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

CENTRE FOR NEWFOUNDLAND STUDIES

TOTAL OF 10 PAGES ONLY
MAY BE XEROXED

(Without Author’s Permission)

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
The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

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

- **Abstract** iii
- **Acknowledgements** v
- **Contents** vi
- **List of Tables** x
- **List of Figures** xii
- **List of Symbols** xv

### 1 Introduction

1.1 General Description 1
1.2 Objective of the Study 6
1.3 Method of Research 7
1.4 Thesis Outline 8

### 2 Description of Study Area

2.1 Introduction 10
2.2 Reservoir and Hydropower Plant Characteristics 10
   2.2.1 System Configuration 10
   2.2.2 Reservoir Characteristics 13
   2.2.3 Hydropower Plant Characteristics 14
2.3 Climate and Hydrology 15
2.4 Current Reservoir Operations 17

vi
<table>
<thead>
<tr>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 Stochastic Modelling of Flows</td>
</tr>
<tr>
<td>3.1 Introduction</td>
</tr>
<tr>
<td>3.2 Statistical Analysis of Historical Flows</td>
</tr>
<tr>
<td>3.2.1 Time Series and Moving Average Plots</td>
</tr>
<tr>
<td>3.2.2 Statistical Characteristics of Historical Annual and Monthly Flows</td>
</tr>
<tr>
<td>3.3 Stochastic Modelling of Annual Flows</td>
</tr>
<tr>
<td>3.3.1 Preliminary Analysis and Model Identification</td>
</tr>
<tr>
<td>3.3.2 Estimation of Parameters</td>
</tr>
<tr>
<td>3.3.3 Model Tests</td>
</tr>
<tr>
<td>3.4 Stochastic Modelling of Monthly Flows</td>
</tr>
<tr>
<td>3.4.1 Method of Fragments</td>
</tr>
<tr>
<td>3.4.2 Two-Tier Model</td>
</tr>
<tr>
<td>3.4.3 Analysis of Results</td>
</tr>
<tr>
<td>3.5 Final Stochastic Models</td>
</tr>
<tr>
<td>4 Reservoir Operations</td>
</tr>
<tr>
<td>4.1 Introduction</td>
</tr>
<tr>
<td>4.2 Operating Policy</td>
</tr>
<tr>
<td>4.3 Simulation</td>
</tr>
<tr>
<td>4.4 Citarum Simulation Model (CTR-SIMOD)</td>
</tr>
<tr>
<td>5 Results and Discussion</td>
</tr>
<tr>
<td>5.1 Introduction</td>
</tr>
<tr>
<td>5.2 Stochastic Flow Modelling</td>
</tr>
</tbody>
</table>
### List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td>Energy production of the Java and Bali System</td>
<td>5</td>
</tr>
<tr>
<td>1-2</td>
<td>Number of units of generator in the Java and Bali Power System</td>
<td>5</td>
</tr>
<tr>
<td>1-3</td>
<td>Installed capacity of generating plants in the Java and Bali Power System</td>
<td>5</td>
</tr>
<tr>
<td>2-1</td>
<td>Water level and the storage capacities of the Saguling, Cirata and Jatiluhur Reservoirs</td>
<td>13</td>
</tr>
<tr>
<td>2-2</td>
<td>Constants $a$, and $b_j$, in Equation 2-1, and $c_j$, $d$, and $e$, in Equation 2-2 for the Saguling, Cirata and Jatiluhur Reservoirs (NEWJEC 1988)</td>
<td>14</td>
</tr>
<tr>
<td>2-3</td>
<td>Characteristics of the Saguling, Cirata and Jatiluhur Hydropower Plants (NEWJEC 1988)</td>
<td>15</td>
</tr>
<tr>
<td>2-4</td>
<td>Evaporation coefficients for the Saguling, Cirata and Jatiluhur Reservoirs</td>
<td>17</td>
</tr>
<tr>
<td>2-5</td>
<td>Five classifications of historical annual flows of the Citarum River at Nanjung</td>
<td>19</td>
</tr>
<tr>
<td>3-1</td>
<td>Summary of Barlett's and Anderson's tests for the historical annual flows</td>
<td>27</td>
</tr>
<tr>
<td>3-2</td>
<td>Summary of the stochastic annual flow models and the respective values of $r$ of residuals, AIC(p) and MS</td>
<td>30</td>
</tr>
<tr>
<td>3-3</td>
<td>Summary of the results of simulation for ARMA(2,0) stochastic annual flow model</td>
<td>31</td>
</tr>
</tbody>
</table>
3-4 Summary of the transformation for the historical monthly flows of the Citarum River at Nanjung ... 36
3-5 Means, standard deviations and coefficients of correlation of transformed flows ... ... ... 36
3-6 Comparisons of the Method of Fragments and the Two-Tier Method for generating synthetic monthly flows ... 38
4-1 Estimated parameters produced by simulation using the CTR-SIMOD ... ... ... ... ... ... ... ... ... 57
5-1 Summary of results of the annual parameters and the related annual energy generation ... ... ... 60
5-2 Load factors of the Saguling, Cirata and Jatiluhur Plants ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... 71
5-3 Comparisons of energy estimates using historical flows ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... 77
5-4 Comparisons of energy estimates using synthetic flows ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... 78
List of Figures

1-1 Map of Indonesia ................................................. 2
1-2 Location of Study Area ........................................ 3
1-3 Method of research ............................................. 9
2-1 Citarum Multi-Reservoir System ................................. 11
2-2 Profile of the Citarum River from the Saguling Reservoir to the Jatiluhur Reservoir .......................... 12
2-3 Drainage area of the Citarum River .............................. 16
2-4 CROG and WRDG for the Saguling, Cirata and Jatiluhur Hydropower Plants ............................................. 20
3-1 Scatter and a 9-point moving average diagram of the historical annual flows of the Citarum River at Nanjung ......................................................... 22
3-2 Auto-correlation function of the historical annual flows of the Citarum River at Nanjung .......................... 23
3-3 Partial auto-correlation function of the historical annual flows of the Citarum River at Nanjung ................... 23
3-4 Historical monthly flows of the Citarum River at Nanjung for the periods of 1928-1987 .......................... 24
3-5 Average of the 60 years of the historical monthly flows of the Citarum River at Nanjung .......................... 24
3-6 Barlett’s and Anderson’s tests for the ACF of the annual flows of the Citarum River at Nanjung ................... 28
3-7 Barlett’s and Anderson’s tests for the PACF of the annual flows of the Citarum River at Nanjung ...

4-1 Schematic of hydropower dam 

4-2 Illustration of the application of a rule curve 

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 

4-4 Five possible outcomes of calculated water available:
   4-4(a) Calculated water available is negative 
   4-4(b) Calculated water available equals to zero 
   4-4(c) Calculated water available is greater than zero but smaller than minimum turbine flow 
   4-4(d) Calculated water available is greater than minimum turbine flow but smaller than maximum turbine flow 
   4-4(e) Calculated water available is greater than maximum turbine flow and: (1) reservoir storage is smaller than maximum reservoir storage, (2) reservoir storage is greater than maximum reservoir storage 

5-1 Energy generation of the Saguling Hydropower Plant 

5-2 Inflows to the Saguling Reservoir 

5-3 Heads on the Saguling Hydropower Plant 

5-4 Turbine flows for the Saguling Hydropower Plant 

5-5 Spill flows from the Saguling Reservoir 

5-6 Energy generation of the Citarum Hydropower System 

5-7 Energy generation of the Saguling Plant using the CROG rule curve
5-8 Energy generation of the Cirata Plant using the CROG rule curve .................. 74
5-9 Energy generation of the Jatiluhur Plant using the CROG rule curve .............. 75
5-10 Total energy generation of the Citarum Hydropower System using the CROG rule curve .................. 76
## List of Symbols

<table>
<thead>
<tr>
<th>Symbols</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>a, b</td>
<td>- constants for equation 2-1</td>
</tr>
<tr>
<td>Ai(t)</td>
<td>- reservoir surface area at the beginning of month t</td>
</tr>
<tr>
<td>Ai(t+1)</td>
<td>- reservoir surface area at the end of month t</td>
</tr>
<tr>
<td>ACC</td>
<td>- Area Control Centre</td>
</tr>
<tr>
<td>ACF</td>
<td>- auto-correlation function</td>
</tr>
<tr>
<td>AIC</td>
<td>- Akaike Information Criterion</td>
</tr>
<tr>
<td>ARMA</td>
<td>- auto-regressive moving average model</td>
</tr>
<tr>
<td>b0</td>
<td>- least square regression coefficient for estimating the (j+1)th flow from</td>
</tr>
<tr>
<td></td>
<td>the jth flow</td>
</tr>
<tr>
<td>c, d, e</td>
<td>- constants for equation 2-2</td>
</tr>
<tr>
<td>ci(t)</td>
<td>- coefficient of evaporation of month t</td>
</tr>
<tr>
<td>CRCG</td>
<td>- current reservoir operating guideline</td>
</tr>
<tr>
<td>C</td>
<td>- coefficient of skewness</td>
</tr>
<tr>
<td>CTR-SIMOD</td>
<td>- Citarum Simulation Model</td>
</tr>
<tr>
<td>CTR-STOCHMOD</td>
<td>- Citarum Stochastic Model</td>
</tr>
<tr>
<td>D</td>
<td>- dry</td>
</tr>
<tr>
<td>η</td>
<td>- plant efficiency</td>
</tr>
<tr>
<td>e(t)</td>
<td>- evaporation coefficients of month t</td>
</tr>
<tr>
<td>E(t)</td>
<td>- evaporation from reservoir i during month t</td>
</tr>
<tr>
<td>g</td>
<td>- acceleration due to gravity (9.81 m/s^2)</td>
</tr>
<tr>
<td>GWh</td>
<td>- Giga Watt hour</td>
</tr>
<tr>
<td>H(t)</td>
<td>- head at the beginning of month t</td>
</tr>
</tbody>
</table>
FSL  - full supply level
i  - node
I_{i0}  - 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_{e(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_{Th(t)}  - turbine flow of hydropower plant during month t
Q_{ThMax,i0}  - maximum turbine flow
Q_{WA(t)}  - available water of month t
r_{phi}  - serial correlation coefficient of month j
r_{k(t)}  - auto-correlation coefficient of lag-k
s_{(j+1)}  - standard deviation of historical flows of month j+1
VD  - very dry
V_{i0}  - volume of reservoir i at the beginning of month t
\( V_{(t+1)} \) - volume of reservoir \( i \) at the end of month \( t \)
\( V_{(t+1)} \) - 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
\( X_{(j)} \) - transformed monthly flow
\( X \) - historical annual flow
\( Y_{(t)} \) - generated monthly flow
\( Y_{(t)} \) - generated annual flow in year \( t \)
\( z (5\%) \) - 5\% significance level of \( z \)
\( z_{(0)} \) - normal random variate with mean zero and variance unity
\( \bar{X}_{(j)} \) - mean of historical flows of month \( j \)
\( \varepsilon \) - independent stochastic component
\( \lambda \) - transformation parameter
\( \mu \) - mean
\( \rho \) - water density (1000 kg/m\(^3\))
\( \sigma \) - 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),
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)

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

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

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

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

<table>
<thead>
<tr>
<th>Power Plant</th>
<th>Area 1</th>
<th>Area 2</th>
<th>Area 3</th>
<th>Area 4</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hydro</td>
<td>33</td>
<td>1412</td>
<td>276</td>
<td>238</td>
<td>1960</td>
</tr>
<tr>
<td>Thermal (Coal)</td>
<td>1600</td>
<td>-</td>
<td>300</td>
<td>750</td>
<td>1900</td>
</tr>
<tr>
<td>Thermal (Oil)</td>
<td>850</td>
<td>-</td>
<td>-</td>
<td>140</td>
<td></td>
</tr>
<tr>
<td>Geothermal</td>
<td>400</td>
<td>80</td>
<td>75</td>
<td>112</td>
<td>667</td>
</tr>
<tr>
<td>Gas Turbine</td>
<td>10</td>
<td>-</td>
<td>-</td>
<td>76</td>
<td>86</td>
</tr>
<tr>
<td>Diesel</td>
<td>2893</td>
<td>1632</td>
<td>652</td>
<td>1177</td>
<td>6353</td>
</tr>
</tbody>
</table>
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.
Analysis of the System Behaviour

Stochastic Modelling

- Statistical Analysis of Historical Flows
- Stochastic Modelling of Annual Flows
- Stochastic Modelling of Monthly Flows

Series of Synthetic Monthly Flows

Simulation of Reservoir Operations Using:
(1) Current Reservoir Operation Guideline Rule Curve
(2) Water Resources Development Guideline Rule Curve
(3) Empty Reservoir Rule Curve
(4) Full Reservoir Rule Curve

Reliability Curves of Annual Energy Production

Analysis of Results

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
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:

$$A_i = a_i V_i^{b_i}$$

in which:

- \(A_i\) = reservoir surface area of the reservoir \(i\) \([10^6 \text{ m}^2]\),
- \(V_i\) = reservoir storage of the reservoir \(i\) \([10^6 \text{ m}^3]\),
- \(a_i, b_i\) = 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>
<thead>
<tr>
<th>Descriptions</th>
<th>Saguling</th>
<th>Cirata</th>
<th>Jatiluhur</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full supply levels (m)</td>
<td>643</td>
<td>220</td>
<td>107</td>
</tr>
<tr>
<td>Low supply levels (m)</td>
<td>623</td>
<td>205</td>
<td>90</td>
</tr>
<tr>
<td>FSL volumes ((10^6 \text{ m}^3))</td>
<td>881</td>
<td>1973</td>
<td>3000</td>
</tr>
<tr>
<td>LSL volumes ((10^6 \text{ m}^3))</td>
<td>272</td>
<td>1177</td>
<td>1200</td>
</tr>
</tbody>
</table>
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:

\[ H_i = c_i V_i^{d_i} + e_i \]  

(2-2)

in which:
- \( H_i \) = head of the reservoir \( i \) [m],
- \( V_i \) = reservoir storage of the reservoir \( i \) \([10^6 \text{ m}^3]\),
- \( c_i, d_i, e_i \) = 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)

