154.2	211.7	142.1	205.4	423.7	396.3	2349
1931	285.5	119.5	48.3	43.1	20.0	106.4	115.0	294.7	187.5	286.8	650.9	313.3	2471
1932	392.4	188.6	54.6	23.6	65.7	51.5	118.7	167.6	261.3	194.6	410.3	488.0	2417
1933	207.2	179.4	77.0	55.4	109.3	108.2	205.1	377.2	362.5	315.4	234.6	334.6	2566
1934	42.0	58.0	29.7	23.9	63.3	89.8	212.0	329.9	371.1	175.2	109.0	324.4	1828
1935	79.3	33.4	64.6	43.6	48.6	112.2	361.4	441.3	259.5	347.0	320.7	328.6	2440
1936	195.2	75.1	33.1	27.6	11.6	68.8	274.5	217.5	161.3	292.3	424.7	303.1	2085
1937	319.9	233.2	41.2	11.3	28.9	114.8	89.3	334.1	186.0	323.6	381.9	354.3	2419
1938	272.6	205.4	167.8	95.6	23.9	25.2	276.6	276.8	313.3	186.0	430.5	253.2	2527
1939	109.0	164.9	192.3	111.1	41.8	91.9	166.3	296.0	147.4	207.0	172.8	186.5	1887
1940	395.6	195.9	115.6	56.2	10.5	16.5	97.4	288.7	197.5	196.2	526.4	406.6	2503
1941	327.5	155.0	56.2	15.0	15.0	73.8	86.7	297.1	160.5	342.8	321.2	418.9	2270
1942	193.3	115.8	31.0	29.2	64.1	169.4	110.8	156.5	345.9	178.6	181.0	149.5	1725
1943	208.8	159.4	69.9	84.6	22.6	184.4	261.1	95.3	152.1	227.7	325.4	355.1	2146
1944	81.9	37.6	4.2	1.1	5.3	31.5	60.9	143.9	184.6	139.2	181.8	225.4	1097
1945	108.7	59.4	29.7	25.7	33.6	47.5	207.5	110.6	168.1	363.8	179.9	247.2	1582
1946	77.0	59.4	39.7	33.6	29.7	25.7	47.5	352.0	195.7	241.1	231.1	154.2	1487
1947	85.1	49.4	27.6	27.6	27.6	138.4	209.6	343.8	428.9	276.8	156.3	231.1	2002
1948	49.4	29.7	23.6	21.8	23.1	17.9	160.2	367.7	197.8	114.5	79.1	65.1	1150
1949	181.8	144.2	55.4	21.8	22.3	21.8	260.8	154.2	306.3	142.4	179.9	270.8	1762
1950	197.5	129.2	118.2	29.7	23.6	75.9	180.4	131.3	256.1	210.1	245.6	290.8	1888
1951	106.1	40.7	32.3	37.8	19.7	25.2	15.8	97.2	226.9	260.0	151.3	106.9	1120
1952	100.1	98.2	24.2	25.5	34.9	39.1	94.0	162.8	201.5	261.9	349.1	289.4	1681
1953	265.5	58.6	15.8	10.2	10.5	1.6	44.1	63.3	187.5	193.1	259.0	283.7	1393
1954	135.3	112.7	50.7	45.2	44.7	52.5	193.3	348.5	167.8	224.6	231.7	145.2	1752
1955	195.9	99.0	136.8	112.4	49.9	85.4	187.5	231.7	244.3	309.7	302.1	311.0	2266
1956	130.3	145.8	100.1	70.4	119.0	120.3	176.2	168.1	240.6	162.3	133.2	184.6	1751
1957	129.8	73.8	139.5	73.3	13.1	11.8	42.8	199.1	168.9	180.7	257.7	264.5	1555
1958	225.6	89.6	116.1	136.6	86.9	105.6	147.6	300.7	179.9	360.4	322.5	329.1	2401
1959	183.6	127.4	63.0	45.2	19.7	17.3	28.4	72.2	204.1	467.0	386.9	214.9	1830
1960	208.8	117.1	40.4	13.1	13.9	20.4	85.9	175.2	188.6	190.2	255.8	300.5	1610
1961	181.2	64.9	6.8	1.8	1.3	3.4	42.8	68.8	262.9	330.4	218.3	197.0	1379
1962	156.0	60.7	63.6	39.1	18.1	35.7	112.4	212.2	95.1	125.5	180.4	276.8	1376
1963	65.7	14.4	1.1	31.8	0.3	11.3	39.1	47.0	201.5	161.5	191.5	121.6	887
1964	138.9	63.3	33.1	19.4	29.4	81.9	110.1	132.1	99.8	80.1	123.7	169.4	1081
1965	44.9	60.7	27.8	5.8	1.8	2.9	33.1	185.7	200.4	232.7	129.2	125.3	1050
1966	151.3	95.9	37.6	9.7	13.1	48.9	108.0	182.8	281.0	282.6	384.0	170.7	1766
1967	123.7	13.9	1.6	0.0	2.1	8.7	41.8	131.3	228.0	121.1	113.5	223.5	1009
1968	387.9	361.2	303.6	348.0	170.2	98.8	221.4	328.3	366.4	178.3	388.2	372.7	3525
1969	137.6	128.7	25.7	6.8	98.0	70.9	193.3	139.5	506.7	465.4	312.8	417.6	2503
1970	379.5	185.4	62.8	16.8	33.9	81.7	333.8	269.2	276.3	209.6	600.2	349.6	2799
1971	192.8	95.1	43.1	23.6	14.4	185.4	368.0	437.3	334.6	292.9	181.5	333.8	2503
1972	192.3	21.5	8.4	11.8	0.5	0.8	84.6	191.7	372.4	341.2	373.2	275.5	1874
1973	448.9	160.0	107.4	65.4	136.8	99.8	205.4	281.8	305.2	332.0	353.0	436.0	2932
1974	287.9	53.8	95.9	123.4	91.1	331.2	321.8	350.9	117.7	208.3	99.3	425.0	2506
1975	218.3	131.3	73.5	55.4	103.5	300.2	417.4	231.4	438.6	420.8	404.5	363.3	3158
1976	122.1	38.9	13.7	27.6	21.3	57.8	230.3	249.5	234.0	166.0	194.6	276.6	1632
1977	213.5	263.7	55.4	35.7	44.4	11.8	90.9	196.5	237.2	354.6	333.0	308.6	2145
1978	208.0	270.8	270.8	157.1	91.9	83.5	203.0	395.6	212.3	116.1	363.0	155.8	2528
1979	321.2	135.3	55.9	60.9	104.3	116.1	168.9	345.7	199.6	343.6	183.1	433.1	2468
1980	131.1	47.0	55.4	74.1	47.3	95.1	211.2	358.0	180.7	70.4	169.4	249.0	1689
1981	194.6	109.5	114.8	64.8	57.0	72.5	153.5	139.2	123.2	116.1	315.2	229.8	1690
1982	104.0	62.8	37.3	23.9	20.0	16.3	25.7	242.2	256.6	264.0	202.8	406.6	1662
1983	278.4	79.1	35.5	3.9	0.8	92.2	322.8	213.3	301.0	414.5	469.9	316.0	2527
1984	303.1	85.1	52.3	56.7	246.1	188.8	184.1	271.6	246.1	369.8	235.3	304.9	2544
1985	102.2	156.5	119.2	51.5	112.9	184.1	130.5	255.8	261.3	230.1	96.9	99.8	1801
1986	172.0	178.2	157.1	117.7	152.9	165.2	321.5	211.7	264.8	214.3	466.0	341.5	2761
1987	152.1	113.5	48.1	23.9	24.2	56.7	109.8	281.8	221.2	199.1	320.4	269.0	1820

(Flows in million cubic metre)

Appendix - B

Statistical Characteristics of the Historical Annual and Monthly Flows of the Citarum River at Nanjung

The appendix gives the statistical characteristics of the historical annual and monthly flows of the Citarum River at Nanjung. The statistical characteristics of the flows were calculated using the MINITAB software package.

Table B: Statistical Characteristics of Historical Annual and Monthly Flows of the Citarum River at Nanjung

	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	Annual Flows
Minimum	42.0	13.9	1.1	0.0	0.3	0.8	15.8	47.0	95.1	70.4	79.1	65.1	886.7
Maximum	507.2	361.2	303.6	348.0	246.1	331.2	417.4	441.3	506.7	467.0	650.9	488.0	3525.1
Mean	193.6	110.6	67.2	48.4	46.7	79.7	163.4	235.3	239.7	246.7	280.5	279.0	1990.8
Standard deviation	105.1	68.6	59.5	53.1	48.5	67.8	98.6	99.7	85.5	93.7	126.7	97.2	568.8
Cs	0.90	1.15	1.93	3.25	1.78	1.49	0.53	0.11	0.86	0.34	0.62	−0.13	0.16
rk1	0.16	0.18	0.07	−0.03	0.22	0.32	0.18	0.19	0.16	−0.09	0.05	0.11	0.29
rk2	0.12	0.08	−0.06	−0.12	0.12	−0.04	0.18	0.25	0.12	0.08	0.11	0.26	0.41
r	0.54	0.70	0.87	0.55	0.56	0.64	0.47	0.16	0.39	0.32	0.45	0.12	–
K													0.86

(Flows in million cubic metre)

Appendix - C

Example of the PPCC Test

The PPCC tests were performed by selecting an appropriate Box-Cox transformation as shown in Table C-1, and testing of the correlation coefficient as shown in Table C-2. The PPCC test were done for the twelve series of the historical monthly flows. The example of the PPCC test presented in Table C-1 and Table C-2 was using historical monthly flows for the month of May.

Table C-1: Selecting an appropriate Box-Cox Transformation of historical monthly flows of the Citarum River at Nanjung. (Example: monthly flows in May)

Year	x	x_t	$(x_t-\bar{x}_t)^3$
1927	125.0	8.95	-1.1
1928	85.6	7.83	-10.2
1929	507.2	14.08	68.0
1930	285.5	11.77	5.6
1931	392.4	13.01	27.3
1932	207.2	10.61	0.2
1933	42.0	5.98	-65.2
1934	79.3	7.62	-13.5
1935	195.2	10.41	0.1
1936	319.9	12.21	10.7
1937	272.6	11.60	4.1
1938	109.0	8.54	-3.1
1939	395.6	13.05	28.2
1940	327.5	12.30	12.1
1941	193.3	10.37	0.1
1942	208.8	10.64	0.3
1943	81.9	7.71	-12.0
1944	108.7	8.53	-3.2
1945	77.0	7.53	-15.0
1946	85.1	7.82	-10.4
1947	49.4	6.37	-47.9
1948	181.8	10.16	0.0
1949	197.5	10.45	0.1
1950	106.1	8.46	-3.7
1951	100.1	8.28	-5.1
1952	265.5	11.50	3.4
1953	135.3	9.20	-0.5
1954	195.9	10.42	0.1
1955	130.3	9.08	-0.8
1956	129.8	9.07	-0.8
1957	225.6	10.91	0.8
1958	183.6	10.20	0.0
1959	208.8	10.64	0.3
1960	181.2	10.15	0.0
1961	156.0	9.66	-0.0
1962	65.7	7.10	-24.3
1963	138.9	9.28	-0.4
1964	44.9	6.14	-57.7
1965	151.3	9.56	-0.1
1966	123.7	8.92	-1.3
1967	387.9	12.97	26.1
1968	137.6	9.25	-0.4
1969	379.5	12.88	23.8
1970	192.8	10.36	0.0
1971	192.3	10.36	0.0
1972	448.9	13.57	45.3
1973	287.9	11.80	5.9
1974	218.3	10.80	0.5
1975	122.1	8.88	-1.4
1976	213.5	10.72	0.4
1977	208.0	10.63	0.2
1978	321.2	12.22	11.0
1979	131.1	9.10	-0.7
1980	194.6	10.40	0.1
1981	104.0	8.40	-4.1
1982	278.4	11.68	4.7
1983	303.1	12.00	8.0
1984	102.2	8.34	-4.5
1985	172.0	9.98	-0.0
1986	152.1	9.57	-0.1

$\lambda = 0.234$

$C_s = -0.00054$

Table C-2: Probability Plot Correlation Coefficient test of monthly flows of Citarum River at Nanjung (Example: monthly flows in May)

Rank	X_i	X_T	P_i	z_i
1	42.0	2.4	0.01	-2.32
2	44.9	2.4	0.03	-1.94
3	49.4	2.5	0.04	-1.72
4	65.7	2.7	0.06	-1.56
5	77.0	2.8	0.08	-1.43
6	79.3	2.8	0.09	-1.32
7	81.9	2.8	0.11	-1.23
8	85.1	2.8	0.13	-1.14
9	85.6	2.8	0.14	-1.06
10	100.1	2.9	0.16	-0.99
11	102.2	3.0	0.18	-0.93
12	104.0	3.0	0.19	-0.86
13	106.1	3.0	0.21	-0.80
14	108.7	3.0	0.23	-0.75
15	109.0	3.0	0.24	-0.69
16	122.1	3.1	0.26	-0.64
17	123.7	3.1	0.28	-0.59
18	125.0	3.1	0.29	-0.54
19	129.8	3.1	0.31	-0.50
20	130.3	3.1	0.33	-0.45
21	131.1	3.1	0.34	-0.40
22	135.3	3.2	0.36	-0.36
23	137.6	3.2	0.38	-0.32
24	138.9	3.2	0.39	-0.27
25	151.3	3.2	0.41	-0.23
26	152.1	3.2	0.43	-0.19
27	156.0	3.3	0.44	-0.15
28	172.0	3.3	0.46	-0.10
29	181.2	3.4	0.48	-0.06
30	181.8	3.4	0.49	-0.02
31	183.6	3.4	0.51	0.02
32	192.3	3.4	0.52	0.06
33	192.8	3.4	0.54	0.10
34	193.3	3.4	0.56	0.15
35	194.6	3.4	0.57	0.19
36	195.2	3.4	0.59	0.23
37	195.9	3.4	0.61	0.27
38	197.5	3.4	0.62	0.32
39	207.2	3.5	0.64	0.36
40	208.0	3.5	0.66	0.40
41	208.8	3.5	0.67	0.45
42	208.8	3.5	0.69	0.50
43	213.5	3.5	0.71	0.54
44	218.3	3.5	0.72	0.59
45	225.6	3.6	0.74	0.64
46	265.5	3.7	0.76	0.69
47	272.6	3.7	0.77	0.75
48	278.4	3.7	0.79	0.80
49	285.5	3.8	0.81	0.86
50	287.9	3.8	0.82	0.93
51	303.1	3.8	0.84	0.99
52	319.9	3.9	0.86	1.06
53	321.2	3.9	0.87	1.14
54	327.5	3.9	0.89	1.23
55	379.5	4.0	0.91	1.32
56	387.9	4.0	0.92	1.43
57	392.4	4.0	0.94	1.56
58	395.6	4.1	0.96	1.72
59	448.9	4.2	0.97	1.94
60	507.2	4.3	0.99	2.32

Regression Output:

Constant	-4.77249566
Std Err of Y Est	0.078679204
R Squared	0.993725581
No. of Observations	60
Degrees of Freedom	58
X Coefficient(s)	1.119658
Std Err of Coef.	0.011682

$r = (R\ Squared)^{0.5}$ 0.996858

At a significance level 5% and a number of data 60, the critical value of r is 0.980

Since calculated r was greater than r(5%), hence the monthly flows in May were fitted by the normal distribution.

Appendix - D

Minitab Listing Program

The appendix gives the Minitab listing program for analyzing the stochastic annual flow model, for transforming monthly flows and for calculating statistical characteristics of the transformed monthly flows. The computer coding for analyzing the time series model of the annual flows of the Citarum River at Nanjung is presented in Computer Coding D-1, for calculating statistical characteristics of the historical monthly flows is presented in Computer Coding D-2, and for transforming the monthly flows and calculating the statistical characteristics of transformed monthly flows are presented in Computer Coding D-3.

Computer Coding D-1

```
NOTE k3 = coefficient of skewness

OUTFILE 'thes-ann.out'
READ 'nan60.dat' into C1
PRINT C1

DESCRIBE C1
NSCORES of C1, put into C2
PLOT C1, C2.
CORRELATION C1 C2
ACF for series in C1, put into C3
PACF for series in C1, put into C4

LET C15 = (C1 - MEAN(C1))**3
LET k1 = (1/60) * SUM(C15)
LET k2 = (STDEV(C1))**3
LET k3 = k1/k2
PRINT k3

BRIEF 1

ARIMA 2 0 0 C1 [put resids in C5]
NSCORES of C5, put into C6
PLOT C5 C6
CORRELATION C5 C6

ARIMA 1 0 1 C1 [put resids in C7]
NSCORES of C7 C8
PLOT C7 C8
CORRELATION C7 C8

ARIMA 1 0 2 C1 [put resids in C9]
NSCORES of C9 C10
PLOT C9 C10
CORRELATION C9 C10

ARIMA 2 0 2  C1 [put resids in C11]
NSCORES of C11 C12
PLOT C11 C12
CORRELATION C11 C12

ARIMA 2 0 1 C1 [put resids in C13]
NSCORES of C13 C14
PLOT C13 C14
CORRELATION C13 C14

END
```

Computer Coding D-2

```
OUTFILE 'month-a.out'
READ 'may1.dat' into C1
LET C2 = (C1 - MEAN(C1))**3
LET k1 = (1/60) * SUM(C2)
LET k2 = (STDEV(C1))**3
LET k3 = k1/k2
PRINT k3

READ 'jun.dat' into C1
LET C2 = (C1 - MEAN(C1))**3
LET k1 = (1/60) * SUM(C2)
LET k2 = (STDEV(C1))**3
LET k3 = k1/k2
PRINT k3

READ 'jul.dat' into C1
LET C2 = (C1 - MEAN(C1))**3
LET k1 = (1/60) * SUM(C2)
LET k2 = (STDEV(C1))**3
LET k3 = k1/k2
PRINT k3

READ 'aug.dat' into C1
LET C2 = (C1 - MEAN(C1))**3
LET k1 = (1/60) * SUM(C2)
LET k2 = (STDEV(C1))**3
LET k3 = k1/k2
PRINT k3

READ 'sep.dat' into C1
LET C2 = (C1 - MEAN(C1))**3
LET k1 = (1/60) * SUM(C2)
LET k2 = (STDEV(C1))**3
LET k3 = k1/k2
PRINT k3

READ 'oct.dat' into C1
LET C2 = (C1 - MEAN(C1))**3
LET k1 = (1/60) * SUM(C2)
LET k2 = (STDEV(C1))**3
LET k3 = k1/k2
PRINT k3

READ 'nov.dat' into C1
LET C2 = (C1 - MEAN(C1))**3
LET k1 = (1/60) * SUM(C2)
LET k2 = (STDEV(C1))**3
LET k3 = k1/k2
PRINT k3
```

```
READ 'dec.dat' into C1
LET C2 = (C1 - MEAN(C1))**3
LET k1 = (1/60) * SUM(C2)
LET k2 = (STDEV(C1))**3
LET k3 = k1/k2
PRINT k3

READ 'jan.dat' into C1
LET C2 = (C1 - MEAN(C1))**3
LET k1 = (1/60) * SUM(C2)
LET k2 = (STDEV(C1))**3
LET k3 = k1/k2
PRINT k3

READ 'feb.dat' into C1
LET C2 = (C1 - MEAN(C1))**3
LET k1 = (1/60) * SUM(C2)
LET k2 = (STDEV(C1))**3
LET k3 = k1/k2
PRINT k3

READ 'mar.dat' into C1
LET C2 = (C1 - MEAN(C1))**3
LET k1 = (1/60) * SUM(C2)
LET k2 = (STDEV(C1))**3
LET k3 = k1/k2
PRINT k3

READ 'apr.dat' into C1
LET C2 = (C1 - MEAN(C1))**3
LET k1 = (1/60) * SUM(C2)
LET k2 = (STDEV(C1))**3
LET k3 = k1/k2
PRINT k3

Note Historical Monthly Flows
READ 'May1.dat' C1
READ 'Jun.dat' C2
READ 'Jul.dat' C3
READ 'Aug.dat' C4
READ 'Sep.dat' C5
READ 'Oct.dat' C6
READ 'Nov.dat' C7
READ 'Dec.dat' C8
READ 'Jan.dat' C9
READ 'Feb.dat' C10
READ 'Mar.dat' C11
READ 'Apr.dat' C12
READ 'May13.dat' C13

PRINT C1-C13
DESCRIBE C1-C13
```

```
ACF 2 for series in C1, put into C21
ACF 2 for series in C2, put into C22
ACF 2 for series in C3, put into C23
ACF 2 for series in C4, put into C24
ACF 2 for series in C5, put into C25
ACF 2 for series in C6, put into C26
ACF 2 for series in C7, put into C27
ACF 2 for series in C8, put into C28
ACF 2 for series in C9, put into C29
ACF 2 for series in C10, put into C30
ACF 2 for series in C11, put into C31
ACF 2 for series in C12, put into C32
ACF 2 for series in C13, put into C33

CORR C1-C13

END
```

Computer coding D-3

```
OUTFILE 'month-b.out'
Note May1.dat
READ 'may1.dat' into C1
LET C21 = C1**(0.234)
LET C2 = (C21 - MEAN(C21))**3
LET k1 = (1/60) * SUM(C2)
LET k2 = (STDEV(C1))**3
LET k3 = k1/k2
PRINT k3

READ 'jun.dat' into C1
NOTE Jun.dat
LET C22 = C1**(0.317)
LET C2 = (C22 - MEAN(C22))**3
LET k1 = (1/60) * SUM(C2)
LET k2 = (STDEV(C1))**3
LET k3 = k1/k2
PRINT k3

READ 'jul.dat' into C1
Note Jul.dat
LET C23 = C1**(0.2966)
LET C2 = (C23 - MEAN(C23))**3
LET k1 = (1/60) * SUM(C2)
LET k2 = (STDEV(C1))**3
LET k3 = k1/k2
PRINT k3

READ 'aug.dat' into C1
Note Aug.dat
LET C24 = C1**(0.303)
LET C2 = (C24 - MEAN(C24))**3
LET k1 = (1/60) * SUM(C2)
LET k2 = (STDEV(C1))**3
LET k3 = k1/k2
PRINT k3

READ 'sep.dat' into C1
Note Sep.dat
LET C25 = C1**(0.293)
LET C2 = (C25 - MEAN(C25))**3
LET k1 = (1/60) * SUM(C2)
LET k2 = (STDEV(C1))**3
LET k3 = k1/k2
PRINT k3

READ 'oct.dat' into C1
Note Oct.dat
LET C26 = C1**(0.405)
LET C2 = (C26 - MEAN(C26))**3
```

```
LET k1 = (1/60) * SUM(C2)
LET k2 = (STDEV(C1))**3
LET k3 = k1/k2
PRINT k3

READ 'nov.dat' into C1
Note Nov.dat
LET C27 = C1**(0.545)
LET C2 = (C27 - MEAN(C27))**3
LET k1 = (1/60) * SUM(C2)
LET k2 = (STDEV(C1))**3
LET k3 = k1/k2
PRINT k3
ACF 2 for series in C1, put into C2

READ 'dec.dat' into C1
Note Dec.dat
LET C28 = C1**(0.860)
LET C2 = (C28 - MEAN(C28))**3
LET k1 = (1/60) * SUM(C2)
LET k2 = (STDEV(C1))**3
LET k3 = k1/k2
PRINT k3

READ 'jan.dat' into C1
Note Jan.dat
LET C29 = C1**(0.073)
LET C2 = (C29 - MEAN(C29))**3
LET k1 = (1/60) * SUM(C2)
LET k2 = (STDEV(C1))**3
LET k3 = k1/k2
PRINT k3

READ 'feb.dat' into C1
Note Feb.dat
LET C30 = C1**(0.592)
LET C2 = (C30 - MEAN(C30))**3
LET k1 = (1/60) * SUM(C2)
LET k2 = (STDEV(C1))**3
LET k3 = k1/k2
PRINT k3

READ 'mar.dat' into C1
Note Mar.dat
LET C31 = C1**(0.360)
LET C2 = (C31 - MEAN(C31))**3
LET k1 = (1/60) * SUM(C2)
LET k2 = (STDEV(C1))**3
LET k3 = k1/k2
PRINT k3

READ 'apr.dat' into C1
```

```
Note Apr.dat
LET C32 = C1**(1.183)
LET C2 = (C32 - MEAN(C32))**3
LET k1 = (1/60) * SUM(C2)
LET k2 = (STDEV(C1))**3
LET k3 = k1/k2
PRINT k3

READ 'May13.dat' C13
Note May13.dat
LET C33 = C1**(0.234)
LET C2 = (C33 - MEAN(C33))**3
LET k1 = (1/60) * SUM(C2)
LET k2 = (STDEV(C1))**3
LET k3 = k1/k2
PRINT k3

Note Transformed Monthly Flows
PRINT C21-C33
DESCRIBE C21-C33

ACF 2 for series in C21, put into C41
ACF 2 for series in C22, put into C42
ACF 2 for series in C23, put into C43
ACF 2 for series in C24, put into C44
ACF 2 for series in C25, put into C45
ACF 2 for series in C26, put into C46
ACF 2 for series in C27, put into C47
ACF 2 for series in C28, put into C48
ACF 2 for series in C29, put into C49
ACF 2 for series in C30, put into C50
ACF 2 for series in C31, put into C51
ACF 2 for series in C32, put into C52
ACF 2 for series in C33, put into C53

CORR C21-C33

END
```

Appendix - E

Graphs Resulting from the Simulation of Reservoir Operations

The appendix gives graphs resulting from the simulation of reservoir operations using the CTR-SIMOD. The output graphs of the simulation using the WRDG and CROG rule curves are presented in Figures E-1 (1) to (10). The output graphs of the simulation using empty reservoir and full reservoir rule curves are presented in Figures E-2 (1) to (16). The reliability curve of the energy generations applying the WRDG and CROG rule curves are presented in Figures E-3 (1) to (4).

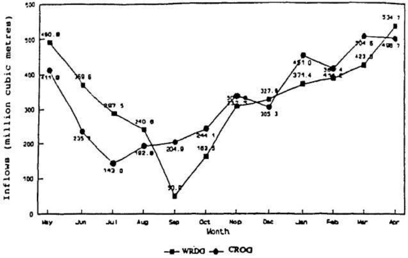


Figure E-1 (1): Inflows to the Cirata Reservoir

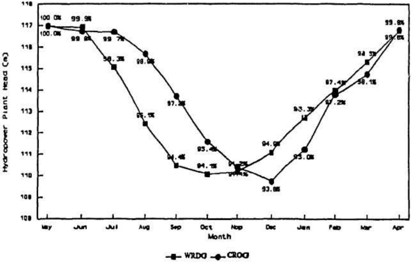


Figure E-1 (2): Heads on the Cirata Hydropower Plant

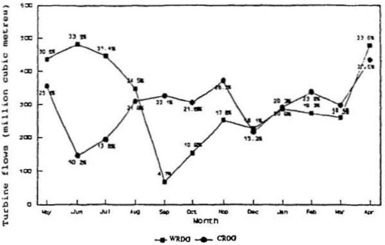


Figure E-1 (3): Turbine flows for the Cirata Hydropower Plant

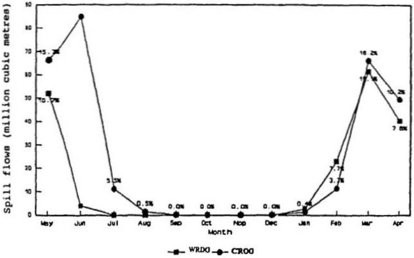


Figure E-1 (4): Spill flows from the Cirata Reservoir

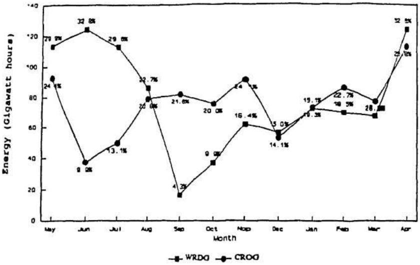


Figure E-1 (5): Energy Generation of the Cirata Hydropower Plant

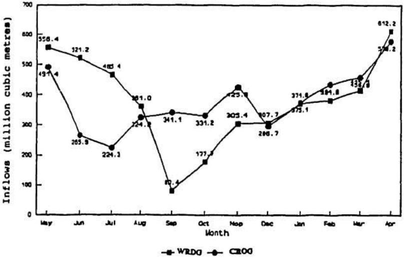


Figure E-1 (6): Inflows to the Jatiluhur Reservoir

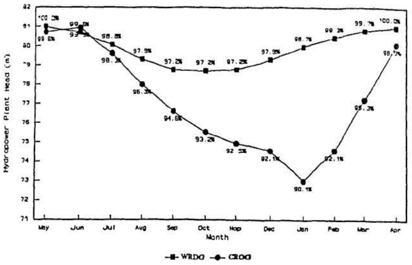


Figure E-1 (7): Heads on the Jatiluhur Hydropower Plant

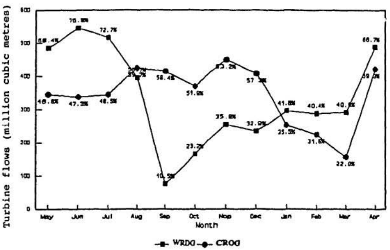


Figure E-1 (8): Turbine flows for the Jatiluhur Hydropower Plant

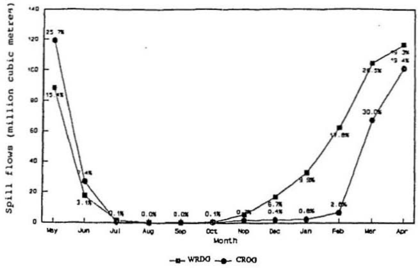


Figure E-1 (9): Spill flows from the Jatiluhur Reservoir

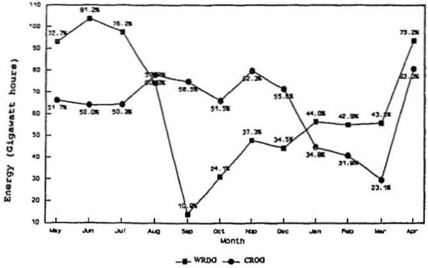


Figure E-1 (10): Energy Generation of the Jatiluhur Hydropower Plant

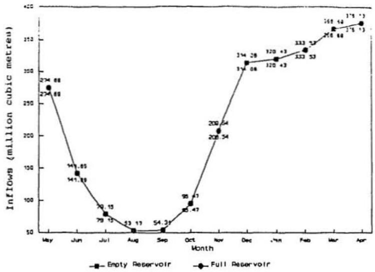


Figure E-2 (1): Inflows to the Saguling Reservoir

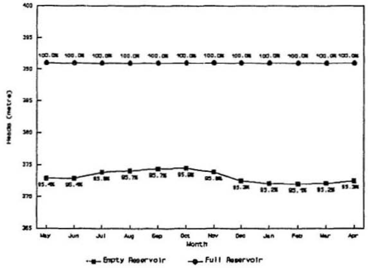


Figure E-2 (2): Heads on the Saguling Hydropower Plant

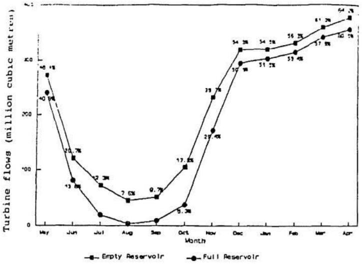


Figure E-2 (3): Turbine flows for the Saguling Hydropower Plant

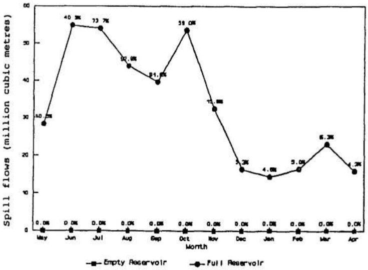


Figure E-2 (4): Spill flows from the Saquling Reservoir

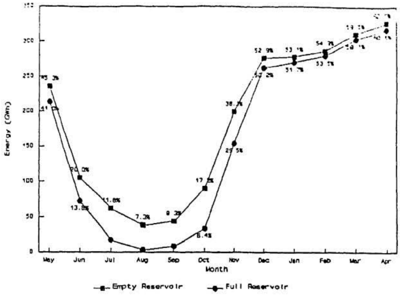


Figure E-2 (5): Energy Generation of the Saguling Hydropower Plant

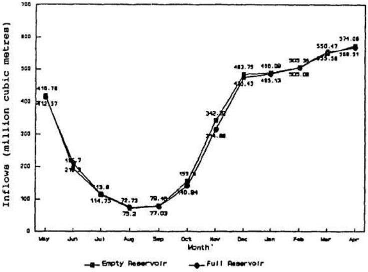


Figure E-2 (6): Inflows to the Cirata Reservoir

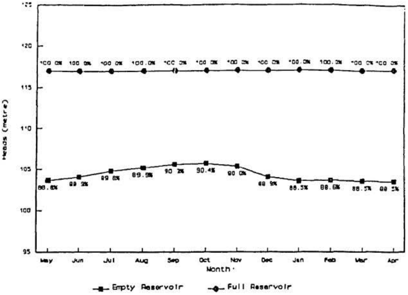


Figure E-2 (7): Heads on the Cirata Hydropower Plant

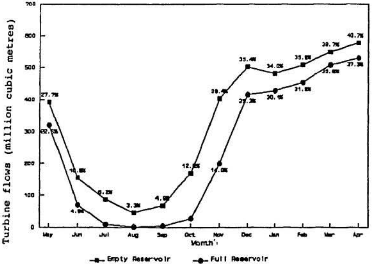


Figure E-2 (8): Turbine flows for the Cirata Hydropower Plant

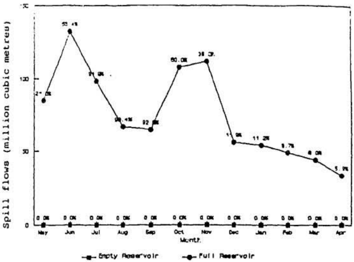


Figure E-2 (9): Spill flows from the Cirata Reservoir

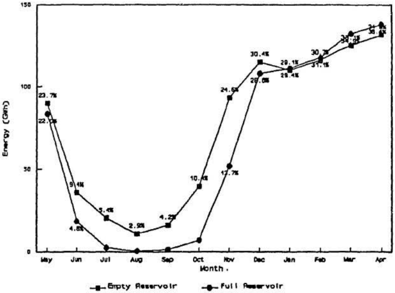


Figure E-2 (10): Energy Generation of the Cirata Hydropower Plant

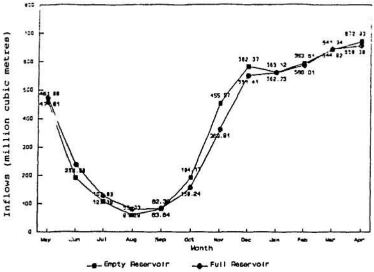


Figure E-2 (11): Inflows to the Jatiluhur Reservoir

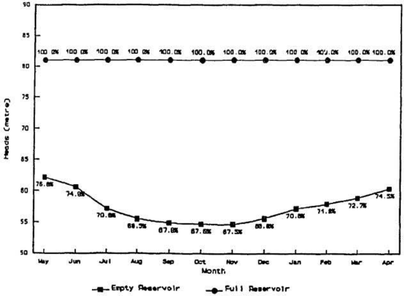


Figure E-2 (12): Heads on the Jatiluhur Hydropower Plant

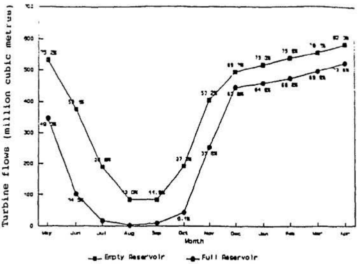


Figure E-2 (13): Turbine flows for the Jatiluhur Hydropower Plant

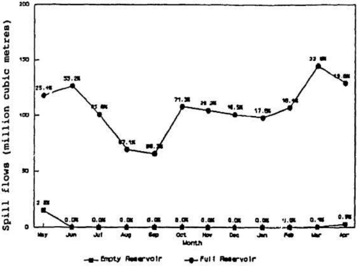


Figure E-2 (14): Spill flows from the Jatiluhur Reservoir

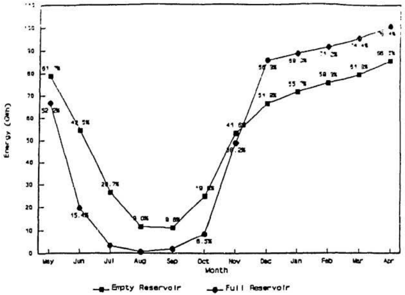


Figure E-2 (15): Energy Generation of the Jatiluhur Hydropower Plant

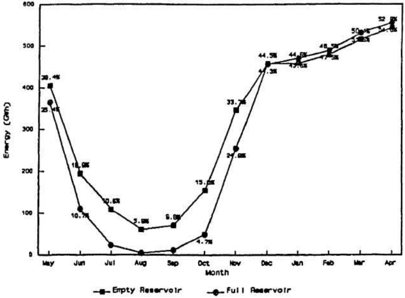


Figure E-2 (16): Total energy generation of the Citarum Hydropower Plant

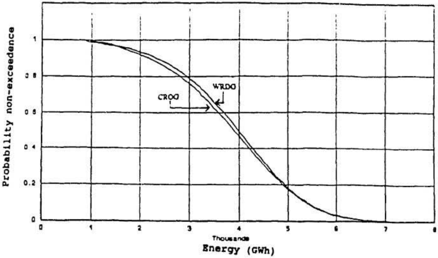


Figure E-3 (1): Total energy generations of the Citarum Hydropower Plant System using the CROG and WRDG rule curves

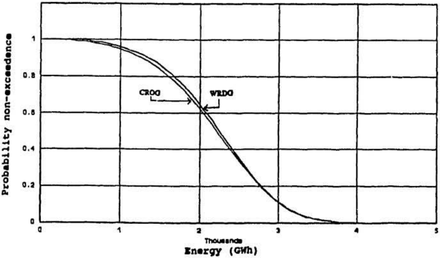


Figure E-3 (2): Energy generations of the Saguling Hydropower Plant using the CROG and WRDG rule curves

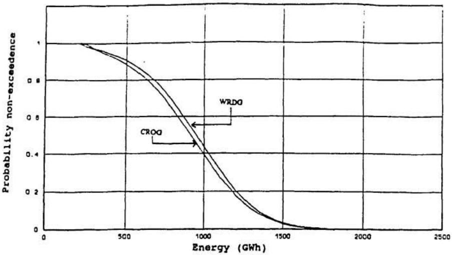


Figure E-3 (3): Energy generations of the Cirata Hydropower Plant using the CROG and WRDG rule curves

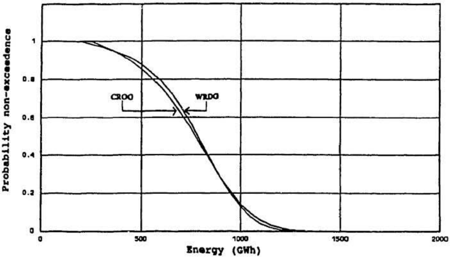


Figure E-3 (4): Energy generations of the Jatiluhur Hydropower Plant using the CROG and WRDG rule curves

Appendix - F

Listing Program of the Citarum Stochastic Model

(CTR-STOCHMOD)

The appendix gives the computer coding for generating synthetic monthly flows of the Citarum River at Nanjung. The stochastic model was the Two-Tier model. The program was written using Quickbasic software package. The inputs to the program were the statistical characteristics of the transformed monthly flows that had been put in the program. The output is a series of synthetic monthly flows of 100 (replications) of 60 (years) of 12 (months).