<table>
<thead>
<tr>
<th>Coefficients</th>
<th>Saguling</th>
<th>Cirata</th>
<th>Jatiluhur</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a_i )</td>
<td>289.7000</td>
<td>17.4600</td>
<td>2.1500</td>
</tr>
<tr>
<td>( b_i )</td>
<td>0.0441</td>
<td>0.2500</td>
<td>0.4523</td>
</tr>
<tr>
<td>( c_i )</td>
<td>0.1130</td>
<td>0.3963</td>
<td>0.9605</td>
</tr>
<tr>
<td>( d_i )</td>
<td>0.8939</td>
<td>0.6667</td>
<td>0.5546</td>
</tr>
<tr>
<td>( e_i )</td>
<td>0.2664</td>
<td>0.6340</td>
<td>0.6214</td>
</tr>
</tbody>
</table>
Table 2-3: Characteristics of the Saguling, Cirata and Jatiluhur Hydropower Plants (NEWJEC, 1988)

<table>
<thead>
<tr>
<th>Descriptions</th>
<th>Saguling</th>
<th>Cirata</th>
<th>Jatiluhur</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. plant capacities (MW)</td>
<td>715</td>
<td>518</td>
<td>175</td>
</tr>
<tr>
<td>Plant efficiencies (%)</td>
<td>0.835</td>
<td>0.816</td>
<td>0.870</td>
</tr>
<tr>
<td>Min. heads (m)</td>
<td>371</td>
<td>102</td>
<td>64</td>
</tr>
<tr>
<td>Max. heads (m)</td>
<td>391</td>
<td>117</td>
<td>81</td>
</tr>
<tr>
<td>Min. turbine flow (10^6 m³)</td>
<td>147</td>
<td>315</td>
<td>315</td>
</tr>
<tr>
<td>Max. turbine flow (10^6 m³)</td>
<td>586</td>
<td>1453</td>
<td>665</td>
</tr>
<tr>
<td>Tail water level (m)</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

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 evaporation rates were calculated using the following equation.

\[ E_i = c_{e,i} A_i \]  

(2-3)

in which:

- \( E_i \) = evaporation of the reservoir \( i \) \( [10^6 \text{ m}^3] \),
- \( A_i \) = surface area of the reservoir \( i \) \( [10^6 \text{ m}^2] \),
- \( c_{e,i} \) = evaporation coefficients of the reservoir \( i \) (see Table 2-4).

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

<table>
<thead>
<tr>
<th>Months</th>
<th>( c_{e,i} ) of Saguling (m)</th>
<th>( c_{e,i} ) of Cirata (m)</th>
<th>( c_{e,i} ) of Jatiluhur (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>May</td>
<td>0.12</td>
<td>0.12</td>
<td>0.12</td>
</tr>
<tr>
<td>June</td>
<td>0.12</td>
<td>0.12</td>
<td>0.12</td>
</tr>
<tr>
<td>July</td>
<td>0.12</td>
<td>0.12</td>
<td>0.12</td>
</tr>
<tr>
<td>August</td>
<td>0.12</td>
<td>0.12</td>
<td>0.12</td>
</tr>
<tr>
<td>September</td>
<td>0.12</td>
<td>0.12</td>
<td>0.12</td>
</tr>
<tr>
<td>October</td>
<td>0.09</td>
<td>0.09</td>
<td>0.09</td>
</tr>
<tr>
<td>November</td>
<td>0.06</td>
<td>0.06</td>
<td>0.06</td>
</tr>
<tr>
<td>December</td>
<td>0.06</td>
<td>0.06</td>
<td>0.06</td>
</tr>
<tr>
<td>January</td>
<td>0.06</td>
<td>0.06</td>
<td>0.06</td>
</tr>
<tr>
<td>February</td>
<td>0.06</td>
<td>0.06</td>
<td>0.06</td>
</tr>
<tr>
<td>March</td>
<td>0.06</td>
<td>0.06</td>
<td>0.06</td>
</tr>
<tr>
<td>April</td>
<td>0.09</td>
<td>0.09</td>
<td>0.09</td>
</tr>
</tbody>
</table>

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)

<table>
<thead>
<tr>
<th>No.</th>
<th>Classifications</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Very dry</td>
<td>(0 - 20)%</td>
</tr>
<tr>
<td>2</td>
<td>Dry</td>
<td>(20 - 45)%</td>
</tr>
<tr>
<td>3</td>
<td>Normal</td>
<td>(45 - 55)%</td>
</tr>
<tr>
<td>4</td>
<td>Wet</td>
<td>(55 - 80)%</td>
</tr>
<tr>
<td>5</td>
<td>Very Wet</td>
<td>(80 - 100)%</td>
</tr>
</tbody>
</table>
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 \( r_k(x_k) \) 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 \( r_m \) 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>
<thead>
<tr>
<th>No</th>
<th>Auto-correlation of:</th>
<th>r</th>
<th>Barlett</th>
<th>Anderson</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>lag-1 (r1)</td>
<td>0.288</td>
<td>0.250</td>
<td>0.240</td>
</tr>
<tr>
<td>2</td>
<td>lag-2 (r2)</td>
<td>0.406</td>
<td>0.250</td>
<td>0.242</td>
</tr>
<tr>
<td>3</td>
<td>lag-3 (r3)</td>
<td>0.229</td>
<td>0.250</td>
<td>0.244</td>
</tr>
<tr>
<td>4</td>
<td>lag-4 (r4)</td>
<td>0.149</td>
<td>0.250</td>
<td>0.246</td>
</tr>
</tbody>
</table>

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.
Barlett's and Anderson's tests for the ACF of the annual flows of the Citarum River at Nanjung

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:

\[ Y_{a(t)} = a_1 Y_{a(t-1)} + a_2 Y_{a(t-2)} + \epsilon_t \]  \hspace{1cm} (3-1)

where:

\[ Y_{a(t)}, Y_{a(t-1)}, Y_{a(t-2)} = \text{generated annual flows in year } t, t-1 \text{ and } t-2 \text{ respectively,} \]

\[ a_1 = 0.1758, \]

\[ a_2 = 0.3737, \]

\[ \epsilon_t \sim N (900.37, 520.422) \]
Table 3-2: Summary of the stochastic annual flow models and the respective values of r of residuals, AIC(p) and MS

<table>
<thead>
<tr>
<th>No</th>
<th>Stochastic Models</th>
<th>AIC(p)</th>
<th>r of residuals</th>
<th>MS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ARMA(2,0)</td>
<td>754.55</td>
<td>0.972</td>
<td>270839</td>
</tr>
<tr>
<td>2</td>
<td>ARMA(1,1)</td>
<td>757.72</td>
<td>0.969</td>
<td>285499</td>
</tr>
<tr>
<td>3</td>
<td>ARMA(1,2)</td>
<td>758.01</td>
<td>0.970</td>
<td>277494</td>
</tr>
<tr>
<td>4</td>
<td>ARMA(2,2)</td>
<td>760.57</td>
<td>0.971</td>
<td>280078</td>
</tr>
<tr>
<td>5</td>
<td>ARMA(2,1)</td>
<td>757.54</td>
<td>0.970</td>
<td>275303</td>
</tr>
</tbody>
</table>

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 autocorrelations, 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.

\[ Y_{a(t)} = 0.1758 Y_{a(t-1)} + 0.3737 Y_{a(t-2)} + 943.87 + 478.056 z_t \quad (3-2) \]

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)

<table>
<thead>
<tr>
<th>No</th>
<th>Moments</th>
<th>Historical flows ($10^6$ m$^3$)</th>
<th>Generated flows ($10^6$ m$^3$)</th>
<th>Lower Bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\mu$</td>
<td>1990.775</td>
<td>2013.878</td>
<td>1705.388</td>
<td>2322.368</td>
</tr>
<tr>
<td></td>
<td>$\sigma$</td>
<td>573.597</td>
<td>556.153</td>
<td>437.097</td>
<td>675.2087</td>
</tr>
<tr>
<td>3</td>
<td>$C_a$</td>
<td>0.169</td>
<td>-0.079</td>
<td>-0.746</td>
<td>0.588</td>
</tr>
<tr>
<td></td>
<td>$r_1$</td>
<td>0.289</td>
<td>0.195</td>
<td>-0.166</td>
<td>0.556</td>
</tr>
<tr>
<td></td>
<td>$r_2$</td>
<td>0.406</td>
<td>0.405</td>
<td>0.162</td>
<td>0.647</td>
</tr>
<tr>
<td>6</td>
<td>$K$</td>
<td>0.868</td>
<td>0.754</td>
<td>0.620</td>
<td>0.887</td>
</tr>
<tr>
<td>7</td>
<td>Minimum</td>
<td>886.700</td>
<td>685.222</td>
<td>158.778</td>
<td>1211.665</td>
</tr>
<tr>
<td>8</td>
<td>Maximum</td>
<td>3525.100</td>
<td>3280.819</td>
<td>2764.829</td>
<td>3796.809</td>
</tr>
</tbody>
</table>

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 (McMahon 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 (McMahon 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 (McMahon 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:

\[ Y_{tm(i+1)} = \bar{x}_{tm(j+1)} + b_j (Y_{tm(i)} - \bar{x}_{tm(j)}) + z_i \left( \sigma_{xtm(j+1)} \sqrt{1 - \gamma^2_{xtm}} \right) \]  

(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, \(1 \leq j \leq 12\),
- \( b_j \) = least squares regression coefficient for estimating the \((j+1)^{th}\) flow from the \(j^{th}\) flow,

\[ b_{j} = \frac{\sigma_{xtm(j+1)}}{\sigma_{xtm(j)}} \]
random normal variate with mean zero and variance unity,

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

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

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).

\[ x_{\text{trans}}(j) = \begin{cases} \frac{\left(x_m(j)^\lambda - 1\right)}{\lambda} & \lambda \neq 0 \\ \log x_m(j) & \lambda = 0 \end{cases} \quad (3-4) \]

where:

\[ x_m(j) = \text{historical monthly flows,} \]

\[ \lambda = \text{transformation parameter,} \]

\[ x_{\text{trans}}(j) = \text{transformed monthly flows.} \]
The transformation parameter $\lambda$ 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

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

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

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

<table>
<thead>
<tr>
<th>No</th>
<th>Descriptions</th>
<th>Method of Fragments</th>
<th>Two-Tier Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>May</td>
<td>r1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>June</td>
<td>r1, r2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>July</td>
<td>r1, r2</td>
<td>σ, Max</td>
</tr>
<tr>
<td>4</td>
<td>August</td>
<td>σ, C, r1, r2, Max</td>
<td>σ, C, Min, Max</td>
</tr>
<tr>
<td>5</td>
<td>September</td>
<td>r1, r2</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>October</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>November</td>
<td>r1</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>December</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>January</td>
<td>σ, r1</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>February</td>
<td>r1, r2</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>March</td>
<td>r1</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>April</td>
<td>r1</td>
<td>r1, r2</td>
</tr>
</tbody>
</table>
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 to 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 incorporated,
(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.
Note:
I: inflows
V: reservoir storage
Q_{SP}: spill flow
Q_{TR}: turbine flow
subscript: node

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.
at the beginning of month $t$, $= a_i V_{t(1)}^{b_i}$,

$$A_{T(t+1)} = \text{water reservoir surface area target for reservoir } i$$

$$a_i = \text{reservoir surface coefficient for reservoir } i,$$

$$b_i = \text{reservoir surface exponent for reservoir } i,$$

$$I_i = \text{inflows during month } t \text{ for reservoir } i.$$
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 \( t \).

\[
Q_{WAI(t)} = V_{i(t)} - V_{Til(t+1)} - E_{i(t)} (V_{i(t)}, V_{Til(t+1)}) + I_{i(t)} \tag{4.1}
\]

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 \( t \),
\( V_{Til(t+1)} \) = reservoir storage target for reservoir \( i \) at the end of month \( t \),
\( E_{i(t)} \) = evaporation from reservoir \( i \) during month \( t \),
\( E_{i(t)} = c_{i(t)} (A_{i(t)} + A_{Til(t+1)}) \times 0.5 \)
\( c_{i(t)} \) = 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 \( t \) \( (V_{t+1}) \) was set at the reservoir storage target at the end of month \( t \) \( (V_{t+1}) \). The turbine flow and spill flow were again set equal to zero.