The stand alone EXE file could be put in a disk drive c and the directory \data and \output have to be created in the disk drive c. All input-output files and output files would be created automatically by the program in the directory \data and \output. The generated monthly flow file, that would be used in reservoir simulation, is a TierGQM.dat. The output test file for generated annual flows is a TierQann.out and for the generated monthly flows is a TierQMon.out.

```
'*******************************************************************
'*                       Main Module
'*******************************************************************
'$DYNAMIC
CLS
OPEN "c:\Output\TierNote.Out" FOR OUTPUT AS #22 'simulation Note
PRINT #22,
PRINT #22, "Output of Two-Tier Program"
PRINT #22, "Start at: ", DATE$, TIME$
PRINT #22,
LOCATE 3, 10
INPUT "Enter NRep.      = ", NRep
PRINT #22, "NRep. = ", NRep
OPEN "c:\data\NRepl.dat" FOR OUTPUT AS #1 'number of replications, years and months
OPEN "c:\data\TierQMon.dat" FOR OUTPUT AS #3 'input-output file
OPEN "c:\data\TierQAnn.dat" FOR OUTPUT AS #71 'input-output file
OPEN "c:\Output\TierTest.out" FOR OUTPUT AS #2 'output test file for replication No. 1

OPEN "c:\data\TierGQM.dat" FOR OUTPUT AS #72 'generated monthly flows
OPEN "c:\Output\TierQann.Out" FOR OUTPUT AS #102 'output test for generated annual flows
OPEN "c:\Output\TierQMon.out" FOR OUTPUT AS #202 'output test for generated monthly flows
LOCATE 7, 10
PRINT "Analysis of Two-Tier Model"
GOSUB Init.Stat.TT
FOR NR = 1 TO NRep
LOCATE 9, 10
PRINT "NRep.            = "; NR
OPEN "c:\data\QAnnAR2.dat" FOR OUTPUT AS #51 'input-output file

GOSUB Generate.Ann

CLOSE #51
OPEN "c:\data\SQMonth.dat" FOR OUTPUT AS #4 'input-output file
OPEN "c:\data\QMonth.dat" FOR OUTPUT AS #52 'input-output file

GOSUB Generate.Month
```

```
'NTTier.Bas

'***************************************************************************
'                         Description of variables
'***************************************************************************
'MTF       - mean of transformed flows
'STF       - standard deviation of transformed flows
'rTF       - coefficient of serial-correlation of transformed flows
'XGQanAR2 - generated annual flows using ARMA(2,0)
'b         - least squares regression coefficient
'XGQM      - generated monthly flows using the Thomas-Fiering Model
'XGQMonth - generated monthly flows using the Two-Tier Model
'NRep      - number of replications
'NYear     - number of year
'Month     - number of month
'Avg       - average
'Min       - minimum
'Max       - maximum
'Std       - standard deviation
'Cs        - coefficient of skewness
'rk1       - auto-correlation coefficient lag-1
'rk2       - auto-correlation coefficient lag-2
'H         - hurst coefficient
```

```
CLOSE #4
CLOSE #52
OPEN "c:\data\SQMonth.dat" FOR INPUT AS #60 'input-output file
OPEN "c:\data\QAnnAR2.dat" FOR INPUT AS #61 'input-output file
OPEN "c:\data\QMonth.dat" FOR INPUT AS #62 'input-output file

GOSUB Two.Tier

CLOSE #60
CLOSE #61
CLOSE #62
GOSUB Statistics.Calc.TT

NEXT NR

CLOSE #1
CLOSE #3
CLOSE #71

LOCATE 11, 10
PRINT "The program of TTier is finish"
LOCATE 15, 10

PRINT "TierQann.Bas"
OPEN "c:\data\TQanStat.dat" FOR OUTPUT AS #103 ' input-output file
OPEN "c:\data\TierQann.dat" FOR INPUT AS #107 ' input-output file

GOSUB Init.Stat.QAnn

FOR NR = 1 TO NRep
LOCATE 13, 10
PRINT "NRep.            = "; NR
GOSUB Statistics.Calc.QAnn
NEXT NR
CLOSE #103
OPEN "c:\data\TQanStat.dat" FOR INPUT AS #104 'input-output file
```

```
GOSUB Statistics.Test.Qann
LOCATE 15, 10
PRINT "The program of TierQann.Bas is finish"

CLOSE #107
CLOSE #104

LOCATE 17, 10
PRINT "Statistical tests of TierQMon.Bas"

OPEN "c:\data\TierQMon.dat" FOR INPUT AS #204 'input-output file
OPEN "c:\data\QHisMon.dat" FOR INPUT AS #215 'input-output file

GOSUB Init.Stat.Mon

GOSUB Statistics.Test.Mon

LOCATE 19, 10
PRINT "The program of TierQMon.Bas is finish"

CLOSE #204
CLOSE #215
CLOSE #2

CLOSE #72
CLOSE #102
CLOSE #202

LOCATE 25, 10
PRINT "The program of Two-Tier is completely finish"
PRINT #22,
PRINT #22, "The simulation is finish at NS: ", NS
PRINT #22, "Finish at: ", DATE$, TIME$
END
'*********************************************************
'*                      Initialization
'*********************************************************
```

```
Init.Stat.TT:
'$DYNAMIC

DIM YGQan(-1 TO 100), XGQan(-1 TO 100), XGQAnn(61), XGQanAR2(-1 TO 61)
DIM XGQM(13, 100), SXGQM(61), b(13), rTF(13), t(13), x(13, 61), MTF(15), STF(13)
DIM MinXGQMonth(13, 100), MaxXGQMonth(13, 100), XGQMonth(13, 60), HXGQMonth(13, 100)
DIM srk1Qana1(13, 100), srk1Qanb1(13, 100), rk1XGQMonth(13, 100), srk2Qana1(13, 100)
DIM srk2Qanb1(13, 100), rk2XGQMonth(13, 100), lmd(13)

DIM SXGQMonth(13, 100), AXGQMonth(13, 100), SDXGQMonthS(13, 100), SDXGQMonthC(13, 100)
DIM DXGQMonthS(13, 100), DXGQMonthC(13, 100), VXGQMonth(13, 100), StdXGQMonth(13, 100)
DIM CsXGQMonth(13, 100)

N = 60
WRITE #1, NRep

'+++++++Read data
'Mean of Transformed Flows = MTF
READ MTF(1), MTF(2), MTF(3), MTF(4), MTF(5), MTF(6), MTF(7), MTF(8), MTF(9), MTF(10),
MTF(11), MTF(12), MTF(13)
DATA 10.001, 10.292, 7.514, 6.362, 5.979, 10.719, 26.220, 124.770, 6.654, 41.560, 17.854,
669.000

'Standard deviation of Transformed Flows = STF
READ STF(1), STF(2), STF(3), STF(4), STF(5), STF(6), STF(7), STF(8), STF(9), STF(10),
STF(11), STF(12), STF(13)
DATA 1.864, 2.766, 2.948, 3.122, 3.194, 5.176, 10.050, 46.840, 0.523, 10.030, 3.479, 269.300

'Coef. of Correlation of Transformed Flows of February-January = rTF(1)
READ rTF(1), rTF(2), rTF(3), rTF(4), rTF(5), rTF(6), rTF(7), rTF(8), rTF(9), rTF(10),
rTF(11), rTF(12)
DATA 0.591, 0.732, 0.825, 0.654, 0.673, 0.695, 0.492, 0.158, 0.392, 0.377, 0.461, 0.975

'Lambda
READ lmd(1), lmd(2), lmd(3), lmd(4), lmd(5), lmd(6), lmd(7), lmd(8), lmd(9), lmd(10),
lmd(11), lmd(12), lmd(13)
DATA 0.234, 0.317, 0.297, 0.304, 0.293, 0.405, 0.545, 0.860, 0.073, 0.592, 0.360, 1.183, 0.234
```

```
RETURN

'***********************************************************************
'*                       Generate Annual Flows
'***********************************************************************
Generate.Ann:

IF NR = 1 THEN
PRINT #2, DATE$, TIME$
PRINT #2,
PRINT #2, "Agung W.H. SOEHARNO"
PRINT #2, "Output Program of Two-Tier Method"
PRINT #2,
PRINT #2, "Statistical Characteristics of Historical Annual Flow"
PRINT #2, "    QanMin   AvgQan   QanMax   StdQan     CsQan      rk1Qan     rk2Qan     HQan"
PRINT #2, USING "######.###"; QanMin; AvgQan; QanMax; StdQan; CsQan; rk1Qan; rk2Qan; HQan
PRINT #2,
PRINT #2, "Stochastic Model: ARMA(2,0)"
PRINT #2, "YGQan(NYear) = .1758 * YGQan(NYear - 1) + .3737 * YGQan(NYear - 2) + 943.87 +
(478.056) * t
XGQanAR2(NYear) = YGQan(NYear)
PRINT #2,
PRINT #2, "XGQanAR2(NYear) = YGQan(NYear)"
ELSE
END IF

XGQanAR2(0) = AvgQan:  XGQanAR2(-1) = AvgQan


FOR NYear = 1 TO N
10 GOSUB random.number
t = z
YGQan(NYear) = .1758 * YGQan(NYear - 1) + .3737 * YGQan(NYear - 2) + 943.87 + (478.056) * t
XGQanAR2(NYear) = YGQan(NYear)

IF XGQanAR2(NYear) < 0 THEN
```

```
GOTO 10
ELSE
END IF

WRITE #51, XGQanAR2(NYear)

IF NR = 1 AND NYear = 1 THEN
PRINT #2,
PRINT #2, "Output test for Number of Replication = 1"
PRINT #2, "    NYear      XGQanAR2(NYear)"


ELSE
END IF

IF NR = 1 THEN
PRINT #2, USING "########.###"; NYear; XGQanAR2(NYear)

ELSE
END IF
NEXT NYear

RETURN
'**************************************************************************
'*                        b Calculation
'**************************************************************************
b.Calculation:
        FOR Month = 1 TO 12
           b(Month) = rTF(Month) * (STF(Month + 1) / STF(Month))
        NEXT Month
RETURN
'**************************************************************************
'*                        Generate Monthly Flows
'**************************************************************************
Generate.Month:
'x(Month=Jan, NYear=1) = Mean Transformed Flow (Jan)
```

```
 x(1, 1) = MTF(1)
'MTH(Month=13) = MTF(Month=1)
 MTF(13) = MTF(1)
'STF(Month=13) = STF(Month=1)
 STF(13) = STF(1)

FOR NYear = 1 TO N
FOR Month = 1 TO 12
20 GOSUB random.number
t(Month) = z
x(Month + 1, NYear) = MTF(Month + 1) + b(Month) * (x(Month, NYear) - MTF(Month)) + t(Month)
* STF(Month + 1) * ((1 - rTF(Month) ^ 2) ^ .5)
IF x(Month + 1, NYear) < 0 THEN
GOTO 20
ELSE
END IF

SELECT CASE Month + 1
CASE 2
XGQM(Month + 1, NYear) = (x(Month + 1, NYear) * .317 + 1) ^ (1 / .317)
CASE 3
XGQM(Month + 1, NYear) = (x(Month + 1, NYear) * .297 + 1) ^ (1 / .297)
CASE 4
XGQM(Month + 1, NYear) = (x(Month + 1, NYear) * .304 + 1) ^ (1 / .304)
CASE 5
XGQM(Month + 1, NYear) = (x(Month + 1, NYear) * .293 + 1) ^ (1 / .293)
CASE 6
XGQM(Month + 1, NYear) = (x(Month + 1, NYear) * .405 + 1) ^ (1 / .405)
CASE 7
XGQM(Month + 1, NYear) = (x(Month + 1, NYear) * .545 + 1) ^ (1 / .545)
CASE 8
XGQM(Month + 1, NYear) = (x(Month + 1, NYear) * .86 + 1) ^ (1 / .86)
CASE 9
XGQM(Month + 1, NYear) = (x(Month + 1, NYear) * .073 + 1) ^ (1 / .073)
CASE 10
XGQM(Month + 1, NYear) = (x(Month + 1, NYear) * .592 + 1) ^ (1 / .592)
CASE 11
```

```
XGQM(Month + 1, NYear) = (x(Month + 1, NYear) * .36 + 1) ^ (1 / .36)
CASE 12
XGQM(Month + 1, NYear) = (x(Month + 1, NYear) * 1.183 + 1) ^ (1 / 1.183)
CASE 13
XGQM(Month + 1, NYear) = (x(Month + 1, NYear) * .234 + 1) ^ (1 / .234)
END SELECT
NEXT Month
x(1, NYear + 1) = x(13, NYear)
NEXT NYear
MTF(1) = 10.001

XGQM(1, 1) = ((10.001) * .234) + 1) ^ (1 / .234)

FOR NYear = 2 TO N
XGQM(1, NYear) = XGQM(13, NYear - 1)
NEXT NYear

FOR NYear = 1 TO N
FOR Month = 1 TO 12
WRITE #52, XGQM(Month, NYear)
NEXT Month
NEXT NYear
FOR NYear = 1 TO N
SXGQM(NYear) = 0
NEXT NYear
FOR NYear = 1 TO N
FOR Month = 1 TO 12
SXGQM(NYear) = SXGQM(NYear) + XGQM(Month, NYear)
NEXT Month
WRITE #4, SXGQM(NYear)
NEXT NYear
RETURN
'*******************************************************
'*              Two-Tier Model
'*******************************************************
Two.Tier:
FOR NYear = 1 TO N
```

```
INPUT #61, XGQanAR2(NYear)
INPUT #60, SXGQM(NYear)
FOR Month = 1 TO 12
INPUT #62, XGQM(Month, NYear)
NEXT Month
NEXT NYear
FOR NYear = 1 TO N
FOR Month = 1 TO 12
XGQMonth(Month, NYear) = (XGQM(Month, NYear) / SXGQM(NYear)) * XGQanAR2(NYear)
WRITE #72, NR, NYear, XGQMonth(Month, NYear)
NEXT Month
NEXT NYear
FOR NYear = 1 TO N
XGQAnn(NYear) = 0
NEXT NYear
FOR NYear = 1 TO N
FOR Month = 1 TO 12
XGQAnn(NYear) = XGQAnn(NYear) + XGQMonth(Month, NYear)
NEXT Month
WRITE #71, XGQAnn(NYear)
NEXT NYear
RETURN
'*******************************************************
'*               Soubroutine Random Number
'*******************************************************
random.number:
        Sum = 0
        RANDOMIZE TIMER
     FOR L = 1 TO 12
        u = RND
        Sum = u + Sum
     NEXT L
        z = Sum - 6
RETURN
'***************************************************************************
'                        Statistics.Calc
'***************************************************************************
```

```
Statistics.Calc.TT:
GOSUB MaxMin.Calc.TT
GOSUB Avg.Calc.TT
GOSUB Sum.Calc.TT
GOSUB Std.Calc.TT
GOSUB Cs.Calc.TT
GOSUB rk1QMonth.TT
GOSUB rk2QMonth.TT
GOSUB Print.Stat.TT
RETURN
'----------------- Max and Min
MaxMin.Calc.TT:
FOR Month = 1 TO 12
MinXGQMonth(Month, NR) = 10000
MaxXGQMonth(Month, NR) = 0
NEXT Month
FOR NYear = 1 TO N
FOR Month = 1 TO 12
IF XGQMonth(Month, NYear) < MinXGQMonth(Month, NR) THEN
MinXGQMonth(Month, NR) = XGQMonth(Month, NYear)
ELSE
END IF
IF XGQMonth(Month, NYear) > MaxXGQMonth(Month, NR) THEN
MaxXGQMonth(Month, NR) = XGQMonth(Month, NYear)
ELSE
END IF
NEXT Month
NEXT NYear
RETURN
'------------------------------ Avg.Calc
Avg.Calc.TT:
'---------- Average of XGQMonth
FOR Month = 1 TO 12
SXGQMonth(Month, NR) = 0
NEXT Month
FOR NYear = 1 TO N
FOR Month = 1 TO 12
```

```
SXGQMonth(Month, NR) = SXGQMonth(Month, NR) + XGQMonth(Month, NYear)
NEXT Month
NEXT NYear
FOR Month = 1 TO 12
AXGQMonth(Month, NR) = SXGQMonth(Month, NR) / N
NEXT Month
RETURN
'------------------------------Sumation
Sum.Calc.TT:
FOR Month = 1 TO 12
SDXGQMonthS(Month, NR) = 0
SDXGQMonthC(Month, NR) = 0
NEXT Month
FOR NYear = 1 TO N
FOR Month = 1 TO 12
DXGQMonthS(Month, NR) = (XGQMonth(Month, NYear) - AXGQMonth(Month, NR)) ^ 2
SDXGQMonthS(Month, NR) = SDXGQMonthS(Month, NR) + DXGQMonthS(Month, NR)
DXGQMonthC(Month, NR) = (XGQMonth(Month, NYear) - AXGQMonth(Month, NR)) ^ 3
SDXGQMonthC(Month, NR) = SDXGQMonthC(Month, NR) + DXGQMonthC(Month, NR)
NEXT Month
NEXT NYear
RETURN
'--------------------------------StdXGQMonth
Std.Calc.TT:
FOR Month = 1 TO 12
VXGQMonth(Month, NR) = SDXGQMonthS(Month, NR) / (N - 1)
StdXGQMonth(Month, NR) = (VXGQMonth(Month, NR)) ^ .5
NEXT Month
RETURN
'----------------------------------CsXGQMonth
Cs.Calc.TT:
FOR Month = 1 TO 12
an = N / ((N - 1) * (N - 2))
a = an * SDXGQMonthC(Month, NR)
CsXGQMonth(Month, NR) = a / (StdXGQMonth(Month, NR)) ^ 3
NEXT Month
RETURN
```

```
'-------------- rk1QMonth
rk1QMonth.TT:
FOR Month = 1 TO 12
srk1Qana1(Month, NR) = 0
srk1Qanb1(Month, NR) = 0
NEXT Month
FOR NYear = 1 TO N - 1
FOR Month = 1 TO 12
srk1Qana1(Month, NR) = srk1Qana1(Month, NR) + (XGQMonth(Month, NYear) - AXGQMonth(Month, NR))
* (XGQMonth(Month + 1, NYear) - AXGQMonth(Month, NR))
NEXT Month
NEXT NYear
FOR NYear = 1 TO N
FOR Month = 1 TO 12
srk1Qanb1(Month, NR) = srk1Qanb1(Month, NR) + (XGQMonth(Month, NYear) - AXGQMonth(Month, NR))
^ 2
NEXT Month
NEXT NYear
FOR Month = 1 TO 12
rk1XGQMonth(Month, NR) = srk1Qana1(Month, NR) / srk1Qanb1(Month, NR)
NEXT Month
RETURN
'-------------- rk2QMonth
rk2QMonth.TT:
FOR Month = 1 TO 12
srk2Qana1(Month, NR) = 0
srk2Qanb1(Month, NR) = 0
NEXT Month
FOR NYear = 1 TO N - 2
FOR Month = 1 TO 12
srk2Qana1(Month, NR) = srk2Qana1(Month, NR) + (XGQMonth(Month, NYear) - AXGQMonth(Month, NR))
* (XGQMonth(Month + 2, NYear) - AXGQMonth(Month, NR))
NEXT Month
NEXT NYear
FOR NYear = 1 TO N
FOR Month = 1 TO 12
srk2Qanb1(Month, NR) = srk2Qanb1(Month, NR) + (XGQMonth(Month, NYear) - AXGQMonth(Month, NR))
```

```
^ 2
NEXT Month
NEXT NYear
FOR Month = 1 TO 12
rk2XGQMonth(Month, NR) = srk2Qana1(Month, NR) / srk2Qanb1(Month, NR)
NEXT Month
RETURN
'--------------Print.Stat
Print.Stat.TT:
IF NR = 1 THEN
PRINT #2, DATE$, TIME$
PRINT #2, "Agung W.H. SOEHARNO"
PRINT #2, "Output of Two - Tier Model"
PRINT #2,
PRINT #2, "Note: Month = 1 is May etc."
PRINT #2,
PRINT #2, "  NR        Month  Min     Avg     Max     Std.Dev    Cs      rk1      rk2"
ELSE
END IF
FOR Month = 1 TO 12
PRINT #2, USING "####.###"; NR; Month; MinXGQMonth(Month, NR); AXGQMonth(Month, NR);
MaxXGQMonth(Month, NR); StdXGQMonth(Month, NR); CsXGQMonth(Month, NR); rk1XGQMonth(Month,
NR); rk2XGQMonth(Month, NR)
WRITE #3, NR, Month, MinXGQMonth(Month, NR), AXGQMonth(Month, NR), MaxXGQMonth(Month, NR),
StdXGQMonth(Month, NR), CsXGQMonth(Month, NR), rk1XGQMonth(Month, NR), rk2XGQMonth(Month, NR)
NEXT Month
RETURN

' TierQann
'*******************************************************
'*                      Initialization
'*******************************************************
Init.Stat.QAnn:
'$DYNAMIC
DIM DH(62), SDH(62), SDHMin(100), SDHMax(100), RnH(100), HXGQan(100)
DIM srk1Qana2(100), srk1Qanb2(100), rk1(100), srk2Qana2(100), srk2Qanb2(100)
DIM rk2(100), MinXGQan(100), MaxXGQan(100), SXGQan(100), AXGQan(100)
```

```
DIM SDXGQanS(100), SDXGQanC(100), DXGQanS(100), DXGQanC(100), VXGQan(100)
DIM StdXGQan(100), CsXGQan(100)

N = 60
AvgQan = 1990.775
StdQan = 573.597
CsQan = .169
rk1Qan = .289
rk2Qan = .406
HQan = .868
MinQan = 886.7
MaxQan = 3525.1
RETURN
'**************************************************************************
'                         Statistics.Calc
'**************************************************************************
Statistics.Calc.QAnn:
FOR NYear = 1 TO N
INPUT #107, XGQan(NYear)
NEXT NYear
GOSUB MaxMin.Calc.QAnn
GOSUB Avg.Calc.QAnn
GOSUB Sum.Calc.QAnn
GOSUB Std.Calc.QAnn
GOSUB Cs.Calc.QAnn
GOSUB rk1Qan.QAnn
GOSUB rk2Qan.QAnn
GOSUB Hurst.QAnn
GOSUB Print.Stat.QAnn
RETURN
'------------------ Max and Min
MaxMin.Calc.QAnn:
Min = 10000
Max = 10
FOR NYear = 1 TO N
IF XGQan(NYear) < Min THEN
Min = XGQan(NYear)
```

```
ELSE
END IF
IF XGQan(NYear) > Max THEN
Max = XGQan(NYear)
ELSE
END IF
NEXT NYear
MinXGQan(NR) = Min
MaxXGQan(NR) = Max
RETURN
'------------------------------ Avg.Calc
Avg.Calc.QAnn:
'---------- Average of XGQan
SXGQan(NR) = 0
FOR NYear = 1 TO N
SXGQan(NR) = SXGQan(NR) + XGQan(NYear)
NEXT NYear
AXGQan(NR) = SXGQan(NR) / N
RETURN
'------------------------------Sumation
Sum.Calc.QAnn:
SDXGQanS(NR) = 0
SDXGQanC(NR) = 0
FOR NYear = 1 TO N
DXGQanS(NR) = (XGQan(NYear) - AXGQan(NR)) ^ 2
SDXGQanS(NR) = SDXGQanS(NR) + DXGQanS(NR)
DXGQanC(NR) = (XGQan(NYear) - AXGQan(NR)) ^ 3
SDXGQanC(NR) = SDXGQanC(NR) + DXGQanC(NR)
NEXT NYear
RETURN
'--------------------------------StdXGQan
Std.Calc.QAnn:
VXGQan(NR) = SDXGQanS(NR) / (N - 1)
StdXGQan(NR) = (VXGQan(NR)) ^ .5
RETURN
'----------------------------------CsXGQan
Cs.Calc.QAnn:
```

```
an = N / ((N - 1) * (N - 2))
a = an * SDXGQanC(NR)
CsXGQan(NR) = a / (StdXGQan(NR)) ^ 3
RETURN
'-------------- rk1Qan
rk1Qan.QAnn:
srk1Qana2(NR) = 0
srk1Qanb2(NR) = 0
FOR NYear = 1 TO N - 1
srk1Qana2(NR) = srk1Qana2(NR) + (XGQan(NYear) - AXGQan(NR)) * (XGQan(NYear + 1) - AXGQan(NR))
NEXT NYear
FOR NYear = 1 TO N
srk1Qanb2(NR) = srk1Qanb2(NR) + (XGQan(NYear) - AXGQan(NR)) ^ 2
NEXT NYear
rk1(NR) = srk1Qana2(NR) / srk1Qanb2(NR)
RETURN
'-------------- rk2Qan
rk2Qan.QAnn:
srk2Qana2(NR) = 0
srk2Qanb2(NR) = 0
FOR NYear = 1 TO N - 2
srk2Qana2(NR) = srk2Qana2(NR) + (XGQan(NYear) - AXGQan(NR)) * (XGQan(NYear + 2) - AXGQan(NR))
NEXT NYear
FOR NYear = 1 TO N
srk2Qanb2(NR) = srk2Qanb2(NR) + (XGQan(NYear) - AXGQan(NR)) ^ 2
NEXT NYear
rk2(NR) = srk2Qana2(NR) / srk2Qanb2(NR)
RETURN
'--------------Hurst Coefficients.
Hurst.QAnn:
SDH(0) = 0
FOR NYear = 1 TO N
DH(NYear) = (XGQan(NYear) - AXGQan(NR))
SDH(NYear) = SDH(NYear - 1) + DH(NYear)
NEXT NYear
Min = 10000
Max = 10
```

```
FOR NYear = 1 TO N
IF SDH(NYear) < Min THEN
Min = SDH(NYear)
ELSE
END IF
IF SDH(NYear) > Max THEN
Max = SDH(NYear)
ELSE
END IF
NEXT NYear
SDHMin(NR) = Min
SDHMax(NR) = Max
RnH(NR) = SDHMax(NR) - SDHMin(NR)
HXGQan(NR) = (LOG(RnH(NR) / StdXGQan(NR)) / (LOG(N / 2)))
RETURN
'--------------Print.Stat
Print.Stat.QAnn:
IF NR = 1 THEN
PRINT #102, DATE$, TIME$
PRINT #102, "Output of TierQann.Bas"
PRINT #102,
PRINT #102,
PRINT #102, "   NR     MinXGQan AvgXGQan MaxXGQan StdXGQan CsXGQan    rk1     rk2     HXGQan"
ELSE
END IF
PRINT #102, USING "#####.###"; NR; MinXGQan(NR); AXGQan(NR); MaxXGQan(NR); StdXGQan(NR);
CsXGQan(NR); rk1(NR); rk2(NR); HXGQan(NR)
WRITE #103, NR, MinXGQan(NR), AXGQan(NR), MaxXGQan(NR), StdXGQan(NR), CsXGQan(NR), rk1(NR),
rk2(NR), HXGQan(NR)
RETURN
'**********************************************************************
'                        Statistics.Test
'**********************************************************************
Statistics.Test.Qann:
CLOSE #103
GOSUB Input.Data.QAnn
GOSUB AvgAvg.Calc.QAnn
```

```
GOSUB SumSum.Calc.QAnn
GOSUB StdStd.Calc.QAnn
GOSUB Confidence.Test.QAnn
GOSUB PrintStat.Test.QAnn
RETURN
'------------------------------ Input.Data
Input.Data.QAnn:
FOR NR = 1 TO NRep
INPUT #104, NR, MinXGQan(NR), AXGQan(NR), MaxXGQan(NR), StdXGQan(NR), CsXGQan(NR), rk1(NR),
rk2(NR), HXGQan(NR)
NEXT NR
RETURN
'------------------------------ AvgAvg.Calc
AvgAvg.Calc.QAnn:
'---------- Average of XGQan
SAXGQan = 0
SStdXGQan = 0
SCsXGQan = 0
Srk1 = 0
Srk2 = 0
SHXGQan = 0
SMinXGQan = 0
SMaxXGQan = 0
FOR NR = 1 TO NRep
SAXGQan = SAXGQan + AXGQan(NR)
SStdXGQan = SStdXGQan + StdXGQan(NR)
SCsXGQan = SCsXGQan + CsXGQan(NR)
Srk1 = Srk1 + rk1(NR)
Srk2 = Srk2 + rk2(NR)
SHXGQan = SHXGQan + HXGQan(NR)
SMinXGQan = SMinXGQan + MinXGQan(NR)
SMaxXGQan = SMaxXGQan + MaxXGQan(NR)
NEXT NR
AAXGQan = SAXGQan / NRep
AStdXGQan = SStdXGQan / NRep
ACsXGQan = SCsXGQan / NRep
Ark1 = Srk1 / NRep
```

```
Ark2 = Srk2 / NRep
AHXGQan = SHXGQan / NRep
AMinXGQan = SMinXGQan / NRep
AMaxXGQan = SMaxXGQan / NRep
RETURN
'------------------------------Sumation of Sumation
SumSum.Calc.QAnn:
SDAXGQanS = 0
SDStdXGQanS = 0
SDCsXGQanS = 0
SDrk1S = 0
SDrk2S = 0
SDHXGQanS = 0
SDMinXGQanS = 0
SDMaxXGQanS = 0
FOR NR = 1 TO NRep
DAXGQanS = (AXGQan(NR) - AAXGQan) ^ 2
SDAXGQanS = SDAXGQanS + DAXGQanS
DStdXGQanS = (StdXGQan(NR) - AStdXGQan) ^ 2
SDStdXGQanS = SDStdXGQanS + DStdXGQanS
DCsXGQanS = (CsXGQan(NR) - ACsXGQan) ^ 2
SDCsXGQanS = SDCsXGQanS + DCsXGQanS
Drk1S = (rk1(NR) - Ark1) ^ 2
SDrk1S = SDrk1S + Drk1S
Drk2S = (rk2(NR) - Ark2) ^ 2
SDrk2S = SDrk2S + Drk2S
DHXGQanS = (HXGQan(NR) - AHXGQan) ^ 2
SDHXGQanS = SDHXGQanS + DHXGQanS
DMinXGQanS = (MinXGQan(NR) - AMinXGQan) ^ 2
SDMinXGQanS = SDMinXGQanS + DMinXGQanS
DMaxXGQanS = (MaxXGQan(NR) - AMaxXGQan) ^ 2
SDMaxXGQanS = SDMaxXGQanS + DMaxXGQanS
NEXT NR
RETURN
'--------------------------------Std of StdXGQan
StdStd.Calc.QAnn:
VAXGQan = SDAXGQanS / (NRep - 1)
```

```
StdAXGQan = (VAXGQan) ^ .5
VStdXGQan = SDStdXGQanS / (NRep - 1)
StdStdXGQan = (VStdXGQan) ^ .5
VCsXGQan = SDCsXGQanS / (NRep - 1)
StdCsXGQan = (VCsXGQan) ^ .5
Vrk1 = SDrk1S / (NRep - 1)
Stdrk1 = (Vrk1) ^ .5
Vrk2 = SDrk2S / (NRep - 1)
Stdrk2 = (Vrk2) ^ .5
VHXGQan = SDHXGQanS / (NRep - 1)
StdHXGQan = (VHXGQan) ^ .5
VMinXGQan = SDMinXGQanS / (NRep - 1)
StdMinXGQan = (VMinXGQan) ^ .5
VMaxXGQan = SDMaxXGQanS / (NRep - 1)
StdMaxXGQan = (VMaxXGQan) ^ .5
RETURN
'------------------ 95 % Confidence Limits Tests
Confidence.Test.QAnn:
LA = AAXGQan - 1.96 * StdAXGQan
UA = AAXGQan + 1.96 * StdAXGQan
LStd = AStdXGQan - 1.96 * StdStdXGQan
UStd = AStdXGQan + 1.96 * StdStdXGQan
LCs = ACsXGQan - 1.96 * StdCsXGQan
UCs = ACsXGQan + 1.96 * StdCsXGQan
Lrk1 = Ark1 - 1.96 * Stdrk1
Urk1 = Ark1 + 1.96 * Stdrk1
Lrk2 = Ark2 - 1.96 * Stdrk2
Urk2 = Ark2 + 1.96 * Stdrk2
LHXGQan = AHXGQan - 1.96 * StdHXGQan
UHXGQan = AHXGQan + 1.96 * StdHXGQan
LMinXGQan = AMinXGQan - 1.96 * StdMinXGQan
UMinXGQan = AMinXGQan + 1.96 * StdMinXGQan
LMaxXGQan = AMaxXGQan - 1.96 * StdMaxXGQan
UMaxXGQan = AMaxXGQan + 1.96 * StdMaxXGQan
RETURN
'------------PrintStat.Test
PrintStat.Test.QAnn:
```

```
PRINT #102,
PRINT #102,
PRINT #102, "-------------> S t a t i s t i c s . T e s t"
PRINT #102,
PRINT #102, "AAXGQan          = "; AAXGQan
PRINT #102, "AStdXGQan        = "; AStdXGQan
PRINT #102, "ACsXGQan         = "; ACsXGQan
PRINT #102, "Ark1             = "; Ark1
PRINT #102, "Ark2             = "; Ark2
PRINT #102, "AHXGQan          = "; AHXGQan
PRINT #102, "AMinXGQan        = "; AMinXGQan
PRINT #102, "AMaxXGQan        = "; AMaxXGQan
PRINT #102,
PRINT #102, "StdAXGQan        = "; StdAXGQan
PRINT #102, "StdStdXGQan      = "; StdStdXGQan
PRINT #102, "StdCsXGQan       = "; StdCsXGQan
PRINT #102, "Stdrk1           = "; Stdrk1
PRINT #102, "Stdrk2           = "; Stdrk2
PRINT #102, "StdHXGQan        = "; StdHXGQan
PRINT #102, "StdMinXGQan      = "; StdMinXGQan
PRINT #102, "StdMaxXGQan      = "; StdMaxXGQan
'-----Lower & Upper Limits
PRINT #102,
PRINT #102, "         L-Bonds     Hist        Gen.    U-Bonds"
PRINT #102, "AAvg  "; USING "#######.###"; LA; AvgQan; AAXGQan; UA
PRINT #102, "AStd  "; USING "#######.###"; LStd; StdQan; AStdXGQan; UStd
PRINT #102, "ACs   "; USING "#######.###"; LCs; CsQan; ACsXGQan; UCs
PRINT #102, "Ark1  "; USING "#######.###"; Lrk1; rk1Qan; Ark1; Urk1
PRINT #102, "Ark2  "; USING "#######.###"; Lrk2; rk2Qan; Ark2; Urk2
PRINT #102, "AHurst"; USING "#######.###"; LHXGQan; HQan; AHXGQan; UHXGQan
PRINT #102, "AMin  "; USING "#######.###"; LMinXGQan; MinQan; AMinXGQan; UMinXGQan
PRINT #102, "AMax  "; USING "#######.###"; LMaxXGQan; MaxQan; AMaxXGQan; UMaxXGQan
PRINT #102,
PRINT #102, "The program of TierQann is finish."
RETURN

'TierQMon
```

```
'*****************************************************
'*                    Initialization
'*****************************************************
Init.Stat.Mon:
'$DYNAMIC
DIM MinQMonth(12), MaxQMonth(12), AvgQMonth(12), StdQMonth(12), rk1QMonth(12)
DIM rk2QMonth(12), CsQMonth(12), LAM(12), UAM(12), LStdM(12), UStdM(12), LCsM(12)
DIM UCsM(12), Lrk1M(12), Urk1M(12), Lrk2M(12), Urk2M(12), LMinXGQM(12), UMinXGQM(12)
DIM LMaxXGQM(12), UMaxXGQM(12), VAXGQM(12), StdAXGQM(12), VStdXGQM(12), StdStdXGQM(12)
DIM VCsXGQM(12), StdCsXGQM(12), Vrk1XGQM(12), Stdrk1XGQM(12), Vrk2XGQM(12), Stdrk2XGQM(12)
DIM VMinXGQM(12), StdMinXGQM(12), VMaxXGQM(12), StdMaxXGQM(12), DAXGQMS(12), SDAXGQMS(12)
DIM DStdXGQMS(12), SDStdXGQMS(12), DCsXGQMS(12), SDCsXGQMS(12), Drk1XGQMS(12), SDrk1XGQMS(12)
DIM Drk2XGQMS(12), SDrk2XGQMS(12), DMinXGQMS(12), SDMinXGQMS(12), DMaxXGQMS(12)
DIM SDMaxXGQMS(12), AAXGQM(12), AStdXGQM(12), ACsXGQM(12), Ark1XGQM(12), Ark2XGQM(12)
DIM AMinXGQM(12), AMaxXGQM(12), SAXGQM(12), SStdXGQM(12), SCsXGQM(12), Srk1XGQM(12)
DIM Srk2XGQM(12), SMinXGQM(12), SMaxXGQM(12), MinXGQM(12, 100), MaxXGQM(12, 100)
DIM rk1XGQM(12, 100), srk2Qana(12, 100), srk2Qanb(12, 100), rk2XGQM(12, 100), AXGQM(12, 100)
DIM SDXGQMS(12, 100), SDXGQMC(12, 100), DXGQMS(12, 100), DXGQMC(12, 100), VXGQM(12, 100)
DIM StdXGQM(12, 100), CsXGQM(12, 100)

N = 60
INPUT #215, QMonth$
INPUT #215, MinQMonth(1), MinQMonth(2), MinQMonth(3), MinQMonth(4), MinQMonth(5),
MinQMonth(6), MinQMonth(7), MinQMonth(8), MinQMonth(9), MinQMonth(10), MinQMonth(11),
MinQMonth(12)
INPUT #215, MaxQMonth(1), MaxQMonth(2), MaxQMonth(3), MaxQMonth(4), MaxQMonth(5),
MaxQMonth(6), MaxQMonth(7), MaxQMonth(8), MaxQMonth(9), MaxQMonth(10), MaxQMonth(10),
MaxQMonth(12)
INPUT #215, AvgQMonth(1), AvgQMonth(2), AvgQMonth(3), AvgQMonth(4), AvgQMonth(5),
AvgQMonth(6), AvgQMonth(7), AvgQMonth(8), AvgQMonth(9), AvgQMonth(10), AvgQMonth(11),
AvgQMonth(12)
INPUT #215, StdQMonth(1), StdQMonth(2), StdQMonth(3), StdQMonth(4), StdQMonth(5),
StdQMonth(6), StdQMonth(7), StdQMonth(8), StdQMonth(9), StdQMonth(10), StdQMonth(11),
StdQMonth(12)
INPUT #215, rk1QMonth(1), rk1QMonth(2), rk1QMonth(3), rk1QMonth(4), rk1QMonth(5),
rk1QMonth(6), rk1QMonth(7), rk1QMonth(8), rk1QMonth(9), rk1QMonth(10), rk1QMonth(11),
rk1QMonth(12)
```

```
INPUT #215, rk2QMonth(1), rk2QMonth(2), rk2QMonth(3), rk2QMonth(4), rk2QMonth(5),
rk2QMonth(6), rk2QMonth(7), rk2QMonth(8), rk2QMonth(9), rk2QMonth(10), rk2QMonth(11),
rk2QMonth(12)
INPUT #215, CsQMonth(1), CsQMonth(2), CsQMonth(3), CsQMonth(4), CsQMonth(5), CsQMonth(6),
CsQMonth(7), CsQMonth(8), CsQMonth(9), CsQMonth(10), CsQMonth(11), CsQMonth(12)
RETURN
'**********************************************************************
'                        Statistics.Test
'**********************************************************************
Statistics.Test.Mon:
GOSUB Input.Data.Mon
GOSUB AvgAvg.Calc.Mon
GOSUB SumSum.Calc.Mon
GOSUB StdStd.Calc.Mon
GOSUB Confidence.Test.Mon
GOSUB PrintStat.Test.Mon
RETURN
'------------------------------ Input.Data
Input.Data.Mon:
FOR NR = 1 TO NRep
FOR Month = 1 TO 12
INPUT #204, NR, Month, MinXGQM(Month, NR), AXGQM(Month, NR), MaxXGQM(Month, NR),
StdXGQM(Month, NR), CsXGQM(Month, NR), rk1XGQM(Month, NR), rk2XGQM(Month, NR)
NEXT Month
NEXT NR
RETURN
'------------------------------ AvgAvg.Calc
AvgAvg.Calc.Mon:
'---------- Average of XGQM(Month)
FOR Month = 1 TO 12
SAXGQM(Month) = 0
SStdXGQM(Month) = 0
SCsXGQM(Month) = 0
Srk1XGQM(Month) = 0
Srk2XGQM(Month) = 0
SMinXGQM(Month) = 0
SMaxXGQM(Month) = 0
```

```
NEXT Month
FOR Month = 1 TO 12
FOR NR = 1 TO NRep
SAXGQM(Month) = SAXGQM(Month) + AXGQM(Month, NR)
SStdXGQM(Month) = SStdXGQM(Month) + StdXGQM(Month, NR)
SCsXGQM(Month) = SCsXGQM(Month) + CsXGQM(Month, NR)
Srk1XGQM(Month) = Srk1XGQM(Month) + rk1XGQM(Month, NR)
Srk2XGQM(Month) = Srk2XGQM(Month) + rk2XGQM(Month, NR)
SMinXGQM(Month) = SMinXGQM(Month) + MinXGQM(Month, NR)
SMaxXGQM(Month) = SMaxXGQM(Month) + MaxXGQM(Month, NR)
NEXT NR
NEXT Month
FOR Month = 1 TO 12
AAXGQM(Month) = SAXGQM(Month) / NRep
AStdXGQM(Month) = SStdXGQM(Month) / NRep
ACsXGQM(Month) = SCsXGQM(Month) / NRep
Ark1XGQM(Month) = Srk1XGQM(Month) / NRep
Ark2XGQM(Month) = Srk2XGQM(Month) / NRep
AMinXGQM(Month) = SMinXGQM(Month) / NRep
AMaxXGQM(Month) = SMaxXGQM(Month) / NRep
NEXT Month
RETURN
'-----------------------------Sumation of Sumation
SumSum.Calc.Mon:
FOR Month = 1 TO 12
SDAXGQMS(Month) = 0
SDStdXGQMS(Month) = 0
SDCsXGQMS(Month) = 0
SDrk1XGQMS(Month) = 0
SDrk2XGQMS(Month) = 0
SDMinXGQMS(Month) = 0
SDMaxXGQMS(Month) = 0
NEXT Month
FOR Month = 1 TO 12
FOR NR = 1 TO NRep
DAXGQMS(Month) = (AXGQM(Month, NR) - AAXGQM(Month)) ^ 2
SDAXGQMS(Month) = SDAXGQMS(Month) + DAXGQMS(Month)
```

```
DStdXGQMS(Month) = (StdXGQM(Month, NR) - AStdXGQM(Month)) ^ 2
SDStdXGQMS(Month) = SDStdXGQMS(Month) + DStdXGQMS(Month)
DCsXGQMS(Month) = (CsXGQM(Month, NR) - ACsXGQM(Month)) ^ 2
SDCsXGQMS(Month) = SDCsXGQMS(Month) + DCsXGQMS(Month)
Drk1XGQMS(Month) = (rk1XGQM(Month, NR) - Ark1XGQM(Month)) ^ 2
SDrk1XGQMS(Month) = SDrk1XGQMS(Month) + Drk1XGQMS(Month)
Drk2XGQMS(Month) = (rk2XGQM(Month, NR) - Ark2XGQM(Month)) ^ 2
SDrk2XGQMS(Month) = SDrk2XGQMS(Month) + Drk2XGQMS(Month)
DMinXGQMS(Month) = (MinXGQM(Month, NR) - AMinXGQM(Month)) ^ 2
SDMinXGQMS(Month) = SDMinXGQMS(Month) + DMinXGQMS(Month)
DMaxXGQMS(Month) = (MaxXGQM(Month, NR) - AMaxXGQM(Month)) ^ 2
SDMaxXGQMS(Month) = SDMaxXGQMS(Month) + DMaxXGQMS(Month)
NEXT NR
NEXT Month
RETURN
'--------------------------------Std of StdXGQM(Month)
StdStd.Calc.Mon:
FOR Month = 1 TO 12
VAXGQM(Month) = SDAXGQMS(Month) / (NRep - 1)
StdAXGQM(Month) = (VAXGQM(Month)) ^ .5
VStdXGQM(Month) = SDStdXGQMS(Month) / (NRep - 1)
StdStdXGQM(Month) = (VStdXGQM(Month)) ^ .5
VCsXGQM(Month) = SDCsXGQMS(Month) / (NRep - 1)
StdCsXGQM(Month) = (VCsXGQM(Month)) ^ .5
Vrk1XGQM(Month) = SDrk1XGQMS(Month) / (NRep - 1)
Stdrk1XGQM(Month) = (Vrk1XGQM(Month)) ^ .5
Vrk2XGQM(Month) = SDrk2XGQMS(Month) / (NRep - 1)
Stdrk2XGQM(Month) = (Vrk2XGQM(Month)) ^ .5
VMinXGQM(Month) = SDMinXGQMS(Month) / (NRep - 1)
StdMinXGQM(Month) = (VMinXGQM(Month)) ^ .5
VMaxXGQM(Month) = SDMaxXGQMS(Month) / (NRep - 1)
StdMaxXGQM(Month) = (VMaxXGQM(Month)) ^ .5
NEXT Month
RETURN
'------------------ 95 % Confidence Limits Tests
Confidence.Test.Mon:
FOR Month = 1 TO 12
```

```
LAM(Month) = AAXGQM(Month) - 1.96 * StdAXGQM(Month)
UAM(Month) = AAXGQM(Month) + 1.96 * StdAXGQM(Month)
LStdM(Month) = AStdXGQM(Month) - 1.96 * StdStdXGQM(Month)
UStdM(Month) = AStdXGQM(Month) + 1.96 * StdStdXGQM(Month)
LCsM(Month) = ACsXGQM(Month) - 1.96 * StdCsXGQM(Month)
UCsM(Month) = ACsXGQM(Month) + 1.96 * StdCsXGQM(Month)
Lrk1M(Month) = Ark1XGQM(Month) - 1.96 * Stdrk1XGQM(Month)
Urk1M(Month) = Ark1XGQM(Month) + 1.96 * Stdrk1XGQM(Month)
Lrk2M(Month) = Ark2XGQM(Month) - 1.96 * Stdrk2XGQM(Month)
Urk2M(Month) = Ark2XGQM(Month) + 1.96 * Stdrk2XGQM(Month)
LMinXGQM(Month) = AMinXGQM(Month) - 1.96 * StdMinXGQM(Month)
UMinXGQM(Month) = AMinXGQM(Month) + 1.96 * StdMinXGQM(Month)
LMaxXGQM(Month) = AMaxXGQM(Month) - 1.96 * StdMaxXGQM(Month)
UMaxXGQM(Month) = AMaxXGQM(Month) + 1.96 * StdMaxXGQM(Month)
NEXT Month

GOSUB PrintStat.Test.Mon
RETURN
'------------PrintStat.Test
PrintStat.Test.Mon:
PRINT #202, DATE$, TIME$
PRINT #202, "Output of TierQMon.Bas"
PRINT #202,
PRINT #202, "--------------> S t a t i s t i c s . T e s t"
'-----Lower & Upper Limits
PRINT #202,
PRINT #202, "95 % Confidence Limits Tests"
PRINT #202, "Lower and Upper Bonds "
PRINT #202,
FOR Month = 1 TO 12
PRINT #202, "Month = ", Month
PRINT #202,
PRINT #202, "         L-Bonds     Hist        Gen.    U-Bonds"
PRINT #202, "AAvg   "; USING "#######.###"; LAM(Month); AvgQMonth(Month); AAXGQM(Month);
UAM(Month)
PRINT #202, "AStd  "; USING "#######.###"; LStdM(Month); StdQMonth(Month); AStdXGQM(Month);
UStdM(Month)
```

```
PRINT #202, "ACs    "; USING "#######.###"; LCsM(Month); CsQMonth(Month); ACsXGQM(Month);
UCsM(Month)
PRINT #202, "Ark1  "; USING "#######.###"; Lrk1M(Month); rk1QMonth(Month); Ark1XGQM(Month);
Urk1M(Month)
PRINT #202, "Ark2  "; USING "#######.###"; Lrk2M(Month); rk2QMonth(Month); Ark2XGQM(Month);
Urk2M(Month)
PRINT  #202,  "AMin    ";  USING  "#######.###";  LMinXGQM(Month);  MinQMonth(Month);
AMinXGQM(Month); UMinXGQM(Month)
PRINT  #202,  "AMax    ";  USING  "#######.###";  LMaxXGQM(Month);  MaxQMonth(Month);
AMaxXGQM(Month); UMaxXGQM(Month)
PRINT #202,
NEXT Month
PRINT #202,
PRINT #202, "Test result of the confidence interval at level 95%"
PRINT #202, "Note Month = 1 is May, etc."
PRINT #202,
FOR Month = 1 TO 12
IF (LAM(Month) <= AvgQMonth(Month)) AND (AvgQMonth(Month) <= UAM(Month)) THEN
PRINT #202, "AvgQMonth of Month         ="; Month; "is ok."
ELSE
PRINT #202, "AvgQMonth of Month         ="; Month; "is not ok."
END IF
IF (LStdM(Month) <= StdQMonth(Month)) AND (StdQMonth(Month) <= UStdM(Month)) THEN
PRINT #202, "StdQMonth of Month         ="; Month; "is ok."
ELSE
PRINT #202, "StdQMonth of Month         ="; Month; "is not ok."
END IF
IF (LCsM(Month) <= CsQMonth(Month)) AND (CsQMonth(Month) <= UCsM(Month)) THEN
PRINT #202, "CsQMonth of Month          ="; Month; "is ok."
ELSE
PRINT #202, "CsQMonth of Month          ="; Month; "is not ok."
END IF
IF (Lrk1M(Month) <= rk1QMonth(Month)) AND (rk1QMonth(Month) <= Urk1M(Month)) THEN
PRINT #202, "rk1QMonth of Month         ="; Month; "is ok."
ELSE
PRINT #202, "rk1QMonth of Month         ="; Month; "is not ok."
END IF
```

```
IF (Lrk2M(Month) <= rk2QMonth(Month)) AND (rk2QMonth(Month) <= Urk2M(Month)) THEN
PRINT #202, "rk2QMonth of Month         ="; Month; "is ok."
ELSE
PRINT #202, "rk2QMonth of Month         ="; Month; "is not ok."
END IF
IF (LMinXGQM(Month) <= MinQMonth(Month)) AND (MinQMonth(Month) <= UMinXGQM(Month)) THEN
PRINT #202, "MinQMonth of Month         ="; Month; "is ok."
ELSE
PRINT #202, "MinQMonth of Month         ="; Month; "is not ok."
END IF
IF (LMaxXGQM(Month) <= MaxQMonth(Month)) AND (MaxQMonth(Month) <= UMaxXGQM(Month)) THEN
PRINT #202, "MaxQMonth of Month         ="; Month; "is ok."
ELSE
PRINT #202, "MaxQMonth of Month         ="; Month; "is ok."
END IF
NEXT Month
PRINT #202,
PRINT #202, "The program of TierQMon is finish."
RETURN
```

Appendix - G

Listing Program of the Citarum Simulation Model (CTR-SIMOD)

The appendix gives the computer coding of the Citarum Simulation Model (CTR-SIMOD) that used for the simulation of reservoir operations of the Citarum Hydropower System. The program was written using the Quickbasic software package. It contained four stand-alone EXE files compiled from *.BAS files. The SAGCTR.EXE simulated reservoir operations for the Saguling Reservoir, the CIRCTR.EXE for Cirata, and the JATCTR.EXE for Jatiluhur. The energy output produced by those three programs were then statistically analyzed by the EGYCTR.EXE. These *.EXE files could be put in a disk drive c. The directories \data, \Sag, \Cir, \Jat, \Egy and \Sum have to be created in the disk drive c.

The input files for the program are:

- series of synthetic monthly flows, generated using the CTR-STOCHMOD, (c:\data\TierGQM.dat),

- rule curves for the Saguling, Cirata and Jatiluhur reservoirs, (c:\sag\RCSag.dat, c:\cir\RCCir.dat, c:\jat\RCJat.dat),

- evaporation coefficients for the Saguling, Cirata and Jatiluhur reservoirs, (c:\sag\EvapSag.dat, c:\cir\EvapCir.dat, c:\jat\EvapJat.dat),

- number of replications, years and months, (c:\data\NRepl.dat).

Other input-output files would be created automatically by the programs.

The four programs were run using a batch file, and the output files, were also copied using a batch file to be one summary file, for example CTR-CROG.OUT. The batch file, CITARUM.BAT, is as follow.

```
: begin
ECHO ON
NSagCtr
NCirCtr
NJatCtr
NEgyCtr
cd sum
copy notesag.out+c:\sag\sgstat.out+sgsum.out file1
copy notecir.out+c:\cir\crstat.out+crsum.out file2
copy notejat.out+c:\jat\jtstat.out+jtsum.out file3
copy file1+file2+file3+esum.out CTR-CROG.OUT
: END
```

The graphs of average monthly inflows, heads, turbine flows, spills and energy generated for the Saguling, Cirata and Jatiluhur are then made using the output file c:\sum\CTR-CROG.Out. The reliability curves of annual energy generated are made using the output files: c:\egy\ESagY.out, ECirY.out and EJatY.out.

```
'SagCtr.Bas
'
'**************************************************************************
'                    Description of Variables
'**************************************************************************
'NIF1      - inflows to Node 1: the Nanjung Gauge Site
'NIF2      - inflows to Node 2: the Saguling Reservoir
'NST2      - reservoir storage Node 2
'NOF2      - turbine flows from Node 2
'NEV2      - evaporations Node 2
'LEG2      - energy produced by hydropower Node 2: the Saguling Hydropower Plant
'ASur2     - reservoir surface area Node 2
'Head2     - heads of hydropower Node 2
'AvgHead2 - average of heads of the beginning and the end of month Node 2
'Spill2    - spill flows Node 2
'RC2       - rule curve or reservoir storage targets Node 2
'MinQTb2   - minimum flow of turbine Node 2
'CapPlant2- installed capacity of hydropower Node 2
'DNST2     - dead storage of Node 2
'Coefa2, Coefb2          - constants for calculating reservoir surface area Node 2
'Coefc2, Coefd2, Coefe3  - constants for calculating hydropower plant head Node 2
'MtpFlow2   -  flow coefficient Node 2
'CoefPlant2 -  plant efficiency Node 2
'NR        - number of replications
'NYear     - number of year
'Month     - number of month
'Avg       - average
'Min       - minimum
'Max       - maximum
'Std       - standard deviation
'
```

```
'*****************************************************
'                Main Module
'*****************************************************

CLS
LOCATE 1, 10
PRINT "Sag-Ctr Program."
GOSUB Initialization.SagCtr

FOR NR = 1 TO NReplicate

GOSUB Simulation.SagCtr
GOSUB Wrap.Up.SagCtr
LOCATE 7, 10
PRINT "The Program of Sag-Ctr is Finish."

IF NR = 1 THEN
        GOSUB Initialization.ASagCtr
ELSE
END IF

LOCATE 9, 10
PRINT "ASagCtr Program."

'Test 1 Year
IF MYear = 1 THEN
   GOTO 100
ELSE
END IF

GOSUB Statistics.Calc.ASagCtr

IF IReplicate = 1 THEN
  GOTO 100
ELSE
END IF
```

```
NEXT NR

GOSUB Outstat.calc.test


CLOSE #38

  GOSUB Statistics.Test.ASagCtr

100  LOCATE 17, 10
     PRINT "The Sub-Program of ASag-Crt is finish."
     LOCATE 21, 10
     PRINT "Press CTRL-C to stop the program !"

END

'*****************************************************
'               Initiation
'*****************************************************
Initialization.SagCtr:

' $DYNAMIC

DIM TNST(61), TNSTm1(61), TNIF(61), TNEV(61), TSpill(61), TNOF(61), CheckTDNST(61), TNOF2(13,
61)
DIM QRSag AS STRING, CRSag AS STRING, NIF1(13, 61), NIF2(13, 61), NST2(13, 61), NOF2(13, 61),
NEV2(13, 61)
DIM  LEG2(13,  61),  ASur2(13),  Head2(13),  AvgHead2(13),  Spill2(13,  61),  CoefEvap2(13),
HOUR(12), QR21(13), RC21(13)

OPEN "c:\data\NRepl.dat" FOR INPUT AS #1 'Number of replication, year & month of simulation
OPEN "c:\data\TierGQM.dat" FOR INPUT AS #2 'Genenerated monthly flows
OPEN "c:\sag\EvapSag.dat" FOR INPUT AS #3 'Coef of Evaporation for the Saguling Reservoir
OPEN "c:\sag\RCSag.dat" FOR INPUT AS #12 'Rule curve for the Saguling Reservoir

OPEN "c:\sag\SgStat.out" FOR OUTPUT AS #17  'Stat. calc. for the Saguling Res.
OPEN "c:\sum\sgsum.out" FOR OUTPUT AS #16 'Summary output for the Saguling Res.
```

```
OPEN "c:\sum\notesag.out" FOR OUTPUT AS #4 'Note for Saguling simulation.

OPEN "c:\sag\SagInp2.Out" FOR OUTPUT AS #6 'Input-output file.
OPEN "c:\sag\SagCir.Out" FOR OUTPUT AS #10 'Input for simulation of the Cirata Res.
OPEN "c:\sag\SgInput3.dat" FOR OUTPUT AS #38 'Input-output file.
OPEN "c:\sag\SagAnn.Out" FOR OUTPUT AS #51 'Annual energy of the Saguling Plant.

        MinQTb2 = 147
        NRTest = 1
        NTest = 1
        CapPlant2 = 715000
        NST2Max = 609
        DNST2 = 272
        Head2Min = 371
        Head2Max = 391
        Coefa2 = .113
        Coefb2 = .8936
        Coefc2 = 289.7
        Coefd2 = .04412
        Coefe2 = .2664
        CoefPlant2 = .835
        MtpFlow2 = 1.3

   PRINT #4, DATE$, TIME$
   PRINT #4,
   PRINT #4,
   PRINT #4, "*************** S A G U L I N G ************************"
   PRINT #4,
   PRINT #4, "******* Print Out Test of Sag-Ctr Program for NYear = "; NTest
   PRINT #4,
        INPUT #1, Month, MYear, NReplicate
   PRINT #4, "Month, NYear, NReplicate"; Month; MYear; NReplicate
   PRINT #4,

            INPUT #3, EvapSag$
            PRINT #4, EvapSag$
```

```
            c$ = "###.## ###.## ###.## ###.## ###.## ###.## ###.## ###.## ###.## ###.##
###.## ###.##"

        INPUT #3, CoefEvap2(1), CoefEvap2(2), CoefEvap2(3), CoefEvap2(4), CoefEvap2(5),
CoefEvap2(6), CoefEvap2(7), CoefEvap2(8), CoefEvap2(9), CoefEvap2(10), CoefEvap2(11),
CoefEvap2(12)
        PRINT #4, " CE(1) CE(2)  CE(3)   CE(4) CE(5)  CE(6)  CE(7)  CE(8)  CE(9)  CE(10)
CE(11) CE(12)"
        PRINT #4, USING c$; CoefEvap2(1); CoefEvap2(2); CoefEvap2(3); CoefEvap2(4);
CoefEvap2(5); CoefEvap2(6); CoefEvap2(7); CoefEvap2(8); CoefEvap2(9); CoefEvap2(10);
CoefEvap2(11); CoefEvap2(12)
        PRINT #4,

        INPUT #12, RCSag$
        INPUT #12, RC21(1), RC21(2), RC21(3), RC21(4), RC21(5), RC21(6), RC21(7),
RC21(8), RC21(9), RC21(10), RC21(11), RC21(12)

        PRINT #4,
        PRINT #4, RCSag$
        PRINT #4, "RC2i1  RC2i2  RC2i3  RC2i4  RC2i5  RC2i6  RC2i7  RC2i8  RC2i9  RC2i10
RC2i11 RC2i12"
        PRINT #4, USING c$; RC21(1); RC21(2); RC21(3); RC21(4); RC21(5); RC21(6);
RC21(7); RC21(8); RC21(9); RC21(10); RC21(11); RC21(12)

        PRINT #4,
        PRINT #4, "Test of Reading Data of NIF1 for NYear = ", NTest
        PRINT #4, " NR   NY  NIF1(1) NIF1(2) NIF1(3) NIF1(4) NIF1(5) NIF1(6) NIF1(7)
NIF1(8) NIF1(9) NIF1(10) NIF1(11) NIF1(12)"
        a$ = "#### #### ####.## ####.## ####.## ####.## ####.## ####.## ####.## ####.##
####.## #####.## #####.## #####.##"

RETURN

'***********************************************************************
'            Simulation
'***********************************************************************
Simulation.SagCtr:
```

```
OPEN "c:\sag\SagInp1.out" FOR OUTPUT AS #5 'input-output file

   FOR NYear = 1 TO MYear
        TNST(NYear) = 0
        TNSTm1(NYear) = 0
        TNIF(NYear) = 0
        TNEV(NYear) = 0
        TSpill(NYear) = 0
        TNOF(NYear) = 0

      FOR Month = 1 TO 12
         INPUT #2, NR, NYear, NIF1(Month, NYear)
         IF NIF1(Month, NYear) < 0 THEN
            NIF1(Month, NYear) = 0
         ELSE
         END IF
          NIF2(Month, NYear) = MtpFlow2 * NIF1(Month, NYear)
      NEXT Month

      IF (NR = NRTest) AND (NYear = NTest) THEN
         PRINT #4, USING a$; NR; NYear; NIF1(1, NYear); NIF1(2, NYear); NIF1(3, NYear);
NIF1(4, NYear); NIF1(5, NYear); NIF1(6, NYear); NIF1(7, NYear); NIF1(8, NYear); NIF1(9,
NYear); NIF1(10, NYear); NIF1(11, NYear); NIF1(12, NYear)
      ELSE
      END IF
   NEXT NYear

   IF (NR = NRTest) THEN
        PRINT #4,
        PRINT #4, "NIF2(month, NYear) = MtpFlow2 * NIF1(month, NYear)"
        PRINT #4, "MtpFlow2 = "; MtpFlow2
        PRINT #4,
        PRINT #4, "Test output of Sag-Ctr program for NYear = "; NTest
        PRINT #4,
        PRINT #4, "NR    NY  Mth   NIF2   NST2    NOF2    NEV2   Spill2   Head2   AvgHead2
  LEG2 "
        b$ = "#### #### #### ####.## ####.## ####.## ####.##  ####.## #####.## ######.##
```

```
##########.##"
   ELSE
   END IF

LOCATE 3, 10
PRINT "Sag-Ctr,             ---> NR      ="; NR

        NST2(1, 1) = NST2Max
        RC21(13) = RC21(1)

FOR NYear = 1 TO MYear
   LOCATE 5, 10
   PRINT "Sag-Ctr,             ---> NYear   = "; NYear

     FOR Month = 1 TO 12

             ASur2(Month) = Coefa2 * (NST2(Month, NYear) + DNST2) ^ Coefb2
             ASur2(Month + 1) = Coefa2 * (RC21(Month + 1) + DNST2) ^ Coefb2
             NEV2(Month, NYear) = CoefEvap2(Month) * (ASur2(Month) + ASur2(Month + 1)) * .5

             NOF2(Month, NYear) = NST2(Month, NYear) - RC21(Month + 1) + NIF2(Month, NYear)
- NEV2(Month, NYear)

              IF (NOF2(Month, NYear) < 0) THEN
                 NST2(Month + 1, NYear) = RC21(Month + 1) + NOF2(Month, NYear)
                 NOF2(Month, NYear) = 0
                 Spill2(Month, NYear) = 0
                 GOTO 101
              ELSE
              END IF

              IF (NOF2(Month, NYear) = 0) THEN
                 NST2(Month + 1, NYear) = RC21(Month + 1) + NOF2(Month, NYear)
                 NOF2(Month, NYear) = 0
                 Spill2(Month, NYear) = 0
                 GOTO 101
              ELSE
```

```
        END IF

        IF (0 < NOF2(Month, NYear) AND NOF2(Month, NYear) < MinQTb2) THEN
           NST2(Month + 1, NYear) = RC21(Month + 1) + NOF2(Month, NYear)
           NOF2(Month, NYear) = 0
           IF NST2(Month + 1, NYear) > NST2Max THEN
              Spill2(Month, NYear) = NST2(Month + 1, NYear) - NST2Max
              NST2(Month + 1, NYear) = NST2Max
              GOTO 101
           ELSE
           END IF
           GOTO 101
        ELSE
        END IF

        Head2(Month) = Coefc2 * (NST2(Month, NYear) + DNST2) ^ Coefd2 + Coefe2
        Head2(Month + 1) = Coefc2 * (RC21(Month + 1) + DNST2) ^ Coefd2 + Coefe2
        AvgHead2 = (Head2(Month) + Head2(Month + 1)) * .5

        QTb2 = (CapPlant2) / (CoefPlant2 * 9.81 * AvgHead2)

  IF (NST2(Month, NYear) > NST2Max OR NST2(Month, NYear) = NST2Max) THEN
        MaxQTb2 = 586.677
     ELSE
        MaxQTb2 = QTb2 * 2.628
  END IF

        IF (NOF2(Month, NYear) > MaxQTb2) THEN
          DNOFMax = NOF2(Month, NYear) - MaxQTb2
          NOF2(Month, NYear) = MaxQTb2
        ELSE
          DNOFMax = 0
        END IF

              NST2(Month + 1, NYear) = RC21(Month + 1) + DNOFMax
              IF NST2(Month + 1, NYear) > NST2Max THEN
                  Spill2(Month, NYear) = NST2(Month + 1, NYear) - NST2Max
```

```
                NST2(Month + 1, NYear) = NST2Max
            ELSE
                Spill2(Month, NYear) = 0
            END IF

       Head2(Month + 1) = Coefc2 * (NST2(Month + 1, NYear) + DNST2) ^ Coefd2 + Coefe2

IF NST2(Month + 1, NYear) <= 0 THEN
   Head2(Month + 1) = Head2Min
ELSE
END IF

IF (NST2(Month + 1, NYear) >= NST2Max) THEN
   Head2(Month + 1) = Head2Max
ELSE
END IF

IF NST2(Month, NYear) <= 0 THEN
   Head2(Month) = Head2Min
ELSE
END IF

IF (NST2(Month, NYear) > NST2Max OR NST2(Month, NYear) = 0) THEN
   Head2(Month) = Head2Max
ELSE
END IF

     AvgHead2 = (Head2(Month) + Head2(Month + 1)) * .5

101    LEG2(Month, NYear) = (2.725 / 1000) * CoefPlant2 * AvgHead2 * NOF2(Month, NYear)

TNOF2(Month, NYear) = NOF2(Month, NYear) + Spill2(Month, NYear)

TNIF(NYear) = TNIF(NYear) + NIF2(Month, NYear)
TNEV(NYear) = TNEV(NYear) + NEV2(Month, NYear)
TSpill(NYear) = TSpill(NYear) + Spill2(Month, NYear)
TNOF(NYear) = TNOF(NYear) + NOF2(Month, NYear)
```

```
        WRITE #5, NIF2(Month, NYear), NST2(Month, NYear), NOF2(Month, NYear), NEV2(Month,
NYear), Spill2(Month, NYear)
        WRITE #6, LEG2(Month, NYear)
        WRITE #10, TNOF2(Month, NYear)

        IF (NYear = NTest AND NR = NRTest) THEN
          PRINT #4, USING b$; NR; NYear; Month; NIF2(Month, NYear); NST2(Month, NYear);
NOF2(Month, NYear); NEV2(Month, NYear); Spill2(Month, NYear); Head2(Month); AvgHead2;
LEG2(Month, NYear)
        ELSE
        END IF

        IF (NYear = NTest + 1) AND (NR = NRTest) THEN
           IF (NYear = NTest + 1) AND (NR = NRTest) AND (Month = NTest) THEN
              PRINT #4, "-----------------------------------------------------------"
              ELSE
              END IF
          PRINT #4, USING b$; NR; NYear; Month; NIF2(Month, NYear); NST2(Month, NYear);
NOF2(Month, NYear); NEV2(Month, NYear); Spill2(Month, NYear); Head2(Month); AvgHead2;
LEG2(Month, NYear)
        ELSE
        END IF

     NEXT Month

       CheckTDNST(NYear) = NST2(1, NYear) - NST2(13, NYear) + TNIF(NYear) - TNEV(NYear) -
TSpill(NYear) - TNOF(NYear)

       NST2(1, NYear + 1) = NST2(13, NYear)

       IF (NYear = NTest AND NR = NRTest) THEN
          PRINT #4, " NR    NY   Month    NST1     NST13     TNIF     TNOF     TNEV     TSpill
CheckTDNST"
          PRINT #4, USING b$; NR; NYear; NMonth; NST2(1, NYear); NST2(13, NYear);
TNIF(NYear); TNOF(NYear); TNEV(NYear); TSpill(NYear); CheckTDNST(NYear)
       ELSE
```

```
        END IF
        WRITE #51, NST2(13, NYear), CheckTDNST(NYear)
NEXT NYear
RETURN
'*******************************************************
'*                       Wrap-Up
'*******************************************************
Wrap.Up.SagCtr:

     CLOSE #1
     CLOSE #5
RETURN


'ASagCtr: Analysis of SagCtr program
'****************************************************
'               Initiation
'****************************************************
Initialization.ASagCtr:
' $DYNAMIC
IMonth = Month: NIYear = MYear: IReplicate = NReplicate

DIM TimeStep AS STRING, Replicate AS STRING, NInflow2(12, 720), NStor2(12, 720)
DIM NOutFlow2(12, 720), NEvapor2(12, 720), Min(IMonth), Max(IMonth), MinNIF2(IMonth),
MaxNIF2(IMonth)
DIM MinNST2(IMonth), MaxNST2(IMonth), MinNOF2(IMonth), MaxNOF2(IMonth), MinNEV2(IMonth),
MaxNEV2(IMonth)
DIM MinSPILL2(IMonth), MaxSPILL2(IMonth), SNIF2(IMonth, IReplicate), ANIF2(IMonth,
IReplicate)
DIM SDNIF2S(IMonth, IReplicate), VNIF2(IMonth, IReplicate), StdNIF2(IMonth, IReplicate),
DNIF2S(IMonth, IReplicate)
DIM SNST2(IMonth, IReplicate), ANST2(IMonth, IReplicate), SDNST2S(IMonth, IReplicate)
DIM VNST2(IMonth, IReplicate), StdNST2(IMonth, IReplicate), DNST2S(IMonth, IReplicate)
DIM SNOF2(IMonth, IReplicate), ANOF2(IMonth, IReplicate), SDNOF2S(IMonth, IReplicate)
DIM VNOF2(IMonth, IReplicate), StdNOF2(IMonth, IReplicate), DNOF2S(IMonth, IReplicate)
DIM SNEV2(IMonth, IReplicate), ANEV2(IMonth, IReplicate), SDNEV2S(IMonth, IReplicate)
DIM VNEV2(IMonth, IReplicate), StdNEV2(IMonth, IReplicate), DNEV2S(IMonth, IReplicate)
```

```
DIM SSPILL2(IMonth, IReplicate), ASPILL2(IMonth, IReplicate), SDSPILL2S(IMonth, IReplicate)
DIM VSPILL2(IMonth, IReplicate), StdSPILL2(IMonth, IReplicate), DSPILL2S(IMonth, IReplicate)

a$ = "#### ######.## ######.## ######.##"
RETURN
'***********************************************************************
'                       Statistics.Calc
'***********************************************************************
Statistics.Calc.ASagCtr:
OPEN "c:\sag\SagInp1.out" FOR INPUT AS #36  'input of stat.calc

   FOR NYear = 1 TO NIYear
      FOR Month = 1 TO 12
         INPUT #36, NIF2(Month, NYear), NST2(Month, NYear), NOF2(Month, NYear), NEV2(Month,
NYear), Spill2(Month, NYear)
      NEXT Month
   NEXT NYear

        LOCATE 11, 10
PRINT "Stat.Calc            ---> NR       = "; NR

        GOSUB MaxMin.Calc
        GOSUB Avg.Calc
        GOSUB Sum.Calc
        GOSUB Std.Calc
        GOSUB OutStat.Calc

CLOSE #36

RETURN

'------------------ Max and Min
MaxMin.Calc:

        FOR Month = 1 TO 12
           MinNIF2(Month) = 10000
           MaxNIF2(Month) = 0
```

```
    MinNST2(Month) = 10000
    MaxNST2(Month) = 0
    MinNOF2(Month) = 10000
    MaxNOF2(Month) = 0
    MinNEV2(Month) = 10000
    MaxNEV2(Month) = 0
    MinSPILL2(Month) = 10000
    MaxSPILL2(Month) = 0
NEXT Month

FOR Month = 1 TO 12
    FOR NYear = 1 TO NIYear
            IF NIF2(Month, NYear) < MinNIF2(Month) THEN
               MinNIF2(Month) = NIF2(Month, NYear)
               ELSE
            END IF
            IF NIF2(Month, NYear) > MaxNIF2(Month) THEN
               MaxNIF2(Month) = NIF2(Month, NYear)
               ELSE
            END IF
            IF NST2(Month, NYear) < MinNST2(Month) THEN
               MinNST2(Month) = NST2(Month, NYear)
               ELSE
            END IF
            IF NST2(Month, NYear) > MaxNST2(Month) THEN
               MaxNST2(Month) = NST2(Month, NYear)
               ELSE
            END IF
            IF NOF2(Month, NYear) < MinNOF2(Month) THEN
               MinNOF2(Month) = NOF2(Month, NYear)
               ELSE
            END IF
            IF NOF2(Month, NYear) > MaxNOF2(Month) THEN
               MaxNOF2(Month) = NOF2(Month, NYear)
               ELSE
            END IF
            IF NEV2(Month, NYear) < MinNEV2(Month) THEN
```

```
                     MinNEV2(Month) = NEV2(Month, NYear)
                     ELSE
                  END IF
                  IF NEV2(Month, NYear) > MaxNEV2(Month) THEN
                     MaxNEV2(Month) = NEV2(Month, NYear)
                     ELSE
                  END IF
                  IF Spill2(Month, NYear) < MinSPILL2(Month) THEN
                     MinSPILL2(Month) = Spill2(Month, NYear)
                     ELSE
                  END IF
                  IF Spill2(Month, NYear) > MaxSPILL2(Month) THEN
                     MaxSPILL2(Month) = Spill2(Month, NYear)
                     ELSE
                  END IF
         NEXT NYear
       NEXT Month

RETURN
'------------------------------ Avg.Calc
Avg.Calc:

'---------- Average of NEV2

        FOR Month = 1 TO 12
           SNIF2(Month, NR) = 0
           SNST2(Month, NR) = 0
           SNOF2(Month, NR) = 0
           SNEV2(Month, NR) = 0
           SSPILL2(Month, NR) = 0
        NEXT Month

     FOR NYear = 1 TO NIYear
        FOR Month = 1 TO 12
           SNIF2(Month, NR) = SNIF2(Month, NR) + NIF2(Month, NYear)
           SNST2(Month, NR) = SNST2(Month, NR) + NST2(Month, NYear)
           SNOF2(Month, NR) = SNOF2(Month, NR) + NOF2(Month, NYear)
```

```
            SNEV2(Month, NR) = SNEV2(Month, NR) + NEV2(Month, NYear)
            SSPILL2(Month, NR) = SSPILL2(Month, NR) + Spill2(Month, NYear)
         NEXT Month
      NEXT NYear

         FOR Month = 1 TO 12
            ANIF2(Month, NR) = SNIF2(Month, NR) / NIYear
            ANST2(Month, NR) = SNST2(Month, NR) / NIYear
            ANOF2(Month, NR) = SNOF2(Month, NR) / NIYear
            ANEV2(Month, NR) = SNEV2(Month, NR) / NIYear
            ASPILL2(Month, NR) = SSPILL2(Month, NR) / NIYear
         NEXT Month

RETURN

'-------------------------------Sumation
Sum.Calc:

         FOR Month = 1 TO 12
            SDNIF2S(Month, NR) = 0
            SDNST2S(Month, NR) = 0
            SDNOF2S(Month, NR) = 0
            SDNEV2S(Month, NR) = 0
            SDSPILL2S(Month, NR) = 0
         NEXT Month

      FOR NYear = 1 TO NIYear
         FOR Month = 1 TO 12
            DNIF2S(Month, NR) = (NIF2(Month, NYear) - ANIF2(Month, NR)) ^ 2
            SDNIF2S(Month, NR) = SDNIF2S(Month, NR) + DNIF2S(Month, NR)
            DNST2S(Month, NR) = (NST2(Month, NYear) - ANST2(Month, NR)) ^ 2
            SDNST2S(Month, NR) = SDNST2S(Month, NR) + DNST2S(Month, NR)
            DNOF2S(Month, NR) = (NOF2(Month, NYear) - ANOF2(Month, NR)) ^ 2
            SDNOF2S(Month, NR) = SDNOF2S(Month, NR) + DNOF2S(Month, NR)
            DNEV2S(Month, NR) = (NEV2(Month, NYear) - ANEV2(Month, NR)) ^ 2
            SDNEV2S(Month, NR) = SDNEV2S(Month, NR) + DNEV2S(Month, NR)
```

```
            DSPILL2S(Month, NR) = (Spill2(Month, NYear) - ASPILL2(Month, NR)) ^ 2
            SDSPILL2S(Month, NR) = SDSPILL2S(Month, NR) + DSPILL2S(Month, NR)
         NEXT Month
      NEXT NYear
RETURN
'--------------------------------StdNEV2
Std.Calc:

         FOR Month = 1 TO 12
            VNIF2(Month, NR) = SDNIF2S(Month, NR) / (NIYear - 1)
            StdNIF2(Month, NR) = (VNIF2(Month, NR)) ^ .5
            VNST2(Month, NR) = SDNST2S(Month, NR) / (NIYear - 1)
            StdNST2(Month, NR) = (VNST2(Month, NR)) ^ .5
            VNOF2(Month, NR) = SDNOF2S(Month, NR) / (NIYear - 1)
            StdNOF2(Month, NR) = (VNOF2(Month, NR)) ^ .5
            VNEV2(Month, NR) = SDNEV2S(Month, NR) / (NIYear - 1)
            StdNEV2(Month, NR) = (VNEV2(Month, NR)) ^ .5
            VSPILL2(Month, NR) = SDSPILL2S(Month, NR) / (NIYear - 1)
            StdSPILL2(Month, NR) = (VSPILL2(Month, NR)) ^ .5
         NEXT Month


RETURN

OutStat.Calc:
IF (NR = NRTest) THEN
          PRINT #17,
          PRINT #17,
          PRINT #17, "Agung W.H. SOEHARNO"
          PRINT #17, DATE$, TIME$
                                                       P    R    I    N    T    #    1    7    ,
"*************************************************************************************"
          PRINT #17, "*            Output of Sag-Ctr Program"
          PRINT #17, "*            Node 2 : S A G U L I N G"
          PRINT #17, "*"
          PRINT #17, "----------------------> Stat.Calc - Replicate no. ", NR
                                                       P    R    I    N    T    #    1    7    ,
```

```
"*****************************************************************************************"
        PRINT #17,
        PRINT #17, " NIF2 = Inflow to Node2: Saguling"
        PRINT #17,
        PRINT #17, "  NYear   May    Jun    Jul    Aug    Sep    Oct    Nov    Dec    Jan
  Feb     Mar    Apr"
        PRINT #17,
   FOR NYear = 1 TO NIYear
       PRINT #17, USING "####.##"; NYear; NIF2(1, NYear); NIF2(2, NYear); NIF2(3, NYear);
NIF2(4, NYear); NIF2(5, NYear); NIF2(6, NYear); NIF2(7, NYear); NIF2(8, NYear); NIF2(9,
NYear); NIF2(10, NYear); NIF2(11, NYear); NIF2(12, NYear)
   NEXT NYear
        PRINT #17,
        PRINT #17, "   NR     May    Jun    Jul    Aug    Sep    Oct    Nov    Dec    Jan
  Feb     Mar    Apr"
        PRINT #17, "Minimum of NIF2"
        PRINT #17, USING "####.##"; NR; MinNIF2(1); MinNIF2(2); MinNIF2(3); MinNIF2(4);
MinNIF2(5); MinNIF2(6); MinNIF2(7); MinNIF2(8); MinNIF2(9); MinNIF2(10); MinNIF2(11);
MinNIF2(12)
        PRINT #17, "ANIF2 = Average of NIF2"
        PRINT #17, USING "####.##"; NR; ANIF2(1, NR); ANIF2(2, NR); ANIF2(3, NR); ANIF2(4,
NR); ANIF2(5, NR); ANIF2(6, NR); ANIF2(7, NR); ANIF2(8, NR); ANIF2(9, NR); ANIF2(10, NR);
ANIF2(11, NR); ANIF2(12, NR)
        PRINT #17, "Maximum of NIF2"
        PRINT #17, USING "####.##"; NR; MaxNIF2(1); MaxNIF2(2); MaxNIF2(3); MaxNIF2(5);
MaxNIF2(5); MaxNIF2(6); MaxNIF2(7); MaxNIF2(8); MaxNIF2(9); MaxNIF2(10); MaxNIF2(11);
MaxNIF2(12)
        PRINT #17, "StdNIF2 = Standard Dev. of NIF2"
        PRINT #17, USING "####.##"; NR; StdNIF2(1, NR); StdNIF2(2, NR); StdNIF2(3, NR);
StdNIF2(4, NR); StdNIF2(5, NR); StdNIF2(6, NR); StdNIF2(7, NR); StdNIF2(8, NR); StdNIF2(9,
NR); StdNIF2(10, NR); StdNIF2(11, NR); StdNIF2(12, NR)

        PRINT #17,
        PRINT #17, " NST2 = End of period Storage of Node 2: Saguling"
        PRINT #17,
        PRINT #17, "  NYear   May    Jun    Jul    Aug    Sep    Oct    Nov    Dec    Jan
  Feb     Mar    Apr"
```

```
        PRINT #17,
   FOR NYear = 1 TO NIYear
        PRINT #17, USING "####.##"; NYear; NST2(1, NYear); NST2(2, NYear); NST2(3, NYear);
NST2(4, NYear); NST2(5, NYear); NST2(6, NYear); NST2(7, NYear); NST2(8, NYear); NST2(9,
NYear); NST2(10, NYear); NST2(11, NYear); NST2(12, NYear)
   NEXT NYear
        PRINT #17,
        PRINT #17, "  NR     May     Jun     Jul     Aug     Sep     Oct     Nov     Dec     Jan
  Feb     Mar     Apr"
        PRINT #17, "Minimum of NST2"
        PRINT #17, USING "####.##"; NR; MinNST2(1); MinNST2(2); MinNST2(3); MinNST2(4);
MinNST2(5); MinNST2(6); MinNST2(7); MinNST2(8); MinNST2(9); MinNST2(10); MinNST2(11);
MinNST2(12)
        PRINT #17, "ANST2 = Average of NST2"
        PRINT #17, USING "####.##"; NR; ANST2(1, NR); ANST2(2, NR); ANST2(3, NR); ANST2(4,
NR); ANST2(5, NR); ANST2(6, NR); ANST2(7, NR); ANST2(8, NR); ANST2(9, NR); ANST2(10, NR);
ANST2(11, NR); ANST2(12, NR)
        PRINT #17, "Maximum of NST2"
        PRINT #17, USING "####.##"; NR; MaxNST2(1); MaxNST2(2); MaxNST2(3); MaxNST2(5);
MaxNST2(5); MaxNST2(6); MaxNST2(7); MaxNST2(8); MaxNST2(9); MaxNST2(10); MaxNST2(11);
MaxNST2(12)
        PRINT #17, "StdNST2 = Standard Dev. of NST2"
        PRINT #17, USING "####.##"; NR; StdNST2(1, NR); StdNST2(2, NR); StdNST2(3, NR);
StdNST2(4, NR); StdNST2(5, NR); StdNST2(6, NR); StdNST2(7, NR); StdNST2(8, NR); StdNST2(9,
NR); StdNST2(10, NR); StdNST2(11, NR); StdNST2(12, NR)

        PRINT #17,
        PRINT #17, " NOF2 = OutFlow from Node 2: Saguling"
        PRINT #17,
        PRINT #17, "  NYear   May     Jun     Jul     Aug     Sep     Oct     Nov     Dec     Jan
  Feb     Mar     Apr"
        PRINT #17,
   FOR NYear = 1 TO NIYear
        PRINT #17, USING "####.##"; NYear; NOF2(1, NYear); NOF2(2, NYear); NOF2(3, NYear);
NOF2(4, NYear); NOF2(5, NYear); NOF2(6, NYear); NOF2(7, NYear); NOF2(8, NYear); NOF2(9,
NYear); NOF2(10, NYear); NOF2(11, NYear); NOF2(12, NYear)
   NEXT NYear
```

```
        PRINT #17,
        PRINT #17, "   NR      May      Jun      Jul      Aug      Sep      Oct      Nov      Dec      Jan
  Feb      Mar      Apr"
        PRINT #17, "Minimum of NOF2"
        PRINT #17, USING "####.##"; NR; MinNOF2(1); MinNOF2(2); MinNOF2(3); MinNOF2(4);
MinNOF2(5);  MinNOF2(6);  MinNOF2(7);  MinNOF2(8);  MinNOF2(9);  MinNOF2(10);  MinNOF2(11);
MinNOF2(12)
        PRINT #17, "ANOF2 = Average of NOF2"
        PRINT #17, USING "####.##"; NR; ANOF2(1, NR); ANOF2(2, NR); ANOF2(3, NR); ANOF2(4,
NR); ANOF2(5, NR); ANOF2(6, NR); ANOF2(7, NR); ANOF2(8, NR); ANOF2(9, NR); ANOF2(10, NR);
ANOF2(11, NR); ANOF2(12, NR)
        PRINT #17, "Maximum of NOF2"
        PRINT #17, USING "####.##"; NR; MaxNOF2(1); MaxNOF2(2); MaxNOF2(3); MaxNOF2(5);
MaxNOF2(5);  MaxNOF2(6);  MaxNOF2(7);  MaxNOF2(8);  MaxNOF2(9);  MaxNOF2(10);  MaxNOF2(11);
MaxNOF2(12)
        PRINT #17, "StdNOF2 = Standard Dev. of NOF2"
        PRINT #17, USING "####.##"; NR; StdNOF2(1, NR); StdNOF2(2, NR); StdNOF2(3, NR);
StdNOF2(4, NR); StdNOF2(5, NR); StdNOF2(6, NR); StdNOF2(7, NR); StdNOF2(8, NR); StdNOF2(9,
NR); StdNOF2(10, NR); StdNOF2(11, NR); StdNOF2(12, NR)

        PRINT #17,
        PRINT #17, "NEV2 = Evaporation of Node 2: Saguling"
        PRINT #17,
        PRINT #17, " NYear    May      Jun      Jul      Aug      Sep      Oct      Nov      Dec      Jan
  Feb      Mar      Apr"
        PRINT #17,
  FOR NYear = 1 TO NIYear
       PRINT #17, USING "####.##"; NYear; NEV2(1, NYear); NEV2(2, NYear); NEV2(3, NYear);
NEV2(4, NYear); NEV2(5, NYear); NEV2(6, NYear), NEV2(7, NYear); NEV2(8, NYear); NEV2(9,
NYear); NEV2(10, NYear); NEV2(11, NYear); NEV2(12, NYear)
   NEXT NYear
        PRINT #17,
        PRINT #17, "   NR      May      Jun      Jul      Aug      Sep      Oct      Nov      Dec      Jan
  Feb      Mar      Apr"
        PRINT #17, "Minimum of NEV2"
        PRINT #17, USING "####.##"; NR; MinNEV2(1); MinNEV2(2); MinNEV2(3); MinNEV2(4);
MinNEV2(5);  MinNEV2(6);  MinNEV2(7);  MinNEV2(8);  MinNEV2(9);  MinNEV2(10);  MinNEV2(11);
```

```
MinNEV2(12)
        PRINT #17, "ANEV2 = Average of NEV2"
        PRINT #17, USING "####.##"; NR; ANEV2(1, NR); ANEV2(2, NR); ANEV2(3, NR); ANEV2(4,
NR); ANEV2(5, NR); ANEV2(6, NR); ANEV2(7, NR); ANEV2(8, NR); ANEV2(9, NR); ANEV2(10, NR);
ANEV2(11, NR); ANEV2(12, NR)
        PRINT #17, "Maximum of NEV2"
        PRINT #17, USING "####.##"; NR; MaxNEV2(1); MaxNEV2(2); MaxNEV2(3); MaxNEV2(5);
MaxNEV2(5); MaxNEV2(6); MaxNEV2(7); MaxNEV2(8); MaxNEV2(9); MaxNEV2(10); MaxNEV2(11);
MaxNEV2(12)
        PRINT #17, "StdNEV2 = Standard Dev. of NEV2"
        PRINT #17, USING "####.##"; NR; StdNEV2(1, NR); StdNEV2(2, NR); StdNEV2(3, NR);
StdNEV2(4, NR); StdNEV2(5, NR); StdNEV2(6, NR); StdNEV2(7, NR); StdNEV2(8, NR); StdNEV2(9,
NR); StdNEV2(10, NR); StdNEV2(11, NR); StdNEV2(12, NR)
        PRINT #17,
        PRINT #17,
        PRINT #17, "SPILL2 = Spill flow of Node 2: Saguling"
        PRINT #17,
        PRINT #17, "  NYear    May    Jun    Jul    Aug    Sep    Oct    Nov    Dec    Jan
  Feb    Mar    Apr"
        PRINT #17,
  FOR NYear = 1 TO NIYear
      PRINT #17, USING "####.##"; NYear; Spill2(1, NYear); Spill2(2, NYear); Spill2(3,
NYear); Spill2(4, NYear); Spill2(5, NYear); Spill2(6, NYear); Spill2(7, NYear); Spill2(8,
NYear); Spill2(9, NYear); Spill2(10, NYear); Spill2(11, NYear); _
Spill2(12, NYear)
  NEXT NYear
        PRINT #17,
        PRINT #17, "   NR    May    Jun    Jul    Aug    Sep    Oct    Nov    Dec    Jan
  Feb    Mar    Apr"
        PRINT #17, "Minimum of SPILL2"
        PRINT #17, USING "####.##"; NR; MinSPILL2(1); MinSPILL2(2); MinSPILL2(3);
MinSPILL2(4); MinSPILL2(5); MinSPILL2(6); MinSPILL2(7); MinSPILL2(8); MinSPILL2(9);
MinSPILL2(10); MinSPILL2(11); MinSPILL2(12)
        PRINT #17, "ASPILL2 = Average of SPILL2"
        PRINT #17, USING "####.##"; NR; ASPILL2(1, NR); ASPILL2(2, NR); ASPILL2(3, NR);
ASPILL2(4, NR); ASPILL2(5, NR); ASPILL2(6, NR); ASPILL2(7, NR); ASPILL2(8, NR); ASPILL2(9,
NR,; ASPILL2(10, NR); ASPILL2(11, NR); ASPILL2(12, NR)
```

```
        PRINT #17, "Maximum of SPILL2"
        PRINT  #17,  USING  "####.##";  NR;  MaxSPILL2(1);  MaxSPILL2(2);  MaxSPILL2(3);
MaxSPILL2(5);  MaxSPILL2(5);  MaxSPILL2(6);  MaxSPILL2(7);  MaxSPILL2(8);  MaxSPILL2(9);
MaxSPILL2(10); MaxSPILL2(11); MaxSPILL2(12)
        PRINT #17, "StdSPILL2 = Standard Dev. of SPILL2"
                          PRINT #17, USING "####.##"; NR; StdSPILL2(1, NR); StdSPILL2(2,
NR); StdSPILL2(3, NR); StdSPILL2(4, NR); StdSPILL2(5, NR); StdSPILL2(6, NR); StdSPILL2(7,
NR); StdSPILL2(8, NR); StdSPILL2(9, NR); StdSPILL2(10, NR);  _
StdSPILL2(11, NR); StdSPILL2(12, NR)
        PRINT #17,
        PRINT #16,
        PRINT #16,
        PRINT #16,
        PRINT #16, "Agung W.H. SOEHARNO"
        PRINT #16, DATE$, TIME$, TIME$
                                                  P   R   I   N   T   #   1   6   ,
"*******************************************************************************************"
        PRINT #16, "*              Output of Sag-Ctr Program"
        PRINT #16, "*              Node 2 : S A G U L I N G"
        PRINT #16, "*"
        PRINT #16, "----------------------> Stat.Calc - Replicate no. ", NR
                                                  P   R   I   N   T   #   1   6   ,
"*******************************************************************************************"
        PRINT #16,
        PRINT #16, " SGSUM = Summary File of Node2: Saguling"
        PRINT #16,
        PRINT #16, "  NYear  May   Jun   Jul   Aug   Sep   Oct   Nov   Dec   Jan
  Feb   Mar   Apr"
        PRINT #16, "ANIF2 = Average of NIF2"
        PRINT #16, USING "####.##"; NR; ANIF2(1, NR); ANIF2(2, NR); ANIF2(3, NR); ANIF2(4,
NR); ANIF2(5, NR); ANIF2(6, NR); ANIF2(7, NR); ANIF2(8, NR); ANIF2(9, NR); ANIF2(10, NR);
ANIF2(11, NR); ANIF2(12, NR)
        PRINT #16, "ANST2 = Average of NST2"
        PRINT #16, USING "####.##"; NR; ANST2(1, NR); ANST2(2, NR); ANST2(3, NR); ANST2(4,
NR); ANST2(5, NR); ANST2(6, NR); ANST2(7, NR); ANST2(8, NR); ANST2(9, NR); ANST2(10, NR);
ANST2(11, NR); ANST2(12, NR)
        PRINT #16, "ANOF2 = Average of NOF2"
```

```
        PRINT #16, USING "####.##"; NR; ANOF2(1, NR); ANOF2(2, NR); ANOF2(3, NR); ANOF2(4,
NR); ANOF2(5, NR); ANOF2(6, NR); ANOF2(7, NR); ANOF2(8, NR); ANOF2(9, NR); ANOF2(10, NR);
ANOF2(11, NR); ANOF2(12, NR)
        PRINT #16, "ANEV2 = Average of NEV2"
        PRINT #16, USING "####.##"; NR; ANEV2(1, NR); ANEV2(2, NR); ANEV2(3, NR); ANEV2(4,
NR); ANEV2(5, NR); ANEV2(6, NR); ANEV2(7, NR); ANEV2(8, NR); ANEV2(9, NR); ANEV2(10, NR);
ANEV2(11, NR); ANEV2(12, NR)
        PRINT #16, "ASPILL2 = Average of SPILL2"
        PRINT #16, USING "####.##"; NR; ASPILL2(1, NR); ASPILL2(2, NR); ASPILL2(3, NR);
ASPILL2(4, NR); ASPILL2(5, NR); ASPILL2(6, NR); ASPILL2(7, NR); ASPILL2(8, NR); ASPILL2(9,
NR); ASPILL2(10, NR); ASPILL2(11, NR); ASPILL2(12, NR)
ELSE
END IF

        WRITE #38, ANIF2(1, NR), ANIF2(2, NR), ANIF2(3, NR), ANIF2(4, NR), ANIF2(5, NR),
ANIF2(6, NR), ANIF2(7, NR), ANIF2(8, NR), ANIF2(9, NR), ANIF2(10, NR), ANIF2(11, NR),
ANIF2(12, NR)
        WRITE #38, ANST2(1, NR), ANST2(2, NR), ANST2(3, NR), ANST2(4, NR), ANST2(5, NR),
ANST2(6, NR), ANST2(7, NR), ANST2(8, NR), ANST2(9, NR), ANST2(10, NR), ANST2(11, NR),
ANST2(12, NR)
        WRITE #38, ANOF2(1, NR), ANOF2(2, NR), ANOF2(3, NR), ANOF2(4, NR), ANOF2(5, NR),
ANOF2(6, NR), ANOF2(7, NR), ANOF2(8, NR), ANOF2(9, NR), ANOF2(10, NR), ANOF2(11, NR),
ANOF2(12, NR)
        WRITE #38, ANEV2(1, NR), ANEV2(2, NR), ANEV2(3, NR), ANEV2(4, NR), ANEV2(5, NR),
ANEV2(6, NR), ANEV2(7, NR), ANEV2(8, NR), ANEV2(9, NR), ANEV2(10, NR), ANEV2(11, NR),
ANEV2(12, NR)
        WRITE #38, ASPILL2(1, NR), ASPILL2(2, NR), ASPILL2(3, NR), ASPILL2(4, NR),
ASPILL2(5, NR), ASPILL2(6, NR), ASPILL2(7, NR), ASPILL2(8, NR), ASPILL2(9, NR), ASPILL2(10,
NR), ASPILL2(11, NR), ASPILL2(12, NR)

        WRITE #38, StdNIF2(1, NR), StdNIF2(2, NR), StdNIF2(3, NR), StdNIF2(4, NR),
StdNIF2(5, NR), StdNIF2(6, NR), StdNIF2(7, NR), StdNIF2(8, NR), StdNIF2(9, NR), StdNIF2(10,
NR), StdNIF2(11, NR), StdNIF2(12, NR)
        WRITE #38, StdNST2(1, NR), StdNST2(2, NR), StdNST2(3, NR), StdNST2(4, NR),
StdNST2(5, NR), StdNST2(6, NR), StdNST2(7, NR), StdNST2(8, NR), StdNST2(9, NR), StdNST2(10,
NR), StdNST2(11, NR), StdNST2(12, NR)
        WRITE #38, StdNOF2(1, NR), StdNOF2(2, NR), StdNOF2(3, NR), StdNOF2(4, NR),
```

```
StdNOF2(5, NR), StdNOF2(6, NR), StdNOF2(7, NR), StdNOF2(8, NR), StdNOF2(9, NR), StdNOF2(10,
NR), StdNOF2(11, NR), StdNOF2(12, NR)
        WRITE #38, StdNEV2(1, NR), StdNEV2(2, NR), StdNEV2(3, NR), StdNEV2(4, NR),
StdNEV2(5, NR), StdNEV2(6, NR), StdNEV2(7, NR), StdNEV2(8, NR), StdNEV2(9, NR), StdNEV2(10,
NR), StdNEV2(11, NR), StdNEV2(12, NR)
        WRITE #38, StdSPILL2(1, NR), StdSPILL2(2, NR), StdSPILL2(3, NR), StdSPILL2(4, NR),
StdSPILL2(5, NR), StdSPILL2(6, NR), StdSPILL2(7, NR), StdSPILL2(8, NR), StdSPILL2(9, NR),
StdSPILL2(10, NR), StdSPILL2(11, NR), StdSPILL2(12, NR)

RETURN

Outstat.calc.test:
FOR NR = 1 TO IReplicate
IF (NR = 1) THEN
        PRINT #17,
        PRINT #17, "  NR      May     Jun     Jul     Aug     Sep     Oct     Nov     Dec     Jan
  Feb     Mar     Apr"
        PRINT #17, "ANIF2 = Average of NIF2"
ELSE
END IF
        PRINT #17, USING "####.##"; NR; ANIF2(1, NR); ANIF2(2, NR); ANIF2(3, NR); ANIF2(4,
NR); ANIF2(5, NR); ANIF2(6, NR); ANIF2(7, NR); ANIF2(8, NR); ANIF2(9, NR); ANIF2(10, NR);
ANIF2(11, NR); ANIF2(12, NR)
NEXT NR

FOR NR = 1 TO IReplicate
IF (NR = 1) THEN
        PRINT #17,
        PRINT #17, "  NR      May     Jun     Jul     Aug     Sep     Oct     Nov     Dec     Jan
  Feb     Mar     Apr"
        PRINT #17, "ANST2 = Average of NST2"
ELSE
END IF
        PRINT #17, USING "####.##"; NR; ANST2(1, NR); ANST2(2, NR); ANST2(3, NR); ANST2(4,
NR); ANST2(5, NR); ANST2(6, NR); ANST2(7, NR); ANST2(8, NR); ANST2(9, NR); ANST2(10, NR);
ANST2(11, NR); ANST2(12, NR)
NEXT NR
```

```
FOR NR = 1 TO IReplicate
IF (NR = 1) THEN
        PRINT #17,
        PRINT #17, "    NR      May     Jun     Jul     Aug     Sep     Oct     Nov     Dec     Jan
  Feb     Mar     Apr"
        PRINT #17, "ANOF2 = Average of NOF2"
ELSE
END IF
        PRINT #17, USING "####.##"; NR; ANOF2(1, NR); ANOF2(2, NR); ANOF2(3, NR); ANOF2(4,
NR); ANOF2(5, NR); ANOF2(6, NR); ANOF2(7, NR); ANOF2(8, NR); ANOF2(9, NR); ANOF2(10, NR);
ANOF2(11, NR); ANOF2(12, NR)
NEXT NR

FOR NR = 1 TO IReplicate
IF (NR = 1) THEN
        PRINT #17,
        PRINT #17, "    NR      May     Jun     Jul     Aug     Sep     Oct     Nov     Dec     Jan
  Feb     Mar     Apr"
        PRINT #17, "ANEV2 = Average of NEV2"
ELSE
END IF
        PRINT #17, USING "####.##"; NR; ANEV2(1, NR); ANEV2(2, NR); ANEV2(3, NR); ANEV2(4,
NR); ANEV2(5, NR); ANEV2(6, NR); ANEV2(7, NR); ANEV2(8, NR); ANEV2(9, NR); ANEV2(10, NR);
ANEV2(11, NR); ANEV2(12, NR)
NEXT NR

FOR NR = 1 TO IReplicate
IF (NR = 1) THEN
        PRINT #17,
        PRINT #17, "    NR      May     Jun     Jul     Aug     Sep     Oct     Nov     Dec     Jan
  Feb     Mar     Apr"
        PRINT #17, "ASPILL2 = Average of SPILL2"
ELSE
END IF
        PRINT #17, USING "####.##"; NR; ASPILL2(1, NR); ASPILL2(2, NR); ASPILL2(3, NR);
ASPILL2(4, NR); ASPILL2(5, NR); ASPILL2(6, NR); ASPILL2(7, NR); ASPILL2(8, NR); ASPILL2(9,
NR); ASPILL2(10, NR); ASPILL2(11, NR); ASPILL2(12, NR)
```

```
NEXT NR
RETURN

'***********************************************************************
'                          Statistics.TESt
'***********************************************************************
Statistics.Test.ASagCtr:
LOCATE 13, 10
PRINT "S t a t i s t i c s . T e s t"
PRINT #17, "--------------> S t a t i s t i c s . T e s t"
PRINT #17,
PRINT #17, "   May     Jun     Jul     Aug     Sep     Oct     Nov     Dec     Jan     Feb     Mar
Apr"
PRINT #17,

OPEN "c:\sag\SgInput3.dat" FOR INPUT AS #39  'input of stat.tes

GOSUB Input.Data
GOSUB AvgAvg.Calc
GOSUB SumSum.Calc
GOSUB StdStd.Calc
GOSUB PrintStat.Tes

CLOSE #39
RETURN

'------------------------------ Input.Data
Input.Data:

     FOR NR = 1 TO IReplicate