**Case three (Figure 4-4 (c)):** if the available water in month \( t \) \( (Q_{wa+t}) \) was greater than zero but smaller than the minimum turbine flow, the reservoir storage at the end of month \( t \) \( (V_{t+1}) \) was set at the reservoir storage target at the end of month \( t \) \( (V_{t+1}) \), plus the available water. The turbine flow was again equal to zero. If the calculated reservoir storage at the end of month \( t \) \( (V_{t+1}) \) 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 \( t \) \( (V_{t+1}) \) 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 \( t \) \( (V_{t+1}) \) was set equal to the reservoir storage target at the end of month \( t \) \( (V_{t+1}) \). The maximum turbine flow was calculated for the average head during the month.
in which:

\[ Q_{\text{TM},i(t)} = \frac{P_{\text{Max},i}}{\eta_i \rho g H_i(t)} \]  

(4-2)

\[ Q_{\text{TM},i(t)} = \text{maximum turbine flow of the reservoir } i, \text{ for month } t \left[10^6 \text{ m}^3\right], \]

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

\[ \eta_i = \text{plant efficiency of hydropower plant } i, \]

\[ \rho = \text{water density} = 1000 \text{ kg/m}^3, \]

\[ g = \text{acceleration due to gravity, (9.81 m/s}^2), \]

\[ \bar{H}_i(t) = \text{average head reservoir } i \text{ for month } t \]

Case five (Figure 4-4 (e)): if the calculated water available in month \( t \) \( (Q_{\text{wai},(t)}) \) 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 \( t \) \( (V_{\text{m+1}}) \) was equal to the reservoir storage target at the end of month \( t \) \( (V_{\text{m+1}}) \) 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):
where:

\[ Q_{Sp,i(t)} = V_{i(t+1)} - V_{Max,i} \]  \hspace{1cm} (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.

\[ P_{i(t)} = \eta_i \rho g Q_{Tul,i(t)} \bar{H}_{i(t)} \]  \hspace{1cm} (4-4)

in which:

- \( P_{i(t)} \) = electric power generated by hydropower plant \( i \) during month \( t \), [Gwh],
- \( \bar{H}_{i(t)} \) = 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,
$Q_{WA,0} < 0$

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

$Q_{WA,0} = 0$

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.
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

<table>
<thead>
<tr>
<th>No</th>
<th>Description</th>
<th>Saguling</th>
<th>Cirata</th>
<th>Jatiluhur</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Inflows</td>
<td>$\mu_1(0)$</td>
<td>$\mu_2(0)$</td>
<td>$\mu_3(0)$</td>
</tr>
<tr>
<td>2</td>
<td>Reservoir storage</td>
<td>$\mu_V(0)$</td>
<td>$\mu_V(0)$</td>
<td>$\mu_V(0)$</td>
</tr>
<tr>
<td>3</td>
<td>Turbine flows</td>
<td>$\mu_QT(0)$</td>
<td>$\mu_QT(0)$</td>
<td>$\mu_QT(0)$</td>
</tr>
<tr>
<td>4</td>
<td>Spill flows</td>
<td>$\mu_QS(0)$</td>
<td>$\mu_QS(0)$</td>
<td>$\mu_QS(0)$</td>
</tr>
<tr>
<td>5</td>
<td>Energy generations</td>
<td>$\mu_E(0)$</td>
<td>$\mu_E(0)$</td>
<td>$\mu_E(0)$</td>
</tr>
</tbody>
</table>
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)

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

energy
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 highest 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 total outflow from the Saguling
reservoir \((29.0 \times 10^6 \text{ m}^3\) of spill flow plus \(239.2 \times 10^6 \text{ m}^3\) of turbine flow). The highest spill flow for the WRDG rule curve occurred in April \((16.2 \times 10^6 \text{ m}^3)\). This was 4.8% of the total outflow from the Saguling node \((16.2 \times 10^6 \text{ m}^3\) of spill flow plus \(321.8 \times 10^6 \text{ m}^3\) 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 the 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 load factor for the Saguling Plant would be 2183 divide by (715 x 365.25 x 24
I, 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

<table>
<thead>
<tr>
<th>No</th>
<th>Rule Curves</th>
<th>Saguling Plant (%)</th>
<th>Cirata Plant (%)</th>
<th>Jatiluhur Plant (%)</th>
<th>Citarum Hydropower System (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>WRDG</td>
<td>36</td>
<td>21</td>
<td>50</td>
<td>32</td>
</tr>
<tr>
<td>2</td>
<td>CROG</td>
<td>35</td>
<td>20</td>
<td>49</td>
<td>31</td>
</tr>
<tr>
<td>3</td>
<td>ER</td>
<td>36</td>
<td>20</td>
<td>42</td>
<td>31</td>
</tr>
<tr>
<td>4</td>
<td>FR</td>
<td>31</td>
<td>17</td>
<td>40</td>
<td>27</td>
</tr>
</tbody>
</table>

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.

- Upper 95% limit
- Lower 5% limit
- 6000 points of generated data
- 60 points of historical data

Proportion non-exceedance vs. Thousands of Energy (GWh)
Figure 5-8: Energy generation of the Cirata plant using the CROG rule curve.
<table>
<thead>
<tr>
<th>Prob. non-exceedence</th>
<th>Confidence level</th>
<th>Saguling Plant</th>
<th>Cirata Plant</th>
<th>Jatiluhur Plant</th>
<th>Citarum Plant System</th>
</tr>
</thead>
<tbody>
<tr>
<td>20%</td>
<td>5%</td>
<td>2400</td>
<td>1050</td>
<td>850</td>
<td>4300</td>
</tr>
<tr>
<td></td>
<td>50%</td>
<td>2750</td>
<td>1200</td>
<td>950</td>
<td>4900</td>
</tr>
<tr>
<td></td>
<td>95%</td>
<td>3000</td>
<td>1350</td>
<td>1100</td>
<td>5450</td>
</tr>
<tr>
<td>50%</td>
<td>5%</td>
<td>1900</td>
<td>750</td>
<td>650</td>
<td>3300</td>
</tr>
<tr>
<td></td>
<td>50%</td>
<td>2200</td>
<td>900</td>
<td>750</td>
<td>3850</td>
</tr>
<tr>
<td></td>
<td>95%</td>
<td>2500</td>
<td>1150</td>
<td>900</td>
<td>4550</td>
</tr>
<tr>
<td>80%</td>
<td>5%</td>
<td>1200</td>
<td>450</td>
<td>400</td>
<td>2050</td>
</tr>
<tr>
<td></td>
<td>50%</td>
<td>1600</td>
<td>650</td>
<td>600</td>
<td>2850</td>
</tr>
<tr>
<td></td>
<td>95%</td>
<td>2000</td>
<td>850</td>
<td>700</td>
<td>3550</td>
</tr>
</tbody>
</table>
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)

<table>
<thead>
<tr>
<th>Prob. non-exceedence</th>
<th>Saguling Plant</th>
<th>Cirata Plant</th>
<th>Jatiluhur Plant</th>
<th>Citarum Plant System</th>
</tr>
</thead>
<tbody>
<tr>
<td>20%</td>
<td>2750</td>
<td>1250</td>
<td>1000</td>
<td>5000</td>
</tr>
<tr>
<td>50%</td>
<td>2100</td>
<td>900</td>
<td>700</td>
<td>3700</td>
</tr>
<tr>
<td>80%</td>
<td>1500</td>
<td>750</td>
<td>600</td>
<td>2850</td>
</tr>
</tbody>
</table>
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


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>
<thead>
<tr>
<th>Year</th>
<th>Jan</th>
<th>Feb</th>
<th>Mar</th>
<th>Apr</th>
<th>May</th>
<th>Jun</th>
<th>Jul</th>
<th>Aug</th>
<th>Sep</th>
<th>Oct</th>
<th>Nov</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>1928</td>
<td>125</td>
<td>85</td>
<td>54</td>
<td>37</td>
<td>38</td>
<td>52</td>
<td>32</td>
<td>77</td>
<td>277</td>
<td>266</td>
<td>396</td>
<td>361</td>
</tr>
<tr>
<td>1929</td>
<td>155</td>
<td>123</td>
<td>46</td>
<td>69</td>
<td>153</td>
<td>126</td>
<td>152</td>
<td>280</td>
<td>285</td>
<td>152</td>
<td>152</td>
<td>319</td>
</tr>
<tr>
<td>1930</td>
<td>152</td>
<td>142</td>
<td>254</td>
<td>437</td>
<td>507</td>
<td>536</td>
<td>536</td>
<td>536</td>
<td>536</td>
<td>536</td>
<td>536</td>
<td>536</td>
</tr>
<tr>
<td>1931</td>
<td>193</td>
<td>184</td>
<td>241</td>
<td>367</td>
<td>325</td>
<td>354</td>
<td>354</td>
<td>354</td>
<td>354</td>
<td>354</td>
<td>354</td>
<td>354</td>
</tr>
<tr>
<td>1932</td>
<td>152</td>
<td>193</td>
<td>171</td>
<td>181</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
</tr>
<tr>
<td>1933</td>
<td>152</td>
<td>193</td>
<td>171</td>
<td>181</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
</tr>
<tr>
<td>1934</td>
<td>152</td>
<td>193</td>
<td>171</td>
<td>181</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
</tr>
<tr>
<td>1935</td>
<td>152</td>
<td>193</td>
<td>171</td>
<td>181</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
</tr>
<tr>
<td>1936</td>
<td>152</td>
<td>193</td>
<td>171</td>
<td>181</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
</tr>
<tr>
<td>1937</td>
<td>152</td>
<td>193</td>
<td>171</td>
<td>181</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
</tr>
<tr>
<td>1938</td>
<td>152</td>
<td>193</td>
<td>171</td>
<td>181</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
</tr>
<tr>
<td>1939</td>
<td>152</td>
<td>193</td>
<td>171</td>
<td>181</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
</tr>
<tr>
<td>1940</td>
<td>152</td>
<td>193</td>
<td>171</td>
<td>181</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
</tr>
<tr>
<td>1941</td>
<td>152</td>
<td>193</td>
<td>171</td>
<td>181</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
</tr>
<tr>
<td>1942</td>
<td>152</td>
<td>193</td>
<td>171</td>
<td>181</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
</tr>
<tr>
<td>1943</td>
<td>152</td>
<td>193</td>
<td>171</td>
<td>181</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
</tr>
<tr>
<td>1944</td>
<td>152</td>
<td>193</td>
<td>171</td>
<td>181</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
</tr>
<tr>
<td>1945</td>
<td>152</td>
<td>193</td>
<td>171</td>
<td>181</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
</tr>
<tr>
<td>1946</td>
<td>152</td>
<td>193</td>
<td>171</td>
<td>181</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
</tr>
<tr>
<td>1947</td>
<td>152</td>
<td>193</td>
<td>171</td>
<td>181</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
</tr>
<tr>
<td>1948</td>
<td>152</td>
<td>193</td>
<td>171</td>
<td>181</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
</tr>
<tr>
<td>1949</td>
<td>152</td>
<td>193</td>
<td>171</td>
<td>181</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
</tr>
<tr>
<td>1950</td>
<td>152</td>
<td>193</td>
<td>171</td>
<td>181</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
</tr>
<tr>
<td>1951</td>
<td>152</td>
<td>193</td>
<td>171</td>
<td>181</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
</tr>
<tr>
<td>1952</td>
<td>152</td>
<td>193</td>
<td>171</td>
<td>181</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
</tr>
<tr>
<td>1953</td>
<td>152</td>
<td>193</td>
<td>171</td>
<td>181</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
</tr>
<tr>
<td>1954</td>
<td>152</td>
<td>193</td>
<td>171</td>
<td>181</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
</tr>
<tr>
<td>1955</td>
<td>152</td>
<td>193</td>
<td>171</td>
<td>181</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
</tr>
<tr>
<td>1956</td>
<td>152</td>
<td>193</td>
<td>171</td>
<td>181</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
</tr>
<tr>
<td>1957</td>
<td>152</td>
<td>193</td>
<td>171</td>
<td>181</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
<td>191</td>
</tr>
</tbody>
</table>

(All flows in million cubic metres)
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>
<thead>
<tr>
<th></th>
<th>May</th>
<th>Jun</th>
<th>Jul</th>
<th>Aug</th>
<th>Sep</th>
<th>Oct</th>
<th>Nov</th>
<th>Dec</th>
<th>Jan</th>
<th>Feb</th>
<th>Mar</th>
<th>Apr</th>
<th>Annual Flows</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Minimum</strong></td>
<td>42.0</td>
<td>13.9</td>
<td>1.1</td>
<td>0.0</td>
<td>0.3</td>
<td>0.8</td>
<td>15.8</td>
<td>47.0</td>
<td>95.1</td>
<td>70.4</td>
<td>79.1</td>
<td>65.1</td>
<td>866.7</td>
</tr>
<tr>
<td><strong>Maximum</strong></td>
<td>507.2</td>
<td>361.2</td>
<td>303.6</td>
<td>348.0</td>
<td>246.1</td>
<td>331.2</td>
<td>417.4</td>
<td>441.3</td>
<td>506.7</td>
<td>467.0</td>
<td>650.9</td>
<td>488.0</td>
<td>3525.1</td>
</tr>
<tr>
<td><strong>Mean</strong></td>
<td>193.6</td>
<td>110.6</td>
<td>67.2</td>
<td>48.4</td>
<td>46.7</td>
<td>79.7</td>
<td>163.4</td>
<td>235.3</td>
<td>239.7</td>
<td>246.7</td>
<td>280.5</td>
<td>279.0</td>
<td>1990.8</td>
</tr>
<tr>
<td><strong>Standard deviation</strong></td>
<td>105.1</td>
<td>68.6</td>
<td>59.5</td>
<td>53.1</td>
<td>48.5</td>
<td>67.8</td>
<td>98.6</td>
<td>99.7</td>
<td>85.5</td>
<td>93.7</td>
<td>126.7</td>
<td>97.2</td>
<td>568.8</td>
</tr>
<tr>
<td><strong>Cs</strong></td>
<td>0.90</td>
<td>1.15</td>
<td>1.93</td>
<td>3.25</td>
<td>1.78</td>
<td>1.49</td>
<td>0.53</td>
<td>0.11</td>
<td>0.86</td>
<td>0.34</td>
<td>0.62</td>
<td>-0.13</td>
<td>0.16</td>
</tr>
<tr>
<td><strong>rk1</strong></td>
<td>0.16</td>
<td>0.18</td>
<td>0.07</td>
<td>-0.03</td>
<td>0.22</td>
<td>0.32</td>
<td>0.18</td>
<td>0.19</td>
<td>0.16</td>
<td>-0.09</td>
<td>0.05</td>
<td>0.11</td>
<td>0.29</td>
</tr>
<tr>
<td><strong>rk2</strong></td>
<td>0.12</td>
<td>0.08</td>
<td>-0.06</td>
<td>-0.12</td>
<td>0.12</td>
<td>-0.04</td>
<td>0.18</td>
<td>0.25</td>
<td>0.12</td>
<td>0.08</td>
<td>0.11</td>
<td>0.26</td>
<td>0.41</td>
</tr>
<tr>
<td><strong>r</strong></td>
<td>0.54</td>
<td>0.70</td>
<td>0.87</td>
<td>0.55</td>
<td>0.56</td>
<td>0.64</td>
<td>0.47</td>
<td>0.16</td>
<td>0.39</td>
<td>0.32</td>
<td>0.45</td>
<td>0.12</td>
<td></td>
</tr>
<tr>
<td><strong>K</strong></td>
<td>0.54</td>
<td>0.70</td>
<td>0.87</td>
<td>0.55</td>
<td>0.56</td>
<td>0.64</td>
<td>0.47</td>
<td>0.16</td>
<td>0.39</td>
<td>0.32</td>
<td>0.45</td>
<td>0.12</td>
<td>0.86</td>
</tr>
</tbody>
</table>

(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)