         INPUT #39, ANIF2(1, NR), ANIF2(2, NR), ANIF2(3, NR), ANIF2(4, NR), ANIF2(5, NR),
ANIF2(6, NR), ANIF2(7, NR), ANIF2(8, NR), ANIF2(9, NR), ANIF2(10, NR), ANIF2(11, NR),
ANIF2(12, NR)
         INPUT #39, ANST2(1, NR), ANST2(2, NR), ANST2(3, NR), ANST2(4, NR), ANST2(5, NR),
ANST2(6, NR), ANST2(7, NR), ANST2(8, NR), ANST2(9, NR), ANST2(10, NR), ANST2(11, NR),
ANST2(12, NR)
         INPUT #39, ANOF2(1, NR), ANOF2(2, NR), ANOF2(3, NR), ANOF2(4, NR), ANOF2(5, NR),
```

```
'---------- Average of NEV2

      FOR Month = 1 TO 12
         SANIF2(Month) = 0
         SStdNIF2(Month) = 0
         SANST2(Month) = 0
         SStdNST2(Month) = 0
         SANOF2(Month) = 0
         SStdNOF2(Month) = 0
         SANEV2(Month) = 0
         SStdNEV2(Month) = 0
         SASPILL2(Month) = 0
         SStdSPILL2(Month) = 0
      NEXT Month

   FOR Month = 1 TO 12
      FOR NR = 1 TO IReplicate
         SANIF2(Month) = SANIF2(Month) + ANIF2(Month, NR)
         SStdNIF2(Month) = SStdNIF2(Month) + StdNIF2(Month, NR)
         SANST2(Month) = SANST2(Month) + ANST2(Month, NR)
         SStdNST2(Month) = SStdNST2(Month) + StdNST2(Month, NR)
         SANOF2(Month) = SANOF2(Month) + ANOF2(Month, NR)
         SStdNOF2(Month) = SStdNOF2(Month) + StdNOF2(Month, NR)
         SANEV2(Month) = SANEV2(Month) + ANEV2(Month, NR)
         SStdNEV2(Month) = SStdNEV2(Month) + StdNEV2(Month, NR)
         SASPILL2(Month) = SASPILL2(Month) + ASPILL2(Month, NR)
         SStdSPILL2(Month) = SStdSPILL2(Month) + StdSPILL2(Month, NR)
      NEXT NR
   NEXT Month

      FOR Month = 1 TO 12
         AANIF2(Month) = SANIF2(Month) / IReplicate
         AStdNIF2(Month) = SStdNIF2(Month) / IReplicate
         AANST2(Month) = SANST2(Month) / IReplicate
         AStdNST2(Month) = SStdNST2(Month) / IReplicate
         AANOF2(Month) = SANOF2(Month) / IReplicate
         AStdNOF2(Month) = SStdNOF2(Month) / IReplicate
```

```
ANOF2(6, NR), ANOF2(7, NR), ANOF2(8, NR), ANOF2(9, NR), ANOF2(10, NR), ANOF2(11, NR),
ANOF2(12, NR)
        INPUT #39, ANEV2(1, NR), ANEV2(2, NR), ANEV2(3, NR), ANEV2(4, NR), ANEV2(5, NR),
ANEV2(6, NR), ANEV2(7, NR), ANEV2(8, NR), ANEV2(9, NR), ANEV2(10, NR), ANEV2(11, NR),
ANEV2(12, NR)
        INPUT #39, ASPILL2(1, NR), ASPILL2(2, NR), ASPILL2(3, NR), ASPILL2(4, NR),
ASPILL2(5, NR), ASPILL2(6, NR), ASPILL2(7, NR), ASPILL2(8, NR), ASPILL2(9, NR), ASPILL2(10,
NR), ASPILL2(11, NR), ASPILL2(12, NR)

        INPUT #39, StdNIF2(1, NR), StdNIF2(2, NR), StdNIF2(3, NR), StdNIF2(4, NR),
StdNIF2(5, NR), StdNIF2(6, NR), StdNIF2(7, NR), StdNIF2(8, NR), StdNIF2(9, NR), StdNIF2(10,
NR), StdNIF2(11, NR), StdNIF2(12, NR)
        INPUT #39, StdNST2(1, NR), StdNST2(2, NR), StdNST2(3, NR), StdNST2(4, NR),
StdNST2(5, NR), StdNST2(6, NR), StdNST2(7, NR), StdNST2(8, NR), StdNST2(9, NR), StdNST2(10,
NR), StdNST2(11, NR), StdNST2(12, NR)
        INPUT #39, StdNOF2(1, NR), StdNOF2(2, NR), StdNOF2(3, NR), StdNOF2(4, NR),
StdNOF2(5, NR), StdNOF2(6, NR), StdNOF2(7, NR), StdNOF2(8, NR), StdNOF2(9, NR), StdNOF2(10,
NR), StdNOF2(11, NR), StdNOF2(12, NR)
        INPUT #39, StdNEV2(1, NR), StdNEV2(2, NR), StdNEV2(3, NR), StdNEV2(4, NR),
StdNEV2(5, NR), StdNEV2(6, NR), StdNEV2(7, NR), StdNEV2(8, NR), StdNEV2(9, NR), StdNEV2(10,
NR), StdNEV2(11, NR), StdNEV2(12, NR)
        INPUT #39, StdSPILL2(1, NR), StdSPILL2(2, NR), StdSPILL2(3, NR), StdSPILL2(4, NR),
StdSPILL2(5, NR), StdSPILL2(6, NR), StdSPILL2(7, NR), StdSPILL2(8, NR), StdSPILL2(9, NR),
StdSPILL2(10, NR), StdSPILL2(11, NR), StdSPILL2(12, NR)
    NEXT NR

RETURN

'------------------------------ AvgAvg.Calc
AvgAvg.Calc:
DIM SANIF2(12), SStdNIF2(12), AANIF2(12), AStdNIF2(12)
DIM SANST2(12), SStdNST2(12), AANST2(12), AStdNST2(12)
DIM SANOF2(12), SStdNOF2(12), AANOF2(12), AStdNOF2(12)

DIM SANEV2(12), SStdNEV2(12), AANEV2(12), AStdNEV2(12)
DIM SASPILL2(12), SStdSPILL2(12), AASPILL2(12), AStdSPILL2(12)
```

```
         AANEV2(Month) = SANEV2(Month) / IReplicate
         AStdNEV2(Month) = SStdNEV2(Month) / IReplicate
         AASPILL2(Month) = SASPILL2(Month) / IReplicate
         AStdSPILL2(Month) = SStdSPILL2(Month) / IReplicate
      NEXT Month

RETURN

'------------------------------Sumation of Sumation
SumSum.Calc:
DIM SDANIF2S(12), SDStdNIF2S(12), DANIF2S(12), DStdNIF2S(12)
DIM SDANST2S(12), SDStdNST2S(12), DANST2S(12), DStdNST2S(12)
DIM SDANOF2S(12), SDStdNOF2S(12), DANOF2S(12), DStdNOF2S(12)
DIM SDANEV2S(12), SDStdNEV2S(12), DANEV2S(12), DStdNEV2S(12)
DIM SDASPILL2S(12), SDStdSPILL2S(12), DASPILL2S(12), DStdSPILL2S(12)

      FOR Month = 1 TO 12
         SDANIF2S(Month) = 0
         SDStdNIF2S(Month) = 0
         SDANST2S(Month) = 0
         SDStdNST2S(Month) = 0
         SDANOF2S(Month) = 0
         SDStdNOF2S(Month) = 0
         SDANEV2S(Month) = 0
         SDStdNEV2S(Month) = 0
         SDASPILL2S(Month) = 0
         SDStdSPILL2S(Month) = 0
      NEXT Month

   FOR Month = 1 TO 12
      FOR NR = 1 TO IReplicate
         DANIF2S(Month) = (ANIF2(Month, NR) - AANIF2(Month)) ^ 2
         SDANIF2S(Month) = SDANIF2S(Month) + DANIF2S(Month)
         DStdNIF2S(Month) = (StdNIF2(Month, NR) - AStdNIF2(Month)) ^ 2
         SDStdNIF2S(Month) = SDStdNIF2S(Month) + DStdNIF2S(Month)
         DANST2S(Month) = (ANST2(Month, NR) - AANST2(Month)) ^ 2
         SDANST2S(Month) = SDANST2S(Month) + DANST2S(Month)
```

```
            DStdNST2S(Month) = (StdNST2(Month, NR) - AStdNST2(Month)) ^ 2
            SDStdNST2S(Month) = SDStdNST2S(Month) + DStdNST2S(Month)
            DANOF2S(Month) = (ANOF2(Month, NR) - AANOF2(Month)) ^ 2
            SDANOF2S(Month) = SDANOF2S(Month) + DANOF2S(Month)
            DStdNOF2S(Month) = (StdNOF2(Month, NR) - AStdNOF2(Month)) ^ 2
            SDStdNOF2S(Month) = SDStdNOF2S(Month) + DStdNOF2S(Month)
            DANEV2S(Month) = (ANEV2(Month, NR) - AANEV2(Month)) ^ 2
            SDANEV2S(Month) = SDANEV2S(Month) + DANEV2S(Month)
            DStdNEV2S(Month) = (StdNEV2(Month, NR) - AStdNEV2(Month)) ^ 2
            SDStdNEV2S(Month) = SDStdNEV2S(Month) + DStdNEV2S(Month)
            DASPILL2S(Month) = (ASPILL2(Month, NR) - AASPILL2(Month)) ^ 2
            SDASPILL2S(Month) = SDASPILL2S(Month) + DASPILL2S(Month)
            DStdSPILL2S(Month) = (StdSPILL2(Month, NR) - AStdSPILL2(Month)) ^ 2
            SDStdSPILL2S(Month) = SDStdSPILL2S(Month) + DStdSPILL2S(Month)
         NEXT NR
      NEXT Month

10 RETURN

'--------------------------------Std of StdNEV2
StdStd.Calc:

DIM VANIF2(12), StdANIF2(12), VStdNIF2(12), StdStdNIF2(12)
DIM VANST2(12), StdANST2(12), VStdNST2(12), StdStdNST2(12)
DIM VANOF2(12), StdANOF2(12), VStdNOF2(12), StdStdNOF2(12)
DIM VANEV2(12), StdANEV2(12), VStdNEV2(12), StdStdNEV2(12)
DIM VASPILL2(12), StdASPILL2(12), VStdSPILL2(12), StdStdSPILL2(12)

         FOR Month = 1 TO 12
            VANIF2(Month) = SDANIF2S(Month) / (IReplicate - 1)
            StdANIF2(Month) = (VANIF2(Month)) ^ .5
            VStdNIF2(Month) = SDStdNIF2S(Month) / (IReplicate - 1)
            StdStdNIF2(Month) = (VStdNIF2(Month)) ^ .5

            VANST2(Month) = SDANST2S(Month) / (IReplicate - 1)
            StdANST2(Month) = (VANST2(Month)) ^ .5
```

```
        VStdNST2(Month) = SDStdNST2S(Month) / (IReplicate - 1)
        StdStdNST2(Month) = (VStdNST2(Month)) ^ .5

        VANOF2(Month) = SDANOF2S(Month) / (IReplicate - 1)
        StdANOF2(Month) = (VANOF2(Month)) ^ .5
        VStdNOF2(Month) = SDStdNOF2S(Month) / (IReplicate - 1)
        StdStdNOF2(Month) = (VStdNOF2(Month)) ^ .5

        VANEV2(Month) = SDANEV2S(Month) / (IReplicate - 1)
        StdANEV2(Month) = (VANEV2(Month)) ^ .5
        VStdNEV2(Month) = SDStdNEV2S(Month) / (IReplicate - 1)
        StdStdNEV2(Month) = (VStdNEV2(Month)) ^ .5

        VASPILL2(Month) = SDASPILL2S(Month) / (IReplicate - 1)
        StdASPILL2(Month) = (VASPILL2(Month)) ^ .5
        VStdSPILL2(Month) = SDStdSPILL2S(Month) / (IReplicate - 1)
        StdStdSPILL2(Month) = (VStdSPILL2(Month)) ^ .5

    NEXT Month

RETURN

'------------------ PrintStat.tes
PrintStat.Tes:
     PRINT #17, "AANIF2 = "
     PRINT #17, USING "####.##"; AANIF2(1); AANIF2(2); AANIF2(3); AANIF2(4); AANIF2(5);
AANIF2(6); AANIF2(7); AANIF2(8); AANIF2(9); AANIF2(10); AANIF2(11); AANIF2(12)
     PRINT #17, "AStdNIF2 ="
     PRINT #17, USING "####.##"; AStdNIF2(1); AStdNIF2(2); AStdNIF2(3); AStdNIF2(4);
AStdNIF2(5); AStdNIF2(6); AStdNIF2(7); AStdNIF2(8); AStdNIF2(9); AStdNIF2(10); AStdNIF2(11);
AStdNIF2(12)

     PRINT #17, "AANST2 = "
     PRINT #17, USING "####.##"; AANST2(1); AANST2(2); AANST2(3); AANST2(4); AANST2(5);
AANST2(6); AANST2(7); AANST2(8); AANST2(9); AANST2(10); AANST2(11); AANST2(12)
     PRINT #17, "AStdNST2 ="
     PRINT #17, USING "####.##"; AStdNST2(1); AStdNST2(2); AStdNST2(3); AStdNST2(4);
```

```
AStdNST2(5); AStdNST2(6); AStdNST2(7); AStdNST2(8); AStdNST2(9); AStdNST2(10); AStdNST2(11);
AStdNST2(12)

     PRINT #17, "AANOF2 = "
     PRINT #17, USING "####.##"; AANOF2(1); AANOF2(2); AANOF2(3); AANOF2(4); AANOF2(5);
AANOF2(6); AANOF2(7); AANOF2(8); AANOF2(9); AANOF2(10); AANOF2(11); AANOF2(12)
     PRINT #17, "AStdNOF2 ="
     PRINT #17, USING "####.##"; AStdNOF2(1); AStdNOF2(2); AStdNOF2(3); AStdNOF2(4);
AStdNOF2(5); AStdNOF2(6); AStdNOF2(7); AStdNOF2(8); AStdNOF2(9); AStdNOF2(10); AStdNOF2(11);
AStdNOF2(12)

     PRINT #17, "AANEV2 = "
     PRINT #17, USING "####.##"; AANEV2(1); AANEV2(2); AANEV2(3); AANEV2(4); AANEV2(5);
AANEV2(6); AANEV2(7); AANEV2(8); AANEV2(9); AANEV2(10); AANEV2(11); AANEV2(12)
     PRINT #17, "AStdNEV2 ="
     PRINT #17, USING "####.##"; AStdNEV2(1); AStdNEV2(2); AStdNEV2(3); AStdNEV2(4);
AStdNEV2(5); AStdNEV2(6); AStdNEV2(7); AStdNEV2(8); AStdNEV2(9); AStdNEV2(10); AStdNEV2(11);
AStdNEV2(12)

     PRINT #17, "AASPILL2 = "
     PRINT #17, USING "####.##"; AASPILL2(1); AASPILL2(2); AASPILL2(3); AASPILL2(4);
AASPILL2(5); AASPILL2(6); AASPILL2(7); AASPILL2(8); AASPILL2(9); AASPILL2(10); AASPILL2(11);
AASPILL2(12)
     PRINT #17, "AStdSPILL2 ="
     PRINT #17, USING "####.##"; AStdSPILL2(1); AStdSPILL2(2); AStdSPILL2(3); AStdSPILL2(4);
AStdSPILL2(5); AStdSPILL2(6); AStdSPILL2(7); AStdSPILL2(8); AStdSPILL2(9); AStdSPILL2(10);
AStdSPILL2(11); AStdSPILL2(12)

     PRINT #17, "StdANIF2 = "
     PRINT #17, USING "####.##"; StdANIF2(1); StdANIF2(2); StdANIF2(3); StdANIF2(4);
StdANIF2(5); StdANIF2(6); StdANIF2(7); StdANIF2(8); StdANIF2(9); StdANIF2(10); StdANIF2(11);
StdANIF2(12)
     PRINT #17, "StdStdNIF2 = "
     PRINT #17, USING "####.##"; StdStdNIF2(1); StdStdNIF2(2); StdStdNIF2(3); StdStdNIF2(4);
StdStdNIF2(5); StdStdNIF2(6); StdStdNIF2(7); StdStdNIF2(8); StdStdNIF2(9); StdStdNIF2(10);
StdStdNIF2(11); StdStdNIF2(12)
```

```
PRINT #17, "StdANST2 = "
PRINT #17, USING "####.##"; StdANST2(1); StdANST2(2); StdANST2(3); StdANST2(4); StdANST2(5); StdANST2(6); StdANST2(7); StdANST2(8); StdANST2(9); StdANST2(10); StdANST2(11); StdANST2(12)
PRINT #17, "StdStdNST2 = "
PRINT #17, USING "####.##"; StdStdNST2(1); StdStdNST2(2); StdStdNST2(3); StdStdNST2(4); StdStdNST2(5); StdStdNST2(6); StdStdNST2(7); StdStdNST2(8); StdStdNST2(9); StdStdNST2(10); StdStdNST2(11); StdStdNST2(12)

PRINT #17, "StdANOF2 = "
PRINT #17, USING "####.##"; StdANOF2(1); StdANOF2(2); StdANOF2(3); StdANOF2(4); StdANOF2(5); StdANOF2(6); StdANOF2(7); StdANOF2(8); StdANOF2(9); StdANOF2(10); StdANOF2(11); StdANOF2(12)
PRINT #17, "StdStdNOF2 = "
PRINT #17, USING "####.##"; StdStdNOF2(1); StdStdNOF2(2); StdStdNOF2(3); StdStdNOF2(4); StdStdNOF2(5); StdStdNOF2(6); StdStdNOF2(7); StdStdNOF2(8); StdStdNOF2(9); StdStdNOF2(10); StdStdNOF2(11); StdStdNOF2(12)

PRINT #17, "StdANEV2 = "
PRINT #17, USING "####.##"; StdANEV2(1); StdANEV2(2); StdANEV2(3); StdANEV2(4); StdANEV2(5); StdANEV2(6); StdANEV2(7); StdANEV2(8); StdANEV2(9); StdANEV2(10); StdANEV2(11); StdANEV2(12)
PRINT #17, "StdStdNEV2 = "
PRINT #17, USING "####.##"; StdStdNEV2(1); StdStdNEV2(2); StdStdNEV2(3); StdStdNEV2(4); StdStdNEV2(5); StdStdNEV2(6); StdStdNEV2(7); StdStdNEV2(8); StdStdNEV2(9); StdStdNEV2(10); StdStdNEV2(11); StdStdNEV2(12)

PRINT #17, "StdASPILL2 = "
PRINT #17, USING "####.##"; StdASPILL2(1); StdASPILL2(2); StdASPILL2(3); StdASPILL2(4); StdASPILL2(5); StdASPILL2(6); StdASPILL2(7); StdASPILL2(8); StdASPILL2(9); StdASPILL2(10); StdASPILL2(11); StdASPILL2(12)
PRINT #17, "StdStdSPILL2 = "
PRINT #17, USING "####.##"; StdStdSPILL2(1); StdStdSPILL2(2); StdStdSPILL2(3); StdStdSPILL2(4); StdStdSPILL2(5); StdStdSPILL2(6); StdStdSPILL2(7); StdStdSPILL2(8); StdStdSPILL2(9); StdStdSPILL2(10); StdStdSPILL2(11); StdStdSPILL2(12)

    PRINT #16,
```

```
        PRINT #16,
        PRINT #16, "Agung W.H. SOEHARNO"
        PRINT #16, DATE$, TIME$
                                          P    R    I    N    T    #    1    6    ,
"*****************************************************************************************"
        PRINT #16, "*            Output of ASag-Ctr Program"
        PRINT #16, "*            Node 2 : S A G U L I N G"
        PRINT #16, "*"
        PRINT #16, "----------------------> Statistics.Test "
                                          P    R    I    N    T    #    1    6    ,
"*****************************************************************************************"
        PRINT #16,
        PRINT #16, " May    Jun    Jul    Aug    Sep    Oct    Nov    Dec    Jan    Feb
 Mar    Apr"
        PRINT #16,

     PRINT #16, "AANIF2 = "
     PRINT #16, USING "####.##"; AANIF2(1); AANIF2(2); AANIF2(3); AANIF2(4); AANIF2(5);
AANIF2(6); AANIF2(7); AANIF2(8); AANIF2(9); AANIF2(10); AANIF2(11); AANIF2(12)

     PRINT #16, "AANST2 = "
     PRINT #16, USING "####.##"; AANST2(1); AANST2(2); AANST2(3); AANST2(4); AANST2(5);
AANST2(6); AANST2(7); AANST2(8); AANST2(9); AANST2(10); AANST2(11); AANST2(12)

     PRINT #16, "AANOF2 = "
     PRINT #16, USING "####.##"; AANOF2(1); AANOF2(2); AANOF2(3); AANOF2(4); AANOF2(5);
AANOF2(6); AANOF2(7); AANOF2(8); AANOF2(9); AANOF2(10); AANOF2(11); AANOF2(12)

     PRINT #16, "AANEV2 = "
     PRINT #16, USING "####.##"; AANEV2(1); AANEV2(2); AANEV2(3); AANEV2(4); AANEV2(5);
AANEV2(6); AANEV2(7); AANEV2(8); AANEV2(9); AANEV2(10); AANEV2(11); AANEV2(12)

     PRINT #16, "AASPILL2 = "
     PRINT #16, USING "####.##"; AASPILL2(1); AASPILL2(2); AASPILL2(3); AASPILL2(4);
AASPILL2(5); AASPILL2(6); AASPILL2(7); AASPILL2(8); AASPILL2(9); AASPILL2(10); AASPILL2(11);
AASPILL2(12)
```

```
    PRINT #17, " The Program ASag-Crt is finish."
    PRINT #16, " The Program ASag-Crt is finish."

RETURN
```

```
'CirCtr.Bas

'****************************************************************************
'                              Description of Variables
'****************************************************************************
'NIF1     - inflows to Node 1: the Nanjung Gauge Site
'NIF3     - inflows to Node 3: the Cirata Reservoir
'NST3     - reservoir storage Node 3
'NOF3     - turbine flows from Node 3
'NEV3     - evaporations Node 3
'LEG3     - energy produced by hydropower Node 3: the Cirata Hydropower Plant
'ASur3    - reservoir surface area Node 3
'Head3    - heads of hydropower Node 3
'AvgHead3 - average of heads of the beginning and the end of month Node 3
'Spill3   - spill flows Node 3
'RC3      - rule curve or reservoir storage targets Node 3
'MinQTb3  - minimum flow of turbine Node 3
'CapPlant3- installed capacity of hydropower Node 3
'DNST3    - dead storage of Node 3
'Coefa3, Coefb3          - constants for calculating reservoir surface area Node 3
'Coefc3, Coefd3, Coefe3  - constants for calculating hydropower plant head Node 3
'LMtpFlow3     - flow coefficient Node 3
'CoefPlant3    - plant efficiency Node 3
'NR       - number of replications
'NYear    - number of year
'Month    - number of month
'Avg      - average
'Min      - minimum
'Max      - maxi um
'Std      - standard deviation
'
```

```
'*****************************************************
'                Main Module
'*****************************************************

CLS
LOCATE 1, 10
PRINT "Cir-Ctr Program."
GOSUB Initialization.CirCtr

FOR NR = 1 TO NReplicate

GOSUB Simulation.CirCtr
GOSUB Wrap.Up.CirCtr
LOCATE 7, 10
PRINT "The Program of Cir-Ctr is Finish."

IF NR = 1 THEN
        GOSUB Initialization.ACirCtr
ELSE
END IF

LOCATE 9, 10
PRINT "ACirCtr Program."

IF MYear = 1 THEN
   GOTO 100
ELSE
END IF

GOSUB Statistics.Calc.ACirCtr

IF IReplicate = 1 THEN
  GOTO 100
ELSE
END IF
```

```
NEXT NR

GOSUB Outstat.calc.test
CLOSE #38

  GOSUB Statistics.Test.ACirCtr

100  LOCATE 17, 10
     PRINT "The Sub-Program of ACir-Crt is finish."
     LOCATE 21, 10
     PRINT "Press CTRL-C to stop the program !"

END

'*****************************************************
'              Initiation
'*****************************************************
Initialization.CirCtr:
' $DYNAMIC

DIM TNST(60), TNSTm1(60), TNIF(60), TNEV(60), TSpill(60), TNOF(60), CheckTDNST(60)
DIM TNOF2(12, 60), TNOF3(12, 60), QRCir AS STRING, CRCir AS STRING, NIF1(13, 60)
DIM NIF3(13, 60), NST3(13, 61), NOF3(12, 60), NEV3(12, 60), LEG3(12, 60), ASur3(13)
DIM Head3(13), AvgHead3(13), Spill3(12, 60), CoefEVap3(13), QR31(13), RC31(13)

OPEN "c:\data\NRepl.dat" FOR INPUT AS #1  'Number of Replication, year and month
OPEN "c:\data\TierGQM.dat" FOR INPUT AS #2 'Gen. monthly flows
OPEN "c:\Cir\EvapCir.dat" FOR INPUT AS #3 'Coef. evaporation for the Cirata Res.
OPEN "c:\Cir\RCCir.dat" FOR INPUT AS #12 'Rule curve for the Cirata Res.
OPEN "c:\Cir\CrStat.out" FOR OUTPUT AS #17  'Output of Stat.calc
OPEN "c:\sum\Crsum.out" FOR OUTPUT AS #16 'Summary output of Cirata simulation
OPEN "c:\sum\noteCir.out" FOR OUTPUT AS #4 'Note of simulation
OPEN "c:\Cir\CirInp2.Out" FOR OUTPUT AS #6 'input-output file
OPEN "c:\Cir\CirJat.Out" FOR OUTPUT AS #10 'input for Jatiluhur simulation
OPEN "c:\Cir\CrInput3.dat" FOR OUTPUT AS #38 'input-output file

OPEN "c:\Sag\SagCir.out" FOR INPUT AS #50 'output from Saguling simulation
```

```
OPEN "c:\Cir\CirAnn.Out" FOR OUTPUT AS #51 'annual energy of Cirata simulation

        NRTest = 1
        NTest = 1
        CapPlant3 = 518000
        MinQTb3 = 315
        NST3Max = 796
        DNST3 = 1177
        Head3Min = 102
        Head3Max = 117
        'Head3Max = 117.08
        Coefc3 = 17.46
        Coefd3 = .25
        Coefa3 = .3963
        Coefb3 = .6667
        Coefe3 = .634025  '!!!
        LMtpFlow3 = .68
        CoefPlant3 = .816

   PRINT #4, DATE$, TIME$
   PRINT #4,
   PRINT #4,
   PRINT #4, "**************** C I R A T A *************************"
   PRINT #4,
   PRINT #4, "******* Print Out Test of Cir-Ctr Program for NYear = "; NTest
   PRINT #4,
        INPUT #1, Month, MYear, NReplicate
   PRINT #4, "Month, NYear, NReplicate"; Month; MYear; NReplicate
   PRINT #4,

            INPUT #3, EvapCir$
            PRINT #4, EvapCir$

                 c$ = "###.## ###.## ###.## ###.## ###.## ###.## ###.## ###.## ###.## ###.##
###.## ###.##"

            INPUT #3, CoefEVap3(1), CoefEVap3(2), CoefEVap3(3), CoefEVap3(4), CoefEVap3(5),
```

```
CoefEVap3(6), CoefEVap3(7), CoefEVap3(8), CoefEVap3(9), CoefEVap3(10), CoefEVap3(11),
CoefEVap3(12)
          PRINT #4, " CE(1) CE(2)  CE(3)   CE(4) CE(5)  CE(6)  CE(7)  CE(8)  CE(9)  CE(10)
CE(11) CE(12)"
          PRINT #4, USING c$; CoefEVap3(1); CoefEVap3(2); CoefEVap3(3); CoefEVap3(4);
CoefEVap3(5); CoefEVap3(6); CoefEVap3(7); CoefEVap3(8); CoefEVap3(9); CoefEVap3(10);
CoefEVap3(11); CoefEVap3(12)
          PRINT #4,

          INPUT #12, RCCir$
          INPUT #12, RC31(1), RC31(2), RC31(3), RC31(4), RC31(5), RC31(6), RC31(7),
RC31(8), RC31(9), RC31(10), RC31(11), RC31(12)

          PRINT #4,
          PRINT #4, RCCir$
          'PRINT #4, "RC(1) = RC(13) = NSTMax"
          PRINT #4, "RC3i1 RC3i2 RC3i3 RC3i4 RC3i5 RC3i6 RC3i7 RC3i8 RC3i9 RC3i10
RC3i11 RC3i12"
          PRINT #4, USING c$; RC31(1); RC31(2); RC31(3); RC31(4); RC31(5); RC31(6);
RC31(7); RC31(8); RC31(9); RC31(10); RC31(11); RC31(12)

          PRINT #4,
          PRINT #4, "Test of Reading Data of NIF1 for NYear = 1"
          PRINT #4, " NR    NY  NIF1(1) NIF1(2) NIF1(3) NIF1(4) NIF1(5) NIF1(6) NIF1(7)
NIF1(8) NIF1(9) NIF1(10) NIF1(11) NIF1(12)"
          a$ = "#### #### ####.## ####.## ####.## ####.## ####.## ####.## ####.## ####.##
####.## #####.## #####.## #####.##"

RETURN
'**********************************************************************
'              Simulation
'**********************************************************************
Simulation.CirCtr:
OPEN "c:\Cir\CirInp1.out" FOR OUTPUT AS #5 'input-output file

FOR NYear = 1 TO MYear
       TNST(NYear) = 0
```

```
        TNST(NYear) = 0
        TNSTm1(NYear) = 0
        TNIF(NYear) = 0
        TNEV(NYear) = 0
        TSpill(NYear) = 0
        TNOF(NYear) = 0

       FOR Month = 1 TO 12
          INPUT #2, NR, NYear, NIF1(Month, NYear)

          IF NIF1(Month, NYear) < 0 THEN
             NIF1(Month, NYear) = 0
          ELSE
          END IF

          INPUT #50, TNOF2(Month, NYear)
          NIF3(Month, NYear) = LMtpFlow3 * NIF1(Month, NYear) + TNOF2(Month, NYear)

       NEXT Month

       IF (NR = NRTest) AND (NYear = NTest) THEN
          PRINT #4, USING a$; NR; NYear; NIF1(1, NYear); NIF1(2, NYear); NIF1(3, NYear);
NIF1(4, NYear); NIF1(5, NYear); NIF1(6, NYear); NIF1(7, NYear); NIF1(8, NYear); NIF1(9,
NYear); NIF1(10, NYear); NIF1(11, NYear); NIF1(12, NYear)
       ELSE
       END IF
NEXT NYear

   IF (NR = NRTest) THEN
        PRINT #4,
        PRINT #4, "NIF3(Month, NYear) = LMtpFlow3 * NIF1(Month, NYear) + TNOF2(Month, NYear)"
        PRINT #4, "LMtpFlow3 = "; LMtpFlow3
        PRINT #4, " NR   NY  TNOF(1) TNOF(2) TNOF(3) TNOF(4) TNOF(5) TNOF(6) TNOF(7) TNOF(8)
TNOF(9) TNOF(10) TNOF(11) TNOF(12)"
        PRINT #4,
   ELSE
   END IF
```

```
LOCATE 3, 10
PRINT "Cir-Ctr,              ---> NR       ="; NR

        NST3(1, 1) = NST3Max
        RC31(13) = RC31(1)

FOR NYear = 1 TO MYear
   LOCATE 5, 10
   PRINT "Cir-Ctr,              ---> NYear    = "; NYear

   IF (NR = NRTest) AND (NYear = NTest) THEN
           PRINT #4, USING a$; NR; NYear; TNOF2(1, NYear); TNOF2(2, NYear); TNOF2(3, NYear);
TNOF2(4, NYear); TNOF2(5, NYear); TNOF2(6, NYear); TNOF2(7, NYear); TNOF2(8, NYear); TNOF2(9,
NYear); TNOF2(10, NYear); TNOF2(11, NYear); TNOF2(12, NYear)
        PRINT #4,
        PRINT #4, "Test output of Cir-Ctr program for NYear = "; NTest
        PRINT #4,
        PRINT #4, "NR    NY  Mth   NIF3   NST3    NOF3    NEV3   Spill3   Head3   AvgHead3
  LEG3 "
        b$ = "#### #### #### ####.## ####.## ####.## ####.##  ####.## #####.## ######.##
##########.##"
   ELSE
   END IF

     FOR Month = 1 TO 12

             ASur3(Month) = Coefa3 * (NST3(Month, NYear) + DNST3) ^ Coefb3
             ASur3(Month + 1) = Coefa3 * (RC31(Month + 1) + DNST3) ^ Coefb3
             NEV3(Month, NYear) = CoefEVap3(Month) * (ASur3(Month) + ASur3(Month + 1)) * .5

             NOF3(Month, NYear) = NST3(Month, NYear) - RC31(Month + 1) + NIF3(Month, NYear)
- NEV3(Month, NYear)

              IF (NOF3(Month, NYear) < 0) THEN
                 NST3(Month + 1, NYear) = RC31(Month + 1) + NOF3(Month, NYear)
                 NOF3(Month, NYear) = 0
                 Spill3(Month, NYear) = 0
```

```
         GOTO 101
      ELSE
      END IF

      IF (NOF3(Month, NYear) = 0) THEN
         NST3(Month + 1, NYear) = RC31(Month + 1) + NOF3(Month, NYear)
         NOF3(Month, NYear) = 0
         Spill3(Month, NYear) = 0
         GOTO 101
      ELSE
      END IF

      IF (0 < NOF3(Month, NYear) AND NOF3(Month, NYear) < MinQTb3) THEN
         NST3(Month + 1, NYear) = RC31(Month + 1) + NOF3(Month, NYear)
         NOF3(Month, NYear) = 0
         IF NST3(Month + 1, NYear) > NST3Max THEN
            Spill3(Month, NYear) = NST3(Month + 1, NYear) - NST3Max
            NST3(Month + 1, NYear) = NST3Max
            GOTO 101
         ELSE
         END IF
         GOTO 101
      ELSE
      END IF

      Head3(Month) = Coefc3 * (NST3(Month, NYear) + DNST3) ^ Coefd3 + Coefe3
      Head3(Month + 1) = Coefc3 * (RC31(Month + 1) + DNST3) ^ Coefd3 + Coefe3
      AvgHead3 = (Head3(Month) + Head3(Month + 1)) * .5

      QTb3 = (CapPlant3) / (CoefPlant3 * 9.81 * AvgHead3)

IF (NST3(Month, NYear) > NST3Max OR NST3(Month, NYear) = 0) THEN
     MaxQTb3 = 1453.483
   ELSE
     MaxQTb3 = QTb3 * 2.628
END IF
```

```
        IF (NOF3(Month, NYear) > MaxQTb3) THEN
          DNOFMax = NOF3(Month, NYear) - MaxQTb3
          NOF3(Month, NYear) = MaxQTb3
        ELSE
          DNOFMax = 0
        END IF

              NST3(Month + 1, NYear) = RC31(Month + 1) + DNOFMax
              IF NST3(Month + 1, NYear) > NST3Max THEN
                  Spill3(Month, NYear) = NST3(Month + 1, NYear) - NST3Max
                  NST3(Month + 1, NYear) = NST3Max
              ELSE
                  Spill3(Month, NYear) = 0
              END IF

       Head3(Month) = Coefc3 * (NST3(Month, NYear) + DNST3) ^ Coefd3 + Coefe3
       Head3(Month + 1) = Coefc3 * (NST3(Month + 1, NYear) + DNST3) ^ Coefd3 + Coefe3

  IF NST3(Month + 1, NYear) = 0 THEN
     Head3(Month + 1) = Head3Min
  ELSE
  END IF

  IF (NST3(Month + 1, NYear) >= NST3Max) THEN
     Head3(Month + 1) = Head3Max
  ELSE
  END IF

  IF NST3(Month, NYear) = 0 THEN
     Head3(Month) = Head3Min
  ELSE
  END IF

  IF (NST3(Month, NYear) >= NST3Max) THEN
     Head3(Month) = Head3Max
  ELSE
  END IF
```

```
            AvgHead3 = (Head3(Month) + Head3(Month + 1)) * .5

101      LEG3(Month, NYear) = (2.725 / 1000) * CoefPlant3 * AvgHead3 * NOF3(Month, NYear)

         TNOF3(Month, NYear) = NOF3(Month, NYear) + Spill3(Month, NYear)

         TNIF(NYear) = TNIF(NYear) + NIF3(Month, NYear)
         TNEV(NYear) = TNEV(NYear) + NEV3(Month, NYear)
         TSpill(NYear) = TSpill(NYear) + Spill3(Month, NYear)
         TNOF(NYear) = TNOF(NYear) + NOF3(Month, NYear)

          WRITE #5, NIF3(Month, NYear), NST3(Month, NYear), NOF3(Month, NYear), NEV3(Month,
NYear), Spill3(Month, NYear)
          WRITE #6, LEG3(Month, NYear)
          WRITE #10, TNOF3(Month, NYear)

          IF (NYear = NTest AND NR = NRTest) THEN
            PRINT #4, USING b$; NR; NYear; Month; NIF3(Month, NYear); NST3(Month, NYear);
NOF3(Month, NYear); NEV3(Month, NYear); Spill3(Month, NYear); Head3(Month); AvgHead3;
LEG3(Month, NYear)
          ELSE
          END IF

          IF (NYear = NTest + 1) AND (NR = NRTest) THEN
             IF (NYear = NTest + 1) AND (NR = NRTest) AND (Month = NTest) THEN
                PRINT #4, "--------------------------------------------------------------"
                ELSE
                END IF
            PRINT #4, USING b$; NR; NYear; Month; NIF3(Month, NYear); NST3(Month, NYear);
NOF3(Month, NYear); NEV3(Month, NYear); Spill3(Month, NYear); Head3(Month); AvgHead3;
LEG3(Month, NYear)
          ELSE
          END IF

      NEXT Month

         CheckTDNST(NYear) = NST3(1, NYear) - NST3(13, NYear) + TNIF(NYear) - TNEV(NYear)
```

```
TSpill(NYear) - TNOF(NYear)
         NST3(1, NYear + 1) = NST3(13, NYear)

        IF (NYear = NTest AND NR = NRTest) THEN
           PRINT #4, " NR   NY  Month   NST3(1)    NST3(13)    TNIF    TNOF    TNEV    TSpill
 CheckTDNST"
           PRINT #4, USING b$; NR; NYear; NMonth; NST3(1, NYear); NST3(13, NYear);
TNIF(NYear); TNOF(NYear); TNEV(NYear); TSpill(NYear); CheckTDNST(NYear)
        ELSE
        END IF
        WRITE #51, NST3(13, NYear), CheckTDNST(NYear)
NEXT NYear
RETURN
'********************************************************
'*                       Wrap-Up
'********************************************************
Wrap.Up.CirCtr:

     CLOSE #1
     CLOSE #5

RETURN


'ACirCtr: Analysis of CirCtr
'*****************************************************
'                Initiation
'*****************************************************
Initialization.ACirCtr:

' $DYNAMIC

IMonth = Month: NIYear = MYear: IReplicate = NReplicate

DIM Blank AS STRING, TimeStep AS STRING, Replicate AS STRING

DIM   Min(IMonth),   Max(IMonth),   MinNIF3(IMonth),   MaxNIF3(IMonth),   MinNST3(IMonth),
```

```
MaxNST3(IMonth)
DIM MinNOF3(IMonth), MaxNOF3(IMonth), MinNEV3(IMonth), MaxNEV3(IMonth), MinSpill3(IMonth),
MaxSpill3(IMonth)
DIM SNIF3(IMonth, IReplicate), ANIF3(IMonth, IReplicate), SDNIF3S(IMonth, IReplicate),
VNIF3(IMonth, IReplicate)
DIM StdNIF3(IMonth, IReplicate), DNIF3S(IMonth, IReplicate), SNST3(IMonth, IReplicate)
DIM ANST3(IMonth, IReplicate), SDNST3S(IMonth, IReplicate), VNST3(IMonth, IReplicate)
DIM StdNST3(IMonth, IReplicate), DNST3S(IMonth, IReplicate), SNOF3(IMonth, IReplicate)
DIM ANOF3(IMonth, IReplicate), SDNOF3S(IMonth, IReplicate), VNOF3(IMonth, IReplicate)
DIM StdNOF3(IMonth, IReplicate), DNOF3S(IMonth, IReplicate), SNEV3(IMonth, IReplicate),
ANEV3(IMonth, IReplicate)
DIM SDNEV3S(IMonth, IReplicate), VNEV3(IMonth, IReplicate), StdNEV3(IMonth, IReplicate)
DIM DNEV3S(IMonth, IReplicate), SSpill3(IMonth, IReplicate), ASpill3(IMonth, IReplicate),
SDSpill3S(IMonth, IReplicate)
DIM VSpill3(IMonth, IReplicate), StdSpill3(IMonth, IReplicate), DSpill3S(IMonth, IReplicate)

a$ = "#### ######.## ######.## ######.##"

RETURN
'*************************************************************************************
'                         Statistics.Calc
'*************************************************************************************
Statistics.Calc.ACirCtr:

OPEN "c:\Cir\CirInp1.out" FOR INPUT AS #36  'input of stat.calc

   FOR NYear = 1 TO NIYear
      FOR Month = 1 TO 12
         INPUT #36, NIF3(Month, NYear), NST3(Month, NYear), NOF3(Month, NYear), NEV3(Month,
NYear), Spill3(Month, NYear)
      NEXT Month
   NEXT NYear

         LOCATE 11, 10
PRINT "Stat.Calc           ---> NR       = "; NR

         GOSUB MaxMin.Calc
```

```
        GOSUB Avg.Calc
        GOSUB Sum.Calc
        GOSUB Std.Calc
        GOSUB OutStat.Calc

CLOSE #36

RETURN

'------------------- Max and Min
MaxMin.Calc:

        FOR Month = 1 TO 12
           MinNIF3(Month) = 10000
           MaxNIF3(Month) = 0
           MinNST3(Month) = 10000
           MaxNST3(Month) = 0
           MinNOF3(Month) = 10000
           MaxNOF3(Month) = 0
           MinNEV3(Month) = 10000
           MaxNEV3(Month) = 0
           MinSpill3(Month) = 10000
           MaxSpill3(Month) = 0
        NEXT Month

        FOR Month = 1 TO 12
           FOR NYear = 1 TO NIYear
                   IF NIF3(Month, NYear) < MinNIF3(Month) THEN
                      MinNIF3(Month) = NIF3(Month, NYear)
                      ELSE
                   END IF
                   IF NIF3(Month, NYear) > MaxNIF3(Month) THEN
                      MaxNIF3(Month) = NIF3(Month, NYear)
                      ELSE
                   END IF
                   IF NST3(Month, NYear) < MinNST3(Month) THEN
                      MinNST3(Month) = NST3(Month, NYear)
```

```
                ELSE
             END IF
             IF NST3(Month, NYear) > MaxNST3(Month) THEN
                MaxNST3(Month) = NST3(Month, NYear)
                ELSE
             END IF
             IF NOF3(Month, NYear) < MinNOF3(Month) THEN
                MinNOF3(Month) = NOF3(Month, NYear)
                ELSE
             END IF
             IF NOF3(Month, NYear) > MaxNOF3(Month) THEN
                MaxNOF3(Month) = NOF3(Month, NYear)
                ELSE
             END IF
             IF NEV3(Month, NYear) < MinNEV3(Month) THEN
                MinNEV3(Month) = NEV3(Month, NYear)
                ELSE
             END IF
             IF NEV3(Month, NYear) > MaxNEV3(Month) THEN
                MaxNEV3(Month) = NEV3(Month, NYear)
                ELSE
             END IF
             IF Spill3(Month, NYear) < MinSpill3(Month) THEN
                MinSpill3(Month) = Spill3(Month, NYear)
                ELSE
             END IF
             IF Spill3(Month, NYear) > MaxSpill3(Month) THEN
                MaxSpill3(Month) = Spill3(Month, NYear)
                ELSE
             END IF
         NEXT NYear
      NEXT Month

RETURN

'------------------------------ Avg.Calc
Avg.Calc:
```

```
'---------- Average of NEV3

        FOR Month = 1 TO 12
           SNIF3(Month, NR) = 0
           SNST3(Month, NR) = 0
           SNOF3(Month, NR) = 0
           SNEV3(Month, NR) = 0
           SSpill3(Month, NR) = 0
        NEXT Month

     FOR NYear = 1 TO NIYear
        FOR Month = 1 TO 12
           SNIF3(Month, NR) = SNIF3(Month, NR) + NIF3(Month, NYear)
           SNST3(Month, NR) = SNST3(Month, NR) + NST3(Month, NYear)
           SNOF3(Month, NR) = SNOF3(Month, NR) + NOF3(Month, NYear)
           SNEV3(Month, NR) = SNEV3(Month, NR) + NEV3(Month, NYear)
           SSpill3(Month, NR) = SSpill3(Month, NR) + Spill3(Month, NYear)
        NEXT Month
     NEXT NYear

        FOR Month = 1 TO 12
           ANIF3(Month, NR) = SNIF3(Month, NR) / NIYear
           ANST3(Month, NR) = SNST3(Month, NR) / NIYear
           ANOF3(Month, NR) = SNOF3(Month, NR) / NIYear
           ANEV3(Month, NR) = SNEV3(Month, NR) / NIYear
           ASpill3(Month, NR) = SSpill3(Month, NR) / NIYear
        NEXT Month


RETURN
'-------------------------------Sumation
Sum.Calc:

        FOR Month = 1 TO 12
           SDNIF3S(Month, NR) = 0
           SDNST3S(Month, NR) = 0
           SDNOF3S(Month, NR) = 0
```

```
            SDNEV3S(Month, NR) = 0
            SDSpill3S(Month, NR) = 0
         NEXT Month

      FOR NYear = 1 TO NIYear
         FOR Month = 1 TO 12
            DNIF3S(Month, NR) = (NIF3(Month, NYear) - ANIF3(Month, NR)) ^ 2
            SDNIF3S(Month, NR) = SDNIF3S(Month, NR) + DNIF3S(Month, NR)
            DNST3S(Month, NR) = (NST3(Month, NYear) - ANST3(Month, NR)) ^ 2
            SDNST3S(Month, NR) = SDNST3S(Month, NR) + DNST3S(Month, NR)
            DNOF3S(Month, NR) = (NOF3(Month, NYear) - ANOF3(Month, NR)) ^ 2
            SDNOF3S(Month, NR) = SDNOF3S(Month, NR) + DNOF3S(Month, NR)
            DNEV3S(Month, NR) = (NEV3(Month, NYear) - ANEV3(Month, NR)) ^ 2
            SDNEV3S(Month, NR) = SDNEV3S(Month, NR) + DNEV3S(Month, NR)
            DSpill3S(Month, NR) = (Spill3(Month, NYear) - ASpill3(Month, NR)) ^ 2
            SDSpill3S(Month, NR) = SDSpill3S(Month, NR) + DSpill3S(Month, NR)
         NEXT Month
      NEXT NYear
RETURN
'-------------------------------StdNEV3
Std.Calc:

         FOR Month = 1 TO 12
            VNIF3(Month, NR) = SDNIF3S(Month, NR) / (NIYear - 1)
            StdNIF3(Month, NR) = (VNIF3(Month, NR)) ^ .5
            VNST3(Month, NR) = SDNST3S(Month, NR) / (NIYear - 1)
            StdNST3(Month, NR) = (VNST3(Month, NR)) ^ .5
            VNOF3(Month, NR) = SDNOF3S(Month, NR) / (NIYear - 1)
            StdNOF3(Month, NR) = (VNOF3(Month, NR)) ^ .5
            VNEV3(Month, NR) = SDNEV3S(Month, NR) / (NIYear - 1)
            StdNEV3(Month, NR) = (VNEV3(Month, NR)) ^ .5
            VSpill3(Month, NR) = SDSpill3S(Month, NR) / (NIYear - 1)
            StdSpill3(Month, NR) = (VSpill3(Month, NR)) ^ .5
         NEXT Month


RETURN
```

```
OutStat.Calc:
IF (NR = NRTest) THEN
        PRINT #17,
        PRINT #17,
        PRINT #17, "Agung W.H. SOEHARNO"
        PRINT #17, DATE$, TIME$
                                                   P    R    I    N    T    #    1    7    ,
"*******************************************************************************************"
        PRINT #17, "*                Output of Cir-Ctr Program"
        PRINT #17, "*                Node 3 : C I R A T A"
        PRINT #17, "*"
        PRINT #17, "-----------------------> Stat.Calc - Replicate no. ", NR
                                                   P    R    I    N    T    #    1    7    ,
"*******************************************************************************************"
        PRINT #17,
        PRINT #17, " NIF3 = Inflow to Node 3: Cirata"
        PRINT #17,
        PRINT #17, "  NYear   May    Jun    Jul    Aug    Sep    Oct    Nov    Dec    Jan
  Feb     Mar    Apr"
        PRINT #17,
   FOR NYear = 1 TO NIYear
       PRINT #17, USING "####.##"; NYear; NIF3(1, NYear); NIF3(2, NYear); NIF3(3, NYear);
NIF3(4, NYear); NIF3(5, NYear); NIF3(6, NYear); NIF3(7, NYear); NIF3(8, NYear); NIF3(9,
NYear); NIF3(10, NYear); NIF3(11, NYear); NIF3(12, NYear)
   NEXT NYear
        PRINT #17,
        PRINT #17, "   NR     May    Jun    Jul    Aug    Sep    Oct    Nov    Dec    Jan
  Feb     Mar    Apr"
        PRINT #17, "Minimum of NIF3"
        PRINT #17, USING "####.##"; NR; MinNIF3(1); MinNIF3(2); MinNIF3(3); MinNIF3(4);
MinNIF3(5); MinNIF3(6); MinNIF3(7); MinNIF3(8); MinNIF3(9); MinNIF3(10); MinNIF3(11);
MinNIF3(12)
        PRINT #17, "ANIF3 = Average of NIF3"
        PRINT #17, USING "####.##"; NR; ANIF3(1, NR); ANIF3(2, NR); ANIF3(3, NR); ANIF3(4,
NR); ANIF3(5, NR); ANIF3(6, NR); ANIF3(7, NR); ANIF3(8, NR); ANIF3(9, NR); ANIF3(10, NR);
ANIF3(11, NR); ANIF3(12, NR)
        PRINT #17, "Maximum of NIF3"
```

```
        PRINT #17, USING "####.##"; NR; MaxNIF3(1); MaxNIF3(2); MaxNIF3(3); MaxNIF3(5);
MaxNIF3(5); MaxNIF3(6); MaxNIF3(7); MaxNIF3(8); MaxNIF3(9); MaxNIF3(10); MaxNIF3(11);
MaxNIF3(12)
        PRINT #17, "StdNIF3 = Standard Dev. of NIF3"
        PRINT #17, USING "####.##"; NR; StdNIF3(1, NR); StdNIF3(2, NR); StdNIF3(3, NR);
StdNIF3(4, NR); StdNIF3(5, NR); StdNIF3(6, NR); StdNIF3(7, NR); StdNIF3(8, NR); StdNIF3(9,
NR); StdNIF3(10, NR); StdNIF3(11, NR); StdNIF3(12, NR)


        PRINT #17,
        PRINT #17, " NST3 = End of period Storage of Node 3: Cirata"
        PRINT #17,
        PRINT #17, "  NYear   May    Jun    Jul    Aug    Sep    Oct    Nov    Dec    Jan
  Feb   Mar   Apr"
        PRINT #17,
   FOR NYear = 1 TO NIYear
       PRINT #17, USING "####.##"; NYear; NST3(1, NYear); NST3(2, NYear); NST3(3, NYear);
NST3(4, NYear); NST3(5, NYear); NST3(6, NYear); NST3(7, NYear); NST3(8, NYear); NST3(9,
NYear); NST3(10, NYear); NST3(11, NYear); NST3(12, NYear)
   NEXT NYear
        PRINT #17,
        PRINT #17, "   NR    May    Jun    Jul    Aug    Sep    Oct    Nov    Dec    Jan
  Feb   Mar   Apr"
        PRINT #17, "Minimum of NST3"
        PRINT #17, USING "####.##"; NR; MinNST3(1); MinNST3(2); MinNST3(3); MinNST3(4);
MinNST3(5); MinNST3(6); MinNST3(7); MinNST3(8); MinNST3(9); MinNST3(10); MinNST3(11);
MinNST3(12)
        PRINT #17, "ANST3 = Average of NST3"
        PRINT #17, USING "####.##"; NR; ANST3(1, NR); ANST3(2, NR); ANST3(3, NR); ANST3(4,
NR); ANST3(5, NR); ANST3(6, NR); ANST3(7, NR); ANST3(8, NR); ANST3(9, NR); ANST3(10, NR);
ANST3(11, NR); ANST3(12, NR)
        PRINT #17, "Maximum of NST3"
        PRINT #17, USING "####.##"; NR; MaxNST3(1); MaxNST3(2); MaxNST3(3); MaxNST3(5);
MaxNST3(5); MaxNST3(6); MaxNST3(7); MaxNST3(8); MaxNST3(9); MaxNST3(10); MaxNST3(11);
MaxNST3(12)
        PRINT #17, "StdNST3 = Standard Dev. of NST3"
        PRINT #17, USING "####.##"; NR; StdNST3(1, NR); StdNST3(2, NR); StdNST3(3, NR);
```

```
StdNST3(4, NR); StdNST3(5, NR); StdNST3(6, NR); StdNST3(7, NR); StdNST3(8, NR); StdNST3(9,
NR); StdNST3(10, NR); StdNST3(11, NR); StdNST3(12, NR)

        PRINT #17,
        PRINT #17, " NOF3 = OutFlow from Node 3: Cirata"
        PRINT #17,
        PRINT #17, "  NYear    May    Jun    Jul    Aug    Sep    Oct    Nov    Dec    Jan
  Feb    Mar    Apr"
        PRINT #17,
   FOR NYear = 1 TO NIYear
       PRINT #17, USING "####.##"; NYear; NOF3(1, NYear); NOF3(2, NYear); NOF3(3, NYear);
NOF3(4, NYear); NOF3(5, NYear); NOF3(6, NYear); NOF3(7, NYear); NOF3(8, NYear); NOF3(9,
NYear); NOF3(10, NYear); NOF3(11, NYear); NOF3(12, NYear)
   NEXT NYear
        PRINT #17,
        PRINT #17, "   NR      May    Jun    Jul    Aug    Sep    Oct    Nov    Dec    Jan
  Feb    Mar    Apr"
        PRINT #17, "Minimum of NOF3"
        PRINT #17, USING "####.##"; NR; MinNOF3(1); MinNOF3(2); MinNOF3(3); MinNOF3(4);
MinNOF3(5); MinNOF3(6); MinNOF3(7); MinNOF3(8); MinNOF3(9); MinNOF3(10); MinNOF3(11);
MinNOF3(12)
        PRINT #17, "ANOF3 = Average of NOF3"
        PRINT #17, USING "####.##"; NR; ANOF3(1, NR); ANOF3(2, NR); ANOF3(3, NR); ANOF3(4,
NR); ANOF3(5, NR); ANOF3(6, NR); ANOF3(7, NR); ANOF3(8, NR); ANOF3(9, NR); ANOF3(10, NR);
ANOF3(11, NR); ANOF3(12, NR)
        PRINT #17, "Maximum of NOF3"
        PRINT #17, USING "####.##"; NR; MaxNOF3(1); MaxNOF3(2); MaxNOF3(3); MaxNOF3(5);
MaxNOF3(5); MaxNOF3(6); MaxNOF3(7); MaxNOF3(8); MaxNOF3(9); MaxNOF3(10); MaxNOF3(11);
MaxNOF3(12)
        PRINT #17, "StdNOF3 = Standard Dev. of NOF3"
        PRINT #17, USING "####.##"; NR; StdNOF3(1, NR); StdNOF3(2, NR); StdNOF3(3, NR);
StdNOF3(4, NR); StdNOF3(5, NR); StdNOF3(6, NR); StdNOF3(7, NR); StdNOF3(8, NR); StdNOF3(9,
NR); StdNOF3(10, NR); StdNOF3(11, NR); StdNOF3(12, NR)

        PRINT #17,
        PRINT #17, "NEV3 = Evaporation of Node 3: Cirata"
        PRINT #17,
```

```
        PRINT #17, "  NYear    May    Jun    Jul    Aug    Sep    Oct    Nov    Dec    Jan
  Feb     Mar    Apr"
        PRINT #17,
   FOR NYear = 1 TO NIYear
       PRINT #17, USING "####.##"; NYear; NEV3(1, NYear); NEV3(2, NYear); NEV3(3, NYear);
NEV3(4, NYear); NEV3(5, NYear); NEV3(6, NYear); NEV3(7, NYear); NEV3(8, NYear); NEV3(9,
NYear); NEV3(10, NYear); NEV3(11, NYear); NEV3(12, NYear)
   NEXT NYear
        PRINT #17,
        PRINT #17, "   NR     May    Jun    Jul    Aug    Sep    Oct    Nov    Dec    Jan
  Feb     Mar    Apr"
        PRINT #17, "Minimum of NEV3"
        PRINT #17, USING "####.##"; NR; MinNEV3(1); MinNEV3(2); MinNEV3(3); MinNEV3(4);
MinNEV3(5);  MinNEV3(6);  MinNEV3(7);  MinNEV3(8);  MinNEV3(9);  MinNEV3(10);  MinNEV3(11);
MinNEV3(12)
        PRINT #17, "ANEV3 = Average of NEV3"
        PRINT #17, USING "####.##"; NR; ANEV3(1, NR); ANEV3(2, NR); ANEV3(3, NR); ANEV3(4,
NR); ANEV3(5, NR); ANEV3(6, NR); ANEV3(7, NR); ANEV3(8, NR); ANEV3(9, NR); ANEV3(10, NR);
ANEV3(11, NR); ANEV3(12, NR)
        PRINT #17, "Maximum of NEV3"
        PRINT #17, USING "####.##"; NR; MaxNEV3(1); MaxNEV3(2); MaxNEV3(3); MaxNEV3(5);
MaxNEV3(5);  MaxNEV3(6);  MaxNEV3(7);  MaxNEV3(8);  MaxNEV3(9);  MaxNEV3(10);  MaxNEV3(11);
MaxNEV3(12)
        PRINT #17, "StdNEV3 = Standard Dev. of NEV3"
        PRINT #17, USING "####.##"; NR; StdNEV3(1, NR); StdNEV3(2, NR); StdNEV3(3, NR);
StdNEV3(4, NR); StdNEV3(5, NR); StdNEV3(6, NR); StdNEV3(7, NR); StdNEV3(8, NR); StdNEV3(9,
NR); StdNEV3(10, NR); StdNEV3(11, NR); StdNEV3(12, NR)
        PRINT #17,
        PRINT #17,
        PRINT #17, "Spill3 = Spill flow of Node 3: Cirata"
        PRINT #17,
        PRINT #17, "  NYear    May    Jun    Jul    Aug    Sep    Oct    Nov    Dec    Jan
  Feb     Mar    Apr"
        PRINT #17,
   FOR NYear = 1 TO NIYear
        PRINT #17, USING "####.##"; NYear; Spill3(1, NYear); Spill3(2, NYear); Spill3(3,
NYear); Spill3(4, NYear); Spill3(5, NYear); Spill3(6, NYear); Spill3(7, NYear); Spill3(8,
```

```
NYear); Spill3(9, NYear); Spill3(10, NYear); Spill3(11, NYear); _
Spill3(12, NYear)
   NEXT NYear
         PRINT #17,
         PRINT #17, "   NR      May     Jun     Jul     Aug     Sep     Oct     Nov     Dec     Jan
  Feb    Mar    Apr"
         PRINT #17, "Minimum of Spill3"
         PRINT  #17,  USING  "####.##";  NR;  MinSpill3(1);  MinSpill3(2);  MinSpill3(3);
MinSpill3(4);  MinSpill3(5);  MinSpill3(6);  MinSpill3(7);  MinSpill3(8);  MinSpill3(9);
MinSpill3(10); MinSpill3(11); MinSpill3(12)
         PRINT #17, "ASpill3 = Average of Spill3"
         PRINT #17, USING "####.##"; NR; ASpill3(1, NR); ASpill3(2, NR); ASpill3(3, NR);
ASpill3(4, NR); ASpill3(5, NR); ASpill3(6, NR); ASpill3(7, NR); ASpill3(8, NR); ASpill3(9,
NR); ASpill3(10, NR); ASpill3(11, NR); ASpill3(12, NR)
         PRINT #17, "Maximum of Spill3"
         PRINT  #17,  USING  "####.##";  NR;  MaxSpill3(1);  MaxSpill3(2);  MaxSpill3(3);
MaxSpill3(5);  MaxSpill3(5);  MaxSpill3(6);  MaxSpill3(7);  MaxSpill3(8);  MaxSpill3(9);
MaxSpill3(10); MaxSpill3(11); MaxSpill3(12)
         PRINT #16,
         PRINT #16,
         PRINT #16,
         PRINT #16, "Agung W.H. SOEHARNO"
         PRINT #16, DATE$, TIME$, TIME$
                                                     P    R    I    N    T    #    1    6    ,
"*****************************************************************************************"
         PRINT #16, "*           Output of Cir-Ctr Program"
         PRINT #16, "*           Node 3 : C I R A T A"
         PRINT #16, "*"
         PRINT #16, "-----------------------> Stat.Calc - Replicate no. ", NR
                                                     P    R    I    N    T    #    1    6    ,
"*****************************************************************************************"
         PRINT #16,
         PRINT #16, " CRSUM = Summary File of Node 3: Cirata"
         PRINT #16,
         PRINT #16, "  NYear   May     Jun     Jul     Aug     Sep     Oct     Nov     Dec     Jan
  Feb    Mar    Apr"
         PRINT #16, "ANIF3 = Average of NIF3"
```

```
        PRINT #16, USING "####.##"; NR; ANIF3(1, NR); ANIF3(2, NR); ANIF3(3, NR); ANIF3(4,
NR); ANIF3(5, NR); ANIF3(6, NR); ANIF3(7, NR); ANIF3(8, NR); ANIF3(9, NR); ANIF3(10, NR);
ANIF3(11, NR); ANIF3(12, NR)
        PRINT #16, "ANST3 = Average of NST3"
        PRINT #16, USING "####.##"; NR; ANST3(1, NR); ANST3(2, NR); ANST3(3, NR); ANST3(4,
NR); ANST3(5, NR); ANST3(6, NR); ANST3(7, NR); ANST3(8, NR); ANST3(9, NR); ANST3(10, NR);
ANST3(11, NR); ANST3(12, NR)
        PRINT #16, "ANOF3 = Average of NOF3"
        PRINT #16, USING "####.##"; NR; ANOF3(1, NR); ANOF3(2, NR); ANOF3(3, NR); ANOF3(4,
NR); ANOF3(5, NR); ANOF3(6, NR); ANOF3(7, NR); ANOF3(8, NR); ANOF3(9, NR); ANOF3(10, NR);
ANOF3(11, NR); ANOF3(12, NR)
        PRINT #16, "ANEV3 = Average of NEV3"
        PRINT #16, USING "####.##"; NR; ANEV3(1, NR); ANEV3(2, NR); ANEV3(3, NR); ANEV3(4,
NR); ANEV3(5, NR); ANEV3(6, NR); ANEV3(7, NR); ANEV3(8, NR); ANEV3(9, NR); ANEV3(10, NR);
ANEV3(11, NR); ANEV3(12, NR)
        PRINT #16, "ASpill3 = Average of Spill3"
        PRINT #16, USING "####.##"; NR; ASpill3(1, NR); ASpill3(2, NR); ASpill3(3, NR);
ASpill3(4, NR); ASpill3(5, NR); ASpill3(6, NR); ASpill3(7, NR); ASpill3(8, NR); ASpill3(9,
NR); ASpill3(10, NR); ASpill3(11, NR); ASpill3(12, NR)
ELSE
END IF
        WRITE #38, ANIF3(1, NR), ANIF3(2, NR), ANIF3(3, NR), ANIF3(4, NR), ANIF3(5, NR),
ANIF3(6, NR), ANIF3(7, NR), ANIF3(8, NR), ANIF3(9, NR), ANIF3(10, NR), ANIF3(11, NR),
ANIF3(12, NR)
        WRITE #38, ANST3(1, NR), ANST3(2, NR), ANST3(3, NR), ANST3(4, NR), ANST3(5, NR),
ANST3(6, NR), ANST3(7, NR), ANST3(8, NR), ANST3(9, NR), ANST3(10, NR), ANST3(11, NR),
ANST3(12, NR)
        WRITE #38, ANOF3(1, NR), ANOF3(2, NR), ANOF3(3, NR), ANOF3(4, NR), ANOF3(5, NR),
ANOF3(6, NR), ANOF3(7, NR), ANOF3(8, NR), ANOF3(9, NR), ANOF3(10, NR), ANOF3(11, NR),
ANOF3(12, NR)
        WRITE #38, ANEV3(1, NR), ANEV3(2, NR), ANEV3(3, NR), ANEV3(4, NR), ANEV3(5, NR),
ANEV3(6, NR), ANEV3(7, NR), ANEV3(8, NR), ANEV3(9, NR), ANEV3(10, NR), ANEV3(11, NR),
ANEV3(12, NR)
        WRITE #38, ASpill3(1, NR), ASpill3(2, NR), ASpill3(3, NR), ASpill3(4, NR),
ASpill3(5, NR), ASpill3(6, NR), ASpill3(7, NR), ASpill3(8, NR), ASpill3(9, NR), ASpill3(10,
NR), ASpill3(11, NR), ASpill3(12, NR)
```

```
        WRITE #38, StdNIF3(1, NR), StdNIF3(2, NR), StdNIF3(3, NR), StdNIF3(4, NR),
StdNIF3(5, NR), StdNIF3(6, NR), StdNIF3(7, NR), StdNIF3(8, NR), StdNIF3(9, NR), StdNIF3(10,
NR), StdNIF3(11, NR), StdNIF3(12, NR)
        WRITE #38, StdNST3(1, NR), StdNST3(2, NR), StdNST3(3, NR), StdNST3(4, NR),
StdNST3(5, NR), StdNST3(6, NR), StdNST3(7, NR), StdNST3(8, NR), StdNST3(9, NR), StdNST3(10,
NR), StdNST3(11, NR), StdNST3(12, NR)
        WRITE #38, StdNOF3(1, NR), StdNOF3(2, NR), StdNOF3(3, NR), StdNOF3(4, NR),
StdNOF3(5, NR), StdNOF3(6, NR), StdNOF3(7, NR), StdNOF3(8, NR), StdNOF3(9, NR), StdNOF3(10,
NR), StdNOF3(11, NR), StdNOF3(12, NR)
        WRITE #38, StdNEV3(1, NR), StdNEV3(2, NR), StdNEV3(3, NR), StdNEV3(4, NR),
StdNEV3(5, NR), StdNEV3(6, NR), StdNEV3(7, NR), StdNEV3(8, NR), StdNEV3(9, NR), StdNEV3(10,
NR), StdNEV3(11, NR), StdNEV3(12, NR)
        WRITE #38, StdSpill3(1, NR), StdSpill3(2, NR), StdSpill3(3, NR), StdSpill3(4, NR),
StdSpill3(5, NR), StdSpill3(6, NR), StdSpill3(7, NR), StdSpill3(8, NR), StdSpill3(9, NR),
StdSpill3(10, NR), StdSpill3(11, NR), StdSpill3(12, NR)

RETURN

Outstat.calc.test:
FOR NR = 1 TO IReplicate
IF (NR = 1) THEN
        PRINT #17,
        PRINT #17, "   NR      May     Jun     Jul     Aug     Sep     Oct     Nov     Dec     Jan
  Feb     Mar     Apr"
        PRINT #17, "ANIF3 = Average of NIF3"
ELSE
END IF
        PRINT #17, USING "####.##"; NR; ANIF3(1, NR); ANIF3(2, NR); ANIF3(3, NR); ANIF3(4,
NR); ANIF3(5, NR); ANIF3(6, NR); ANIF3(7, NR); ANIF3(8, NR); ANIF3(9, NR); ANIF3(10, NR);
ANIF3(11, NR); ANIF3(12, NR)
NEXT NR

FOR NR = 1 TO IReplicate
IF (NR = 1) THEN
        PRINT #17,
        PRINT #17, "   NR      May     Jun     Jul     Aug     Sep     Oct     Nov     Dec     Jan
  Feb     Mar     Apr"
```

```
        PRINT #17, "ANST3 = Average of NST3"
ELSE
END IF
        PRINT #17, USING "####.##"; NR; ANST3(1, NR); ANST3(2, NR); ANST3(3, NR); ANST3(4,
NR); ANST3(5, NR); ANST3(6, NR); ANST3(7, NR); ANST3(8, NR); ANST3(9, NR); ANST3(10, NR);
ANST3(11, NR); ANST3(12, NR)
NEXT NR

FOR NR = 1 TO IReplicate
IF (NR = 1) THEN
        PRINT #17,
        PRINT #17, "    NR      May      Jun      Jul      Aug      Sep      Oct      Nov      Dec      Jan
  Feb     Mar     Apr"
        PRINT #17, "ANEV3 = Average of NEV3"
ELSE
END IF
        PRINT #17, USING "####.##"; NR; ANEV3(1, NR); ANEV3(2, NR); ANEV3(3, NR); ANEV3(4,
NR); ANEV3(5, NR); ANEV3(6, NR); ANEV3(/, NR); ANEV3(8, NR); ANEV3(9, NR); ANEV3(10, NR);
ANEV3(11, NR); ANEV3(12, NR)
NEXT NR

FOR NR = 1 TO IReplicate
IF (NR = 1) THEN
        PRINT #17,
        PRINT #17, "    NR      May      Jun      Jul      Aug      Sep      Oct      Nov      Dec      Jan
  Feb     Mar     Apr"
        PRINT #17, "ASpill3 = Average of Spill3"
ELSE
END IF
        PRINT #17, USING "####.##"; NR; ASpill3(1, NR); ASpill3(2, NR); ASpill3(3, NR);
ASpill3(4, NR); ASpill3(5, NR); ASpill3(6, NR); ASpill3(7, NR); ASpill3(8, NR); ASpill3(9,
NR); ASpill3(10, NR); ASpill3(11, NR); ASpill3(12, NR)
NEXT NR
RETURN

'****************************************************************************
'                         Statistics.TESt
```

```
'**************************************************************************
Statistics.Test.ACirCtr:
LOCATE 13, 10
PRINT "S t a t i s t i c s . T e s t"
IF (NR = NRTest) THEN
PRINT #17, "-------------> S t a t i s t i c s . T e s t"
PRINT #17,
PRINT #17, "   May    Jun    Jul    Aug    Sep    Oct    Nov    Dec    Jan    Feb    Mar
Apr"
PRINT #17,
ELSE
END IF

OPEN "c:\Cir\CrInput3.dat" FOR INPUT AS #39  'input of stat.tes

GOSUB Input.Data
GOSUB AvgAvg.Calc
GOSUB SumSum.Calc
GOSUB StdStd.Calc
GOSUB PrintStat.Tes

CLOSE #39
RETURN

'------------------------------ Input.Data
Input.Data:

     FOR NR = 1 TO IReplicate
         INPUT #39, ANIF3(1, NR), ANIF3(2, NR), ANIF3(3, NR), ANIF3(4, NR), ANIF3(5, NR),
ANIF3(6, NR), ANIF3(7, NR), ANIF3(8, NR), ANIF3(9, NR), ANIF3(10, NR), ANIF3(11, NR),
ANIF3(12, NR)
         INPUT #39, ANST3(1, NR), ANST3(2, NR), ANST3(3, NR), ANST3(4, NR), ANST3(5, NR),
ANST3(6, NR), ANST3(7, NR), ANST3(8, NR), ANST3(9, NR), ANST3(10, NR), ANST3(11, NR),
ANST3(12, NR)
         INPUT #39, ANOF3(1, NR), ANOF3(2, NR), ANOF3(3, NR), ANOF3(4, NR), ANOF3(5, NR),
ANOF3(6, NR), ANOF3(7, NR), ANOF3(8, NR), ANOF3(9, NR), ANOF3(10, NR), ANOF3(11, NR),
ANOF3(12, NR)
```

```
        INPUT #39, ANEV3(1, NR), ANEV3(2, NR), ANEV3(3, NR), ANEV3(4, NR), ANEV3(5, NR),
ANEV3(6, NR), ANEV3(7, NR), ANEV3(8, NR), ANEV3(9, NR), ANEV3(10, NR), ANEV3(11, NR),
ANEV3(12, NR)
        INPUT #39, ASpill3(1, NR), ASpill3(2, NR), ASpill3(3, NR), ASpill3(4, NR),
ASpill3(5, NR), ASpill3(6, NR), ASpill3(7, NR), ASpill3(8, NR), ASpill3(9, NR), ASpill3(10,
NR), ASpill3(11, NR), ASpill3(12, NR)

        INPUT #39, StdNIF3(1, NR), StdNIF3(2, NR), StdNIF3(3, NR), StdNIF3(4, NR),
StdNIF3(5, NR), StdNIF3(6, NR), StdNIF3(7, NR), StdNIF3(8, NR), StdNIF3(9, NR), StdNIF3(10,
NR), StdNIF3(11, NR), StdNIF3(12, NR)
        INPUT #39, StdNST3(1, NR), StdNST3(2, NR), StdNST3(3, NR), StdNST3(4, NR),
StdNST3(5, NR), StdNST3(6, NR), StdNST3(7, NR), StdNST3(8, NR), StdNST3(9, NR), StdNST3(10,
NR), StdNST3(11, NR), StdNST3(12, NR)
        INPUT #39, StdNOF3(1, NR), StdNOF3(2, NR), StdNOF3(3, NR), StdNOF3(4, NR),
StdNOF3(5, NR), StdNOF3(6, NR), StdNOF3(7, NR), StdNOF3(8, NR), StdNOF3(9, NR), StdNOF3(10,
NR), StdNOF3(11, NR), StdNOF3(12, NR)
        INPUT #39, StdNEV3(1, NR), StdNEV3(2, NR), StdNEV3(3, NR), StdNEV3(4, NR),
StdNEV3(5, NR), StdNEV3(6, NR), StdNEV3(7, NR), StdNEV3(8, NR), StdNEV3(9, NR), StdNEV3(10,
NR), StdNEV3(11, NR), StdNEV3(12, NR)
        INPUT #39, StdSpill3(1, NR), StdSpill3(2, NR), StdSpill3(3, NR), StdSpill3(4, NR),
StdSpill3(5, NR), StdSpill3(6, NR), StdSpill3(7, NR), StdSpill3(8, NR), StdSpill3(9, NR),
StdSpill3(10, NR), StdSpill3(11, NR), StdSpill3(12, NR)
     NEXT NR

RETURN
'------------------------------ AvgAvg.Calc
AvgAvg.Calc:
DIM SANIF3(12), SStdNIF3(12), AANIF3(12), AStdNIF3(12), SANST3(12)
DIM SStdNST3(12), AANST3(12), AStdNST3(12), SANOF3(12), SStdNOF3(12), AANOF3(12)
DIM AStdNOF3(12), SANEV3(12), SStdNEV3(12), AANEV3(12), AStdNEV3(12)
DIM SASpill3(12), SStdSpill3(12), AASpill3(12), AStdSpill3(12)

'---------- Average of NEV3

        FOR Month = 1 TO 12
           SANIF3(Month) = 0
           SStdNIF3(Month) = 0
```

```
      SANST3(Month) = 0
      SStdNST3(Month) = 0
      SANOF3(Month) = 0
      SStdNOF3(Month) = 0
      SANEV3(Month) = 0
      SStdNEV3(Month) = 0
      SASpill3(Month) = 0
      SStdSpill3(Month) = 0
   NEXT Month

FOR Month = 1 TO 12
   FOR NR = 1 TO IReplicate
      SANIF3(Month) = SANIF3(Month) + ANIF3(Month, NR)
      SStdNIF3(Month) = SStdNIF3(Month) + StdNIF3(Month, NR)
      SANST3(Month) = SANST3(Month) + ANST3(Month, NR)
      SStdNST3(Month) = SStdNST3(Month) + StdNST3(Month, NR)
      SANOF3(Month) = SANOF3(Month) + ANOF3(Month, NR)
      SStdNOF3(Month) = SStdNOF3(Month) + StdNOF3(Month, NR)
      SANEV3(Month) = SANEV3(Month) + ANEV3(Month, NR)
      SStdNEV3(Month) = SStdNEV3(Month) + StdNEV3(Month, NR)
      SASpill3(Month) = SASpill3(Month) + ASpill3(Month, NR)
      SStdSpill3(Month) = SStdSpill3(Month) + StdSpill3(Month, NR)
   NEXT NR
NEXT Month

   FOR Month = 1 TO 12
      AANIF3(Month) = SANIF3(Month) / IReplicate
      AStdNIF3(Month) = SStdNIF3(Month) / IReplicate
      AANST3(Month) = SANST3(Month) / IReplicate
      AStdNST3(Month) = SStdNST3(Month) / IReplicate
      AANOF3(Month) = SANOF3(Month) / IReplicate
      AStdNOF3(Month) = SStdNOF3(Month) / IReplicate
      AANEV3(Month) = SANEV3(Month) / IReplicate
      AStdNEV3(Month) = SStdNEV3(Month) / IReplicate
      AASpill3(Month) = SASpill3(Month) / IReplicate
      AStdSpill3(Month) = SStdSpill3(Month) / IReplicate
   NEXT Month
```

```
RETURN

'------------------------------Sumation of Sumation
SumSum.Calc:
DIM SDANIF3S(12), SDStdNIF3S(12), DANIF3S(12), DStdNIF3S(12)
DIM SDANST3S(12), SDStdNST3S(12), DANST3S(12), DStdNST3S(12)
DIM SDANOF3S(12), SDStdNOF3S(12), DANOF3S(12), DStdNOF3S(12)
DIM SDANEV3S(12), SDStdNEV3S(12), DANEV3S(12), DStdNEV3S(12)
DIM SDASpill3S(12), SDStdSpill3S(12), DASpill3S(12), DStdSpill3S(12)

        FOR Month = 1 TO 12
           SDANIF3S(Month) = 0
           SDStdNIF3S(Month) = 0
           SDANST3S(Month) = 0
           SDStdNST3S(Month) = 0
           SDANOF3S(Month) = 0
           SDStdNOF3S(Month) = 0
           SDANEV3S(Month) = 0
           SDStdNEV3S(Month) = 0
           SDASpill3S(Month) = 0
           SDStdSpill3S(Month) = 0
        NEXT Month

     FOR Month = 1 TO 12
        FOR NR = 1 TO IReplicate
           DANIF3S(Month) = (ANIF3(Month, NR) - AANIF3(Month)) ^ 2
           SDANIF3S(Month) = SDANIF3S(Month) + DANIF3S(Month)
           DStdNIF3S(Month) = (StdNIF3(Month, NR) - AStdNIF3(Month)) ^ 2
           SDStdNIF3S(Month) = SDStdNIF3S(Month) + DStdNIF3S(Month)
           DANST3S(Month) = (ANST3(Month, NR) - AANST3(Month)) ^ 2
           SDANST3S(Month) = SDANST3S(Month) + DANST3S(Month)
           DStdNST3S(Month) = (StdNST3(Month, NR) - AStdNST3(Month)) ^ 2
           SDStdNST3S(Month) = SDStdNST3S(Month) + DStdNST3S(Month)
           DANOF3S(Month) = (ANOF3(Month, NR) - AANOF3(Month)) ^ 2
           SDANOF3S(Month) = SDANOF3S(Month) + DANOF3S(Month)
           DStdNOF3S(Month) = (StdNOF3(Month, NR) - AStdNOF3(Month)) ^ 2
```

```
            SDStdNOF3S(Month) = SDStdNOF3S(Month) + DStdNOF3S(Month)
            DANEV3S(Month) = (ANEV3(Month, NR) - AANEV3(Month)) ^ 2
            SDANEV3S(Month) = SDANEV3S(Month) + DANEV3S(Month)
            DStdNEV3S(Month) = (StdNEV3(Month, NR) - AStdNEV3(Month)) ^ 2
            SDStdNEV3S(Month) = SDStdNEV3S(Month) + DStdNEV3S(Month)
            DASpill3S(Month) = (ASpill3(Month, NR) - AASpill3(Month)) ^ 2
            SDASpill3S(Month) = SDASpill3S(Month) + DASpill3S(Month)
            DStdSpill3S(Month) = (StdSpill3(Month, NR) - AStdSpill3(Month)) ^ 2
            SDStdSpill3S(Month) = SDStdSpill3S(Month) + DStdSpill3S(Month)
         NEXT NR
      NEXT Month

10 RETURN
'--------------------------------Std of StdNEV3
StdStd.Calc:

DIM VANIF3(12), StdANIF3(12), VStdNIF3(12), StdStdNIF3(12)
DIM VANST3(12), StdANST3(12), VStdNST3(12), StdStdNST3(12)
DIM VANOF3(12), StdANOF3(12), VStdNOF3(12), StdStdNOF3(12)
DIM VANEV3(12), StdANEV3(12), VStdNEV3(12), StdStdNEV3(12)
DIM VASpill3(12), StdASpill3(12), VStdSpill3(12), StdStdSpill3(12)

         FOR Month = 1 TO 12
            VANIF3(Month) = SDANIF3S(Month) / (IReplicate - 1)
            StdANIF3(Month) = (VANIF3(Month)) ^ .5
            VStdNIF3(Month) = SDStdNIF3S(Month) / (IReplicate - 1)
            StdStdNIF3(Month) = (VStdNIF3(Month)) ^ .5

            VANST3(Month) = SDANST3S(Month) / (IReplicate - 1)
            StdANST3(Month) = (VANST3(Month)) ^ .5
            VStdNST3(Month) = SDStdNST3S(Month) / (IReplicate - 1)
            StdStdNST3(Month) = (VStdNST3(Month)) ^ .5

            VANOF3(Month) = SDANOF3S(Month) / (IReplicate - 1)
            StdANOF3(Month) = (VANOF3(Month)) ^ .5
            VStdNOF3(Month) = SDStdNOF3S(Month) / (IReplicate - 1)
            StdStdNOF3(Month) = (VStdNOF3(Month)) ^ .5
```

```
            VANEV3(Month) = SDANEV3S(Month) / (IReplicate - 1)
            StdANEV3(Month) = (VANEV3(Month)) ^ .5
            VStdNEV3(Month) = SDStdNEV3S(Month) / (IReplicate - 1)
            StdStdNEV3(Month) = (VStdNEV3(Month)) ^ .5

            VASpill3(Month) = SDASpill3S(Month) / (IReplicate - 1)
            StdASpill3(Month) = (VASpill3(Month)) ^ .5
            VStdSpill3(Month) = SDStdSpill3S(Month) / (IReplicate - 1)
            StdStdSpill3(Month) = (VStdSpill3(Month)) ^ .5

        NEXT Month

RETURN
'------------------ PrintStat.tes
PrintStat.Tes:

IF (NR = NRTest) THEN

     PRINT #17, "AANIF3 = "
     PRINT #17, USING "####.##"; AANIF3(1); AANIF3(2); AANIF3(3); AANIF3(4); AANIF3(5);
AANIF3(6); AANIF3(7); AANIF3(8); AANIF3(9); AANIF3(10); AANIF3(11); AANIF3(12)
     PRINT #17, "AStdNIF3 ="
     PRINT #17, USING "####.##"; AStdNIF3(1); AStdNIF3(2); AStdNIF3(3); AStdNIF3(4);
AStdNIF3(5); AStdNIF3(6); AStdNIF3(7); AStdNIF3(8); AStdNIF3(9); AStdNIF3(10); AStdNIF3(11);
AStdNIF3(12)

     PRINT #17, "AANST3 = "
     PRINT #17, USING "####.##"; AANST3(1); AANST3(2); AANST3(3); AANST3(4); AANST3(5);
AANST3(6); AANST3(7); AANST3(8); AANST3(9); AANST3(10); AANST3(11); AANST3(12)
     PRINT #17, "AStdNST3 ="
     PRINT #17, USING "####.##"; AStdNST3(1); AStdNST3(2); AStdNST3(3); AStdNST3(4);
AStdNST3(5); AStdNST3(6); AStdNST3(7); AStdNST3(8); AStdNST3(9); AStdNST3(10); AStdNST3(11);
AStdNST3(12)

     PRINT #17, "AANOF3 = "
     PRINT #17, USING "####.##"; AANOF3(1); AANOF3(2); AANOF3(3); AANOF3(4); AANOF3(5);
AANOF3(6); AANOF3(7); AANOF3(8); AANOF3(9); AANOF3(10); AANOF3(11); AANOF3(12)
```

```
      PRINT #17, "AStdNOF3 ="
      PRINT #17, USING "####.##"; AStdNOF3(1); AStdNOF3(2); AStdNOF3(3); AStdNOF3(4);
AStdNOF3(5); AStdNOF3(6); AStdNOF3(7); AStdNOF3(8); AStdNOF3(9); AStdNOF3(10); AStdNOF3(11);
AStdNOF3(12)

      PRINT #17, "AANEV3 = "
      PRINT #17, USING "####.##"; AANEV3(1); AANEV3(2); AANEV3(3); AANEV3(4); AANEV3(5);
AANEV3(6); AANEV3(7); AANEV3(8); AANEV3(9); AANEV3(10); AANEV3(11); AANEV3(12)
      PRINT #17, "AStdNEV3 ="
      PRINT #17, USING "####.##"; AStdNEV3(1); AStdNEV3(2); AStdNEV3(3); AStdNEV3(4);
AStdNEV3(5); AStdNEV3(6); AStdNEV3(7); AStdNEV3(8); AStdNEV3(9); AStdNEV3(10); AStdNEV3(11);
AStdNEV3(12)

      PRINT #17, "AASpill3 = "
      PRINT #17, USING "####.##"; AASpill3(1); AASpill3(2); AASpill3(3); AASpill3(4);
AASpill3(5); AASpill3(6); AASpill3(7); AASpill3(8); AASpill3(9); AASpill3(10); AASpill3(11);
AASpill3(12)
      PRINT #17, "AStdSpill3 ="
      PRINT #17, USING "####.##"; AStdSpill3(1); AStdSpill3(2); AStdSpill3(3); AStdSpill3(4);
AStdSpill3(5); AStdSpill3(6); AStdSpill3(7); AStdSpill3(8); AStdSpill3(9); AStdSpill3(10);
AStdSpill3(11); AStdSpill3(12)

      PRINT #17, "StdANIF3 = "
      PRINT #17, USING "####.##"; StdANIF3(1); StdANIF3(2); StdANIF3(3); StdANIF3(4);
StdANIF3(5); StdANIF3(6); StdANIF3(7); StdANIF3(8); StdANIF3(9); StdANIF3(10); StdANIF3(11);
StdANIF3(12)
      PRINT #17, "StdStdNIF3 = "
      PRINT #17, USING "####.##"; StdStdNIF3(1); StdStdNIF3(2); StdStdNIF3(3); StdStdNIF3(4);
StdStdNIF3(5); StdStdNIF3(6); StdStdNIF3(7); StdStdNIF3(8); StdStdNIF3(9); StdStdNIF3(10);
StdStdNIF3(11); StdStdNIF3(12)

      PRINT #17, "StdANST3 = "
      PRINT #17, USING "####.##"; StdANST3(1); StdANST3(2); StdANST3(3); StdANST3(4);
StdANST3(5); StdANST3(6); StdANST3(7); StdANST3(8); StdANST3(9); StdANST3(10); StdANST3(11);
StdANST3(12)
      PRINT #17, "StdStdNST3 = "
      PRINT #17, USING "####.##"; StdStdNST3(1); StdStdNST3(2); StdStdNST3(3); StdStdNST3(4);
```

```
StdStdNST3(5); StdStdNST3(6); StdStdNST3(7); StdStdNST3(8); StdStdNST3(9); StdStdNST3(10);
StdStdNST3(11); StdStdNST3(12)

     PRINT #17, "StdANOF3 = "
     PRINT #17, USING "####.##"; StdANOF3(1); StdANOF3(2); StdANOF3(3); StdANOF3(4);
StdANOF3(5); StdANOF3(6); StdANOF3(7); StdANOF3(8); StdANOF3(9); StdANOF3(10); StdANOF3(11);
StdANOF3(12)
     PRINT #17, "StdStdNOF3 = "
     PRINT #17, USING "####.##"; StdStdNOF3(1); StdStdNOF3(2); StdStdNOF3(3); StdStdNOF3(4);
StdStdNOF3(5); StdStdNOF3(6); StdStdNOF3(7); StdStdNOF3(8); StdStdNOF3(9); StdStdNOF3(10);
StdStdNOF3(11); StdStdNOF3(12)

     PRINT #17, "StdANEV3 = "
     PRINT #17, USING "####.##"; StdANEV3(1); StdANEV3(2); StdANEV3(3); StdANEV3(4);
StdANEV3(5); StdANEV3(6); StdANEV3(7); StdANEV3(8); StdANEV3(9); StdANEV3(10); StdANEV3(11);
StdANEV3(12)
     PRINT #17, "StdStdNEV3 = "
     PRINT #17, USING "####.##"; StdStdNEV3(1); StdStdNEV3(2); StdStdNEV3(3); StdStdNEV3(4);
StdStdNEV3(5); StdStdNEV3(6); StdStdNEV3(7); StdStdNEV3(8); StdStdNEV3(9); StdStdNEV3(10);
StdStdNEV3(11); StdStdNEV3(12)

     PRINT #17, "StdASpill3 = "
     PRINT #17, USING "####.##"; StdASpill3(1); StdASpill3(2); StdASpill3(3); StdASpill3(4);
StdASpill3(5); StdASpill3(6); StdASpill3(7); StdASpill3(8); StdASpill3(9); StdASpill3(10);
StdASpill3(11); StdASpill3(12)
     PRINT #17, "StdStdSpill3 = "
     PRINT #17, USING "####.##"; StdStdSpill3(1); StdStdSpill3(2); StdStdSpill3(3);
StdStdSpill3(4); StdStdSpill3(5); StdStdSpill3(6); StdStdSpill3(7); StdStdSpill3(8);
StdStdSpill3(9); StdStdSpill3(10); StdStdSpill3(11); StdStdSpill3(12)
ELSE
END IF
        PRINT #16,
        PRINT #16,
        PRINT #16, "Agung W.H. SOEHARNO"
        PRINT #16, DATE$, TIME$
                                                 P  R  I  N  T  #  1  6  ,
"*****************************************************************************************"
```

```
'JatCtr.Bas
'
'***************************************************************************
'                              Description of Variables
'***************************************************************************
'NIF1      -     inflows to Node 1: the Nanjung Gauge Site
'NIF4      - inflows to Node 4: the Jatiluhur Reservoir
'NST4      - reservoir storage Node 4
'NOF4      - turbine flows from Node 4
'NEV4      - evaporations Node 4
'LEG4      - energy produced by hydropower Node 4: the Jatiluhur Hydropower Plant
'ASur4     - reservoir surface area Node 4
'Head4     -     heads of hydropower Node 4
'AvgHead4 -     average of heads of the beginning and the end of month Node 4
'Spill4    - spill flows Node 4
'RC4       - rule curve or reservoir storage targets Node 4
'MinQTb4   - minimum flow of turbine Node 4
'CapPlant4-     installed capacity of hydropower Node 4
'DNST4     - dead storage of Node 4
'Coefa4, Coefb4            - constants for calculating reservoir surface area Node 4
'Coefc4, Coefd4, Coefe3  - constants for calculating hydropower plant head Node 4
'HtpFlow4      -     flow coefficient Node 4
'CoefPlant4 -  plant efficiency Node 4
'NR        - number of replications
'NYear     - number of year
'Month     - number of month
'Avg       - average
'Min       - minimum
'Max       - maximum
'Std       - standard deviation
'
```

```
        PRINT #16, "*              Output of ACir-Ctr Program"
        PRINT #16, "*              Node 3 : C I R A T A"
        PRINT #16, "*"
        PRINT #16, "----------------------> Statistics.Test "
                                                                P    R    I    N    T    #    1    6    ,
"*****************************************************************************************************"
        PRINT #16,
        PRINT #16, " May    Jun    Jul    Aug    Sep    Oct    Nov    Dec    Jan    Feb
 Mar    Apr"
        PRINT #16,

    PRINT #16, "AANIF3 = "
    PRINT #16, USING "####.##"; AANIF3(1); AANIF3(2); AANIF3(3); AANIF3(4); AANIF3(5);
AANIF3(6); AANIF3(7); AANIF3(8); AANIF3(9); AANIF3(10); AANIF3(11); AANIF3(12)

    PRINT #16, "AANST3 = "
    PRINT #16, USING "####.##"; AANST3(1); AANST3(2); AANST3(3); AANST3(4); AANST3(5);
AANST3(6); AANST3(7); AANST3(8); AANST3(9); AANST3(10); AANST3(11); AANST3(12)

    PRINT #16, "AANOF3 = "
    PRINT #16, USING "####.##"; AANOF3(1); AANOF3(2); AANOF3(3); AANOF3(4); AANOF3(5);
AANOF3(6); AANOF3(7); AANOF3(8); AANOF3(9); AANOF3(10); AANOF3(11); AANOF3(12)

    PRINT #16, "AANEV3 = "
    PRINT #16, USING "####.##"; AANEV3(1); AANEV3(2); AANEV3(3); AANEV3(4); AANEV3(5);
AANEV3(6); AANEV3(7); AANEV3(8); AANEV3(9); AANEV3(10); AANEV3(11); AANEV3(12)

    PRINT #16, "AASpill3 = "
    PRINT #16, USING "####.##"; AASpill3(1); AASpill3(2); AASpill3(3); AASpill3(4);
AASpill3(5); AASpill3(6); AASpill3(7); AASpill3(8); AASpill3(9); AASpill3(10); AASpill3(11);
AASpill3(12)

    PRINT #17, " The Program ACir-Crt is finish."
    PRINT #16, " The Program ACir-Crt is finish."

RETURN
```

```
'*****************************************************
'                  Main Module
'*****************************************************

CLS
LOCATE 1, 10
PRINT "Jat-Ctr Program."
GOSUB Initialization.JatCtr

FOR NR = 1 TO NReplicate

GOSUB Simulation.JatCtr
GOSUB Wrap.Up.JatCtr
LOCATE 7, 10
PRINT "The Program of Jat-Ctr is Finish."

IF NR = 1 THEN
        GOSUB Initialization.AJatCtr
ELSE
END IF

LOCATE 9, 10
PRINT "AJatCtr Program."

IF MYear = 1 THEN
   GOTO 100
ELSE
END IF

GOSUB Statistics.Calc.AJatCtr

IF IReplicate = 1 THEN
  GOTO 100
ELSE
END IF

NEXT NR
```

```
GOSUB Outstat.calc.test

CLOSE #38

GOSUB Statistics.Test.AJatCtr

100  LOCATE 17, 10
     PRINT "The Sub-Program of AJat-Ctr is finish."
     LOCATE 21, 10
     PRINT "Press CTRL-C to stop the program !"

END

'****************************************************
'                Initiation
'****************************************************
Initialization.JatCtr:
' $DYNAMIC

DIM TNST(61), TNSTm1(61), TNIF(61), TNEV(61), TSpill(61), TNOF(61), CheckTDNST(61)
DIM TNOF3(13, 61), TNOF4(13, 61), QRJat AS STRING, CRJat AS STRING, NIF1(13, 61)
DIM NIF4(13, 61), NST4(13, 61), NOF4(13, 61), NEV4(13, 61), LEG4(13, 61), ASur4(13)
DIM Head4(13), AvgHead4(13), Spill4(13, 61), CoefEvap4(13), QR41(13), RC41(13)

OPEN "c:\data\NRepl.dat" FOR INPUT AS #1  'Number of replication, nyear
OPEN "c:\data\TierGQM.dat" FOR INPUT AS #2 'Generated monthly flows
OPEN "c:\Jat\EvapJat.dat" FOR INPUT AS #3 'Coef. of evaporation of the Jatiluhur Res.
OPEN "c:\Jat\RCJat.dat" FOR INPUT AS #12 'Rule curve of the Jatiluhur Res.

OPEN "c:\Jat\JtStat.out" FOR OUTPUT AS #17  'Output of Stat.calc
OPEN "c:\sum\Jtsum.out" FOR OUTPUT AS #16 'Summary of Jatiluhur simulation
OPEN "c:\sum\noteJat.out" FOR OUTPUT AS #4 ' Note for Jatiluhur simulation

OPEN "c:\Jat\JatInp2.Out" FOR OUTPUT AS #6 'input-output file
OPEN "c:\Jat\JatJat.Out" FOR OUTPUT AS #10 'input-output file
OPEN "c:\Jat\JtInput3.dat" FOR OUTPUT AS #38 ' for output of stat.TES
```

```
OPEN "c:\Cir\CirJat.out" FOR INPUT AS #50 'outfput from Cirata simulation
OPEN "c:\Jat\JatAnn.Out" FOR OUTPUT AS #51 'simulated annual energy of the Jatiluhur Plant

        NRTest = 1
        NTest = 1
        MinQTb4 = 315
        NST4Max = 1800
        DNST4 = 1200
        Head4Min = 53
        Head4Max = 81
        Coefc4 = 2.15
        Coefd4 = .4523
        Coefa4 = .9605
        Coefb4 = .5546
        Coefe4 = .621406
        LMtpFlow4 = .33
        CoefPlant4 = .87


   PRINT #4, DATE$, TIME$
   PRINT #4,
   PRINT #4,
   PRINT #4, "**************** J A T I L U H U R ************************"
   PRINT #4,
   PRINT #4, "******* Print Out Test of Jat-Ctr Program for NYear = "; NTest
   PRINT #4,
        INPUT #1, Month, MYear, NReplicate
   PRINT #4, "Month, NYear, NReplicate"; Month; MYear; NReplicate
   PRINT #4,

            INPUT #3, EvapJat$
            PRINT #4, EvapJat$

                 c$ = "###.## ###.## ###.## ###.## ###.## ###.## ###.## ###.## ###.## ###.##
###.## ###.##"

            INPUT #3, CoefEvap4(1), CoefEvap4(2), CoefEvap4(3), CoefEvap4(4), CoefEvap4(5),
```

```
CoefEvap4(6), CoefEvap4(7), CoefEvap4(8), CoefEvap4(9), CoefEvap4(10), CoefEvap4(11),
CoefEvap4(12)
          PRINT #4, " CE(1) CE(2)  CE(3)   CE(4) CE(5)  CE(6)  CE(7)  CE(8)  CE(9)  CE(10)
CE(11) CE(12)"
          PRINT #4, USING c$; CoefEvap4(1); CoefEvap4(2); CoefEvap4(3); CoefEvap4(4);
CoefEvap4(5); CoefEvap4(6); CoefEvap4(7); CoefEvap4(8); CoefEvap4(9); CoefEvap4(10);
CoefEvap4(11); CoefEvap4(12)
          PRINT #4,

          INPUT #12, RCJat$
          INPUT #12, RC41(1), RC41(2), RC41(3), RC41(4), RC41(5), RC41(6), RC41(7),
RC41(8), RC41(9), RC41(10), RC41(11), RC41(12)

          PRINT #4,
          PRINT #4, RCJat$
          'PRINT #4, "RC(1) = RC(13) = NSTMax"
          PRINT #4, "RC4i1  RC4i2  RC4i3  RC4i4  RC4i5  RC4i6  RC4i7  RC4i8  RC4i9  RC4i10
RC4i11 RC4i12"
          PRINT #4, USING c$; RC41(1); RC41(2); RC41(3); RC41(4); RC41(5); RC41(6);
RC41(7); RC41(8); RC41(9); RC41(10); RC41(11); RC41(12)

          PRINT #4,
          PRINT #4, "Test of Reading Data of NIF1 for NYear = 1"
          PRINT #4, " NR    NY  NIF1(1) NIF1(2) NIF1(3) NIF1(4) NIF1(5) NIF1(6) NIF1(7)
NIF1(8) NIF1(9) NIF1(10) NIF1(11) NIF1(12)"
          a$ = "#### #### ####.## ####.## ####.## ####.## ####.## ####.## ####.## ####.##
####.## #####.## #####.## #####.##"
RETURN

'**********************************************************************
'               Simulation
'**********************************************************************
Simulation.JatCtr:
OPEN "c:\Jat\JatInpl.out" FOR OUTPUT AS #5 'input-output file

FOR NYear = 1 TO MYear
        TNST(NYear) = 0
```

```
        TNSTm1(NYear) = 0
        TNIF(NYear) = 0
        TNEV(NYear) = 0
        TSpill(NYear) = 0
        TNOF(NYear) = 0

       FOR Month = 1 TO 12
          INPUT #2, NR, NYear, NIF1(Month, NYear)
          IF NIF1(Month, NYear) < 0 THEN
             NIF1(Month, NYear) = 0
          ELSE
          END IF

        INPUT #50, TNOF3(Month, NYear)
        NIF4(Month, NYear) = LMtpFlow4 * NIF1(Month, NYear) + TNOF3(Month, NYear)
       NEXT Month

       IF (NR = NRTest) AND (NYear = NTest) THEN
          PRINT #4, USING a$; NR; NYear; NIF1(1, NYear); NIF1(2, NYear); NIF1(3, NYear);
NIF1(4, NYear); NIF1(5, NYear); NIF1(6, NYear); NIF1(7, NYear); NIF1(8, NYear); NIF1(9,
NYear); NIF1(10, NYear); NIF1(11, NYear); NIF1(12, NYear)
       ELSE
       END IF
NEXT NYear

   IF (NR = NRTest) THEN
        PRINT #4,
        PRINT #4, "NIF4(Month, NYear) = LMtpFlow4 * NIF1(Month, NYear) + TNOF3(Month, NYear)"
        PRINT #4, "LMtpFlow4 = "; LMtpFlow4
        PRINT #4,
            PRINT #4, " NR    NY   TNOF(1) TNOF(2) TNOF(3) TNOF(4) TNOF(5) TNOF(6) TNOF(7)
TNOF(8) TNOF(9) TNOF(10) TNOF(11) TNOF(12)"
        PRINT #4,
   ELSE
   END IF

LOCATE 3, 10
```

```
PRINT "Jat-Ctr,              ---> NR       ="; NR

        NST4(1, 1) = NST4Max
        RC41(13) = RC41(1)

FOR NYear = 1 TO MYear

        LOCATE 5, 10
        PRINT "Jat-Ctr,              ---> NYear    = "; NYear

   IF (NR = NRTest) AND (NYear = NTest) THEN
        PRINT #4, USING a$; NR; NYear; TNOF3(1, NYear); TNOF3(2, NYear); TNOF3(3, NYear);
TNOF3(4, NYear); TNOF3(5, NYear); TNOF3(6, NYear); TNOF3(7, NYear); TNOF3(8, NYear); TNOF3(9,
NYear); TNOF3(10, NYear); TNOF3(11, NYear); TNOF3(12, NYear)
        PRINT #4,
        PRINT #4, "Test output of Jat-Ctr program for NYear = "; NTest
        PRINT #4,
        PRINT #4, "NR   NY  Mth   NIF4   NST4     NOF4     NEV4   Spill4   Head4   AvgHead4
  LEG4 "
        b$ = "#### #### #### ####.## ####.## ####.## ####.##  ####.## #####.## ######.##
##########.##"
   ELSE
   END IF

     FOR Month = 1 TO 12

          ASur4(Month) = Coefa4 * (NST4(Month, NYear) + DNST4) ^ Coefb4
          ASur4(Month + 1) = Coefa4 * (RC41(Month + 1) + DNST4) ^ Coefb4
          NEV4(Month, NYear) = CoefEvap4(Month) * (ASur4(Month) + ASur4(Month + 1)) * .5

          NOF4(Month, NYear) = NST4(Month, NYear) - RC41(Month + 1) + NIF4(Month, NYear)
- NEV4(Month, NYear)

           IF NOF4(Month, NYear) < 0 THEN
              NST4(Month + 1, NYear) = RC41(Month + 1) + NOF4(Month, NYear)
              NOF4(Month, NYear) = 0
              Spill4(Month, NYear) = 0
```

```
   GOTO 101
ELSE
END IF

IF NOF4(Month, NYear) = 0 THEN
   NST4(Month + 1, NYear) = RC41(Month + 1) + NOF4(Month, NYear)
   NOF4(Month, NYear) = 0
   Spill4(Month, NYear) = 0
   GOTO 101
ELSE
END IF

IF ((0 < NOF4(Month, NYear)) AND (NOF4(Month, NYear) < MinQTb4)) THEN
   NST4(Month + 1, NYear) = RC41(Month + 1) + NOF4(Month, NYear)
   NOF4(Month, NYear) = 0
   IF NST4(Month + 1, NYear) > NST4Max THEN
      Spill4(Month, NYear) = NST4(Month + 1, NYear) - NST4Max
      NST4(Month + 1, NYear) = NST4Max
      GOTO 101
   ELSE
      Spill4(Month, NYear) = 0
   END IF
   GOTO 101
ELSE
END IF

Head4(Month) = Coefc4 * (NST4(Month, NYear) + DNST4) ^ Coefd4 + Coefe4
Head4(Month + 1) = Coefc4 * (NST4(Month + 1, NYear) + DNST4) ^ Coefd4 + Coefe4
AvgHead4 = (Head4(Month) + Head4(Month + 1)) * .5

QTb4 = (CapPlant4) / (CoefPlant4 * 9.81 * AvgHead4)

IF (NST4(Month, NYear) > 1165 OR NST4(Month, NYear) = 1165) THEN
      MaxQTb4 = 665.258
   ELSE
      CapPlant4 = NST4(Month, NYear) * 54.79 + 108758.2
      QTb4 = (CapPlant4) / (CoefPlant4 * 9.81 * AvgHead4)
```

```
      MaxQTb4 = QTb4 * 2.628
END IF

      IF (NOF4(Month, NYear) > MaxQTb4) THEN
        DNOFMax = NOF4(Month, NYear) - MaxQTb4
        NOF4(Month, NYear) = MaxQTb4
      ELSE
        DNOFMax = 0
      END IF