<table>
<thead>
<tr>
<th>Year</th>
<th>$x$</th>
<th>$x_1$</th>
<th>$(x_1 - x)^\lambda$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1927</td>
<td>126.0</td>
<td>8.35</td>
<td>-1.1</td>
</tr>
<tr>
<td>1928</td>
<td>85.6</td>
<td>7.83</td>
<td>-1.2</td>
</tr>
<tr>
<td>1929</td>
<td>507.2</td>
<td>14.08</td>
<td>66.0</td>
</tr>
<tr>
<td>1930</td>
<td>285.5</td>
<td>11.77</td>
<td>5.6</td>
</tr>
<tr>
<td>1931</td>
<td>322.4</td>
<td>12.01</td>
<td>27.3</td>
</tr>
<tr>
<td>1932</td>
<td>207.2</td>
<td>10.61</td>
<td>0.2</td>
</tr>
<tr>
<td>1933</td>
<td>42.0</td>
<td>5.98</td>
<td>-55.2</td>
</tr>
<tr>
<td>1934</td>
<td>79.3</td>
<td>7.62</td>
<td>-135</td>
</tr>
<tr>
<td>1935</td>
<td>195.2</td>
<td>10.41</td>
<td>0.1</td>
</tr>
<tr>
<td>1936</td>
<td>319.9</td>
<td>12.21</td>
<td>10.7</td>
</tr>
<tr>
<td>1937</td>
<td>274.6</td>
<td>11.60</td>
<td>4.1</td>
</tr>
<tr>
<td>1938</td>
<td>109.0</td>
<td>5.54</td>
<td>-3.1</td>
</tr>
<tr>
<td>1939</td>
<td>193.6</td>
<td>13.05</td>
<td>28.2</td>
</tr>
<tr>
<td>1940</td>
<td>327.5</td>
<td>12.30</td>
<td>12.1</td>
</tr>
<tr>
<td>1941</td>
<td>193.3</td>
<td>10.37</td>
<td>0.1</td>
</tr>
<tr>
<td>1942</td>
<td>206.8</td>
<td>10.64</td>
<td>0.3</td>
</tr>
<tr>
<td>1943</td>
<td>81.9</td>
<td>7.71</td>
<td>-12.0</td>
</tr>
<tr>
<td>1944</td>
<td>108.7</td>
<td>6.53</td>
<td>-3.2</td>
</tr>
<tr>
<td>1945</td>
<td>77.0</td>
<td>7.53</td>
<td>-15.0</td>
</tr>
<tr>
<td>1946</td>
<td>85.1</td>
<td>7.92</td>
<td>-10.4</td>
</tr>
<tr>
<td>1947</td>
<td>49.4</td>
<td>6.37</td>
<td>-47.9</td>
</tr>
<tr>
<td>1948</td>
<td>181.8</td>
<td>10.18</td>
<td>0.0</td>
</tr>
<tr>
<td>1949</td>
<td>197.5</td>
<td>10.45</td>
<td>0.1</td>
</tr>
<tr>
<td>1950</td>
<td>106.1</td>
<td>8.46</td>
<td>-3.7</td>
</tr>
<tr>
<td>1951</td>
<td>100.1</td>
<td>8.28</td>
<td>-3.1</td>
</tr>
<tr>
<td>1952</td>
<td>265.5</td>
<td>11.50</td>
<td>3.4</td>
</tr>
<tr>
<td>1953</td>
<td>135.3</td>
<td>9.20</td>
<td>-0.5</td>
</tr>
<tr>
<td>1954</td>
<td>195.9</td>
<td>10.42</td>
<td>0.1</td>
</tr>
<tr>
<td>1955</td>
<td>130.3</td>
<td>9.08</td>
<td>-0.8</td>
</tr>
<tr>
<td>1956</td>
<td>123.8</td>
<td>9.07</td>
<td>-0.8</td>
</tr>
<tr>
<td>1957</td>
<td>225.6</td>
<td>10.91</td>
<td>0.8</td>
</tr>
<tr>
<td>1958</td>
<td>183.6</td>
<td>10.20</td>
<td>0.0</td>
</tr>
<tr>
<td>1959</td>
<td>206.8</td>
<td>10.64</td>
<td>0.3</td>
</tr>
<tr>
<td>1960</td>
<td>121.2</td>
<td>10.15</td>
<td>0.0</td>
</tr>
<tr>
<td>1961</td>
<td>158.0</td>
<td>9.66</td>
<td>0.0</td>
</tr>
<tr>
<td>1962</td>
<td>65.7</td>
<td>7.10</td>
<td>-24.3</td>
</tr>
<tr>
<td>1963</td>
<td>138.9</td>
<td>9.28</td>
<td>-0.4</td>
</tr>
<tr>
<td>1964</td>
<td>44.9</td>
<td>6.14</td>
<td>-57.7</td>
</tr>
<tr>
<td>1965</td>
<td>151.3</td>
<td>9.56</td>
<td>-0.1</td>
</tr>
<tr>
<td>1966</td>
<td>123.7</td>
<td>8.92</td>
<td>-1.3</td>
</tr>
<tr>
<td>1967</td>
<td>367.9</td>
<td>12.97</td>
<td>26.1</td>
</tr>
<tr>
<td>1968</td>
<td>137.8</td>
<td>9.25</td>
<td>-1.4</td>
</tr>
<tr>
<td>1969</td>
<td>379.5</td>
<td>12.58</td>
<td>23.8</td>
</tr>
<tr>
<td>1970</td>
<td>192.8</td>
<td>10.36</td>
<td>0.0</td>
</tr>
<tr>
<td>1971</td>
<td>192.3</td>
<td>10.36</td>
<td>0.0</td>
</tr>
<tr>
<td>1972</td>
<td>448.9</td>
<td>13.57</td>
<td>45.3</td>
</tr>
<tr>
<td>1973</td>
<td>267.9</td>
<td>11.80</td>
<td>5.9</td>
</tr>
<tr>
<td>1974</td>
<td>218.3</td>
<td>10.80</td>
<td>0.5</td>
</tr>
<tr>
<td>1975</td>
<td>122.1</td>
<td>8.88</td>
<td>-1.4</td>
</tr>
<tr>
<td>1976</td>
<td>213.5</td>
<td>10.72</td>
<td>0.4</td>
</tr>
<tr>
<td>1977</td>
<td>203.0</td>
<td>10.63</td>
<td>0.2</td>
</tr>
<tr>
<td>1978</td>
<td>321.2</td>
<td>12.22</td>
<td>11.0</td>
</tr>
<tr>
<td>1979</td>
<td>131.1</td>
<td>9.10</td>
<td>-0.7</td>
</tr>
<tr>
<td>1980</td>
<td>194.6</td>
<td>10.40</td>
<td>0.1</td>
</tr>
<tr>
<td>1981</td>
<td>104.0</td>
<td>8.40</td>
<td>-4.1</td>
</tr>
<tr>
<td>1982</td>
<td>279.4</td>
<td>11.68</td>
<td>4.7</td>
</tr>
<tr>
<td>1983</td>
<td>303.1</td>
<td>12.00</td>
<td>8.0</td>
</tr>
<tr>
<td>1984</td>
<td>102.2</td>
<td>8.34</td>
<td>-4.5</td>
</tr>
<tr>
<td>1985</td>
<td>172.0</td>
<td>9.98</td>
<td>-0.0</td>
</tr>
<tr>
<td>1986</td>
<td>152.1</td>
<td>9.57</td>
<td>-0.1</td>
</tr>
</tbody>
</table>

$\lambda = 0.234$

$c = -0.00054$
Table C-2: Probability Plot Correlation Coefficient test of monthly flows of Citarum River at Nanjung
(Example: monthly flows in May)

<table>
<thead>
<tr>
<th>Rank</th>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$P_i$</th>
<th>$z_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>42.9</td>
<td>2.4</td>
<td>0.01</td>
<td>-2.92</td>
</tr>
<tr>
<td>2</td>
<td>44.9</td>
<td>2.4</td>
<td>0.03</td>
<td>-2.94</td>
</tr>
<tr>
<td>3</td>
<td>49.4</td>
<td>2.5</td>
<td>0.04</td>
<td>-1.72</td>
</tr>
<tr>
<td>4</td>
<td>65.7</td>
<td>2.7</td>
<td>0.06</td>
<td>-1.56</td>
</tr>
<tr>
<td>5</td>
<td>77.0</td>
<td>2.8</td>
<td>0.08</td>
<td>-1.43</td>
</tr>
<tr>
<td>6</td>
<td>79.3</td>
<td>2.8</td>
<td>0.09</td>
<td>-1.32</td>
</tr>
<tr>
<td>7</td>
<td>81.9</td>
<td>2.8</td>
<td>0.11</td>
<td>-1.23</td>
</tr>
<tr>
<td>8</td>
<td>85.1</td>
<td>2.8</td>
<td>0.13</td>
<td>-1.14</td>
</tr>
<tr>
<td>9</td>
<td>85.6</td>
<td>2.8</td>
<td>0.14</td>
<td>-1.06</td>
</tr>
<tr>
<td>10</td>
<td>100.1</td>
<td>2.9</td>
<td>0.16</td>
<td>-0.99</td>
</tr>
<tr>
<td>11</td>
<td>102.2</td>
<td>3.0</td>
<td>0.18</td>
<td>-0.93</td>
</tr>
<tr>
<td>12</td>
<td>104.0</td>
<td>3.0</td>
<td>0.19</td>
<td>-0.86</td>
</tr>
<tr>
<td>13</td>
<td>105.1</td>
<td>3.0</td>
<td>0.21</td>
<td>-0.80</td>
</tr>
<tr>
<td>14</td>
<td>108.7</td>
<td>3.0</td>
<td>0.23</td>
<td>-0.75</td>
</tr>
<tr>
<td>15</td>
<td>109.24</td>
<td>3.0</td>
<td>0.24</td>
<td>-0.69</td>
</tr>
<tr>
<td>16</td>
<td>122.1</td>
<td>3.1</td>
<td>0.26</td>
<td>-0.64</td>
</tr>
<tr>
<td>17</td>
<td>123.7</td>
<td>3.1</td>
<td>0.28</td>
<td>-0.59</td>
</tr>
<tr>
<td>18</td>
<td>125.0</td>
<td>3.1</td>
<td>0.29</td>
<td>-0.54</td>
</tr>
<tr>
<td>19</td>
<td>129.8</td>
<td>3.1</td>
<td>0.31</td>
<td>-0.50</td>
</tr>
<tr>
<td>20</td>
<td>130.3</td>
<td>3.1</td>
<td>0.33</td>
<td>-0.45</td>
</tr>
<tr>
<td>21</td>
<td>131.1</td>
<td>3.1</td>
<td>0.34</td>
<td>-0.40</td>
</tr>
<tr>
<td>22</td>
<td>135.6</td>
<td>3.2</td>
<td>0.36</td>
<td>-0.36</td>
</tr>
<tr>
<td>23</td>
<td>137.6</td>
<td>3.2</td>
<td>0.38</td>
<td>-0.32</td>
</tr>
<tr>
<td>24</td>
<td>138.9</td>
<td>3.2</td>
<td>0.39</td>
<td>-0.27</td>
</tr>
<tr>
<td>25</td>
<td>151.3</td>
<td>3.2</td>
<td>0.41</td>
<td>-0.23</td>
</tr>
<tr>
<td>26</td>
<td>152.1</td>
<td>3.2</td>
<td>0.43</td>
<td>-0.19</td>
</tr>
<tr>
<td>27</td>
<td>155.0</td>
<td>3.3</td>
<td>0.44</td>
<td>-0.15</td>
</tr>
<tr>
<td>28</td>
<td>172.0</td>
<td>3.3</td>
<td>0.46</td>
<td>-0.10</td>
</tr>
<tr>
<td>29</td>
<td>181.2</td>
<td>3.4</td>
<td>0.48</td>
<td>-0.06</td>
</tr>
<tr>
<td>30</td>
<td>181.8</td>
<td>3.4</td>
<td>0.49</td>
<td>-0.02</td>
</tr>
<tr>
<td>31</td>
<td>183.6</td>
<td>3.4</td>
<td>0.51</td>
<td>0.02</td>
</tr>
<tr>
<td>32</td>
<td>192.3</td>
<td>3.4</td>
<td>0.52</td>
<td>0.06</td>
</tr>
<tr>
<td>33</td>
<td>192.8</td>
<td>3.4</td>
<td>0.54</td>
<td>0.10</td>
</tr>
<tr>
<td>34</td>
<td>193.3</td>
<td>3.4</td>
<td>0.56</td>
<td>0.15</td>
</tr>
<tr>
<td>35</td>
<td>194.6</td>
<td>3.4</td>
<td>0.57</td>
<td>0.19</td>
</tr>
<tr>
<td>36</td>
<td>195.2</td>
<td>3.4</td>
<td>0.59</td>
<td>0.23</td>
</tr>
<tr>
<td>37</td>
<td>195.2</td>
<td>3.4</td>
<td>0.61</td>
<td>0.27</td>
</tr>
<tr>
<td>38</td>
<td>197.5</td>
<td>3.4</td>
<td>0.62</td>
<td>0.32</td>
</tr>
<tr>
<td>39</td>
<td>207.2</td>
<td>3.5</td>
<td>0.64</td>
<td>0.36</td>
</tr>
<tr>
<td>40</td>
<td>208.0</td>
<td>3.5</td>
<td>0.66</td>
<td>0.40</td>
</tr>
<tr>
<td>41</td>
<td>208.8</td>
<td>3.5</td>
<td>0.67</td>
<td>0.45</td>
</tr>
<tr>
<td>42</td>
<td>208.8</td>
<td>3.5</td>
<td>0.69</td>
<td>0.50</td>
</tr>
<tr>
<td>43</td>
<td>213.7</td>
<td>3.5</td>
<td>0.71</td>
<td>0.54</td>
</tr>
<tr>
<td>44</td>
<td>218.3</td>
<td>3.5</td>
<td>0.72</td>
<td>0.59</td>
</tr>
<tr>
<td>45</td>
<td>225.6</td>
<td>3.6</td>
<td>0.74</td>
<td>0.64</td>
</tr>
<tr>
<td>46</td>
<td>255.5</td>
<td>3.7</td>
<td>0.76</td>
<td>0.69</td>
</tr>
<tr>
<td>47</td>
<td>272.8</td>
<td>3.7</td>
<td>0.77</td>
<td>0.75</td>
</tr>
<tr>
<td>48</td>
<td>278.4</td>
<td>3.7</td>
<td>0.79</td>
<td>0.80</td>
</tr>
<tr>
<td>49</td>
<td>285.5</td>
<td>3.8</td>
<td>0.81</td>
<td>0.86</td>
</tr>
<tr>
<td>50</td>
<td>287.9</td>
<td>3.8</td>
<td>0.82</td>
<td>0.93</td>
</tr>
<tr>
<td>51</td>
<td>303.1</td>
<td>3.6</td>
<td>0.84</td>
<td>0.99</td>
</tr>
<tr>
<td>52</td>
<td>319.9</td>
<td>3.9</td>
<td>0.86</td>
<td>1.06</td>
</tr>
<tr>
<td>53</td>
<td>321.2</td>
<td>3.9</td>
<td>0.87</td>
<td>1.14</td>
</tr>
<tr>
<td>54</td>
<td>327.5</td>
<td>3.9</td>
<td>0.89</td>
<td>1.23</td>
</tr>
<tr>
<td>55</td>
<td>379.5</td>
<td>4.0</td>
<td>0.91</td>
<td>1.32</td>
</tr>
<tr>
<td>56</td>
<td>379.9</td>
<td>4.0</td>
<td>0.92</td>
<td>1.43</td>
</tr>
<tr>
<td>57</td>
<td>392.4</td>
<td>4.0</td>
<td>0.94</td>
<td>1.53</td>
</tr>
<tr>
<td>58</td>
<td>395.6</td>
<td>4.1</td>
<td>0.95</td>
<td>1.72</td>
</tr>
<tr>
<td>59</td>
<td>448.9</td>
<td>4.2</td>
<td>0.97</td>
<td>1.94</td>
</tr>
<tr>
<td>60</td>
<td>507.2</td>
<td>4.3</td>
<td>0.99</td>
<td>2.32</td>
</tr>
</tbody>
</table>
Regression Output:

<table>
<thead>
<tr>
<th></th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>-4.77249565</td>
</tr>
<tr>
<td>Std Err of Y Est</td>
<td>0.078679204</td>
</tr>
<tr>
<td>R Squared</td>
<td>0.993725581</td>
</tr>
<tr>
<td>No. of Observations</td>
<td>60</td>
</tr>
<tr>
<td>Degrees of Freedom</td>
<td>58</td>
</tr>
<tr>
<td>X Coefficient(s)</td>
<td>1.119658</td>
</tr>
<tr>
<td>Std Err of Coef.</td>
<td>0.011682</td>
</tr>
</tbody>
</table>

\[ r = (R \text{ Squared})^{0.5} = 0.996858 \]

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
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(C2))**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(C2))**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 Saguling 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\TierGQH.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\SOMonth.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-Fieiring 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
'rkl  - 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\QAmmAR2.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
"<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
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)

'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)
\[ x(1, 1) = \text{MTF}(1) \]
\[ 'MTH(\text{Month}=13) = \text{MTF}(\text{Month}=1) \]
\[ \text{MTF}(13) = \text{MTF}(1) \]
\[ 'STF(\text{Month}=13) = \text{STF}(\text{Month}=1) \]
\[ \text{STF}(13) = \text{STF}(1) \]

FOR \text{NYear} = 1 \text{ TO } N
FOR \text{Month} = 1 \text{ \ TO \ 12}
20 \text{ GOSUB } \text{ random. number}
\text{t(Month)} = z
\text{x(Month + 1, NYear)} = \text{MTF(Month + 1)} + b(\text{Month}) \times (\text{x(Month, NYear)} - \text{MTF(Month)}) + \text{t(Month)}
\times \text{STF(Month + 1)} \times ((1 - \text{RTF(Month)} \times 2) \times 0.5)
\text{IF } \text{x(Month + 1, NYear)} < 0 \text{ THEN}
\text{GOTO } 20
\text{ELSE}
\text{END IF}

\text{SELECT CASE Month + 1}
\text{CASE 2}
\text{XGQM(Month + 1, NYear)} = (\text{x(Month + 1, NYear)} \times 0.317 + 1) \times (1 / 0.317)
\text{CASE 3}
\text{XGQM(Month + 1, NYear)} = (\text{x(Month + 1, NYear)} \times 0.297 + 1) \times (1 / 0.297)
\text{CASE 4}
\text{XGQM(Month + 1, NYear)} = (\text{x(Month + 1, NYear)} \times 0.304 + 1) \times (1 / 0.304)
\text{CASE 5}
\text{XGQM(Month + 1, NYear)} = (\text{x(Month + 1, NYear)} \times 0.293 + 1) \times (1 / 0.293)
\text{CASE 6}
\text{XGQM(Month + 1, NYear)} = (\text{x(Month + 1, NYear)} \times 0.405 + 1) \times (1 / 0.405)
\text{CASE 7}
\text{XGQM(Month + 1, NYear)} = (\text{x(Month + 1, NYear)} \times 0.545 + 1) \times (1 / 0.545)
\text{CASE 8}
\text{XGQM(Month + 1, NYear)} = (\text{x(Month + 1, NYear)} \times 0.86 + 1) \times (1 / 0.86)
\text{CASE 9}
\text{XGQM(Month + 1, NYear)} = (\text{x(Month + 1, NYear)} \times 0.073 + 1) \times (1 / 0.073)
\text{CASE 10}
\text{XGQM(Month + 1, NYear)} = (\text{x(Month + 1, NYear)} \times 0.592 + 1) \times (1 / 0.592)
\text{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

'******************************************************************************
'*
' subroutine 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
srk1Qanal(Month, NR) = 0
srk1Qanbl(Month, NR) = 0
NEXT Month
FOR NYear = 1 TO N - 1
FOR Month = 1 TO 12
srk1Qanal(Month, NR) = srk1Qanal(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
srk1Qanbl(Month, NR) = srk1Qanbl(Month, NR) + (XGQMonth(Month, NYear) - AXGQMonth(Month, NR)) ^ 2
NEXT Month
NEXT NYear
FOR Month = 1 TO 12
rk1XGQMonth(Month, NR) = srk1Qanal(Month, NR) / srk1Qanbl(Month, NR)
NEXT Month
RETURN
'---------- rk2QMonth
rk2QMonth.TT:
FOR Month = 1 TO 12
srk2Qanal(Month, NR) = 0
srk2Qanbl(Month, NR) = 0
NEXT Month
FOR NYear = 1 TO N - 2
FOR Month = 1 TO 12
srk2Qanal(Month, NR) = srk2Qanal(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
srk2Qanbl(Month, NR) = srk2Qanbl(Month, NR) + (XGQMonth(Month, NYear) - AXGQMonth(Month, NR))
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, DATES$, TIMES$
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.TQann:
'$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

'*****************************************************************************

Function for Statistics Calculation
*****************************************************************************'

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

'------------------------------ Std.XGQan
Std.Calc.QAnn:
VXGQan(NR) = SDXGQanS(NR) / (N - 1)
StdXGQan(NR) = (VXGQan(NR))^1/2
RETURN

'------------------------------ Cs.XGQan
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
Srkl = 0
Srkl2 = 0
SHXGQan = 0
SMinXGQan = 0
SMaxXGQan = 0
FOR NR " 1 TO NRep
   SAXGQan = SAXGQan + AXGQan(NR)
   SStdXGQan = SStdXGQan + StdXGQan(NR)
   SCsXGQan = SCsXGQan + CsXGQan(NR)
   Srkl = Srkl + rk1(NR)
   Srkl2 = Srkl2 + 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 = Srkl / 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 = SDStdXGQan / (NRep - 1)
StdStdXGQan = (VStdXGQan) ^ .5
VCsXGQan = SDCsXGQan / (NRep - 1)
StdCsXGQan = (VCsXGQan) ^ .5
Vrk1 = SDrk1 / (NRep - 1)
Stdrk1 = (Vrk1) ^ .5
Vrk2 = SDrk2 / (NRep - 1)
Stdrk2 = (Vrk2) ^ .5
VHXGQan = SDHXGQan / (NRep - 1)
StdHXGQan = (VHXGQan) ^ .5
VMinXGQan = SDMinXGQan / (NRep - 1)
StdMinXGQan = (VMinXGQan) ^ .5
VMaxXGQan = SDMaxXGQan / (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.QAnn:
PRINT #102, "----------> Statistics Test"
PRINT #102, "AAvgXGQan = "; AAVgXGQan
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, "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, "L-Bonds Hist Gen. U-Bonds"
PRINT #102, "AAvg "; USING "####.####"; LAA; AvgXGQan; UA
PRINT #102, "AStd "; USING "####.####"; LAStd; StdXGQan; AStdXGQan; UStd
PRINT #102, "ACs "; USING "####.####"; LCS; CsXGQan; ACSXGQan; UCs
PRINT #102, "Ark1 "; USING "####.####"; LRk1; rk1XGQan; Ark1; Urk1
PRINT #102, "Ark2 "; USING "####.####"; LRk2; rk2XGQan; Ark2; Urk2
PRINT #102, "AHurst"; USING "####.####"; LHXMLGQan; HQan; AHXGQan; UHXGQan
PRINT #102, "AMin "; USING "####.####"; LMINXGQan; MINQan; AMinXGQan; UMINXGQan
PRINT #102, "AMax "; USING "####.####"; LMAXXGQan; MAXQan; AMAXXGQan; UMAXXGQan
PRINT #102, "The program of TierQan is finish."
RETURN
Initialization

Init.Stat.Mon:

`$DYNAMIC`

```
DIM MinQMonth(12), MaxQMonth(12), AvgQMonth(12), StdQMonth(12), rklQMonth(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 VCxsXGQM(12), StdCsXGQM(12), Vrk1XGQM(12), Stdrk1XGQM(12), Vrk2XGQM(12), Stdrk2XGQM(12)
DIM VMInXGQM(12), StdMinXGQM(12), VMaxXGQM(12), StdMaxXGQM(12), DAXGQM(12), SDAXGQM(12)
DIM DStdXGQM(12), SDstdXGQM(12), DCsXGQM(12), SDCsXGQM(12), Drk1XGQM(12), SDrk1XGQM(12)
DIM Drk2XGQM(12), SDrk2XGQM(12), SDMinXGQM(12), SDMaxXGQM(12), DMaxXGQM(12)
DIM SDMaxXGQM(12), AAXGQM(12), AStdXGQM(12), ACSXGQM(12), Arrk1XGQM(12), Arrk2XGQM(12)
DIM AMInXGQM(12), AMaxXGQM(12), SAXGQM(12), SStdXGQM(12), SCsXGQM(12), Arrk1XGQM(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 SDXGQM(12, 100), SDXGQM(12, 100), DXGQM(12, 100), DXGQM(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(11)

INPUT #215, AvgQMonth(1), AvgQMonth(2), AvgQMonth(3), AvgQMonth(4), AvgQMonth(5), AvgQMonth(6), AvgQMonth(7), AvgQMonth(8), AvgQMonth(9), AvgQMonth(10), AvgQMonth(11)

INPUT #215, StdQMonth(1), StdQMonth(2), StdQMonth(3), StdQMonth(4), StdQMonth(5), StdQMonth(6), StdQMonth(7), StdQMonth(8), StdQMonth(9), StdQMonth(10), StdQMonth(11)

INPUT #215, rklQMonth(1), rklQMonth(2), rklQMonth(3), rklQMonth(4), rklQMonth(5), rklQMonth(6), rklQMonth(7), rklQMonth(8), rklQMonth(9), rklQMonth(10), rklQMonth(11), rklQMonth(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
SrklXGQM(Month) = 0
Srkl2XGQM(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)
SrklXGQM(Month) = SrklXGQM(Month) + rk1XGQM(Month, NR)
SrJc2XGQM(Month) = SrJc2XGQM(Month) + rJc2XGQM(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
AAXGOM(Month) = SAXGQM(Month) / NRep
AStdXGQMS(Month) = SStdXGQM(Month) / NRep
ACsXGQM(Month) = SCsXGQM(Month) / NRep
Ark1XGQM(Month) = SrklXGQM(Month) / NRep
Ark2XGQM(Month) = SrJc2XGQM(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) = SDStdXGQS(Month) + DStdXGQS(Month)
DCsXGQS(Month) = (CsXGQM(Month, NR) - ACsXQG(Month))^2
SDCsXGQS(Month) = SDCsXGQS(Month) + DCsXGQS(Month)
Drk1XGQM(Month) = (rk1XGQM(Month, NR) - Ark1XGQM(Month))^2
SDrk1XGQMS(Month) = SDrk1XGQMS(Month) + Drk1XGQMS(Month)
Drk2XGQM(Month) = (rk2XGQM(Month, NR) - Ark2XGQM(Month))^2
SDrk2XGQMS(Month) = SDrk2XGQMS(Month) + Drk2XGQMS(Month)
DMinXGQMS(Month) = (MinXGQ(Month, NR) - AMinXGQ(Month))^2
SDMinXGQMS(Month) = SDMinXGQMS(Month) + DMinXGQMS(Month)
DMaxXGQMS(Month) = (MaxXGQ(Month, NR) - AMaxXGQ(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)
StdStdXGQMS(Month) = (VStdXGQM(Month))^ .5
VCsXGQM(Month) = SDcsXGQMS(Month) / (NRep - 1)
StdCsXGQM(Month) = (VCsXGQM(Month))^ .5
VrklXGQM(Month) = SDrk1XGQMS(Month) / (NRep - 1)
Stdtk1XGQM(Month) = (VrklXGQM(Month))^ .5
Vrk2XGQM(Month) = SDrk2XGQMS(Month) / (NRep - 1)
Stdtk2XGQM(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, "---------------> Statistics Test"
'-------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, "AAAvg "; USING "###########"; LAM(Month); AvgQMonth(Month); AAXGQM(Month);
UAM(Month)
PRINT #202, "AStd "; USING "###########"; LStdM(Month); StdQMonth(Month); AStdXGQM(Month);
UStdM(Month)

143
PRINT #202, "ACs "; USING "####.###"; UCsM(Month); CsQMonth(Month); ACsXGQM(Month); Lrk1M(Month); rk1QMonth(Month); ArklXGQM(Month); Lrk2M(Month); rk2QMonth(Month); ArklXGQM(Month); Urk1M(Month); Lrk1M(Month); rk1QMonth(Month); ArklXGQM(Month); Urk2M(Month); Lrk2M(Month); rk2QMonth(Month); ArklXGQM(Month); AMinXGQM(Month); UMinXGQM(Month); LMaxXGQM(Month); UMaxXGQM(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, "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 (LMaxXGQH(Month) <= MaxQMonth(Month)) AND (MaxQMonth(Month) <= UMaxXGQH(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.
Description of Variables

NIF1  - inflows to Node 1: the Nanjunq 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
DNS2  - dead storage of Node 2
Coef2, Coefb2  - constants for calculating reservoir surface area Node 2
Coefc2, Coefd2, Coefe2  - 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.ASagCt

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.SagCt:

' $DYNAMIC

DIM TNST(61), TNSTm1(61), TNIF(61), TNEV(61), TSpill(61), TNOF(61), CheckTDNST(61), TNOF2(13, 61)
DIM QRSaq 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\RCSaq.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:\st\m\notesag.out" FOR OUTPUT AS #4 'Note for Saguling simulation.
OPEN "c:\sag\SagInp2.0ut" FOR OUTPUT AS #6 'Input-output file.
OPEN "c:\sag\SagCir.0ut" 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.

MinQTb:?
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, "************************ 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) 
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$; RC21(1); RC21(2); RC21(3); RC21(4); RC21(5); RC21(6); RC21(7); RC21(8); RC21(9); RC21(10); RC21(11); RC21(12) 
PRINT #4, 
PRINT #4, "Test of Reading Data of NIF1 for NYear = ", NT 
PRINT #4, " N IF1(1) NIF1(2) NIF1(3) NIF1(4) NIF1(5) NIF1(6) NIF1(7) NIF1(8) NIF1(9) NIF1(10) NIF1(11) NIF1(12)" 
a$ = "###.## ####.## ####.## ####.## ####.## ####.## ####.## ####.## ####.## ####.## ###.##" 

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"
        b$ = "#### #### #### #### #### #### #### #### #### #### ####

153
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) = Coef2a * (NST2(Month, NYear) + DNST2) ^ Coefb2
    ASur2(Month + 1) = Coef2a * (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
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

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
\[
\text{NST2}(\text{Month } + 1, \text{NYear}) = \text{NST2Max} \\
\text{ELSE} \\
\text{Spill2}(\text{Month}, \text{NYear}) = 0 \\
\text{END IF} \\
\text{Head2}(\text{Month } + 1) = \text{CoefC2} \times (\text{NST2}(\text{Month } + 1, \text{NYear}) + \text{DNST2}) \times \text{Coefd2} + \text{Coefc2} \\
\text{IF NST2}(\text{Month } + 1, \text{NYear}) \leq 0 \text{ THEN} \\
\text{Head2}(\text{Month } + 1) = \text{Head2Min} \\
\text{ELSE} \\
\text{END IF} \\
\text{IF NST2}(\text{Month } + 1, \text{NYear}) \geq \text{NST2Max} \text{ THEN} \\
\text{Head2}(\text{Month } + 1) = \text{Head2Max} \\
\text{ELSE} \\
\text{END IF} \\
\text{IF NST2}(\text{Month}, \text{NYear}) \leq 0 \text{ THEN} \\
\text{Head2}(\text{Month}) = \text{Head2Min} \\
\text{ELSE} \\
\text{END IF} \\
\text{IF NST2}(\text{Month}, \text{NYear}) > \text{NST2Max OR NST2}(\text{Month}, \text{NYear}) = 0 \text{ THEN} \\
\text{Head2}(\text{Month}) = \text{Head2Max} \\
\text{ELSE} \\
\text{END IF} \\
\text{AvgHead2} = (\text{Head2}(\text{Month}) + \text{Head2}(\text{Month } + 1)) \times .5 \\
\text{LEG2}(\text{Month}, \text{NYear}) = (2.725 / 1000) \times \text{CoeffPlant2} \times \text{AvgHead2} \times \text{NOF2}(\text{Month}, \text{NYear}) \\
\text{TNOF2}(\text{Month}, \text{NYear}) = \text{NOF2}(\text{Month}, \text{NYear}) + \text{Spill2}(\text{Month}, \text{NYear}) \\
\text{TNIF}(\text{NYear}) = \text{TNIF}(\text{NYear}) + \text{NIF2}(\text{Month}, \text{NYear}) \\
\text{TNEV}(\text{NYear}) = \text{TNEV}(\text{NYear}) + \text{NEV2}(\text{Month}, \text{NYear}) \\
\text{TSpill}(\text{NYear}) = \text{TSpill}(\text{NYear}) + \text{Spill2}(\text{Month}, \text{NYear}) \\
\text{TNOF}(\text{NYear}) = \text{TNOF}(\text{NYear}) + \text{NOF2}(\text{Month}, \text{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 NYear Month NST1 NST13 TNIF TNOF TNEV TSpill"
  PRINT #4, USING b$; NR; NYear; NMonth; NST2(1, NYear); NST2(13, NYear); TNIF(NYear); TNOF(NYear); TNEV(NYear); TSpill(NYear); CheckTDNST(NYear)
ELSE

157
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, HNF2(Month, NYear), HST2(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
  NEXT Month
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
  NEXT NYear
NEXT Month
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 NYear
    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)
    NEXT Month
NEXT NYear
SNEV2(Month, NR) = SNEV2(Month, NR) + NEV2(Month, N\text{Year})
SSPILL2(Month, NR) = SSPILL2(Month, NR) + Spill2(Month, N\text{Year})
NEXT Month
NEXT N\text{Year}

FOR Month = 1 TO 12
ANIF2(Month, NR) = SNIF2(Month, NR) / N\text{Year}
ANST2(Month, NR) = SNST2(Month, NR) / N\text{Year}
ANOF2(Month, NR) = SNOF2(Month, NR) / N\text{Year}
ANEV2(Month, NR) = SNEV2(Month, NR) / N\text{Year}
ASPILL2(Month, NR) = SSPILL2(Month, NR) / N\text{Year}
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 N\text{Year} = 1 TO N\text{I\text{Year}}
FOR Month = 1 TO 12
DNIF2S(Month, NR) = (NIF2(Month, N\text{Year}) - ANIF2(Month, NR)) ^ 2
SDNIF2S(Month, NR) = SDNIF2S(Month, NR) + DNIF2S(Month, NR)
DNST2S(Month, NR) = (NST2(Month, N\text{Year}) - ANST2(Month, NR)) ^ 2
SDNST2S(Month, NR) = SDNST2S(Month, NR) + DNST2S(Month, NR)
DNOF2S(Month, NR) = (NOF2(Month, N\text{Year}) - ANOF2(Month, NR)) ^ 2
SDNOF2S(Month, NR) = SDNOF2S(Month, NR) + DNOF2S(Month, NR)
DNEV2S(Month, NR) = (NEV2(Month, N\text{Year}) - ANEV2(Month, NR)) ^ 2
SDNEV2S(Month, NR) = SDNEV2S(Month, NR) + DNEV2S(Month, NR)
NEXT Month
NEXT N\text{Year}
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 = NRTtest) THEN
    PRINT #17, "Agung W.H. SOEHARNO"
    PRINT #17, DATE$, TIME$
    PRINT #17, "-----------------------------------------------Stat.Calc - Replicate no. ", NR
    PRINT #17, "-----------------------------------------------"
PRINT #17, "NIF2 = Inflow to Node2: Saguling"
PRINT #17, "NYear May Jun Jul Aug Sep Oct Nov Dec Jan"
Feb Mar Apr
PRINT #17, "Minimum of NIF2"
PRINT #17, "ANIF2 = Average of NIF2"
PRINT #17, "StdNIF2 = Standard Dev. of NIF2"
PRINT #17, "NST2 = End of period Storage of Node 2: Saguling"
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"
PRINT #17, "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(4); 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, "NOF2 = OutFlow from Node 2: Saguling"
PRINT #17, "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, " 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, "NEV2 = Evaporation of Node 2: Saguling"
PRINT #17, " NYear May Jun Jul Aug Sep Oct Nov Dec Jan
Feb Mar Apr"
FOR NYear = 1 TO NYear
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, " 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(4); 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, 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)

167
PRINT #17, "Maximum of SPILL2"
PRINT #17, USING "###.##"; NR; MaxSPILL2(1); MaxSPILL2(2); MaxSPILL2(3); MaxSPILL2(4); 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, "Agung W.H. SOEHARNO"
PRINT #16, DATE$, TIME$, TIME$
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, 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)
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, HR), StdNEV2(2, NR), StdNEV2(3, HR), 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, HR), StdSPILL2(3, NR), StdSPILL2(4, HR), 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, " NR May Jun Jul Aug Sep Oct Nov Dec Jan
   Feb Mar Apr"
PRINT #17, "ANIF2 = Average of NIF2"
ELSE
PRINT #17, "ANST2 = Average of NST2"
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, " NR May Jun Jul Aug Sep Oct Nov Dec Jan
   Feb Mar Apr"
ELSE
PRINT #17, "ANST2 = Average of NIF2"
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"
ELSE
  PRINT #17, "ANOF2 = Average of NOF2"
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"
ELSE
  PRINT #17, "ANEV2 = Average of NEV2"
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"
ELSE
  PRINT #17, "ASPILL2 = Average of SPILL2"
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)

171
NEXT NR
RETURN

'****************************************************************************************
' Statistics.Test
'****************************************************************************************
Statistics.Test.ASagCtr:
LOCATE 13, 10
PRINT "Statistics. Test"
PRINT #17, "------------> Statistics. Test"
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
  AANEV2(Month) = SANEV2(Month) / IReplicate
  AStdNEV2(Month) = SStdNEV2(Month) / IReplicate
  AASPILL2(Month) = SASPILL2(Month) / IReplicate
  AStdSPILL2(Month) = SStdSPILL2(Month) / IReplicate
END
ANOF2(6, NR), ANOF2(7, NR), ANOF2(8, NR), ANOF2(9, NR), ANOF2(10, NR), ANOF2(11, NR), ANOF2(12, NR)

INPUT #39, ANEY2(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)

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 SANEY2(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
AASPIILL2(Month) = SASPIILL2(Month) / IReplicate
AStdSPIILL2(Month) = SStdSPIILL2(Month) / IReplicate

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), SDStdSPIILL2S(12), DASPILL2S(12), DStdSPIILL2S(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
SDStdSPIILL2S(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)
SDStdNST2S(Month) = SDStdNST2S(Month) + DStdNST2S(Month)
SDASPILL2S(Month) = SDASPILL2S(Month) + DASPILL2S(Month)
SDStdSPIILL2S(Month) = SDStdSPIILL2S(Month) + DStdSPIILL2S(Month)
NEXT NR
NEXT 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
    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

176
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);
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, "StdStdOF2 = ">
PRINT #17, USING "###.###"; StdStdOF2(1); StdStdOF2(2); StdStdOF2(3); StdStdOF2(4);
StdStdOF2(5); StdStdOF2(6); StdStdOF2(7); StdStdOF2(8); StdStdOF2(9); StdStdOF2(10);
StdStdOF2(11); StdStdOF2(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, "StdStdEV2 = ">
PRINT #17, USING "###.###"; StdStdEV2(1); StdStdEV2(2); StdStdEV2(3); StdStdEV2(4);
StdStdEV2(5); StdStdEV2(6); StdStdEV2(7); StdStdEV2(8); StdStdEV2(9); StdStdEV2(10);
StdStdEV2(11); StdStdEV2(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, "StdStdSILL2 = ">
PRINT #17, USING "###.###"; StdStdSILL2(1); StdStdSILL2(2); StdStdSILL2(3);
StdStdSILL2(4); StdStdSILL2(5); StdStdSILL2(6); StdStdSILL2(7); StdStdSILL2(8);
StdStdSILL2(9); StdStdSILL2(10); StdStdSILL2(11); StdStdSILL2(12)
PRINT #16,
PRINT #16,
PRINT #16, "Agung W.H. SOEHARNO"
PRINT #16, DATE$, TIME$

"********************** Output of ASag-Ctr Program **********************"
PRINT #16, "* Node 2 : S A G U L I N G "
PRINT #16, "* "
PRINT #16, "-----------------------> Statistics. Test "

"********************** May Jun Jul Aug Sep Oct Nov Dec Jan Feb**********************"
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, "AAEVE2 = 
PRINT #16, USING "###.###"; AAEVE2(1); AAEVE2(2); AAEVE2(3); AAEVE2(4); AAEVE2(5);
AAEVE2(6); AAEVE2(7); AAEVE2(8); AAEVE2(9); AAEVE2(10); AAEVE2(11); AAEVE2(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 #16, "The Program ASag-Crt is finish."