            NST4(Month + 1, NYear) = RC41(Month + 1) + DNOFMax

            IF NST4(Month + 1, NYear) > NST4Max THEN
                Spill4(Month, NYear) = NST4(Month + 1, NYear) - NST4Max
                NST4(Month + 1, NYear) = NST4Max
            ELSE
                Spill4(Month, NYear) = 0
            END IF

     Head4(Month) = Coefc4 * (NST4(Month, NYear) + DNST4) ^ Coefd4 + Coefe4
     Head4(Month + 1) = Coefc4 * (NST4(Month + 1, NYear) + DNST4) ^ Coefd4 + Coefe4

IF NST4(Month + 1, NYear) = 0 THEN
   Head4(Month + 1) = Head4Min
ELSE
END IF

IF (NST4(Month + 1, NYear) >= NST4Max) THEN
   Head4(Month + 1) = Head4Max
ELSE
END IF

IF NST4(Month, NYear) = 0 THEN
   Head4(Month) = Head4Min
ELSE
END IF
```

```
        IF (NST4(Month, NYear) >= NST4Max) THEN
           Head4(Month) = Head4Max
        ELSE
        END IF

             AvgHead4 = (Head4(Month) + Head4(Month + 1)) * .5

101     LEG4(Month, NYear) = (2.725 / 1000) * CoefPlant4 * AvgHead4 * NOF4(Month, NYear)

        TNOF4(Month, NYear) = NOF4(Month, NYear) + Spill4(Month, NYear)

        TNIF(NYear) = TNIF(NYear) + NIF4(Month, NYear)
        TNEV(NYear) = TNEV(NYear) + NEV4(Month, NYear)
        TSpill(NYear) = TSpill(NYear) + Spill4(Month, NYear)
        TNOF(NYear) = TNOF(NYear) + NOF4(Month, NYear)

         WRITE #5, NIF4(Month, NYear), NST4(Month, NYear), NOF4(Month, NYear), NEV4(Month,
NYear), Spill4(Month, NYear)
         WRITE #6, LEG4(Month, NYear)
         WRITE #10, TNOF4(Month, NYear)

         IF (NYear = NTest AND NR = NRTest) THEN
           PRINT #4, USING b$; NR; NYear; Month; NIF4(Month, NYear); NST4(Month, NYear);
NOF4(Month, NYear); NEV4(Month, NYear); Spill4(Month, NYear); Head4(Month); AvgHead4;
LEG4(Month, NYear)
         ELSE
         END IF

         IF (NYear = NTest + 1) AND (NR = NRTest) THEN
            IF (NYear = NTest + 1) AND (NR = NRTest) AND (Month = NTest) THEN
               PRINT #4, "------------------------------------------------------------"
               ELSE
               END IF
           PRINT #4, USING b$; NR; NYear; Month; NIF4(Month, NYear); NST4(Month, NYear);
NOF4(Month, NYear); NEV4(Month, NYear); Spill4(Month, NYear); Head4(Month); AvgHead4;
LEG4(Month, NYear)
         ELSE
```

```
        END IF

     NEXT Month

        CheckTDNST(NYear) = NST4(1, NYear) - NST4(13, NYear) + TNIF(NYear) - TNEV(NYear) -
TSpill(NYear) - TNOF(NYear)

         NST4(1, NYear + 1) = NST4(13, NYear)

        IF (NYear = NTest AND NR = NRTest) THEN
           PRINT #4, " NR    NY  Month NST4(1) NST4(13) TNIF TNOF TNEV  TSpill  CheckTDNST"
           PRINT #4, USING b$; NR; NYear; NMonth; NST4(1, NYear); NST4(13, NYear);
TNIF(NYear); TNOF(NYear); TNEV(NYear); TSpill(NYear); CheckTDNST(NYear)
        ELSE
        END IF
        WRITE #51, NST4(13, NYear), CheckTDNST(NYear)

NEXT NYear
RETURN
'*******************************************************
'*                       Wrap-Up
'*******************************************************
Wrap.Up.JatCtr:

     CLOSE #1
     CLOSE #5

RETURN


'AJatCtr: Analysis of JatCtr.
'****************************************************
'                Initiation
'****************************************************
Initialization.AJatCtr:

IMonth = Month: NIYear = MYear: IReplicate = NReplicate
```

```
DIM Blank AS STRING, TimeStep AS STRING, Replicate AS STRING, KStep0 AS INTEGER
DIM   Min(IMonth),   Max(IMonth),   MinNIF4(IMonth),   MaxNIF4(IMonth),   MinNST4(IMonth),
MaxNST4(IMonth)
DIM MinNOF4(IMonth), MaxNOF4(IMonth), MinNEV4(IMonth), MaxNEV4(IMonth), MinSpill4(IMonth),
MaxSpill4(IMonth)
DIM SNIF4(IMonth, IReplicate), ANIF4(IMonth, IReplicate), SDNIF4S(IMonth, IReplicate)
DIM VNIF4(IMonth, IReplicate), StdNIF4(IMonth, IReplicate), DNIF4S(IMonth, IReplicate)
DIM SNST4(IMonth, IReplicate), ANST4(IMonth, IReplicate), SDNST4S(IMonth, IReplicate)
DIM VNST4(IMonth, IReplicate), StdNST4(IMonth, IReplicate), DNST4S(IMonth, IReplicate)
DIM SNOF4(IMonth, IReplicate), ANOF4(IMonth, IReplicate), SDNOF4S(IMonth, IReplicate)
DIM VNOF4(IMonth, IReplicate), StdNOF4(IMonth, IReplicate), DNOF4S(IMonth, IReplicate)
DIM SNEV4(IMonth, IReplicate), ANEV4(IMonth, IReplicate), SDNEV4S(IMonth, IReplicate)
DIM VNEV4(IMonth, IReplicate), StdNEV4(IMonth, IReplicate), DNEV4S(IMonth, IReplicate)
DIM SSpill4(IMonth, IReplicate), ASpill4(IMonth, IReplicate), SDSpill4S(IMonth, IReplicate)
DIM VSpill4(IMonth, IReplicate), StdSpill4(IMonth, IReplicate), DSpill4S(IMonth, IReplicate)

a$ = "#### ######.## ######.## ######.##"

RETURN

'*****************************************************************************
'                              Statistics.Calc
'*****************************************************************************
Statistics.Calc.AJatCtr:

OPEN "c:\Jat\JatInp1.out" FOR INPUT AS #36  'input of stat.calc

   FOR NYear = 1 TO NIYear
      FOR Month = 1 TO 12
        INPUT #36, NIF4(Month, NYear), NST4(Month, NYear), NOF4(Month, NYear), NEV4(Month,
NYear), Spill4(Month, NYear)
      NEXT Month
   NEXT NYear

        LOCATE 11, 10
PRINT "Stat.Calc             ---> NR       = "; NR
```

```
        GOSUB MaxMin.Calc
        GOSUB Avg.Calc
        GOSUB Sum.Calc
        GOSUB Std.Calc
        GOSUB OutStat.Calc

CLOSE #36

RETURN

'------------------ Max and Min
MaxMin.Calc:

        FOR Month = 1 TO 12
           MinNIF4(Month) = 10000
           MaxNIF4(Month) = 0
           MinNST4(Month) = 10000
           MaxNST4(Month) = 0
           MinNOF4(Month) = 10000
           MaxNOF4(Month) = 0
           MinNEV4(Month) = 10000
           MaxNEV4(Month) = 0
           MinSpill4(Month) = 10000
           MaxSpill4(Month) = 0
        NEXT Month

        FOR Month = 1 TO 12
           FOR NYear = 1 TO NIYear
                      IF NIF4(Month, NYear) < MinNIF4(Month) THEN
                         MinNIF4(Month) = NIF4(Month, NYear)
                         ELSE
                      END IF
                      IF NIF4(Month, NYear) > MaxNIF4(Month) THEN
                         MaxNIF4(Month) = NIF4(Month, NYear)
                         ELSE
                      END IF
                      IF NST4(Month, NYear) < MinNST4(Month) THEN
```

```
            MinNST4(Month) = NST4(Month, NYear)
            ELSE
         END IF
         IF NST4(Month, NYear) > MaxNST4(Month) THEN
            MaxNST4(Month) = NST4(Month, NYear)
            ELSE
         END IF
         IF NOF4(Month, NYear) < MinNOF4(Month) THEN
            MinNOF4(Month) = NOF4(Month, NYear)
            ELSE
         END IF
         IF NOF4(Month, NYear) > MaxNOF4(Month) THEN
            MaxNOF4(Month) = NOF4(Month, NYear)
            ELSE
         END IF
         IF NEV4(Month, NYear) < MinNEV4(Month) THEN
            MinNEV4(Month) = NEV4(Month, NYear)
            ELSE
         END IF
         IF NEV4(Month, NYear) > MaxNEV4(Month) THEN
            MaxNEV4(Month) = NEV4(Month, NYear)
            ELSE
         END IF
         IF Spill4(Month, NYear) < MinSpill4(Month) THEN
            MinSpill4(Month) = Spill4(Month, NYear)
            ELSE
         END IF
         IF Spill4(Month, NYear) > MaxSpill4(Month) THEN
            MaxSpill4(Month) = Spill4(Month, NYear)
            ELSE
         END IF
      NEXT NYear
    NEXT Month

RETURN

'------------------------------ Avg.Calc
```

```
Avg.Calc:

'---------- Average of NEV4

        FOR Month = 1 TO 12
           SNIF4(Month, NR) = 0
           SNST4(Month, NR) = 0
           SNOF4(Month, NR) = 0
           SNEV4(Month, NR) = 0
           SSpill4(Month, NR) = 0
        NEXT Month

     FOR NYear = 1 TO NIYear
        FOR Month = 1 TO 12
           SNIF4(Month, NR) = SNIF4(Month, NR) + NIF4(Month, NYear)
           SNST4(Month, NR) = SNST4(Month, NR) + NST4(Month, NYear)
           SNOF4(Month, NR) = SNOF4(Month, NR) + NOF4(Month, NYear)
           SNEV4(Month, NR) = SNEV4(Month, NR) + NEV4(Month, NYear)
           SSpill4(Month, NR) = SSpill4(Month, NR) + Spill4(Month, NYear)
        NEXT Month
     NEXT NYear

        FOR Month = 1 TO 12
           ANIF4(Month, NR) = SNIF4(Month, NR) / NIYear
           ANST4(Month, NR) = SNST4(Month, NR) / NIYear
           ANOF4(Month, NR) = SNOF4(Month, NR) / NIYear
           ANEV4(Month, NR) = SNEV4(Month, NR) / NIYear
           ASpill4(Month, NR) = SSpill4(Month, NR) / NIYear
        NEXT Month

RETURN

'------------------------------Sumation
Sum.Calc:

        FOR Month = 1 TO 12
```

```
         SDNIF4S(Month, NR) = 0
         SDNST4S(Month, NR) = 0
         SDNOF4S(Month, NR) = 0
         SDNEV4S(Month, NR) = 0
         SDSpill4S(Month, NR) = 0
      NEXT Month

   FOR NYear = 1 TO NIYear
      FOR Month = 1 TO 12
         DNIF4S(Month, NR) = (NIF4(Month, NYear) - ANIF4(Month, NR)) ^ 2
         SDNIF4S(Month, NR) = SDNIF4S(Month, NR) + DNIF4S(Month, NR)
         DNST4S(Month, NR) = (NST4(Month, NYear) - ANST4(Month, NR)) ^ 2
         SDNST4S(Month, NR) = SDNST4S(Month, NR) + DNST4S(Month, NR)
         DNOF4S(Month, NR) = (NOF4(Month, NYear) - ANOF4(Month, NR)) ^ 2
         SDNOF4S(Month, NR) = SDNOF4S(Month, NR) + DNOF4S(Month, NR)
         DNEV4S(Month, NR) = (NEV4(Month, NYear) - ANEV4(Month, NR)) ^ 2
         SDNEV4S(Month, NR) = SDNEV4S(Month, NR) + DNEV4S(Month, NR)
         DSpill4S(Month, NR) = (Spill4(Month, NYear) - ASpill4(Month, NR)) ^ 2
         SDSpill4S(Month, NR) = SDSpill4S(Month, NR) + DSpill4S(Month, NR)
      NEXT Month
   NEXT NYear
RETURN
'--------------------------------StdNEV4
Std.Calc:

      FOR Month = 1 TO 12
         VNIF4(Month, NR) = SDNIF4S(Month, NR) / (NIYear - 1)
         StdNIF4(Month, NR) = (VNIF4(Month, NR)) ^ .5
         VNST4(Month, NR) = SDNST4S(Month, NR) / (NIYear - 1)
         StdNST4(Month, NR) = (VNST4(Month, NR)) ^ .5
         VNOF4(Month, NR) = SDNOF4S(Month, NR) / (NIYear - 1)
         StdNOF4(Month, NR) = (VNOF4(Month, NR)) ^ .5
         VNEV4(Month, NR) = SDNEV4S(Month, NR) / (NIYear - 1)
         StdNEV4(Month, NR) = (VNEV4(Month, NR)) ^ .5
         VSpill4(Month, NR) = SDSpill4S(Month, NR) / (NIYear - 1)
         StdSpill4(Month, NR) = (VSpill4(Month, NR)) ^ .5
      NEXT Month
```

```
RETURN

OutStat.Calc:
IF (NR = NRTest) THEN
        PRINT #17,
        PRINT #17,
        PRINT #17, "Agung W.H. SOEHARNO"
        PRINT #17, DATE$, TIME$
                                                    P    R    I    N    T    #    1    7    ,
"*********************************************************************************"
        PRINT #17, "*          Output of Jat-Ctr Program"
        PRINT #17, "*          Node 4 : J A T I L U H U R"
        PRINT #17, "*"
        PRINT #17, "-----------------------> Stat.Calc - Replicate no. ", NR
                                                    P    R    I    N    T    #    1    7    ,
"*********************************************************************************"
        PRINT #17,
        PRINT #17, " NIF4 = Inflow to Node 4: Jatiluhur"
        PRINT #17,
        PRINT #17, "  NYear   May    Jun    Jul    Aug    Sep    Oct    Nov    Dec    Jan
  Feb    Mar    Apr"
        PRINT #17,
   FOR NYear = 1 TO NIYear
       PRINT #17, USING "######.#"; NYear; NIF4(1, NYear); NIF4(2, NYear); NIF4(3, NYear);
NIF4(4, NYear); NIF4(5, NYear); NIF4(6, NYear); NIF4(7, NYear); NIF4(8, NYear); NIF4(9,
NYear); NIF4(10, NYear); NIF4(11, NYear); NIF4(12, NYear)
   NEXT NYear
        PRINT #17,
        PRINT #17, "   NR     May    Jun    Jul    Aug    Sep    Oct    Nov    Dec    Jan
  Feb    Mar    Apr"
        PRINT #17, "Minimum of NIF4"
        PRINT #17, USING "######.#"; NR; MinNIF4(1); MinNIF4(2); MinNIF4(3); MinNIF4(4);
MinNIF4(5); MinNIF4(6); MinNIF4(7); MinNIF4(8); MinNIF4(9); MinNIF4(10); MinNIF4(11);
MinNIF4(12)
        PRINT #17, "ANIF4 = Average of NIF4"
```

```
        PRINT #17, USING "######.#"; NR; ANIF4(1, NR); ANIF4(2, NR); ANIF4(3, NR); ANIF4(4,
NR); ANIF4(5, NR); ANIF4(6, NR); ANIF4(7, NR); ANIF4(8, NR); ANIF4(9, NR); ANIF4(10, NR);
ANIF4(11, NR); ANIF4(12, NR)
        PRINT #17, "Maximum of NIF4"
        PRINT #17, USING "######.#"; NR; MaxNIF4(1); MaxNIF4(2); MaxNIF4(3); MaxNIF4(5);
MaxNIF4(5);  MaxNIF4(6);  MaxNIF4(7);  MaxNIF4(8);  MaxNIF4(9);  MaxNIF4(10);  MaxNIF4(11);
MaxNIF4(12)
        PRINT #17, "StdNIF4 = Standard Dev. of NIF4"
        PRINT #17, USING "######.#"; NR; StdNIF4(1, NR); StdNIF4(2, NR); StdNIF4(3, NR);
StdNIF4(4, NR); StdNIF4(5, NR); StdNIF4(6, NR); StdNIF4(7, NR); StdNIF4(8, NR); StdNIF4(9,
NR); StdNIF4(10, NR); StdNIF4(11, NR); StdNIF4(12, NR)


        PRINT #17,
        PRINT #17, " NST4 = End of period Storage of Node 4: Jatiluhur"
        PRINT #17,
        PRINT #17, "  NYear   May    Jun    Jul    Aug    Sep    Oct    Nov    Dec    Jan
  Feb   Mar    Apr"
        PRINT #17,
   FOR NYear = 1 TO NIYear
       PRINT #17, USING "######.#"; NYear; NST4(1, NYear); NST4(2, NYear); NST4(3, NYear);
NST4(4, NYear); NST4(5, NYear); NST4(6, NYear); NST4(7, NYear); NST4(8, NYear); NST4(9,
NYear); NST4(10, NYear); NST4(11, NYear); NST4(12, NYear)
   NEXT NYear
        PRINT #17,
        PRINT #17, "   NR     May    Jun    Jul    Aug    Sep    Oct    Nov    Dec    Jan
  Feb   Mar    Apr"
        PRINT #17, "Minimum of NST4"
        PRINT #17, USING "######.#"; NR; MinNST4(1); MinNST4(2); MinNST4(3); MinNST4(4);
MinNST4(5);  MinNST4(6);  MinNST4(7);  MinNST4(8);  MinNST4(9);  MinNST4(10);  MinNST4(11);
MinNST4(12)
        PRINT #17, "ANST4 = Average of NST4"
        PRINT #17, USING "######.#"; NR; ANST4(1, NR); ANST4(2, NR); ANST4(3, NR); ANST4(4,
NR); ANST4(5, NR); ANST4(6, NR); ANST4(7, NR); ANST4(8, NR); ANST4(9, NR); ANST4(10, NR);
ANST4(11, NR); ANST4(12, NR)
        PRINT #17, "Maximum of NST4"
        PRINT #17, USING "######.#"; NR; MaxNST4(1); MaxNST4(2); MaxNST4(3); MaxNST4(5);
```

```
MaxNST4(5); MaxNST4(6); MaxNST4(7); MaxNST4(8); MaxNST4(9); MaxNST4(10); MaxNST4(11);
MaxNST4(12)
        PRINT #17, "StdNST4 = Standard Dev. of NST4"
        PRINT #17, USING "######.#"; NR; StdNST4(1, NR); StdNST4(2, NR); StdNST4(3, NR);
StdNST4(4, NR); StdNST4(5, NR); StdNST4(6, NR); StdNST4(7, NR); StdNST4(8, NR); StdNST4(9,
NR); StdNST4(10, NR); StdNST4(11, NR); StdNST4(12, NR)

        PRINT #17,
        PRINT #17, " NOF4 = OutFlow from Node 4: Jatiluhur"
        PRINT #17,
        PRINT #17, "  NYear   May    Jun    Jul    Aug    Sep    Oct    Nov    Dec    Jan
  Feb    Mar    Apr"
        PRINT #17,
   FOR NYear = 1 TO NIYear
       PRINT #17, USING "######.#"; NYear; NOF4(1, NYear); NOF4(2, NYear); NOF4(3, NYear);
NOF4(4, NYear); NOF4(5, NYear); NOF4(6, NYear); NOF4(7, NYear); NOF4(8, NYear); NOF4(9,
NYear); NOF4(10, NYear); NOF4(11, NYear); NOF4(12, NYear)
   NEXT NYear
        PRINT #17,
        PRINT #17, "   NR    May    Jun    Jul    Aug    Sep    Oct    Nov    Dec    Jan
  Feb    Mar    Apr"
        PRINT #17, "Minimum of NOF4"
        PRINT #17, USING "######.#"; NR; MinNOF4(1); MinNOF4(2); MinNOF4(3); MinNOF4(4);
MinNOF4(5); MinNOF4(6); MinNOF4(7); MinNOF4(8); MinNOF4(9); MinNOF4(10); MinNOF4(11);
MinNOF4(12)
        PRINT #17, "ANOF4 = Average of NOF4"
        PRINT #17, USING "######.#"; NR; ANOF4(1, NR); ANOF4(2, NR); ANOF4(3, NR); ANOF4(4,
NR); ANOF4(5, NR); ANOF4(6, NR); ANOF4(7, NR); ANOF4(8, NR); ANOF4(9, NR); ANOF4(10, NR);
ANOF4(11, NR); ANOF4(12, NR)
        PRINT #17, "Maximum of NOF4"
        PRINT #17, USING "######.#"; NR; MaxNOF4(1); MaxNOF4(2); MaxNOF4(3); MaxNOF4(5);
MaxNOF4(5); MaxNOF4(6); MaxNOF4(7); MaxNOF4(8); MaxNOF4(9); MaxNOF4(10); MaxNOF4(11);
MaxNOF4(12)
        PRINT #17, "StdNOF4 = Standard Dev. of NOF4"
        PRINT #17, USING "######.#"; NR; StdNOF4(1, NR); StdNOF4(2, NR); StdNOF4(3, NR);
StdNOF4(4, NR); StdNOF4(5, NR); StdNOF4(6, NR); StdNOF4(7, NR); StdNOF4(8, NR); StdNOF4(9,
NR); StdNOF4(10, NR); StdNOF4(11, NR); StdNOF4(12, NR)
```

```
        PRINT #17,
        PRINT #17, "NEV4 = Evaporation of Node 4: Jatiluhur"
        PRINT #17,
        PRINT #17, "  NYear   May     Jun     Jul     Aug     Sep     Oct     Nov     Dec     Jan
  Feb     Mar     Apr"
        PRINT #17,
   FOR NYear = 1 TO NIYear
       PRINT #17, USING "######.#"; NYear; NEV4(1, NYear); NEV4(2, NYear); NEV4(3, NYear);
NEV4(4, NYear); NEV4(5, NYear); NEV4(6, NYear); NEV4(7, NYear); NEV4(8, NYear); NEV4(9,
NYear); NEV4(10, NYear); NEV4(11, NYear); NEV4(12, NYear)
   NEXT NYear
        PRINT #17,
        PRINT #17, "    NR     May     Jun     Jul     Aug     Sep     Oct     Nov     Dec     Jan
  Feb     Mar     Apr"
        PRINT #17, "Minimum of NEV4"
        PRINT #17, USING "######.#"; NR; MinNEV4(1); MinNEV4(2); MinNEV4(3); MinNEV4(4);
MinNEV4(5); MinNEV4(6); MinNEV4(7); MinNEV4(8); MinNEV4(9); MinNEV4(10); MinNEV4(11);
MinNEV4(12)
        PRINT #17, "ANEV4 = Average of NEV4"
        PRINT #17, USING "######.#"; NR; ANEV4(1, NR); ANEV4(2, NR); ANEV4(3, NR); ANEV4(4,
NR); ANEV4(5, NR); ANEV4(6, NR); ANEV4(7, NR); ANEV4(8, NR); ANEV4(9, NR); ANEV4(10, NR);
ANEV4(11, NR); ANEV4(12, NR)
        PRINT #17, "Maximum of NEV4"
        PRINT #17, USING "######.#"; NR; MaxNEV4(1); MaxNEV4(2); MaxNEV4(3); MaxNEV4(5);
MaxNEV4(5); MaxNEV4(6); MaxNEV4(7); MaxNEV4(8); MaxNEV4(9); MaxNEV4(10); MaxNEV4(11);
MaxNEV4(12)
        PRINT #17, "StdNEV4 = Standard Dev. of NEV4"
        PRINT #17, USING "######.#"; NR; StdNEV4(1, NR); StdNEV4(2, NR); StdNEV4(3, NR);
StdNEV4(4, NR); StdNEV4(5, NR); StdNEV4(6, NR); StdNEV4(7, NR); StdNEV4(8, NR); StdNEV4(9,
NR); StdNEV4(10, NR); StdNEV4(11, NR); StdNEV4(12, NR)
        PRINT #17,
        PRINT #17,
        PRINT #17, "Spill4 = Spill flow of Node 4: Jatiluhur"
        PRINT #17,
        PRINT #17, "  NYear   May     Jun     Jul     Aug     Sep     Oct     Nov     Dec     Jan
  Feb     Mar     Apr"
```

```
        PRINT #17,
   FOR NYear = 1 TO NIYear
                  PRINT #17, USING "######.#"; NYear; Spill4(1, NYear); Spill4(2, NYear);
Spill4(3, NYear); Spill4(4, NYear); Spill4(5, NYear); Spill4(6, NYear); Spill4(7, NYear);
Spill4(8, NYear); Spill4(9, NYear); Spill4(10, NYear); Spill4(11,  _
NYear); Spill4(12, NYear)
   NEXT NYear
        PRINT #17,
        PRINT #17, "   NR     May    Jun    Jul    Aug    Sep    Oct    Nov    Dec    Jan
  Feb     Mar    Apr"
        PRINT #17, "Minimum of Spill4"
        PRINT  #17,  USING  "######.#";  NR;  MinSpill4(1);  MinSpill4(2);  MinSpill4(3);
MinSpill4(4);  MinSpill4(5);  MinSpill4(6);  MinSpill4(7);  MinSpill4(8);  MinSpill4(9);
MinSpill4(10); MinSpill4(11); MinSpill4(12)
        PRINT #17, "ASpill4 = Average of Spill4"
        PRINT #17, USING "######.#"; NR; ASpill4(1, NR); ASpill4(2, NR); ASpill4(3, NR);
ASpill4(4, NR); ASpill4(5, NR); ASpill4(6, NR); ASpill4(7, NR); ASpill4(8, NR); ASpill4(9,
NR); ASpill4(10, NR); ASpill4(11, NR); ASpill4(12, NR)
        PRINT #17, "Maximum of Spill4"
        PRINT  #17,  USING  "######.#";  NR;  MaxSpill4(1);  MaxSpill4(2);  MaxSpill4(3);
MaxSpill4(5);  MaxSpill4(5);  MaxSpill4(6);  MaxSpill4(7);  MaxSpill4(8);  MaxSpill4(9);
MaxSpill4(10); MaxSpill4(11); MaxSpill4(12)
        PRINT #17, "StdSpill4 = Standard Dev. of Spill4"
                  PRINT #17, USING "######.#"; NR; StdSpill4(1, NR); StdSpill4(2, NR);
StdSpill4(3, NR); StdSpill4(4, NR); StdSpill4(5, NR); StdSpill4(6, NR); StdSpill4(7, NR);
StdSpill4(8, NR); StdSpill4(9, NR); StdSpill4(10, NR); StdSpill4(11, NR _
); StdSpill4(12, NR)
        PRINT #17,
        PRINT #16,
        PRINT #16,
        PRINT #16,
        PRINT #16, "Agung W.H. SOEHARNO"
        PRINT #16, DATE$, TIME$, TIME$
                                          P   R   I   N   T   #   1   6   ,
"*******************************************************************************************"
        PRINT #16, "*           Output of Jat-Ctr Program"
        PRINT #16, "*           Node 4 : J A T I L U H U R"
```

```
        PRINT #16, "*"
        PRINT #16, "-----------------------> Stat.Calc - Replicate no. ", NR
                                            P    R    I    N    T    #    1    6    ,
"*************************************************************************************"
        PRINT #16,
        PRINT #16, " JTSUM = Summary File of Node 4: Jatiluhur"
        PRINT #16,
        PRINT #16, "  NYear   May    Jun    Jul    Aug    Sep    Oct    Nov    Dec    Jan
  Feb    Mar    Apr"
        PRINT #16, "ANIF4 = Average of NIF4"
        PRINT #16, USING "######.#"; NR; ANIF4(1, NR); ANIF4(2, NR); ANIF4(3, NR); ANIF4(4,
NR); ANIF4(5, NR); ANIF4(6, NR); ANIF4(7, NR); ANIF4(8, NR); ANIF4(9, NR); ANIF4(10, NR);
ANIF4(11, NR); ANIF4(12, NR)
        PRINT #16, "ANST4 = Average of NST4"
        PRINT #16, USING "######.#"; NR; ANST4(1, NR); ANST4(2, NR); ANST4(3, NR); ANST4(4,
NR); ANST4(5, NR); ANST4(6, NR); ANST4(7, NR); ANST4(8, NR); ANST4(9, NR); ANST4(10, NR);
ANST4(11, NR); ANST4(12, NR)
        PRINT #16, "ANOF4 = Average of NOF4"
        PRINT #16, USING "######.#"; NR; ANOF4(1, NR); ANOF4(2, NR); ANOF4(3, NR); ANOF4(4,
NR); ANOF4(5, NR); ANOF4(6, NR); ANOF4(7, NR); ANOF4(8, NR); ANOF4(9, NR); ANOF4(10, NR);
ANOF4(11, NR); ANOF4(12, NR)
        PRINT #16, "ANEV4 = Average of NEV4"
        PRINT #16, USING "######.#"; NR; ANEV4(1, NR); ANEV4(2, NR); ANEV4(3, NR); ANEV4(4,
NR); ANEV4(5, NR); ANEV4(6, NR); ANEV4(7, NR); ANEV4(8, NR); ANEV4(9, NR); ANEV4(10, NR);
ANEV4(11, NR); ANEV4(12, NR)
        PRINT #16, "ASpill4 = Average of Spill4"
        PRINT #16, USING "######.#"; NR; ASpill4(1, NR); ASpill4(2, NR); ASpill4(3, NR);
ASpill4(4, NR); ASpill4(5, NR); ASpill4(6, NR); ASpill4(7, NR); ASpill4(8, NR); ASpill4(9,
NR); ASpill4(10, NR); ASpill4(11, NR); ASpill4(12, NR)
ELSE
END IF

        WRITE #38, ANIF4(1, NR), ANIF4(2, NR), ANIF4(3, NR), ANIF4(4, NR), ANIF4(5, NR),
ANIF4(6, NR), ANIF4(7, NR), ANIF4(8, NR), ANIF4(9, NR), ANIF4(10, NR), ANIF4(11, NR),
ANIF4(12, NR)
        WRITE #38, ANST4(1, NR), ANST4(2, NR), ANST4(3, NR), ANST4(4, NR), ANST4(5, NR),
ANST4(6, NR), ANST4(7, NR), ANST4(8, NR), ANST4(9, NR), ANST4(10, NR), ANST4(11, NR),
```

```
ANST4(12, NR)
        WRITE #38, ANOF4(1, NR), ANOF4(2, NR), ANOF4(3, NR), ANOF4(4, NR), ANOF4(5, NR),
ANOF4(6, NR), ANOF4(7, NR), ANOF4(8, NR), ANOF4(9, NR), ANOF4(10, NR), ANOF4(11, NR),
ANOF4(12, NR)
        WRITE #38, ANEV4(1, NR), ANEV4(2, NR), ANEV4(3, NR), ANEV4(4, NR), ANEV4(5, NR),
ANEV4(6, NR), ANEV4(7, NR), ANEV4(8, NR), ANEV4(9, NR), ANEV4(10, NR), ANEV4(11, NR),
ANEV4(12, NR)
        WRITE #38, ASpill4(1, NR), ASpill4(2, NR), ASpill4(3, NR), ASpill4(4, NR),
ASpill4(5, NR), ASpill4(6, NR), ASpill4(7, NR), ASpill4(8, NR), ASpill4(9, NR), ASpill4(10,
NR), ASpill4(11, NR), ASpill4(12, NR)

        WRITE #38, StdNIF4(1, NR), StdNIF4(2, NR), StdNIF4(3, NR), StdNIF4(4, NR),
StdNIF4(5, NR), StdNIF4(6, NR), StdNIF4(7, NR), StdNIF4(8, NR), StdNIF4(9, NR), StdNIF4(10,
NR), StdNIF4(11, NR), StdNIF4(12, NR)
        WRITE #38, StdNST4(1, NR), StdNST4(2, NR), StdNST4(3, NR), StdNST4(4, NR),
StdNST4(5, NR), StdNST4(6, NR), StdNST4(7, NR), StdNST4(8, NR), StdNST4(9, NR), StdNST4(10,
NR), StdNST4(11, NR), StdNST4(12, NR)
        WRITE #38, StdNOF4(1, NR), StdNOF4(2, NR), StdNOF4(3, NR), StdNOF4(4, NR),
StdNOF4(5, NR), StdNOF4(6, NR), StdNOF4(7, NR), StdNOF4(8, NR), StdNOF4(9, NR), StdNOF4(10,
NR), StdNOF4(11, NR), StdNOF4(12, NR)
        WRITE #38, StdNEV4(1, NR), StdNEV4(2, NR), StdNEV4(3, NR), StdNEV4(4, NR),
StdNEV4(5, NR), StdNEV4(6, NR), StdNEV4(7, NR), StdNEV4(8, NR), StdNEV4(9, NR), StdNEV4(10,
NR), StdNEV4(11, NR), StdNEV4(12, NR)
        WRITE #38, StdSpill4(1, NR), StdSpill4(2, NR), StdSpill4(3, NR), StdSpill4(4, NR),
StdSpill4(5, NR), StdSpill4(6, NR), StdSpill4(7, NR), StdSpill4(8, NR), StdSpill4(9, NR),
StdSpill4(10, NR), StdSpill4(11, NR), StdSpill4(12, NR)

RETURN

Outstat.calc.test:
FOR NR = 1 TO IReplicate
IF (NR = 1) THEN
        PRINT #17,
        PRINT #17, "   NR     May     Jun     Jul     Aug     Sep     Oct     Nov     Dec     Jan
  Feb     Mar     Apr"
        PRINT #17, "ANIF4 = Average of NIF4"
ELSE
```

```
END IF
        PRINT #17, USING "######.#"; NR; ANIF4(1, NR); ANIF4(2, NR); ANIF4(3, NR); ANIF4(4,
NR); ANIF4(5, NR); ANIF4(6, NR); ANIF4(7, NR); ANIF4(8, NR); ANIF4(9, NR); ANIF4(10, NR);
ANIF4(11, NR); ANIF4(12, NR)
NEXT NR

FOR NR = 1 TO IReplicate
IF (NR = 1) THEN
        PRINT #17,
        PRINT #17, "   NR      May      Jun      Jul      Aug      Sep      Oct      Nov      Dec      Jan
  Feb      Mar      Apr"
        PRINT #17, "ANST4 = Average of NST4"
ELSE
END IF
        PRINT #17, USING "######.#"; NR; ANST4(1, NR); ANST4(2, NR); ANST4(3, NR); ANST4(4,
NR); ANST4(5, NR); ANST4(6, NR); ANST4(7, NR); ANST4(8, NR); ANST4(9, NR); ANST4(10, NR);
ANST4(11, NR); ANST4(12, NR)
NEXT NR

FOR NR = 1 TO IReplicate
IF (NR = 1) THEN
        PRINT #17,
        PRINT #17, "   NR      May      Jun      Jul      Aug      Sep      Oct      Nov      Dec      Jan
  Feb      Mar      Apr"
        PRINT #17, "ANEV4 = Average of NEV4"
ELSE
END IF
        PRINT #17, USING "######.#"; NR; ANEV4(1, NR); ANEV4(2, NR); ANEV4(3, NR); ANEV4(4,
NR); ANEV4(5, NR); ANEV4(6, NR); ANEV4(7, NR); ANEV4(8, NR); ANEV4(9, NR); ANEV4(10, NR);
ANEV4(11, NR); ANEV4(12, NR)
NEXT NR

FOR NR = 1 TO IReplicate
IF (NR = 1) THEN
        PRINT #17,
        PRINT #17, "   NR      May      Jun      Jul      Aug      Sep      Oct      Nov      Dec      Jan
  Feb      Mar      Apr"
```

```
        PRINT #17, "ASpill4 = Average of Spill4"
ELSE
END IF
        PRINT #17, USING "######.#"; NR; ASpill4(1, NR); ASpill4(2, NR); ASpill4(3, NR);
ASpill4(4, NR); ASpill4(5, NR); ASpill4(6, NR); ASpill4(7, NR); ASpill4(8, NR); ASpill4(9,
NR); ASpill4(10, NR); ASpill4(11, NR); ASpill4(12, NR)
NEXT NR
RETURN

'**************************************************************************
'                       Statistics.TESt
'**************************************************************************

Statistics.Test.AJatCtr:
LOCATE 13, 10
PRINT "S t a t i s t i c s . T e s t"
PRINT #17, "-------------> S t a t i s t i c s . T e s t"
PRINT #17,
PRINT #17, "   May    Jun    Jul    Aug    Sep    Oct    Nov    Dec    Jan    Feb    Mar
Apr"
PRINT #17,

OPEN "c:\Jat\JtInput3.dat" FOR INPUT AS #39  'input of stat.tes

GOSUB Input.Data
GOSUB AvgAvg.Calc
GOSUB SumSum.Calc
GOSUB StdStd.Calc
GOSUB PrintStat.Tes

CLOSE #39
RETURN

'----------------------------- Input.Data
Input.Data:
```

```
    FOR NR = 1 TO IReplicate
        INPUT #39, ANIF4(1, NR), ANIF4(2, NR), ANIF4(3, NR), ANIF4(4, NR), ANIF4(5, NR),
ANIF4(6, NR), ANIF4(7, NR), ANIF4(8, NR), ANIF4(9, NR), ANIF4(10, NR), ANIF4(11, NR),
ANIF4(12, NR)
        INPUT #39, ANST4(1, NR), ANST4(2, NR), ANST4(3, NR), ANST4(4, NR), ANST4(5, NR),
ANST4(6, NR), ANST4(7, NR), ANST4(8, NR), ANST4(9, NR), ANST4(10, NR), ANST4(11, NR),
ANST4(12, NR)
        INPUT #39, ANOF4(1, NR), ANOF4(2, NR), ANOF4(3, NR), ANOF4(4, NR), ANOF4(5, NR),
ANOF4(6, NR), ANOF4(7, NR), ANOF4(8, NR), ANOF4(9, NR), ANOF4(10, NR), ANOF4(11, NR),
ANOF4(12, NR)
        INPUT #39, ANEV4(1, NR), ANEV4(2, NR), ANEV4(3, NR), ANEV4(4, NR), ANEV4(5, NR),
ANEV4(6, NR), ANEV4(7, NR), ANEV4(8, NR), ANEV4(9, NR), ANEV4(10, NR), ANEV4(11, NR),
ANEV4(12, NR)
        INPUT #39, ASpill4(1, NR), ASpill4(2, NR), ASpill4(3, NR), ASpill4(4, NR),
ASpill4(5, NR), ASpill4(6, NR), ASpill4(7, NR), ASpill4(8, NR), ASpill4(9, NR), ASpill4(10,
NR), ASpill4(11, NR), ASpill4(12, NR)

        INPUT #39, StdNIF4(1, NR), StdNIF4(2, NR), StdNIF4(3, NR), StdNIF4(4, NR),
StdNIF4(5, NR), StdNIF4(6, NR), StdNIF4(7, NR), StdNIF4(8, NR), StdNIF4(9, NR), StdNIF4(10,
NR), StdNIF4(11, NR), StdNIF4(12, NR)
        INPUT #39, StdNST4(1, NR), StdNST4(2, NR), StdNST4(3, NR), StdNST4(4, NR),
StdNST4(5, NR), StdNST4(6, NR), StdNST4(7, NR), StdNST4(8, NR), StdNST4(9, NR), StdNST4(10,
NR), StdNST4(11, NR), StdNST4(12, NR)
        INPUT #39, StdNOF4(1, NR), StdNOF4(2, NR), StdNOF4(3, NR), StdNOF4(4, NR),
StdNOF4(5, NR), StdNOF4(6, NR), StdNOF4(7, NR), StdNOF4(8, NR), StdNOF4(9, NR), StdNOF4(10,
NR), StdNOF4(11, NR), StdNOF4(12, NR)
        INPUT #39, StdNEV4(1, NR), StdNEV4(2, NR), StdNEV4(3, NR), StdNEV4(4, NR),
StdNEV4(5, NR), StdNEV4(6, NR), StdNEV4(7, NR), StdNEV4(8, NR), StdNEV4(9, NR), StdNEV4(10,
NR), StdNEV4(11, NR), StdNEV4(12, NR)
        INPUT #39, StdSpill4(1, NR), StdSpill4(2, NR), StdSpill4(3, NR), StdSpill4(4, NR),
StdSpill4(5, NR), StdSpill4(6, NR), StdSpill4(7, NR), StdSpill4(8, NR), StdSpill4(9, NR),
StdSpill4(10, NR), StdSpill4(11, NR), StdSpill4(12, NR)
    NEXT NR

RETURN

'------------------------------ AvgAvg.Calc
```

```
AvgAvg.Calc:
DIM SANIF4(12), SStdNIF4(12), AANIF4(12), AStdNIF4(12), SANST4(12), SStdNST4(12)
DIM AANST4(12), AStdNST4(12), SANOF4(12), SStdNOF4(12), AANOF4(12), AStdNOF4(12)
DIM SANEV4(12), SStdNEV4(12), AANEV4(12), AStdNEV4(12), SASpill4(12), SStdSpill4(12)
DIM AASpill4(12), AStdSpill4(12)

'---------- Average of NEV4

        FOR Month = 1 TO 12
           SANIF4(Month) = 0
           SStdNIF4(Month) = 0
           SANST4(Month) = 0
           SStdNST4(Month) = 0
           SANOF4(Month) = 0
           SStdNOF4(Month) = 0
           SANEV4(Month) = 0
           SStdNEV4(Month) = 0
           SASpill4(Month) = 0
           SStdSpill4(Month) = 0
        NEXT Month

     FOR Month = 1 TO 12
        FOR NR = 1 TO IReplicate
           SANIF4(Month) = SANIF4(Month) + ANIF4(Month, NR)
           SStdNIF4(Month) = SStdNIF4(Month) + StdNIF4(Month, NR)
           SANST4(Month) = SANST4(Month) + ANST4(Month, NR)
           SStdNST4(Month) = SStdNST4(Month) + StdNST4(Month, NR)
           SANOF4(Month) = SANOF4(Month) + ANOF4(Month, NR)
           SStdNOF4(Month) = SStdNOF4(Month) + StdNOF4(Month, NR)
           SANEV4(Month) = SANEV4(Month) + ANEV4(Month, NR)
           SStdNEV4(Month) = SStdNEV4(Month) + StdNEV4(Month, NR)
           SASpill4(Month) = SASpill4(Month) + ASpill4(Month, NR)
           SStdSpill4(Month) = SStdSpill4(Month) + StdSpill4(Month, NR)
        NEXT NR
     NEXT Month
```

```
            DANIF4S(Month) = (ANIF4(Month, NR) - AANIF4(Month)) ^ 2
            SDANIF4S(Month) = SDANIF4S(Month) + DANIF4S(Month)
            DStdNIF4S(Month) = (StdNIF4(Month, NR) - AStdNIF4(Month)) ^ 2
            SDStdNIF4S(Month) = SDStdNIF4S(Month) + DStdNIF4S(Month)
            DANST4S(Month) = (ANST4(Month, NR) - AANST4(Month)) ^ 2
            SDANST4S(Month) = SDANST4S(Month) + DANST4S(Month)
            DStdNST4S(Month) = (StdNST4(Month, NR) - AStdNST4(Month)) ^ 2
            SDStdNST4S(Month) = SDStdNST4S(Month) + DStdNST4S(Month)
            DANOF4S(Month) = (ANOF4(Month, NR) - AANOF4(Month)) ^ 2
            SDANOF4S(Month) = SDANOF4S(Month) + DANOF4S(Month)
            DStdNOF4S(Month) = (StdNOF4(Month, NR) - AStdNOF4(Month)) ^ 2
            SDStdNOF4S(Month) = SDStdNOF4S(Month) + DStdNOF4S(Month)
            DANEV4S(Month) = (ANEV4(Month, NR) - AANEV4(Month)) ^ 2
            SDANEV4S(Month) = SDANEV4S(Month) + DANEV4S(Month)
            DStdNEV4S(Month) = (StdNEV4(Month, NR) - AStdNEV4(Month)) ^ 2
            SDStdNEV4S(Month) = SDStdNEV4S(Month) + DStdNEV4S(Month)
            DASpill4S(Month) = (ASpill4(Month, NR) - AASpill4(Month)) ^ 2
            SDASpill4S(Month) = SDASpill4S(Month) + DASpill4S(Month)
            DStdSpill4S(Month) = (StdSpill4(Month, NR) - AStdSpill4(Month)) ^ 2
            SDStdSpill4S(Month) = SDStdSpill4S(Month) + DStdSpill4S(Month)
         NEXT NR
      NEXT Month

10 RETURN
'---------------------------------Std of StdNEV4
StdStd.Calc:

DIM VANIF4(12), StdANIF4(12), VStdNIF4(12), StdStdNIF4(12), VANST4(12), StdANST4(12)
DIM VStdNST4(12), StdStdNST4(12), VANOF4(12), StdANOF4(12), VStdNOF4(12), StdStdNOF4(12)
DIM VANEV4(12), StdANEV4(12), VStdNEV4(12), StdStdNEV4(12), VASpill4(12), StdASpill4(12)
DIM VStdSpill4(12), StdStdSpill4(12)

        FOR Month = 1 TO 12
           VANIF4(Month) = SDANIF4S(Month) / (IReplicate - 1)
           StdANIF4(Month) = (VANIF4(Month)) ^ .5
           VStdNIF4(Month) = SDStdNIF4S(Month) / (IReplicate - 1)
```

```
        FOR Month = 1 TO 12
           AANIF4(Month) = SANIF4(Month) / IReplicate
           AStdNIF4(Month) = SStdNIF4(Month) / IReplicate
           AANST4(Month) = SANST4(Month) / IReplicate
           AStdNST4(Month) = SStdNST4(Month) / IReplicate
           AANOF4(Month) = SANOF4(Month) / IReplicate
           AStdNOF4(Month) = SStdNOF4(Month) / IReplicate
           AANEV4(Month) = SANEV4(Month) / IReplicate
           AStdNEV4(Month) = SStdNEV4(Month) / IReplicate
           AASpill4(Month) = SASpill4(Month) / IReplicate
           AStdSpill4(Month) = SStdSpill4(Month) / IReplicate
        NEXT Month

RETURN

'-------------------------------Sumation of Sumation
SumSum.Calc:
DIM SDANIF4S(12), SDStdNIF4S(12), DANIF4S(12), DStdNIF4S(12), SDANST4S(12), SDStdNST4S(12)
DIM DANST4S(12), DStdNST4S(12), SDANOF4S(12), SDStdNOF4S(12), DANOF4S(12), DStdNOF4S(12)
DIM SDANEV4S(12), SDStdNEV4S(12), DANEV4S(12), DStdNEV4S(12), SDASpill4S(12)
DIM SDStdSpill4S(12), DASpill4S(12), DStdSpill4S(12)

        FOR Month = 1 TO 12
           SDANIF4S(Month) = 0
           SDStdNIF4S(Month) = 0
           SDANST4S(Month) = 0
           SDStdNST4S(Month) = 0
           SDANOF4S(Month) = 0
           SDStdNOF4S(Month) = 0
           SDANEV4S(Month) = 0
           SDStdNEV4S(Month) = 0
           SDASpill4S(Month) = 0
           SDStdSpill4S(Month) = 0
        NEXT Month

     FOR Month = 1 TO 12
        FOR NR = 1 TO IReplicate
```

```
            StdStdNIF4(Month) = (VStdNIF4(Month)) ^ .5

            VANST4(Month) = SDANST4S(Month) / (IReplicate - 1)
            StdANST4(Month) = (VANST4(Month)) ^ .5
            VStdNST4(Month) = SDStdNST4S(Month) / (IReplicate - 1)
            StdStdNST4(Month) = (VStdNST4(Month)) ^ .5

            VANOF4(Month) = SDANOF4S(Month) / (IReplicate - 1)
            StdANOF4(Month) = (VANOF4(Month)) ^ .5
            VStdNOF4(Month) = SDStdNOF4S(Month) / (IReplicate - 1)
            StdStdNOF4(Month) = (VStdNOF4(Month)) ^ .5

            VANEV4(Month) = SDANEV4S(Month) / (IReplicate - 1)
            StdANEV4(Month) = (VANEV4(Month)) ^ .5
            VStdNEV4(Month) = SDStdNEV4S(Month) / (IReplicate - 1)
            StdStdNEV4(Month) = (VStdNEV4(Month)) ^ .5

            VASpill4(Month) = SDASpill4S(Month) / (IReplicate - 1)
            StdASpill4(Month) = (VASpill4(Month)) ^ .5
            VStdSpill4(Month) = SDStdSpill4S(Month) / (IReplicate - 1)
            StdStdSpill4(Month) = (VStdSpill4(Month)) ^ .5

        NEXT Month

RETURN

'------------------ PrintStat.tes
PrintStat.Tes:
     PRINT #17, "AANIF4 = "
     PRINT #17, USING "######.#"; AANIF4(1); AANIF4(2); AANIF4(3); AANIF4(4); AANIF4(5);
AANIF4(6); AANIF4(7); AANIF4(8); AANIF4(9); AANIF4(10); AANIF4(11); AANIF4(12)
     PRINT #17, "AStdNIF4 ="
     PRINT #17, USING "######.#"; AStdNIF4(1); AStdNIF4(2); AStdNIF4(3); AStdNIF4(4);
AStdNIF4(5); AStdNIF4(6); AStdNIF4(7); AStdNIF4(8); AStdNIF4(9); AStdNIF4(10); AStdNIF4(11);
AStdNIF4(12)

     PRINT #17, "AANST4 = "
```

```
     PRINT #17, USING "######.#"; AANST4(1); AANST4(2); AANST4(3); AANST4(4); AANST4(5);
AANST4(6); AANST4(7); AANST4(8); AANST4(9); AANST4(10); AANST4(11); AANST4(12)
     PRINT #17, "AStdNST4 ="
     PRINT #17, USING "######.#"; AStdNST4(1); AStdNST4(2); AStdNST4(3); AStdNST4(4);
AStdNST4(5); AStdNST4(6); AStdNST4(7); AStdNST4(8); AStdNST4(9); AStdNST4(10); AStdNST4(11);
AStdNST4(12)

     PRINT #17, "AANOF4 = "
     PRINT #17, USING "######.#"; AANOF4(1); AANOF4(2); AANOF4(3); AANOF4(4); AANOF4(5);
AANOF4(6); AANOF4(7); AANOF4(8); AANOF4(9); AANOF4(10); AANOF4(11); AANOF4(12)
     PRINT #17, "AStdNOF4 ="
     PRINT #17, USING "######.#"; AStdNOF4(1); AStdNOF4(2); AStdNOF4(3); AStdNOF4(4);
AStdNOF4(5); AStdNOF4(6); AStdNOF4(7); AStdNOF4(8); AStdNOF4(9); AStdNOF4(10); AStdNOF4(11);
AStdNOF4(12)

     PRINT #17, "AANEV4 = "
     PRINT #17, USING "######.#"; AANEV4(1); AANEV4(2); AANEV4(3); AANEV4(4); AANEV4(5);
AANEV4(6); AANEV4(7); AANEV4(8); AANEV4(9); AANEV4(10); AANEV4(11); AANEV4(12)
     PRINT #17, "AStdNEV4 ="
     PRINT #17, USING "######.#"; AStdNEV4(1); AStdNEV4(2); AStdNEV4(3); AStdNEV4(4);
AStdNEV4(5); AStdNEV4(6); AStdNEV4(7); AStdNEV4(8); AStdNEV4(9); AStdNEV4(10); AStdNEV4(11);
AStdNEV4(12)

     PRINT #17, "AASpill4 = "
     PRINT #17, USING "######.#"; AASpill4(1); AASpill4(2); AASpill4(3); AASpill4(4);
AASpill4(5); AASpill4(6); AASpill4(7); AASpill4(8); AASpill4(9); AASpill4(10); AASpill4(11);
AASpill4(12)
     PRINT #17, "AStdSpill4 ="
     PRINT #17, USING "######.#"; AStdSpill4(1); AStdSpill4(2); AStdSpill4(3); AStdSpill4(4);
AStdSpill4(5); AStdSpill4(6); AStdSpill4(7); AStdSpill4(8); AStdSpill4(9); AStdSpill4(10);
AStdSpill4(11); AStdSpill4(12)

     PRINT #17, "StdANIF4 = "
     PRINT #17, USING "######.#"; StdANIF4(1); StdANIF4(2); StdANIF4(3); StdANIF4(4);
StdANIF4(5); StdANIF4(6); StdANIF4(7); StdANIF4(8); StdANIF4(9); StdANIF4(10); StdANIF4(11);
StdANIF4(12)
     PRINT #17, "StdStdNIF4 = "
```

```
PRINT #17, USING "######.#"; StdStdNIF4(1); StdStdNIF4(2); StdStdNIF4(3); StdStdNIF4(4);
StdStdNIF4(5); StdStdNIF4(6); StdStdNIF4(7); StdStdNIF4(8); StdStdNIF4(9); StdStdNIF4(10);
StdStdNIF4(11); StdStdNIF4(12)

PRINT #17, "StdANST4 = "
PRINT #17, USING "######.#"; StdANST4(1); StdANST4(2); StdANST4(3); StdANST4(4);
StdANST4(5); StdANST4(6); StdANST4(7); StdANST4(8); StdANST4(9); StdANST4(10); StdANST4(11);
StdANST4(12)
PRINT #17, "StdStdNST4 = "
PRINT #17, USING "######.#"; StdStdNST4(1); StdStdNST4(2); StdStdNST4(3); StdStdNST4(4);
StdStdNST4(5); StdStdNST4(6); StdStdNST4(7); StdStdNST4(8); StdStdNST4(9); StdStdNST4(10);
StdStdNST4(11); StdStdNST4(12)

PRINT #17, "StdANOF4 = "
PRINT #17, USING "######.#"; StdANOF4(1); StdANOF4(2); StdANOF4(3); StdANOF4(4);
StdANOF4(5); StdANOF4(6); StdANOF4(7); StdANOF4(8); StdANOF4(9); StdANOF4(10); StdANOF4(11);
StdANOF4(12)
PRINT #17, "StdStdNOF4 = "
PRINT #17, USING "######.#"; StdStdNOF4(1); StdStdNOF4(2); StdStdNOF4(3); StdStdNOF4(4);
StdStdNOF4(5); StdStdNOF4(6); StdStdNOF4(7); StdStdNOF4(8); StdStdNOF4(9); StdStdNOF4(10);
StdStdNOF4(11); StdStdNOF4(12)

PRINT #17, "StdANEV4 = "
PRINT #17, USING "######.#"; StdANEV4(1); StdANEV4(2); StdANEV4(3); StdANEV4(4);
StdANEV4(5); StdANEV4(6); StdANEV4(7); StdANEV4(8); StdANEV4(9); StdANEV4(10); StdANEV4(11);
StdANEV4(12)
PRINT #17, "StdStdNEV4 = "
PRINT #17, USING "######.#"; StdStdNEV4(1); StdStdNEV4(2); StdStdNEV4(3); StdStdNEV4(4);
StdStdNEV4(5); StdStdNEV4(6); StdStdNEV4(7); StdStdNEV4(8); StdStdNEV4(9); StdStdNEV4(10);
StdStdNEV4(11); StdStdNEV4(12)

PRINT #17, "StdASpill4 = "
PRINT #17, USING "######.#"; StdASpill4(1); StdASpill4(2); StdASpill4(3); StdASpill4(4);
StdASpill4(5); StdASpill4(6); StdASpill4(7); StdASpill4(8); StdASpill4(9); StdASpill4(10);
StdASpill4(11); StdASpill4(12)
PRINT #17, "StdStdSpill4 = "
PRINT #17, USING "######.#"; StdStdSpill4(1); StdStdSpill4(2); StdStdSpill4(3);
```

```
StdStdSpill4(4);  StdStdSpill4(5);  StdStdSpill4(6);  StdStdSpill4(7);  StdStdSpill4(8);
StdStdSpill4(9); StdStdSpill4(10); StdStdSpill4(11); StdStdSpill4(12)

        PRINT #16,
        PRINT #16,
        PRINT #16, "Agung W.H. SOEHARNO"
        PRINT #16, DATE$, TIME$
                                                P   R   I   N   T   #   1   6   ,
"********************************************************************************"
        PRINT #16, "*           Output of AJat-Ctr Program"
        PRINT #16, "*           Node 4 : J A T I L U H U R"
        PRINT #16, "*"
        PRINT #16, "-----------------------> Statistics.Test "
                                                P   R   I   N   T   #   1   6   ,
"********************************************************************************"
        PRINT #16,
        PRINT #16, " May    Jun    Jul    Aug    Sep    Oct    Nov    Dec    Jan    Feb
 Mar     Apr"
        PRINT #16,

    PRINT #16, "AANIF4 = "
    PRINT #16, USING "######.#"; AANIF4(1); AANIF4(2); AANIF4(3); AANIF4(4); AANIF4(5);
AANIF4(6); AANIF4(7); AANIF4(8); AANIF4(9); AANIF4(10); AANIF4(11); AANIF4(12)

    PRINT #16, "AANST4 = "
    PRINT #16, USING "######.#"; AANST4(1); AANST4(2); AANST4(3); AANST4(4); AANST4(5);
AANST4(6); AANST4(7); AANST4(8); AANST4(9); AANST4(10); AANST4(11); AANST4(12)

    PRINT #16, "AANOF4 = "
    PRINT #16, USING "######.#"; AANOF4(1); AANOF4(2); AANOF4(3); AANOF4(4); AANOF4(5);
AANOF4(6); AANOF4(7); AANOF4(8); AANOF4(9); AANOF4(10); AANOF4(11); AANOF4(12)

    PRINT #16, "AANEV4 = "
    PRINT #16, USING "######.#"; AANEV4(1); AANEV4(2); AANEV4(3); AANEV4(4); AANEV4(5);
AANEV4(6); AANEV4(7); AANEV4(8); AANEV4(9); AANEV4(10); AANEV4(11); AANEV4(12)

    PRINT #16, "AASpill4 = "
```

```
     PRINT #16, USING "######.#"; AASpill4(1); AASpill4(2); AASpill4(3); AASpill4(4);
AASpill4(5); AASpill4(6); AASpill4(7); AASpill4(8); AASpill4(9); AASpill4(10); AASpill4(11);
AASpill4(12)

     PRINT #17, " The Program AJat-Ctr is finish."
     PRINT #16, " The Program AJat-Ctr is finish."

RETURN
```

```
'EgyCtr.Bas
'
'*************************************************************************
'                          Description of Variables
'*************************************************************************
'LEG2     - energy produced by hydropower Node 2: the Saguling Hydropower Plant
'LEG3     - energy produced by hydropower Node 3: the Cirata Hydropower Plant
'LEG4     - energy produced by hydropower Node 4: the Jatiluhur Hydropower Plant
'TE       - total energy of the Citarum Hydropower System
'NR       - number of replications
'NYear    - number of year
'Month    - number of month
'Avg      - average
'Min      - minimum
'Max      - maximum
'Std      - standard deviation
'
```

```
'*****************************************************
'                Main Module
'*****************************************************
CLS
CLEAR , , 3000

LOCATE 9, 10
PRINT "AEgyCtr Program"
LOCATE 11, 10
PRINT "Be Patient Please !!!"
GOSUB Initialization

FOR NR = 1 TO IReplicate
GOSUB Statistics.Calc
IF IReplicate = 1 THEN
   CLS
   PRINT #7, "Total Hydro Electric Power Within Year (TATE) = "; TATE
   GOTO 100
ELSE
END IF
NEXT NR

   GOSUB Statistics.TESt
   PRINT #7, "Total Hydro Electric Power Within Year (TAATE) = "; TAATE

100 GOSUB Wrap.Up

     LOCATE 15, 10
     PRINT "The Sub-Program of Egy-Ctr is finish."
END

'*****************************************************
'                Initiation
'*****************************************************
Initialization:

' $DYNAMIC
```

```
OPEN "c:\data\NRepl.dat" FOR INPUT AS #11 'number of replication, year and months
INPUT #11, IMonth, NIYear, IReplicate
CLOSE #11

DIM E2(60), E3(60), E4(60), ET(60), TALEG2(100), TALEG3(100), TALEG4(100), TATE(10)
DIM Blank AS STRING, TimeSTEp AS STRING, Replicate AS STRING, KSTEp0 AS INTEGER
DIM MHTE(12), SHTE(12), LEG2(12, 60), LEG3(12, 60), LEG4(12, 60), TE(12, 60)
DIM LEnergy2(12, 720), LEnergy3(12, 720), LEnergy4(12, 720)
DIM   Min(IMonth),   Max(IMonth),   MinLEG2(IMonth),   MaxLEG2(IMonth),   MinLEG3(IMonth),
MaxLEG3(IMonth)
DIM MinLEG4(IMonth), MaxLEG4(IMonth), MinTE(IMonth), MaxTE(IMonth), SLEG2(IMonth, IReplicate)
DIM ALEG2(IMonth, IReplicate), SDLEG2S(IMonth, IReplicate), VLEG2(IMonth, IReplicate)
DIM StdLEG2(IMonth, IReplicate), DLEG2S(IMonth, IReplicate), SLEG3(IMonth, IReplicate)
DIM ALEG3(IMonth, IReplicate), SDLEG3S(IMonth, IReplicate), VLEG3(IMonth, IReplicate)
DIM StdLEG3(IMonth, IReplicate), DLEG3S(IMonth, IReplicate), SLEG4(IMonth, IReplicate)
DIM ALEG4(IMonth, IReplicate), SDLEG4S(IMonth, IReplicate), VLEG4(IMonth, IReplicate)
DIM StdLEG4(IMonth, IReplicate), DLEG4S(IMonth, IReplicate), STE(IMonth, IReplicate)
DIM ATE(IMonth, IReplicate), SDTES(IMonth, IReplicate), VTE(IMonth, IReplicate)
DIM StdTE(IMonth, IReplicate), DTES(IMonth, IReplicate)

OPEN "c:\sag\SagInp2.out" FOR INPUT AS #2  ' input file from Saguling simulation
OPEN "c:\cir\CirInp2.out" FOR INPUT AS #3  ' input file from Cirata simulation
OPEN "c:\jat\JatInp2.out" FOR INPUT AS #4  ' input file from Jatiluhur simulation
OPEN "c:\egy\EStat.out" FOR OUTPUT AS #7 'output of stat.calc

OPEN "c:\egy\EInput3.dat" FOR OUTPUT AS #8 'input-output file

a$ = "#### ######.## ######.## ######.##"

OPEN "c:\egy\EInput2.dat" FOR OUTPUT AS #5 'input-output file
OPEN "c:\sum\ESUM.OUT" FOR OUTPUT AS #16 'summary output
OPEN "c:\EGY\ESagY.out" FOR OUTPUT AS #41 'annual energy of the Saguling Plant
OPEN "c:\EGY\ECirY.out" FOR OUTPUT AS #42 'annual energy of the Cirata Plant
OPEN "c:\EGY\EJatY.out" FOR OUTPUT AS #43 'annual energy of the Jatiluhur Plant
OPEN "c:\EGY\ETotY.out" FOR OUTPUT AS #44 'annual energy of the Citarum Hyd. Sys.

RETURN
```

```
'*****************************************************************************
'                         Statistics.Calc
'*****************************************************************************
Statistics.Calc:

   FOR N = 1 TO NIYear
       E2(N) = 0
       E3(N) = 0
       E4(N) = 0
       ET(N) = 0
   NEXT N

   FOR N = 1 TO NIYear
      FOR Month = 1 TO 12
        INPUT #2, LEG2(Month, N)
        E2(N) = E2(N) + LEG2(Month, N)

        INPUT #3, LEG3(Month, N)
        E3(N) = E3(N) + LEG3(Month, N)

        INPUT #4, LEG4(Month, N)
        E4(N) = E4(N) + LEG4(Month, N)
      NEXT Month
        ET(N) = E2(N) + E3(N) + E4(N)
        WRITE #41, E2(N)
        WRITE #42, E3(N)
        WRITE #43, E4(N)
        WRITE #44, ET(N)
   NEXT N

   FOR NYear = 1 TO NIYear
      FOR Month = 1 TO 12
        TE(Month, NYear) = LEG2(Month, NYear) + LEG3(Month, NYear) + LEG4(Month, NYear)
      NEXT Month
   NEXT NYear

        LOCATE 13, 10
```

```
        PRINT "----> Stat.Calc - Replicate no. "; NR

        GOSUB MaxMin.Calc
        GOSUB Avg.Calc
        GOSUB Sum.Calc
        GOSUB Std.Calc
        GOSUB OutStat.Calc

RETURN

'------------------- Max and Min
MaxMin.Calc:

        FOR Month = 1 TO 12
           MinLEG2(Month) = 1000000
           MaxLEG2(Month) = 0
           MinLEG3(Month) = 1000000
           MaxLEG3(Month) = 0
           MinLEG4(Month) = 1000000
           MaxLEG4(Month) = 0
           MinTE(Month) = 1000000
           MaxTE(Month) = 0
        NEXT Month


        FOR Month = 1 TO 12
           FOR NYear = 1 TO NIYear
                   IF LEG2(Month, NYear) < MinLEG2(Month) THEN
                      MinLEG2(Month) = LEG2(Month, NYear)
                      ELSE
                   END IF
                   IF LEG2(Month, NYear) > MaxLEG2(Month) THEN
                      MaxLEG2(Month) = LEG2(Month, NYear)
                      ELSE
                   END IF
                   IF LEG3(Month, NYear) < MinLEG3(Month) THEN
                      MinLEG3(Month) = LEG3(Month, NYear)
```

```
                ELSE
             END IF
             IF LEG3(Month, NYear) > MaxLEG3(Month) THEN
                MaxLEG3(Month) = LEG3(Month, NYear)
                ELSE
             END IF
             IF LEG4(Month, NYear) < MinLEG4(Month) THEN
                MinLEG4(Month) = LEG4(Month, NYear)
                ELSE
             END IF
             IF LEG4(Month, NYear) > MaxLEG4(Month) THEN
                MaxLEG4(Month) = LEG4(Month, NYear)
                ELSE
             END IF
             IF TE(Month, NYear) < MinTE(Month) THEN
                MinTE(Month) = TE(Month, NYear)
                ELSE
             END IF
             IF TE(Month, NYear) > MaxTE(Month) THEN
                MaxTE(Month) = TE(Month, NYear)
                ELSE
             END IF
      NEXT NYear
     NEXT Month

RETURN

'------------------------------ Avg.Calc
Avg.Calc:
'---------- Average of TE

    FOR Month = 1 TO 12
       SLEG2(Month, NR) = 0
       SLEG3(Month, NR) = 0
       SLEG4(Month, NR) = 0
       STE(Month, NR) = 0
    NEXT Month
```

```
    FOR NYear = 1 TO NIYear
       FOR Month = 1 TO 12
          SLEG2(Month, NR) = SLEG2(Month, NR) + LEG2(Month, NYear)
          SLEG3(Month, NR) = SLEG3(Month, NR) + LEG3(Month, NYear)
          SLEG4(Month, NR) = SLEG4(Month, NR) + LEG4(Month, NYear)
          STE(Month, NR) = STE(Month, NR) + TE(Month, NYear)
       NEXT Month
    NEXT NYear

       TALEG2(NR) = 0
       TALEG3(NR) = 0
       TALEG4(NR) = 0

       FOR Month = 1 TO 12
          ALEG2(Month, NR) = SLEG2(Month, NR) / NIYear
          TALEG2(NR) = TALEG2(NR) + ALEG2(Month, NR)
          ALEG3(Month, NR) = SLEG3(Month, NR) / NIYear
          TALEG3(NR) = TALEG3(NR) + ALEG3(Month, NR)
          ALEG4(Month, NR) = SLEG4(Month, NR) / NIYear
          TALEG4(NR) = TALEG4(NR) + ALEG4(Month, NR)
          ATE(Month, NR) = STE(Month, NR) / NIYear
       NEXT Month


RETURN

'------------------------------Sumation
Sum.Calc:

       FOR Month = 1 TO 12
          SDLEG2S(Month, NR) = 0
          SDLEG3S(Month, NR) = 0
          SDLEG4S(Month, NR) = 0
          SDTES(Month, NR) = 0
       NEXT Month
```

```
      FOR NYear = 1 TO NIYear
         FOR Month = 1 TO 12
            DLEG2S(Month, NR) = (LEG2(Month, NYear) - ALEG2(Month, NR)) ^ 2
            SDLEG2S(Month, NR) = SDLEG2S(Month, NR) + DLEG2S(Month, NR)
            DLEG3S(Month, NR) = (LEG3(Month, NYear) - ALEG3(Month, NR)) ^ 2
            SDLEG3S(Month, NR) = SDLEG3S(Month, NR) + DLEG3S(Month, NR)
            DLEG4S(Month, NR) = (LEG4(Month, NYear) - ALEG4(Month, NR)) ^ 2
            SDLEG4S(Month, NR) = SDLEG4S(Month, NR) + DLEG4S(Month, NR)
            DTES(Month, NR) = (TE(Month, NYear) - ATE(Month, NR)) ^ 2
            SDTES(Month, NR) = SDTES(Month, NR) + DTES(Month, NR)
         NEXT Month
      NEXT NYear
RETURN
'--------------------------------StdTE
Std.Calc:

         FOR Month = 1 TO 12
            VLEG2(Month, NR) = SDLEG2S(Month, NR) / (NIYear - 1)
            StdLEG2(Month, NR) = (VLEG2(Month, NR)) ^ .5
            VLEG3(Month, NR) = SDLEG3S(Month, NR) / (NIYear - 1)
            StdLEG3(Month, NR) = (VLEG3(Month, NR)) ^ .5
            VLEG4(Month, NR) = SDLEG4S(Month, NR) / (NIYear - 1)
            StdLEG4(Month, NR) = (VLEG4(Month, NR)) ^ .5
            VTE(Month, NR) = SDTES(Month, NR) / (NIYear - 1)
            StdTE(Month, NR) = (VTE(Month, NR)) ^ .5
         NEXT Month


RETURN

OutStat.Calc:
            ATE1 = ATE(1, NR)
            ATE2 = ATE(2, NR)
            ATE3 = ATE(3, NR)
            ATE4 = ATE(4, NR)
            ATE5 = ATE(5, NR)
            ATE6 = ATE(6, NR)
```

```
            ATE7 = ATE(7, NR)
            ATE8 = ATE(8, NR)
            ATE9 = ATE(9, NR)
            ATE10 = ATE(10, NR)
            ATE11 = ATE(11, NR)
            ATE12 = ATE(12, NR)

            TATE = ATE1 + ATE2 + ATE3 + ATE4 + ATE5 + ATE6 + ATE7 + ATE8 + ATE9 + ATE10 +
ATE11 + ATE12


IF (NR = NRTest) THEN
        PRINT #7,
        PRINT #7,
        PRINT #7, "Agung W.H. SOEHARNO"
        PRINT #7, DATE$
                                                          P    R    I    N    T    #    7    ,
"******************************************************************************************"
        PRINT #7, "*            Output of Egy-Ctr Program"
        PRINT #7, "*"
        PRINT #7, "----------------------> Stat.Calc - Replicate no. ", NR
                                                          P    R    I    N    T    #    7    ,
"******************************************************************************************"

        PRINT #7,
        PRINT #7, " LEG2 = Energy of Link 2: Saguling Hydro Electric Power"
        PRINT #7,
        PRINT #7, "  NYear   May    Jun    Jul    Aug    Sep    Oct    Nov    Dec    Jan
 Feb    Mar    Apr"
        PRINT #7,
   FOR NYear = 1 TO NIYear
       PRINT #7, USING "#####.#"; NYear; LEG2(1, NYear); LEG2(2, NYear); LEG2(3, NYear);
LEG2(4, NYear); LEG2(5, NYear); LEG2(6, NYear); LEG2(7, NYear); LEG2(8, NYear); LEG2(9,
NYear); LEG2(10, NYear); LEG2(11, NYear); LEG2(12, NYear)
   NEXT NYear
        PRINT #7,
        PRINT #7, "    NR     May    Jun    Jul    Aug    Sep    Oct    Nov    Dec    Jan
```

```
 Feb     Mar     Apr"
        PRINT #7, "Minimum of LEG2"
        PRINT #7, USING "#####.#"; NR; MinLEG2(1); MinLEG2(2); MinLEG2(3); MinLEG2(4);
MinLEG2(5); MinLEG2(6); MinLEG2(7); MinLEG2(8); MinLEG2(9); MinLEG2(10); MinLEG2(11);
MinLEG2(12)
        PRINT #7, "ALEG2 = Average of LEG2"
        PRINT #7, USING "#####.#"; NR; ALEG2(1, NR); ALEG2(2, NR); ALEG2(3, NR); ALEG2(4,
NR); ALEG2(5, NR); ALEG2(6, NR); ALEG2(7, NR); ALEG2(8, NR); ALEG2(9, NR); ALEG2(10, NR);
ALEG2(11, NR); ALEG2(12, NR)
        PRINT #7, "Maximum of LEG2"
        PRINT #7, USING "#####.#"; NR; MaxLEG2(1); MaxLEG2(2); MaxLEG2(3); MaxLEG2(5);
MaxLEG2(5); MaxLEG2(6); MaxLEG2(7); MaxLEG2(8); MaxLEG2(9); MaxLEG2(10); MaxLEG2(11);
MaxLEG2(12)
        PRINT #7, "StdLEG2 = Standard Dev. of LEG2"
        PRINT #7, USING "#####.#"; NR; StdLEG2(1, NR); StdLEG2(2, NR); StdLEG2(3, NR);
StdLEG2(4, NR); StdLEG2(5, NR); StdLEG2(6, NR); StdLEG2(7, NR); StdLEG2(8, NR); StdLEG2(9,
NR); StdLEG2(10, NR); StdLEG2(11, NR); StdLEG2(12, NR)

        PRINT #7,
        PRINT #7, " LEG3 = Energy of Link 3: Cirata Hydro Electric Power"
        PRINT #7,
        PRINT #7, "  NYear    May     Jun     Jul     Aug     Sep     Oct     Nov     Dec     Jan
 Feb     Mar     Apr"
        PRINT #7,
   FOR NYear = 1 TO NIYear
       PRINT #7, USING "#####.#"; NYear; LEG3(1, NYear); LEG3(2, NYear); LEG3(3, NYear);
LEG3(4, NYear); LEG3(5, NYear); LEG3(6, NYear); LEG3(7, NYear); LEG3(8, NYear); LEG3(9,
NYear); LEG3(10, NYear); LEG3(11, NYear); LEG3(12, NYear)
   NEXT NYear
        PRINT #7,
        PRINT #7, "   NR      May     Jun     Jul     Aug     Sep     Oct     Nov     Dec     Jan
 Feb     Mar     Apr"
        PRINT #7, "Minimum of LEG3"
        PRINT #7, USING "#####.#"; NR; MinLEG3(1); MinLEG3(2); MinLEG3(3); MinLEG3(4);
MinLEG3(5); MinLEG3(6); MinLEG3(7); MinLEG3(8); MinLEG3(9); MinLEG3(10); MinLEG3(11);
MinLEG3(12)
        PRINT #7, "ALEG3 = Average of LEG3"
```

```
        PRINT #7, USING "#####.#"; NR; ALEG3(1, NR); ALEG3(2, NR); ALEG3(3, NR); ALEG3(4,
NR); ALEG3(5, NR); ALEG3(6, NR); ALEG3(7, NR); ALEG3(8, NR); ALEG3(9, NR); ALEG3(10, NR);
ALEG3(11, NR); ALEG3(12, NR)
        PRINT #7, "Maximum of LEG3"
        PRINT #7, USING "#####.#"; NR; MaxLEG3(1); MaxLEG3(2); MaxLEG3(3); MaxLEG3(5);
MaxLEG3(5); MaxLEG3(6); MaxLEG3(7); MaxLEG3(8); MaxLEG3(9); MaxLEG3(10); MaxLEG3(11);
MaxLEG3(12)
        PRINT #7, "StdLEG3 = Standard Dev. of LEG3"
        PRINT #7, USING "#####.#"; NR; StdLEG3(1, NR); StdLEG3(2, NR); StdLEG3(3, NR);
StdLEG3(4, NR); StdLEG3(5, NR); StdLEG3(6, NR); StdLEG3(7, NR); StdLEG3(8, NR); StdLEG3(9,
NR); StdLEG3(10, NR); StdLEG3(11, NR); StdLEG3(12, NR)

        PRINT #7,
        PRINT #7, " LEG4 = Energy of Link 4: Jatiluhur Hydro Electric Power"
        PRINT #7,
        PRINT #7, "  NYear   May    Jun    Jul    Aug    Sep    Oct    Nov    Dec    Jan
 Feb    Mar    Apr"
        PRINT #7,
   FOR NYear = 1 TO NIYear
       PRINT #7, USING "#####.#"; NYear; LEG4(1, NYear); LEG4(2, NYear); LEG4(3, NYear);
LEG4(4, NYear); LEG4(5, NYear); LEG4(6, NYear); LEG4(7, NYear); LEG4(8, NYear); LEG4(9,
NYear); LEG4(10, NYear); LEG4(11, NYear); LEG4(12, NYear)
   NEXT NYear
        PRINT #7,
        PRINT #7, "   NR     May    Jun    Jul    Aug    Sep    Oct    Nov    Dec    Jan
 Feb    Mar    Apr"
        PRINT #7, "Minimum of LEG4"
        PRINT #7, USING "#####.#"; NR; MinLEG4(1); MinLEG4(2); MinLEG4(3); MinLEG4(4);
MinLEG4(5); MinLEG4(6); MinLEG4(7); MinLEG4(8); MinLEG4(9); MinLEG4(10); MinLEG4(11);
MinLEG4(12)
        PRINT #7, "ALEG4 = Average of LEG4"
        PRINT #7, USING "#####.#"; NR; ALEG4(1, NR); ALEG4(2, NR); ALEG4(3, NR); ALEG4(4,
NR); ALEG4(5, NR); ALEG4(6, NR); ALEG4(7, NR); ALEG4(8, NR); ALEG4(9, NR); ALEG4(10, NR);
ALEG4(11, NR); ALEG4(12, NR)
        PRINT #7, "Maximum of LEG4"
        PRINT #7, USING "#####.#"; NR; MaxLEG4(1); MaxLEG4(2); MaxLEG4(3); MaxLEG4(5);
MaxLEG4(5); MaxLEG4(6); MaxLEG4(7); MaxLEG4(8); MaxLEG4(9); MaxLEG4(10); MaxLEG4(11);
```

```
MaxLEG4(12)
        PRINT #7, "StdLEG4 = Standard Dev. of LEG4"
        PRINT #7, USING "#####.#"; NR; StdLEG4(1, NR); StdLEG4(2, NR); StdLEG4(3, NR);
StdLEG4(4, NR); StdLEG4(5, NR); StdLEG4(6, NR); StdLEG4(7, NR); StdLEG4(8, NR); StdLEG4(9,
NR); StdLEG4(10, NR); StdLEG4(11, NR); StdLEG4(12, NR)

        PRINT #7,
        PRINT #7, "TE = Total Energy of Saguling, Cirata and Jatiluhur"
        PRINT #7,
        PRINT #7, "  NYear    May     Jun     Jul     Aug     Sep     Oct     Nov     Dec     Jan
 Feb     Mar     Apr"
        PRINT #7,
   FOR NYear = 1 TO NIYear
       PRINT #7, USING "#####.#"; NYear; TE(1, NYear); TE(2, NYear); TE(3, NYear); TE(4,
NYear); TE(5, NYear); TE(6, NYear); TE(7, NYear); TE(8, NYear); TE(9, NYear); TE(10, NYear);
TE(11, NYear); TE(12, NYear)
   NEXT NYear
        PRINT #7,
        PRINT #7, "   NR     May     Jun     Jul     Aug     Sep     Oct     Nov     Dec     Jan
 Feb     Mar     Apr"
        PRINT #7, "Minimum of TE"
        PRINT #7, USING "#####.#"; NR; MinTE(1); MinTE(2); MinTE(3); MinTE(4); MinTE(5);
MinTE(6); MinTE(7); MinTE(8); MinTE(9); MinTE(10); MinTE(11); MinTE(12)
        PRINT #7, "ATE = Average of TE"
        PRINT #7, USING "#####.#"; NR; ATE(1, NR); ATE(2, NR); ATE(3, NR); ATE(4, NR);
ATE(5, NR); ATE(6, NR); ATE(7, NR); ATE(8, NR); ATE(9, NR); ATE(10, NR); ATE(11, NR); ATE(12,
NR)
        PRINT #7, "Maximum of TE"
        PRINT #7, USING "#####.#"; NR; MaxTE(1); MaxTE(2); MaxTE(3); MaxTE(5); MaxTE(5);
MaxTE(6); MaxTE(7). MaxTE(8); MaxTE(9); MaxTE(10); MaxTE(11); MaxTE(12)
        PRINT #7, "StdTE = Standard Dev. of TE"
        PRINT #7, USING "#####.#"; NR; StdTE(1, NR); StdTE(2, NR); StdTE(3, NR); StdTE(4,
NR); StdTE(5, NR); StdTE(6, NR); StdTE(7, NR); StdTE(8, NR); StdTE(9, NR); StdTE(10, NR);
StdTE(11, NR); StdTE(12, NR)
        PRINT #7,

        PRINT #16,
```

```
        PRINT #16,
        PRINT #16, "Agung W.H. SOEHARNO"
        PRINT #16, DATE$, TIME$
                                                      P    R    I    N    T    #    1    6    ,
"*****************************************************************************************"
        PRINT #16, "*            Output of Egy-Ctr Program"
        PRINT #16, "*"
        PRINT #16, "----------------------> Stat.Calc - Replicate no. ", NR
                                                      P    R    I    N    T    #    1    6    ,
"*****************************************************************************************"
        PRINT #16,
        PRINT #16, " ESUM = Summary Files of Link Energy"
        PRINT #16,
        PRINT #16, " NYear   May    Jun    Jul    Aug    Sep    Oct    Nov    Dec    Jan
  Feb    Mar    Apr"
        PRINT #16, "ALEG2 = Average of LEG2"
        PRINT #16, USING "#####.#"; NR; ALEG2(1, NR); ALEG2(2, NR); ALEG2(3, NR); ALEG2(4,
NR); ALEG2(5, NR); ALEG2(6, NR); ALEG2(7, NR); ALEG2(8, NR); ALEG2(9, NR); ALEG2(10, NR);
ALEG2(11, NR); ALEG2(12, NR)
        PRINT #16, "TALEG2(NR)"; NR; TALEG2(NR)
        PRINT #16, "ALEG3 = Average of LEG3"
        PRINT #16, USING "#####.#"; NR; ALEG3(1, NR); ALEG3(2, NR); ALEG3(3, NR); ALEG3(4,
NR); ALEG3(5, NR); ALEG3(6, NR); ALEG3(7, NR); ALEG3(8, NR); ALEG3(9, NR); ALEG3(10, NR);
ALEG3(11, NR); ALEG3(12, NR)
        PRINT #16, "TALEG3(NR)"; NR; TALEG3(NR)
        PRINT #16, "ALEG4 = Average of LEG4"
        PRINT #16, USING "#####.#"; NR; ALEG4(1, NR); ALEG4(2, NR); ALEG4(3, NR); ALEG4(4,
NR); ALEG4(5, NR); ALEG4(6, NR); ALEG4(7, NR); ALEG4(8, NR); ALEG4(9, NR); ALEG4(10, NR);
ALEG4(11, NR); ALEG4(12, NR)
        PRINT #16, "TALEG4(NR)"; NR; TALEG4(NR)
        PRINT #16, "ATE = Average of TE"
        PRINT #16, USING "#####.#"; NR; ATE(1, NR); ATE(2, NR); ATE(3, NR); ATE(4, NR);
ATE(5, NR); ATE(6, NR); ATE(7, NR); ATE(8, NR); ATE(9, NR); ATE(10, NR); ATE(11, NR); ATE(12,
NR)
        PRINT #16, "TATE = Total of Average of TE"
        PRINT #16, USING "############.###"; TATE
ELSE
```

```
END IF
        WRITE #8, ALEG2(1, NR), ALEG2(2, NR), ALEG2(3, NR), ALEG2(4, NR), ALEG2(5, NR),
ALEG2(6, NR), ALEG2(7, NR), ALEG2(8, NR), ALEG2(9, NR), ALEG2(10, NR), ALEG2(11, NR),
ALEG2(12, NR)
        WRITE #8, ALEG3(1, NR), ALEG3(2, NR), ALEG3(3, NR), ALEG3(4, NR), ALEG3(5, NR),
ALEG3(6, NR), ALEG3(7, NR), ALEG3(8, NR), ALEG3(9, NR), ALEG3(10, NR), ALEG3(11, NR),
ALEG3(12, NR)
        WRITE #8, ALEG4(1, NR), ALEG4(2, NR), ALEG4(3, NR), ALEG4(4, NR), ALEG4(5, NR),
ALEG4(6, NR), ALEG4(7, NR), ALEG4(8, NR), ALEG4(9, NR), ALEG4(10, NR), ALEG4(11, NR),
ALEG4(12, NR)
        WRITE #8, ATE(1, NR), ATE(2, NR), ATE(3, NR), ATE(4, NR), ATE(5, NR), ATE(6, NR),
ATE(7, NR), ATE(8, NR), ATE(9, NR), ATE(10, NR), ATE(11, NR), ATE(12, NR)

        WRITE #8, StdLEG2(1, NR), StdLEG2(2, NR), StdLEG2(3, NR), StdLEG2(4, NR), StdLEG2(5,
NR), StdLEG2(6, NR), StdLEG2(7, NR), StdLEG2(8, NR), StdLEG2(9, NR), StdLEG2(10, NR),
StdLEG2(11, NR), StdLEG2(12, NR)
        WRITE #8, StdLEG3(1, NR), StdLEG3(2, NR), StdLEG3(3, NR), StdLEG3(4, NR), StdLEG3(5,
NR), StdLEG3(6, NR), StdLEG3(7, NR), StdLEG3(8, NR), StdLEG3(9, NR), StdLEG3(10, NR),
StdLEG3(11, NR), StdLEG3(12, NR)
        WRITE #8, StdLEG4(1, NR), StdLEG4(2, NR), StdLEG4(3, NR), StdLEG4(4, NR), StdLEG4(5,
NR), StdLEG4(6, NR), StdLEG4(7, NR), StdLEG4(8, NR), StdLEG4(9, NR), StdLEG4(10, NR),
StdLEG4(11, NR), StdLEG4(12, NR)
        WRITE #8, StdTE(1, NR), StdTE(2, NR), StdTE(3, NR), StdTE(4, NR), StdTE(5, NR),
StdTE(6, NR), StdTE(7, NR), StdTE(8, NR), StdTE(9, NR), StdTE(10, NR), StdTE(11, NR),
StdTE(12, NR)
RETURN

'**************************************************************************
'                         Statistics.TEst
'**************************************************************************

Statistics.TESt:

PRINT "---------------> S t a t i s t i c s . T e s t"
PRINT #7, "--------------> S t a t i s t i c s . T e s t"
PRINT #7,
```

```
PRINT #7, "    May    Jun    Jul    Aug    Sep    Oct    Nov    Dec    Jan    Feb    Mar
Apr"
PRINT #7,

GOSUB Input.Data
GOSUB AvgAvg.Calc
GOSUB SumSum.Calc
GOSUB StdStd.Calc
GOSUB PrintStat.TEs
RETURN

'---------------------------- Input.Data
Input.Data:

CLOSE #8

OPEN "c:\egy\EInput3.dat" FOR INPUT AS #9  'input-output file

    FOR NR = 1 TO IReplicate
        INPUT #9, ALEG2(1, NR), ALEG2(2, NR), ALEG2(3, NR), ALEG2(4, NR), ALEG2(5, NR),
ALEG2(6, NR), ALEG2(7, NR), ALEG2(8, NR), ALEG2(9, NR), ALEG2(10, NR), ALEG2(11, NR),
ALEG2(12, NR)
        INPUT #9, ALEG3(1, NR), ALEG3(2, NR), ALEG3(3, NR), ALEG3(4, NR), ALEG3(5, NR),
ALEG3(6, NR), ALEG3(7, NR), ALEG3(8, NR), ALEG3(9, NR), ALEG3(10, NR), ALEG3(11, NR),
ALEG3(12, NR)
        INPUT #9, ALEG4(1, NR), ALEG4(2, NR), ALEG4(3, NR), ALEG4(4, NR), ALEG4(5, NR),
ALEG4(6, NR), ALEG4(7, NR), ALEG4(8, NR), ALEG4(9, NR), ALEG4(10, NR), ALEG4(11, NR),
ALEG4(12, NR)
        INPUT #9, ATE(1, NR), ATE(2, NR), ATE(3, NR), ATE(4, NR), ATE(5, NR), ATE(6, NR),
ATE(7, NR), ATE(8, NR), ATE(9, NR), ATE(10, NR), ATE(11, NR), ATE(12, NR)

        INPUT #9, StdLEG2(1, NR), StdLEG2(2, NR), StdLEG2(3, NR), StdLEG2(4, NR), StdLEG2(5,
NR), StdLEG2(6, NR), StdLEG2(7, NR), StdLEG2(8, NR), StdLEG2(9, NR), StdLEG2(10, NR),
StdLEG2(11, NR), StdLEG2(12, NR)
        INPUT #9, StdLEG3(1, NR), StdLEG3(2, NR), StdLEG3(3, NR), StdLEG3(4, NR), StdLEG3(5,
NR), StdLEG3(6, NR), StdLEG3(7, NR), StdLEG3(8, NR), StdLEG3(9, NR), StdLEG3(10, NR),
StdLEG3(11, NR), StdLEG3(12, NR)
```

```
        INPUT #9, StdLEG4(1, NR), StdLEG4(2, NR), StdLEG4(3, NR), StdLEG4(4, NR), StdLEG4(5,
NR), StdLEG4(6, NR), StdLEG4(7, NR), StdLEG4(8, NR), StdLEG4(9, NR), StdLEG4(10, NR),
StdLEG4(11, NR), StdLEG4(12, NR)
        INPUT #9, StdTE(1, NR), StdTE(2, NR), StdTE(3, NR), StdTE(4, NR), StdTE(5, NR),
StdTE(6, NR), StdTE(7, NR), StdTE(8, NR), StdTE(9, NR), StdTE(10, NR), StdTE(11, NR),
StdTE(12, NR)
    NEXT NR
    CLOSE #9

RETURN

'------------------------------ AvgAvg.Calc
AvgAvg.Calc:
DIM SALEG2(100), SStdLEG2(100), AALEG2(100), AStdLEG2(100), SALEG3(100), SStdLEG3(100)
DIM AALEG3(100), AStdLEG3(100), SALEG4(100), SStdLEG4(100), AALEG4(100), AStdLEG4(100)
DIM SATE(100), SStdTE(100), AATE(100), AStdTE(100)

'---------- Average of TE

      FOR Month = 1 TO 12
         SALEG2(Month) = 0
         SStdLEG2(Month) = 0
         SALEG3(Month) = 0
         SStdLEG3(Month) = 0
         SALEG4(Month) = 0
         SStdLEG4(Month) = 0
         SATE(Month) = 0
         SStdTE(Month) = 0
      NEXT Month

   FOR Month = 1 TO 12
      FOR NR = 1 TO IReplicate
         SALEG2(Month) = SALEG2(Month) + ALEG2(Month, NR)
         SStdLEG2(Month) = SStdLEG2(Month) + StdLEG2(Month, NR)
         SALEG3(Month) = SALEG3(Month) + ALEG3(Month, NR)
         SStdLEG3(Month) = SStdLEG3(Month) + StdLEG3(Month, NR)
         SALEG4(Month) = SALEG4(Month) + ALEG4(Month, NR)
```

```
            SStdLEG4(Month) = SStdLEG4(Month) + StdLEG4(Month, NR)
            SATE(Month) = SATE(Month) + ATE(Month, NR)
            SStdTE(Month) = SStdTE(Month) + StdTE(Month, NR)
        NEXT NR
      NEXT Month

        FOR Month = 1 TO 12
            AALEG2(Month) = SALEG2(Month) / IReplicate
            AStdLEG2(Month) = SStdLEG2(Month) / IReplicate
            AALEG3(Month) = SALEG3(Month) / IReplicate
            AStdLEG3(Month) = SStdLEG3(Month) / IReplicate
            AALEG4(Month) = SALEG4(Month) / IReplicate
            AStdLEG4(Month) = SStdLEG4(Month) / IReplicate
            AATE(Month) = SATE(Month) / IReplicate
            AStdTE(Month) = SStdTE(Month) / IReplicate
        NEXT Month

RETURN

'------------------------------Sumation of Sumation
SumSum.Calc:
DIM SDALEG2S(100), SDStdLEG2S(100), DALEG2S(100), DStdLEG2S(100), SDALEG3S(100)
DIM SDStdLEG3S(100), DALEG3S(100), DStdLEG3S(100), SDALEG4S(100), SDStdLEG4S(100)
DIM DALEG4S(100), DStdLEG4S(100), SDATES(100), SDStdTES(100), DATES(100), DStdTES(100)

        FOR Month = 1 TO 100
            SDALEG2S(Month) = 0
            SDStdLEG2S(Month) = 0
            SDALEG3S(Month) = 0
            SDStdLEG3S(Month) = 0
            SDALEG4S(Month) = 0
            SDStdLEG4S(Month) = 0
            SDATES(Month) = 0
            SDStdTES(Month) = 0
        NEXT Month
```

```
     FOR Month = 1 TO 12
        FOR NR = 1 TO IReplicate
           DALEG2S(Month) = (ALEG2(Month, NR) - AALEG2(Month)) ^ 2
           SDALEG2S(Month) = SDALEG2S(Month) + DALEG2S(Month)
           DStdLEG2S(Month) = (StdLEG2(Month, NR) - AStdLEG2(Month)) ^ 2
           SDStdLEG2S(Month) = SDStdLEG2S(Month) + DStdLEG2S(Month)
           DALEG3S(Month) = (ALEG3(Month, NR) - AALEG3(Month)) ^ 2
           SDALEG3S(Month) = SDALEG3S(Month) + DALEG3S(Month)
           DStdLEG3S(Month) = (StdLEG3(Month, NR) - AStdLEG3(Month)) ^ 2
           SDStdLEG3S(Month) = SDStdLEG3S(Month) + DStdLEG3S(Month)
           DALEG4S(Month) = (ALEG4(Month, NR) - AALEG4(Month)) ^ 2
           SDALEG4S(Month) = SDALEG4S(Month) + DALEG4S(Month)
           DStdLEG4S(Month) = (StdLEG4(Month, NR) - AStdLEG4(Month)) ^ 2
           SDStdLEG4S(Month) = SDStdLEG4S(Month) + DStdLEG4S(Month)
           DATES(Month) = (ATE(Month, NR) - AATE(Month)) ^ 2
           SDATES(Month) = SDATES(Month) + DATES(Month)
           DStdTES(Month) = (StdTE(Month, NR) - AStdTE(Month)) ^ 2
           SDStdTES(Month) = SDStdTES(Month) + DStdTES(Month)
        NEXT NR
     NEXT Month

10 RETURN

'---------------------------------Std of StdTE
StdStd.Calc:
DIM VALEG2(100), StdALEG2(100), VStdLEG2(100), StdStdLEG2(100), VALEG3(100)
DIM StdALEG3(100), VStdLEG3(100), StdStdLEG3(100), VALEG4(100), StdALEG4(100)
DIM VStdLEG4(100), StdStdLEG4(100), VATE(100), StdATE(100), VStdTE(100), StdStdTE(100)

        FOR Month = 1 TO 12
           VALEG2(Month) = SDALEG2S(Month) / (IReplicate - 1)
           StdALEG2(Month) = (VALEG2(Month)) ^ .5
           VStdLEG2(Month) = SDStdLEG2S(Month) / (IReplicate - 1)
           StdStdLEG2(Month) = (VStdLEG2(Month)) ^ .5
```

```
        VALEG3(Month) = SDALEG3S(Month) / (IReplicate - 1)
        StdALEG3(Month) = (VALEG3(Month)) ^ .5
        VStdLEG3(Month) = SDStdLEG3S(Month) / (IReplicate - 1)
        StdStdLEG3(Month) = (VStdLEG3(Month)) ^ .5

        VALEG4(Month) = SDALEG4S(Month) / (IReplicate - 1)
        StdALEG4(Month) = (VALEG4(Month)) ^ .5
        VStdLEG4(Month) = SDStdLEG4S(Month) / (IReplicate - 1)
        StdStdLEG4(Month) = (VStdLEG4(Month)) ^ .5

        VATE(Month) = SDATES(Month) / (IReplicate - 1)
        StdATE(Month) = (VATE(Month)) ^ .5
        VStdTE(Month) = SDStdTES(Month) / (IReplicate - 1)
        StdStdTE(Month) = (VStdTE(Month)) ^ .5

    NEXT Month

RETURN

'------------------ PrintStat.TEs
PrintStat.TEs:
    PRINT #7, "AALEG2 = "
    PRINT #7, USING "#####.#"; AALEG2(1); AALEG2(2); AALEG2(3); AALEG2(4); AALEG2(5);
AALEG2(6); AALEG2(7); AALEG2(8); AALEG2(9); AALEG2(10); AALEG2(11); AALEG2(12)
    PRINT #7, "AStdLEG2 ="
    PRINT #7, USING "#####.#"; AStdLEG2(1); AStdLEG2(2); AStdLEG2(3); AStdLEG2(4);
AStdLEG2(5); AStdLEG2(6); AStdLEG2(7); AStdLEG2(8); AStdLEG2(9); AStdLEG2(10); AStdLEG2(11);
AStdLEG2(12)

    PRINT #7, "AALEG3 = "
    PRINT #7, USING "#####.#"; AALEG3(1); AALEG3(2); AALEG3(3); AALEG3(4); AALEG3(5);
AALEG3(6); AALEG3(7); AALEG3(8); AALEG3(9); AALEG3(10); AALEG3(11); AALEG3(12)
    PRINT #7, "AStdLEG3 ="
    PRINT #7, USING "#####.#"; AStdLEG3(1); AStdLEG3(2); AStdLEG3(3); AStdLEG3(4);
AStdLEG3(5); AStdLEG3(6); AStdLEG3(7); AStdLEG3(8); AStdLEG3(9); AStdLEG3(10); AStdLEG3(11);
AStdLEG3(12)
```

```
    PRINT #7, "AALEG4 = "
    PRINT #7, USING "#####.#"; AALEG4(1); AALEG4(2); AALEG4(3); AALEG4(4); AALEG4(5);
AALEG4(6); AALEG4(7); AALEG4(8); AALEG4(9); AALEG4(10); AALEG4(11); AALEG4(12)
    PRINT #7, "AStdLEG4 ="
    PRINT #7, USING "#####.#"; AStdLEG4(1); AStdLEG4(2); AStdLEG4(3); AStdLEG4(4);
AStdLEG4(5); AStdLEG4(6); AStdLEG4(7); AStdLEG4(8); AStdLEG4(9); AStdLEG4(10); AStdLEG4(11);
AStdLEG4(12)

    PRINT #7, "AATE = "
    PRINT #7, USING "#####.#"; AATE(1); AATE(2); AATE(3); AATE(4); AATE(5); AATE(6);
AATE(7); AATE(8); AATE(9); AATE(10); AATE(11); AATE(12)
    PRINT #7, "AStdTE ="
    PRINT #7, USING "#####.#"; AStdTE(1); AStdTE(2); AStdTE(3); AStdTE(4); AStdTE(5);
AStdTE(6); AStdTE(7); AStdTE(8); AStdTE(9); AStdTE(10); AStdTE(11); AStdTE(12)
    TAATE = AATE(1) + AATE(2) + AATE(3) + AATE(4) + AATE(5) + AATE(6) + AATE(7) + AATE(8)
+ AATE(9) + AATE(10) + AATE(11) + AATE(12)
    PRINT #7, "TAATE"
    PRINT #7, TAATE

    PRINT #7, "StdALEG2 = "
    PRINT #7, USING "#####.#"; StdALEG2(1); StdALEG2(2); StdALEG2(3); StdALEG2(4);
StdALEG2(5); StdALEG2(6); StdALEG2(7); StdALEG2(8); StdALEG2(9); StdALEG2(10); StdALEG2(11);
StdALEG2(12)
    PRINT #7, "StdStdLEG2 = "
    PRINT #7, USING "#####.#"; StdStdLEG2(1); StdStdLEG2(2); StdStdLEG2(3); StdStdLEG2(4);
StdStdLEG2(5); StdStdLEG2(6); StdStdLEG2(7); StdStdLEG2(8); StdStdLEG2(9); StdStdLEG2(10);
StdStdLEG2(11); StdStdLEG2(12)

    PRINT #7, "StdALEG3 = "
    PRINT #7, USING "#####.#"; StdALEG3(1); StdALEG3(2); StdALEG3(3); StdALEG3(4);
StdALEG3(5); StdALEG3(6); StdALEG3(7); StdALEG3(8); StdALEG3(9); StdALEG3(10); StdALEG3(11);
StdALEG3(12)
    PRINT #7, "StdStdLEG3 = "
    PRINT #7, USING "#####.#"; StdStdLEG3(1); StdStdLEG3(2); StdStdLEG3(3); StdStdLEG3(4);
StdStdLEG3(5); StdStdLEG3(6); StdStdLEG3(7); StdStdLEG3(8); StdStdLEG3(9); StdStdLEG3(10);
StdStdLEG3(11); StdStdLEG3(12)
```

```
     PRINT #7, "StdALEG4 = "
     PRINT #7, USING "#####.#"; StdALEG4(1); StdALEG4(2); StdALEG4(3); StdALEG4(4);
StdALEG4(5); StdALEG4(6); StdALEG4(7); StdALEG4(8); StdALEG4(9); StdALEG4(10); StdALEG4(11);
StdALEG4(12)
     PRINT #7, "StdStdLEG4 = "
     PRINT #7, USING "#####.#"; StdStdLEG4(1); StdStdLEG4(2); StdStdLEG4(3); StdStdLEG4(4);
StdStdLEG4(5); StdStdLEG4(6); StdStdLEG4(7); StdStdLEG4(8); StdStdLEG4(9); StdStdLEG4(10);
StdStdLEG4(11); StdStdLEG4(12)

     PRINT #7, "StdATE = "
     PRINT #7, USING "#####.#"; StdATE(1); StdATE(2); StdATE(3); StdATE(4); StdATE(5);
StdATE(6); StdATE(7); StdATE(8); StdATE(9); StdATE(10); StdATE(11); StdATE(12)
     PRINT #7, "StdStdTE = "
     PRINT #7, USING "#####.#"; StdStdTE(1); StdStdTE(2); StdStdTE(3); StdStdTE(4);
StdStdTE(5); StdStdTE(6); StdStdTE(7); StdStdTE(8); StdStdTE(9); StdStdTE(10); StdStdTE(11);
StdStdTE(12)

         PRINT #16,
         PRINT #16,
         PRINT #16, "Agung W.H. SOEHARNO"
         PRINT #16, DATE$, TIME$
                                                     P    R    I    N    T    #    1    6    ,
"**************************************************************************************"
         PRINT #16, "*          Output of Egy-Ctr Program"
         PRINT #16, "*"
         PRINT #16, "-----------------------> Statistics.Test"
                                                     P    R    I    N    T    #    1    6    ,
"**************************************************************************************"
         PRINT #16,

     PRINT #16, "AALEG2 = "
     PRINT #16, USING "#####.#"; AALEG2(1); AALEG2(2); AALEG2(3); AALEG2(4); AALEG2(5);
AALEG2(6); AALEG2(7); AALEG2(8); AALEG2(9); AALEG2(10); AALEG2(11); AALEG2(12)
     TAALEG2 = AALEG2(1) + AALEG2(2) + AALEG2(3) + AALEG2(4) + AALEG2(5) + AALEG2(6) +
AALEG2(7) + AALEG2(8) + AALEG2(9) + AALEG2(10) + AALEG2(11) + AALEG2(12)
     PRINT #16, "StdALEG2 = "
```

```
     PRINT #16,
     PRINT #16,

RETURN

'*******************************************************
'*                      Wrap-Up
'*******************************************************
Wrap.Up:

     PRINT #7, "     The Program of EGY-Ctr is finish."

RETURN
```

```
     PRINT #16, USING "#####.#"; StdALEG2(1); StdALEG2(2); StdALEG2(3); StdALEG2(4);
StdALEG2(5); StdALEG2(6); StdALEG2(7); StdALEG2(8); StdALEG2(9); StdALEG2(10); StdALEG2(11);
StdALEG2(12)

     PRINT #16, "AALEG3 = "
     PRINT #16, USING "#####.#"; AALEG3(1); AALEG3(2); AALEG3(3); AALEG3(4); AALEG3(5);
AALEG3(6); AALEG3(7); AALEG3(8); AALEG3(9); AALEG3(10); AALEG3(11); AALEG3(12)
     TAALEG3 = AALEG3(1) + AALEG3(2) + AALEG3(3) + AALEG3(4) + AALEG3(5) + AALEG3(6) +
AALEG3(7) + AALEG3(8) + AALEG3(9) + AALEG3(10) + AALEG3(11) + AALEG3(12)
     PRINT #16, "StdALEG3 = "
     PRINT #16, USING "#####.#"; StdALEG3(1); StdALEG3(2); StdALEG3(3); StdALEG3(4);
StdALEG3(5); StdALEG3(6); StdALEG3(7); StdALEG3(8); StdALEG3(9); StdALEG3(10); StdALEG3(11);
StdALEG3(12)

     PRINT #16, "AALEG4 = "
     PRINT #16, USING "#####.#"; AALEG4(1); AALEG4(2); AALEG4(3); AALEG4(4); AALEG4(5);
AALEG4(6); AALEG4(7); AALEG4(8); AALEG4(9); AALEG4(10); AALEG4(11); AALEG4(12)
     TAALEG4 = AALEG4(1) + AALEG4(2) + AALEG4(3) + AALEG4(4) + AALEG4(5) + AALEG4(6) +
AALEG4(7) + AALEG4(8) + AALEG4(9) + AALEG4(10) + AALEG4(11) + AALEG4(12)
     PRINT #16, "StdALEG4 = "
     PRINT #16, USING "#####.#"; StdALEG4(1); StdALEG4(2); StdALEG4(3); StdALEG4(4);
StdALEG4(5); StdALEG4(6); StdALEG4(7); StdALEG4(8); StdALEG4(9); StdALEG4(10); StdALEG4(11);
StdALEG4(12)

     PRINT #16, "AATE = "
     PRINT #16, USING "#####.#"; AATE(1); AATE(2); AATE(3); AATE(4); AATE(5); AATE(6);
AATE(7); AATE(8); AATE(9); AATE(10); AATE(11); AATE(12)
     PRINT #16, "StdATE = "
     PRINT #16, USING "#####.#"; StdATE(1); StdATE(2); StdATE(3); StdATE(4); StdATE(5);
StdATE(6); StdATE(7); StdATE(8); StdATE(9); StdATE(10); StdATE(11); StdATE(12)

     PRINT #16,
     PRINT #16,
     PRINT #16, "TAALEG2 = "; TAALEG2
     PRINT #16, "TAALEG3 = "; TAALEG3
     PRINT #16, "TAALEG4 = "; TAALEG4
     PRINT #16, "TAATE   = "; TAATE
```