PRINT #17, "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 - evaporation 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 - maximum
'Std - standard deviation
CLS
LOCATE 1, 10
PRINT "Cir-Ctr Program."
FOR NR = 1 TO NReplicate
GOSUB Simulation.CirCtr
GOSUB Wrap.Up.CirCtr
LOCATE 7, 10
PRINT "The Program of Cir-Ctr is Finish."
END IF
LOCATE 9, 10
PRINT "ACirCtr Program."
END IF
IF "Year" 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.CirCtrl:
' $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), LG3(12, 60), ASur3(13)
DIM Head3(13), AvgHead3(13), Spill13(12, 60), CoefEVap3(13), QR31(13), RC31(13)

OPEN "c:\data\NRrepl.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\Crlnp2.OUT" FOR OUTPUT AS #6 'input-output file
OPEN "c:\Cir\Crlnp3.OUT" FOR OUTPUT AS #10 'Input for Jatiluhur simulation
OPEN "c:\Cir\Crlnp3.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

NRTTest = 1
NTTest = 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, "************ C I R A T A **********"  
PRINT #4,
PRINT #4, "***** Print Out Test of Cir-Ctr Program for NYear = "; NTTest
PRINT #4,
PRINT #4, "Month, NYear, NReplicate"; Month; HYear; NReplicate
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),
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, "RC31 RC312 RC3i3 RC314 RC315 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
TNSTml(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 NYear
  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) = Coef3 * (NST3(Month, NYear) + DNST3) ^ Coefb3
    ASur3(Month + 1) = Coef3 * (RC31(Month + 1) + DNST3) ^ Coefb3
    NEV3(Month, NYear) = CcefEvap3(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
  END FOR

  IF (NOF3(Month, NYear) < 0) THEN
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
    END IF
ELSE
    GOTO 101
END IF

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

MaxQTb3 = (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
    DNOF3Max = NOF3(Month, NYear) - MaxQTb3
    NOF3(Month, NYear) = MaxQTb3
ELSE
    DNOF3Max = 0
END IF

NST3(Month + 1, NYear) = RC31(Month + 1) + DNOF3Max
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
Average HEAD3 = (HEAD3(Month) + HEAD3(Month + 1)) \times 0.5

101 LEG3(Month, NYear) = (2.725 / 1000) \times \text{CoefPlant3} \times \text{AvgHead3} \times \text{NOF3(Month, NYear)}

TNOF3(Month, NYear) = \text{NOF3(Month, NYear)} + \text{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
  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(11, NYear) + NOF3(1, NYear) - TNIF(1, 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 bS; 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 SHIF3(IMonth, IReplicate), ANIF3(IMonth, IReplicate), SDNIF3S(IMonth, IReplicate), VNIF3(IMonth, IReplicate)
DIM StdNIF3(IMonth, IReplicate), DNIF3S(IMonth, IReplicate), STNST3(IMonth, IReplicate), VNST3(IMonth, IReplicate)
DIM StdNST3(IMonth, IReplicate), DNST3S(IMonth, IReplicate), SNOF3(IMonth, IReplicate), VNOF3(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), SDSpill3S(IMonth, IReplicate), ASpill3(IMonth, IReplicate), DSpill3S(IMonth, IReplicate)
DIM VSpill3(IMonth, IReplicate), StdSpill3(IMonth, IReplicate), DSSpill3S(IMonth, IReplicate)

a$ = "##### ###### ####### ####### #######"
RETURN

'******************************************************************************
                  Statistics.Calc
'******************************************************************************
Statistics.Calc.ACircle:

OPEN "c:\Cir\CirInpl.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 NYear
    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
ELSE
END IF
IF NST3(Month, NYear) > MaxNST3(Month) THEN
    MaxNST3(Month) = NST3(Month, NYear)
ELSE
END IF
IF NOP3(Month, NYear) < MinNOP3(Month) THEN
    MinNOP3(Month) = NOP3(Month, NYear)
ELSE
END IF
IF NOP3(Month, NYear) > MaxNOP3(Month) THEN
    MaxNOP3(Month) = NOP3(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 NEF3

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 NYear

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) / NYear
ANST3(Month, NR) = SNST3(Month, NR) / NYear
ANOF3(Month, NR) = SNOF3(Month, NR) / NYear
ANEV3(Month, NR) = SNEV3(Month, NR) / NYear
ASpill3(Month, NR) = SSpill3(Month, NR) / NYear

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
SDNEV3S(Month, NR) = 0
SDSpilll3S(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)
    DSpilll3S(Month, NR) = (Spill3(Month, NYear) - ASpil13(Month, NR)) ^ 2
    SDSpilll3S(Month, NR) = SDSpilll3S(Month, NR) + DSpilll3S(Month, NR)
  NEXT Month
NEXT NYear
RETURN

'-----------------------------------------------StdNEV3
Std.Calc:

FOR Month = 1 TO 12
  VNIF3(Month, NR) = SDNIF3S(Month, NR) / (N!Year - 1)
  StdNIF3(Month, NR) = (VNIF3(Month, NR)) ^ .5
  VNST3(Month, NR) = SDNST3S(Month, NR) / (N!Year - 1)
  StdNST3(Month, NR) = (VNST3(Month, NR)) ^ .5
  VNOF3(Month, NR) = SDNOF3S(Month, NR) / (N!Year - 1)
  StdNOF3(Month, NR) = (VNOF3(Month, NR)) ^ .5
  VNEV3(Month, NR) = SDNEV3S(Month, NR) / (N!Year - 1)
  StdNEV3(Month, NR) = (VNEV3(Month, NR)) ^ .5
  VSpi1l3(Month, NR) = SDSpilll3S(Month, NR) / (N!Year - 1)
  StdSpilll3(Month, NR) = (VSpi1l3(Month, NR)) ^ .5

NEXT Month

RETURN
OutStat.Calc:
IF (NR = NRTest) THEN
  PRINT #17,
  PRINT #17,
  PRINT #17, "Agung W.H. SOEKARNO"
  PRINT #17, TIME$, DATE$
  PRINT #17, "***************************************************
  PRINT #17, "********** Output of Cir-Ctr Program**********"
  PRINT #17, "* Node 3: CIRATA*
  PRINT #17, "**-> Stat.Calc - Replicate no. ", NR
  PRINT #17, "*******************************
PRINT #17,
  FOR NYear - 1 TO NYear
    PRINT #17, 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"
  PRINT #17, " ********** Minimum of NIF3 **********"
  PRINT #17, 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, " ********** Average of NIF3 **********"
  PRINT #17, 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"

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"

FOR NR = 1 TO 12
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"
PRINT #17, PRINT #17, "Minimum of NOF3"
PRINT #17, PRINT #17, "Average of NOF3"
PRINT #17, PRINT #17, "Maximum of NOF3"
PRINT #17, PRINT #17, "StdNOF3 = Standard Dev. of NOF3"
PRINT #17, PRINT #17, "NEV3 = Evaporation of Node 3: Cirata"
PRINT #17, " NYear May Jun Jul Aug Sep Oct Nov Dec Jan
Feb Mar Apr"
PRINT #17,
FOR NYear = 1 TO NYear
  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(4);
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 NYear
  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(4); 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, DATES$, TIME$, TIMES$
PRINT #16, "---------------------------------------------------------------------"
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
PRINT #16, "---------------------------------------------------------------------"
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, StdNIF(1, NR), StdNIF(2, NR), StdNIF(3, NR), StdNIF(4, NR), StdNIF(5, NR), StdNIF(6, NR), StdNIF(7, NR), StdNIF(8, NR), StdNIF(9, NR), StdNIF(10, NR), StdNIF(11, NR), StdNIF(12, NR)
WRITE #38, StdNST(1, NR), StdNST(2, NR), StdNST(3, NR), StdNST(4, NR), StdNST(5, NR), StdNST(6, NR), StdNST(7, NR), StdNST(8, NR), StdNST(9, NR), StdNST(10, NR), StdNST(11, NR), StdNST(12, NR)
WRITE #38, StdNOF(1, NR), StdNOF(2, NR), StdNOF(3, NR), StdNOF(4, NR), StdNOF(5, NR), StdNOF(6, NR), StdNOF(7, NR), StdNOF(8, NR), StdNOF(9, NR), StdNOF(10, NR), StdNOF(11, NR), StdNOF(12, NR)
WRITE #38, StdNEV(1, NR), StdNEV(2, NR), StdNEV(3, NR), StdNEV(4, NR), StdNEV(5, NR), StdNEV(6, NR), StdNEV(7, NR), StdNEV(8, NR), StdNEV(9, NR), StdNEV(10, NR), StdNEV(11, NR), StdNEV(12, NR)
WRITE #38, StdSpil(1, NR), StdSpil(2, NR), StdSpil(3, NR), StdSpil(4, NR), StdSpil(5, NR), StdSpil(6, NR), StdSpil(7, NR), StdSpil(8, NR), StdSpil(9, NR), StdSpil(10, NR), StdSpil(11, NR), StdSpil(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, "ANIF = Average of NIF3"
ELSE
ENDIF
PRINT #17, USING "####.####"; NR; ANIF(1, NR); ANIF(2, NR); ANIF(3, NR); ANIF(4, NR); ANIF(5, NR); ANIF(6, NR); ANIF(7, NR); ANIF(8, NR); ANIF(9, NR); ANIF(10, NR); ANIF(11, NR); ANIF(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"
ENDIF
PRINT #17, USING "###.###"; NR; ANIF(1, NR); ANIF(2, NR); ANIF(3, NR); ANIF(4, NR); ANIF(5, NR); ANIF(6, NR); ANIF(7, NR); ANIF(8, NR); ANIF(9, NR); ANIF(10, NR); ANIF(11, NR); ANIF(12, NR) Return
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"
ELSE
PRINT #17, "ANEV3 = Average of NEV3"
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(7, 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"
ELSE
PRINT #17, "ASpill3 = Average of Spill3"
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 "Statistics.Test"
IF (NR = NRTest) THEN
PRINT #17, "------------> Statistics.Test"
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 statistics

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, HR), StdNST3(2, HR), StdNST3(3, HR), StdNST3(4, HR), StdNST3(5, HR), StdNST3(6, HR), StdNST3(7, HR), StdNST3(8, HR), StdNST3(9, HR), StdNST3(10, HR), StdNST3(11, HR), StdNST3(12, HR)

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, HR), StdNEV3(2, HR), StdNEV3(3, HR), StdNEV3(4, HR), StdNEV3(5, HR), StdNEV3(6, HR), StdNEV3(7, HR), StdNEV3(8, HR), StdNEV3(9, HR), StdNEV3(10, HR), StdNEV3(11, HR), StdNEV3(12, HR)

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

NEXT NR
SANSTJ(Month) = 0
SStdNSTJ(Month) = 0
SANOFJ(Month) = 0
SStdNOFJ(Month) = 0
SANEV3(Month) = 0
SStdNEVJ(Month) = 0
SASpillJ(Month) = 0
SStdspillJ(Month) = 0
NEXT Month

FOR Month = 1 TO 12
    FOR NR = 1 TO IReplicate
        SANIFJ(Month) = SANIFJ(Month) + ANIFJ(Month, NR)
        SStdNIFJ(Month) = SStdNIFJ(Month) + StdNIFJ(Month, NR)
        SANSTJ(Month) = SANSTJ(Month) + ANSTJ(Month, NR)
        SStdNSTJ(Month) = SStdNSTJ(Month) + StdNSTJ(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)
        SASpill13(Month) = SASpill13(Month) + ASpill13(Month, NR)
        SStdSpill13(Month) = SStdSpill13(Month) + StdSpill13(Month, NR)
    NEXT NR
    NEXT Month

FOR Month = 1 TO 12
    AANIFJ(Month) = SANIFJ(Month) / IReplicate
    AAstdNIFJ(Month) = SStdNIFJ(Month) / IReplicate
    AANSTJ(Month) = SANSTJ(Month) / IReplicate
    AAstdNSTJ(Month) = SStdNSTJ(Month) / IReplicate
    AANOF3(Month) = SANOF3(Month) / IReplicate
    AAstdNOF3(Month) = SStdNOF3(Month) / IReplicate
    AANEV3(Month) = SANEV3(Month) / IReplicate
    AASpill13(Month) = SASpill13(Month) / IReplicate
    AasDspill13(Month) = SStdSpill13(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 SDASpil13S(12), SDStdSpil13S(12), DASpil13S(12), DStdSpil13S(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
    SDASpil13S(Month) = 0
    SDStdSpil13S(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)
    NEXT NR
NEXT Month
SDStdN0FJS(Month) = SDStdN0FJS(Month) + DStdN0FJS(Month)
DANEVJS(Month) = (ANEV3(Month, NR) - AANEV3(Month)) ^ 2
SDANEVJS(Month) = SDANEVJS(Month) + DANEVJS(Month)
DStdN0FJS(Month) = (StdN0FJS(Month, NR) - AStdN0FJS(Month)) ^ 2
SDStdN0FJS(Month) = SDStdN0FJS(Month) + DStdN0FJS(Month)
DASpillJS(Month) = (ASpillJS(Month, NR) - AASpillJS(Month)) ^ 2
SDASpillJS(Month) = SDASpillJS(Month) + DASpillJS(Month)
DStdSpillJS(Month) = (StdSpillJS(Month, NR) - AStdSpillJS(Month)) ^ 2
SDStdSpillJS(Month) = SDStdSpillJS(Month) + DStdSpillJS(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 VASpillJ(12), StdASpillJ(12), VStdSpillJ(12), StdStdSpillJ(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
VANEVJ3(Month) = SDANENVJ3S(Month) / (IREplicate - 1)
StdANENVJ3(Month) = (VANEVJ3(Month)) ^ .5
VSTDANENVJ3(Month) = SDSTDANENVJ3S(Month) / (IREplicate - 1)
STDSTDANENVJ3(Month) = (VSTDANENVJ3(Month)) ^ .5

VASpill13(Month) = SDASpill13S(Month) / (IREplicate - 1)
StdASpill13(Month) = (VASpill13(Month)) ^ .5
VSTDASpill13(Month) = SDSTDASpill13S(Month) / (IREplicate - 1)
STDSTDASpill13(Month) = (VSTDASpill13(Month)) ^ .5

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

IF (NR = NRTest) THEN

PRINT #17, "AANIFJ3 = "
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, "AStdNIFJ3 ="
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, "AANSTJ3 = "
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, "AStdNSTJ3 ="
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, "AANOFJ3 = "
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, "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, "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, "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);
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, "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, "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, "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, "StdASpillJ = "
PRINT #17, USING "###.###"; StdASpillJ(1); StdASpillJ(2); StdASpillJ(3); StdASpillJ(4);
StdASpillJ(5); StdASpillJ(6); StdASpillJ(7); StdASpillJ(8); StdASpillJ(9); StdASpillJ(10);
StdASpillJ(11); StdASpillJ(12)
PRINT #17, "StdASpillJ = "
PRINT #17, USING "###.###"; StdASpillJ(1); StdASpillJ(2); StdASpillJ(3); StdASpillJ(4);
StdASpillJ(5); StdASpillJ(6); StdASpillJ(7); StdASpillJ(8); StdASpillJ(9); StdASpillJ(10);
StdASpillJ(11); StdASpillJ(12)
ELSE
END IF
PRINT #16,
PRINT #16,
PRINT #16, "Agung W.H. SOEHARNO"
PRINT #16, DATE$, TIME$
JatCtn.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 - evaporation 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
Coef4a, Coefb4 - constants for calculating reservoir surface area Node 4
Coefc4, Coefd4, Coefe4 - 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 "
PRINT #16, "P R I N T # 1 6 ,
PRINT #16, " May Jun Jul Aug Sep Oct Nov Dec Jan Feb"
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."
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.AJatCtr:
' $DYNAHIC

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(l3), AvgHead4(l3), 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 'output 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$

###.

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$
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.JatCtt:
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 as; 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)"
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"

    b$ = "### #### ##.## ##.## ##.## #.## #### ######## ######## ########

    ELSE
        END IF

FOR Month = 1 TO 12

    ASur4(Month) = Coef4 * (NST4(Month, NYear) + DNST4) ^ Coefb4
    ASur4(Month + 1) = Coef4 * (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

221
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) < MinQTh4)) 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 + 108.758.2