Bibliography

1. Alexsandrovskii, A. YU.; PIMKIN, A. V. and Chernenko, G. F. (1986). "Comparison of Various Methods of Controlling the Long-Term Operating Regimes of Hydroelectric Stations", *Hydrotech Contruction*, Vol. 20, No. 2, pp. 67-69.

2. Burges, Stephen J. (1972). "Some Problems with Log-Normal Markov Runoff Models", *Journal of the Hydraulics Division*. Vol. HY 9, pp. 1487-1495.

3. Croley II, Thomas E. and Rao, K. N. Raja, (1977). *Stochastic Trade-offs for Reservoir Operation*. Iowa Institute of Hydraulic Research, The University of Iowa, Iowa City, 143 p.

4. Croley II, Thomas E., (1978). "Risk in Reservoir Design and Operation: A State-of-the-art Review". *International Simposium on Risk and Reliability in Water Resources*. University of Waterloo, Waterloo, Ontario, pp. 156-176.

5. Devore, J. L., (1987). *Probability and Statistics for Engineering and the Sciences*, Brook/Cole Publishing Company, Moterey, California, 672 p.

6. Djordjevic, B., Jovanovic, S., Opricovic, S. and Simonovic, S., (1975). "Simulation Models of the Velika Morava Basin for Design of a Multipurpose Reservoir System", *Proceedings of the Bratislava Symposium,* IAHS-AISH Pub. No. 115, pp. 85-94.

7. Evenson, Donald E. and Mosely, Joe C., (1970). "Simulation / Optimization Techniques for Multi-Basin Water Resource Planning", *Water Resources Bulletin*. Vol. 6, No. 5, pp. 725-736.

8. Gan, Thian Y., and Beard, Leo R., (1980). *Reliability of Stochastic Models Generating Hydrologic Series*, Centre for Research in Water Resources, The Univ. of Texas at Austin, Austin Texas, 64 p.

9. Jackson, Barbara Bund, (1975). "The Use of Streamflow Models in Planning". *Water Resources Research*, Vol. 11, No. 1, pp. 54-63.

10. Kariyawasam, H. and Beard, Leo R., (1980). *Evaluation of Hydropower Potential in a River Basin*, Center for Research in Water Resources, The University of Texas at Austin, Austin Texas, 141 p.

11. Kite, G. W., (1988). *Frequency and Risk Analyses in Hydrology*, Water Resources Publications, Colorado, U.S.A., 257 p.

12. Kottegoda, N. T., (1980). *Stochastic Water Resources Technology*, A Halsted Press Book, John Wiley & Sons, New York, 384 p.

13. Kuo, Jan-Tai and Yang, Feng-Rong, (1987). "The Generation of Ten-day Period Streamflow of the Tanshui River Basin Using Disaggregation Models". *Journal of the Chinese Institute of Engineers*. Vol. 10, No. 1, pp. 1-12.

14. Kuo, Jan-Tai., Hsu, Nien-Sheng., Chu, Wen-sen, Wan, Shian and Lin, Youn-Jan, (1990). "Real-Time Operation of Tanshui River Reservoirs", *Journal of Water Resources Planning and Management*, Vol. 116, No. 3, pp. 349-361.

15. Lee, Han-Lin, Brill, Jr. E Downey and Liebman, Jon C. (1987). *Reliability of Reservoir Operation Under Hydrologic Uncertainty*. Water Resources Centre, University of Illinois, Urbana, I.L., 172 p.

16. Mc Bean, E. A.; Hipel, K. W. and Unny, T. E., (1979). *Reliability in Water Resources Management*, Water Resources Publications, Fort Collins, Colorado, U.S.A., 407 p.

17. Mejia, Jose M; Egli, Paul and Leclerc, Andre. (1974), "Evaluating Multireservoir Operating Rules", *Water Resources Research*, Vol. 10, No. 6, pp. 1090-1098.

18. Pegram, G. G. S and McKenzie, R.S., (1991). "Synthetic Streamflow generation in the Vaal River System Study". *The Civil Engineer in South Africa*. Vol. 31, No. 1, pp. 15-24.

19. Rozhdestvensky, A.V. and Zvereva, V.M., (1975). "Statistical Modelling of river flow by the Monte Carlo Method and Evaluation of Distribution Parameters and Quantiles", *Proceedings of the Bratislava Symposium*, IAHS-AISH Pub. No. 115, pp. 203-207.

20. Shen, H.W., *Stochastic Approaches to Water Resources Volume I*, H.W. SHEN, Fort Collins, Colorado, 1976, pp.4-1 to 4-41.

21. Sigvaldason, O. T., (1976). "Simulation Model for Operating a Multipurpose Multireservoir System", *Water*

Resources Research, Vol. 12, No. 2, pp. 263-278.

22. Tingsanchali, Tawatchai and Song, Lin Wen, (1980). "A Simulation Model for Serially Linked Reservoir System Ta Chia River Basin", *Proceeding of the International Conference on Water Resources Development*, Taipei, Taiwan, pp. 197-206.

23. Vansteenkiste, G. C., (1976). *System Simulation in Water Resources*, North-Holand Publishing Company, Amsterdam, Oxford, New York, 417 p.

24. Viesmann Jr., Warren, (1990). "Water Management: Challenge and Opportunity", *Journal of Water Resources Planning and Management*, Vol. 116, No. 2, pp. 155-169.

25. Vogel, Richard M. and Stedinger, Jery R., (1988). "The Value of Stochastic Streamflow Model in Overyear Reservoir Design Applications", *Journal of American Geophysical Union*, Vol. 24, No. 9, pp. 1483-1490.

26. Yazicigil, Hasan, Houck, Mark H. and Toebes, Gerrit H., (1983). "Daily Operation of a Multipurpose Reservoir System", *Water Resources Research*, Vol. 19, No. 1, pp. 1-13.

27. Yevjevich, V., (1972). *Stochastic Processes in Hydrology*, V. Yevjevich, Water Resources Publications, Fort Collins, Colorado, U.S.A., 300 p.

28. Yeh, William W., (1985). "Reservoir Management and Operations Models: A State-of-the-art Review", *Water Resources Research*, Vol. 21, No. 12, pp. 1797-1818.