    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
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 = NRTTest) 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.JatCt:

  CLOSE #1
  CLOSE #5

RETURN

'AJatCt: Analysis of JatCt.
'******************************************************************************
'  * Initiation
'******************************************************************************
Initialization.AJatCt:

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 SN1F4(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 VSpill4(IMonth, IReplicate), ASpill4(IMonth, IReplicate), SDspill4S(IMonth, IReplicate)
ad = """
RETURN

'**************************************************************************
' Statistics.Calc
'**************************************************************************
Statistics.Calc.AJetCtr:
OPEN "c:\Jet\JetInpl.out" FOR INPUT AS #36 'input of stat.calc
FOR NYear = 1 TO N1Year
    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

228
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
    AHEV4(Month, NR) = SNEV4(Month, NR) / NIYear
    ASpil14(Month, NR) = SSpil14(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) = DNIF4S(Month, NR) + DNIF4S(Month, NR)^2
DNST4S(Month, NR) = (NST4(Month, NYear) - ANST4(Month, NR))^2
SDNST4S(Month, NR) = DNST4S(Month, NR) + DNST4S(Month, NR)^2
DNOF4S(Month, NR) = (NOF4(Month, NYear) - ANOF4(Month, NR))^2
SDNOF4S(Month, NR) = DNOF4S(Month, NR) + DNOF4S(Month, NR)^2
DNEV4S(Month, NR) = (NEV4(Month, NYear) - ANEV4(Month, NR))^2
SDNEV4S(Month, NR) = DNEV4S(Month, NR) + DNEV4S(Month, NR)^2
DSpill4S(Month, NR) = (Spill4(Month, NYear) - ASpill4(Month, NR))^2
SDSpill4S(Month, NR) = DSpill4S(Month, NR) + DSpill4S(Month, NR)^2

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, DATES$, TIMES$

"******************************************************************"  PRINT #17, "* Output of Jat-Ctr Program"
"**************************************************************"  PRINT #17, "* Node 4 : J A T I L U H U R"
"**************************************************************"  PRINT #17, "------------------------------------------ Stat.Calc - Replicate no. ", NR
  PRINT #17, "NIF4 • Inflow to Node 4: Jatiluhur"
  PRINT #17, "NYear May Jun Jul Aug Sep Oct Nov Dec Jan"

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, "NR May Jun Jul Aug Sep Oct Nov Dec Jan"

FOR NYear = 1 TO NIYear
  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)
NEXT NYear

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(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, PRINT #17,"
FOR NYear = 1 TO NYear
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,"
Feb Mar Apr
PRINT #17, 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, 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, PRINT #17, "Maximum of NST4"
PRINT #17, USING "#####.#"; NR; MaxNST4(1); MaxNST4(2); MaxNST4(3); MaxNST4(5); 232
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: Jatiiluhur"
PRINT #17, PRINT #17, " NYear May Jun Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr"
FOR NYear = 1 TO NYear
PRINT #17, 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, 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, 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, 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"
FOR NYear = 1 TO NIYear
PRINT #17, USING "#0000.#"; 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"
FOR NYear = 1 TO NIYear
PRINT #17, USING "#0000.#"; 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 NYear
PRINT #17, "Minimum of NEV4"
PRINT #17, USING "#0000.#"; 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 "#0000.#"; 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 "#0000.#"; NR; MaxNEV4(1); MaxNEV4(2); MaxNEV4(3); MaxNEV4(4); 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 "#0000.#"; 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"
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, " 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(4); 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)
PRINT #17,
PRINT #16,
PRINT #16,
PRINT #16,
PRINT #16, "Agung W.H. SOEHARNO"
PRINT #16, DATE$, TIME$, TIME$

PRINT #16, " Agung W.H. SOEHARNO"
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

"*****
PRINT #16, "JTSUM = Summary File of Node 4: Jatiluhur"
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),
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; 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"
ELSE
PRINT #17, "ANST4 = Average of NST4"
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"
ELSE
PRINT #17, "ANEV4 = Average of NEV4"
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, "ASp1 = Average of Spill14"
ELSE
END IF
PRINT #17, USING "#####.##"; NR; ASp1(1, NR); ASp1(2, NR); ASp1(3, NR);
ASp1(4, NR); ASp1(5, NR); ASp1(6, NR); ASp1(7, NR); ASp1(8, NR);
ASp1(9, NR); ASp1(10, NR); ASp1(11, NR); ASp1(12, NR)
NEXT NR
RETURN

'*************************************************************************
' Statistics.TEST
'*************************************************************************

Statistics.TEST.AJatCtr:
LOCATE 13, 10
PRINT "Statistics.TEST"
PRINT #17, "-----------> Statistics.TEST"
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.Test
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, ASPi14(1, NR), ASPi14(2, NR), ASPi14(3, NR), ASPi14(4, NR),
          ASPi14(5, NR), ASPi14(6, NR), ASPi14(7, NR), ASPi14(8, NR), ASPi14(9, NR), ASPi14(10, NR),
          ASPi14(11, NR), ASPi14(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, StdSpi14(1, NR), StdSpi14(2, NR), StdSpi14(3, NR), StdSpi14(4, NR),
          StdSpi14(5, NR), StdSpi14(6, NR), StdSpi14(7, NR), StdSpi14(8, NR), StdSpi14(9, NR), StdSpi14(10, NR),
          StdSpi14(11, NR), StdSpi14(12, NR)
NEXT NR
RETURN

'------------------------------------ AvgAvg.Calc
AvgAvg.Calc:
DIM SANIF4(12), SStdNIF4(12), AAIF4(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), SASpil14(12), SStdSpil14(12)
DIM AASpil14(12), AStdSpil14(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
    SASpil14(Month) = 0
    SStdSpil14(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)
        SASpil14(Month) = SASpil14(Month) + ASpil14(Month, NR)
        SStdSpil14(Month) = SStdSpil14(Month) + StdSpil14(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)
DASppl14S(Month) = (ASppl14(Month, NR) - AASppl14(Month)) ^ 2
SDASppl14S(Month) = SDASppl14S(Month) + DASppl14S(Month)
DStdSppl14S(Month) = (StdSppl14(Month, NR) - AStdSppl14(Month)) ^ 2
SDStdSppl14S(Month) = SDStdSppl14S(Month) + DStdSppl14S(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),VASppl14(12), StdASppl14(12)
DIM VStdSppl14(12), StdStdSppl14(12)

FOR Month = 1 TO 12
  VANIF4(Month) = SDANIF4S(Month) / (IREplicate - 1)
  StdANIF4(Month) = (VANIF4(Month))^ .5
  VStdNIF4(Month) = SDStdNIF4S(Month) / (IREplicate - 1)

243
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
    AASpi1l4(Month) = SASpi1l4(Month) / IReplicate
    AStdSpi1l4(Month) = SStdSpi1l4(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), DStdNOF4S(12), SDANOF4S(12), DStdNOF4S(12)
DIM SDANEV4S(12), SDStdNEV4S(12), DANEV4S(12), DStdNEV4S(12), SDASpi1l4S(12)
DIM SDStdSpi1l4S(12), DASpi1l4S(12), DStdSpi1l4S(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
    SDASpi1l4S(Month) = 0
    SDStdSpi1l4S(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

VASpil14 (Month) = SDASpil14S (Month) / (IReplicate - 1)
STAdSpil14 (Month) = (VASpil14 (Month)) ^ .5
VSTDSpil14 (Month) = SDSTDSpil14S (Month) / (IReplicate - 1)
STDSTDSpil14 (Month) = (VSTDSpil14 (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, "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, "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, "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, "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, "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, "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, "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, "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, "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, "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, "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, "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, "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, "AANOF4 ="
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, "StdStdOF4 = "
PRINT #17, USING "#.#"; StdStdOF4(1); StdStdOF4(2); StdStdOF4(3); StdStdOF4(4); StdStdOF4(5); StdStdOF4(6); StdStdOF4(7); StdStdOF4(8); StdStdOF4(9); StdStdOF4(10); StdStdOF4(11); StdStdOF4(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, "StdStdEV4 = "
PRINT #17, USING "#.#"; StdStdEV4(1); StdStdEV4(2); StdStdEV4(3); StdStdEV4(4); StdStdEV4(5); StdStdEV4(6); StdStdEV4(7); StdStdEV4(8); StdStdEV4(9); StdStdEV4(10); StdStdEV4(11); StdStdEV4(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 #17, "StdStdSpiI14 = "
PRINT #17, USING "#.#"; StdStdSpiI14(1); StdStdSpiI14(2); StdStdSpiI14(3); StdStdSpiI14(4); StdStdSpiI14(5); StdStdSpiI14(6); StdStdSpiI14(7); StdStdSpiI14(8); StdStdSpiI14(9); StdStdSpiI14(10); StdStdSpiI14(11); StdStdSpiI14(12)

PRINT #17, "StdStdSpi114 = "
PRINT #17, USING "#.#"; StdStdSpi114(1); StdStdSpi114(2); StdStdSpi114(3); StdStdSpi114(4); StdStdSpi114(5); StdStdSpi114(6); StdStdSpi114(7); StdStdSpi114(8); StdStdSpi114(9); StdStdSpi114(10); StdStdSpi114(11); StdStdSpi114(12)

PRINT #17, "StdStdSpiI14 = "
PRINT #17, USING "#.#"; StdStdSpiI14(1); StdStdSpiI14(2); StdStdSpiI14(3); StdStdSpiI14(4); StdStdSpiI14(5); StdStdSpiI14(6); StdStdSpiI14(7); StdStdSpiI14(8); StdStdSpiI14(9); StdStdSpiI14(10); StdStdSpiI14(11); StdStdSpiI14(12)
PRINT #16,
PRINT #16,
PRINT #16, "Agung W.H. SOEHARNO"
PRINT #16, DATE$, TIME$

PRINT #16, "******************************************************************************************"
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 "

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, "AASpid14 = "

247
PRINT #16, USING "##.#-##"; AASpil14(1); AASpil14(2); AASpil14(3); AASpil14(4); AASpil14(5); AASpil14(6); AASpil14(7); AASpil14(8); AASpil14(9); AASpil14(10); AASpil14(11); AASpil14(12)

PRINT #17, " The Program AJat-Ctr is finish."
PRINT #16, " The Program AJat-Ctr is finish."

RETURN
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
END FOR NR

GOSUB Statistics.TEST
PRINT #7, "Total Hydro Electric Power Within Year (TATE) = "; TATE

100 GOSUB Wrap.Up
    LOCATE 15, 10
    PRINT "The Sub-Program of Egy-Ctr is finish."
END

Initialization:

$DYNAMIC
OPEN "c:\data\NRepL.dat" FOR INPUT AS #11 'number of replication, year and months
INPUT #11, IMonth, NYear, 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),
MaxLEG4(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), STE(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
NEX T 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)
  
NEX T 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
NEX T NYear

LOCATE 13, 10

252
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 NYear
        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
    NEXT NYear
NEXT Month
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) / NYear
    TALEG2 (NR) = TALEG2 (NR) + ALEG2 (Month, NR)
    ALEG3 (Month, NR) = SLEG3 (Month, NR) / NYear
    TALEG3 (NR) = TALEG3 (NR) + ALEG3 (Month, NR)
    ALEG4 (Month, NR) = SLEG4 (Month, NR) / NYear
    TALEG4 (NR) = TALEG4 (NR) + ALEG4 (Month, NR)
    ATE (Month, NR) = STE (Month, NR) / NYear
NEXT Month

RETURN

--- 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 = NRTes) THEN
    PRINT #7,
    PRINT #7,
    PRINT #7, "Agung W.H. SOEHARNO"
    PRINT #7, DATE$
    PRINT #7, "************************************************************
            "
    PRINT #7, "* Output of Egy-Ctr Program"
    PRINT #7, "*
    PRINT #7, "--------------------------------- Stat.Calc - Replicate no. ", NR
    PRINT #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 NNIYear
        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

257
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
FOR NYear = 1 TO NYear
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; 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, "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)
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(4); 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$, TIME$
PRINT #16, " ****************************************************
PRINT #16, " Output of Egy-Ctr Program"
PRINT #16, " ****************************************************
PRINT #16, "------------------ Stat. Calc - Replicate no. ", NR
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 "-------------------> Statistics.Test"
PRINT #7, "-------------------> Statistics.Test"
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)

263
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, HR), StdTE(2, HR), StdTE(3, HR), 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)
    NEXT NR
NEXT Month
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) = SLEG2(Month) / IReplicate
AStdLEG2(Month) = SStdLEG2(Month) / IReplicate
AALEG3(Month) = SLEG3(Month) / IReplicate
AStdLEG3(Month) = SStdLEG3(Month) / IReplicate
AALEG4(Month) = SLEG4(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), DATES(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, "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, "StdDATE = " 
PRINT #7, USING "##...##"; StdDATE(1); StdDATE(2); StdDATE(3); StdDATE(4); StdDATE(5); StdDATE(6); StdDATE(7); StdDATE(8); StdDATE(9); StdDATE(10); StdDATE(11); StdDATE(12) 
PRINT #7, "StdDATE = " 
PRINT #7, USING "##...##"; StdDATE(1); StdDATE(2); StdDATE(3); StdDATE(4); StdDATE(5); StdDATE(6); StdDATE(7); StdDATE(8); StdDATE(9); StdDATE(10); StdDATE(11); StdDATE(12) 
PRINT #16, PRINT #16, "Agung W.H. SOEHARNO" PRINT #16, DATE$, TIME$ PR I N T # 1 6 , "**************************************************" PRINT #16, "* Output of Egy-Ctr Program" PRINT #16, "**" PRINT #16, "-----------------> Statistics.Test" PR 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) TAALLEG2 = 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, "StdALEG2 = " 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, 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)

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, "TAALEG2 = "; TAALEG2
PRINT #16, "TAALEG3 = "; TAALEG3
PRINT #16, "TAALEG4 = "; TAALEG4
PRINT #16, "TAATE = "; TAATE
Bibliography


