

**INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO (S-1)
KONSENTRASI TEKNIK ENERGI LISTRIK**



**KOMITMEN UNIT PEMBANGKIT TERMAL DENGAN METODE
INTEGER-CODED GENETIC ALGORITHM PADA
PT. PEMBANGKITAN JAWA-BALI**

SKRIPSI

Disusun oleh :

NAMA : ALIEF CATUR WIBOWO

NIM : 00.12.051

SEPTEMBER 2006

DALAM LANGKAH PENYEMPURNAAN
KEMERDEKAAN NEGARA
(1) PERSEKUTUAN NEGARA
MELAYU DAN NEGARA NEGERI
SABAH DAN SARAWAK

BUKUKAN MANDIRI KEMERDEKAAN TANGGAH
KEMERDEKAAN NEGARA
KEMERDEKAAN NEGARA
KEMERDEKAAN NEGARA

1975

1975

1975

1975

1975

LEMBAR PERSETUJUAN

**KOMITMEN UNIT PEMBANGKIT TERMAL DENGAN
METODE *INTEGER-CODED GENETIC ALGORITHM* PADA
PT. PEMBANGKITAN JAWA-BALI**

SKRIPSI

*Disusun Guna Melengkapi dan Memenuhi Syarat-Syarat
Guna Mencapai Gelar Sarjana Teknik*

Disusun Oleh :

**ALIEF CATUR WIBOWO
NIM. 00.12.051**



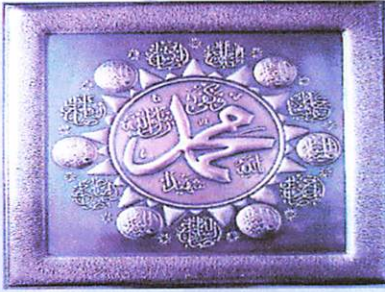
**Mengetahui,
Ketua Jurusan Teknik Elektro**

**Ir.F Yudi Limpraptono, MT
NIP.Y. 103 950 0274**

**Menyetujui,
Dosen Pembimbing**

**Ir. Yusuf Ismail Nakhoda, MT
NIP.Y. 101 880 0189**

**JURUSAN TEKNIK ELEKTRO (S-1)
KONSENTRASI TEKNIK ENERGI LISTRIK
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG**



ALHAMDULILLAHIRABBIL 'ALAMIIN

Terima kasih Ya Allah S.W.T berkat rahmat dan hidayah yang Engkau berikan hingga aku sekarang bisa menempuh Sarjana Teknik (ST), Semoga ilmu yang ku peroleh dapat berguna untuk diriku dan orang lain. Taklupa pada junjungan ku nabi Muhammad S.A.W.

*To Ibu **TERCINTA** : Karena air susu aku bisa hidup, karena keringatmu aku bisa berhasil, karena doamu aku bisa tenang dan karena usahamu aku bisa menjadi seperti sekarang.*

*To Bapak **TERCINTA** : Engkaulah penyemangat hidupku, kaulah orang yang pantas ditiru, kaulah orang yang bisa membimbingku, kaulah orang yang membuat aku lebih dewasa. engkaulah berdua yang membimbingku dari mulai aku kecil sampai menjadi Sarjana, Walau sering aku tak menuruti apa yang engkau inginkan dan yang engkau harapkan. Tak akan ku lupakan jasa2mu dan pengorbananmu...*

To My Brother and My Sister : Mas Andi (SE) & Mbak Nining (ST), Mas Agus (SH) & Mbak Erin, Mas Dody (ST) & Mbak Yuni (SS), yang selama ini mendukung ku dan selalu mendo'akan ku.

THANK'S FOR ALL.....

Entah apa yang bisa aku perbuat untuk membalas semua jasmu dan ilmu yang telah engkau berikan kepadaku selama ini. Sampai saat ini aku hanya bisa mengucapkan terima kasih dan bersyukur yang sebesar-besarnya kepada ALLAH SWT karena telah diberi orang tua dan keluarga yang selalu memperhatikan ku dan mendo'akan ku.



BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI


Nama Mahasiswa : ALIEF CATUR WIBOWO
NIM : 00.12.051
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Energi Listrik
Judul Skripsi : KOMITMEN UNIT PEMBANGKIT
TERMAL DENGAN METODE *INTEGER-
CODED GENETIC ALGORITHM* PADA
PT. PEMBANGKITAN JAWA-BALI.

Dipertahankan Dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1)

Hari : Jum'at
Tanggal : 22 September 2006
Dengan Nilai : 81,10 (A)


Panitia Ujian Skripsi

Ketua Majelis Penguji



(Ir. Mochtar Asroni, MSME)
NIP. Y. 101 8100 036

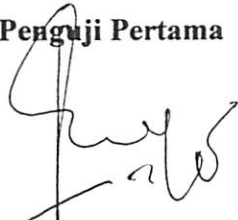
Sekretaris Majelis Penguji



(Ir. F. Yudi Limpraptono, MT)
NIP. Y. 103 9500 274


Anggota Penguji

Penguji Pertama



(Ir. Eko Nurcahyo)
NIP. 102 870 072

Penguji Kedua



(Ir. Djojo Priatmono, MT)
NIP. Y. 101 8500 107



LEMBAR BIMBINGAN SKRIPSI

1. Nama : ALIEF CATUR WIBOWO
2. NIM : 00.12.051
3. Jurusan : Teknik Elektro S-1
4. Konsentrasi : Teknik Energi Listrik
5. Judul Skripsi : KOMITMEN UNIT PEMBANGKIT
TERMAL DENGAN METODE *INTEGER-CODED GENETIC ALGORITHM* PADA
PT. PEMBANGKITAN JAWA BALI
6. Tanggal Mengajukan Skripsi : 18 Februari 2006
7. Tanggal Menyelesaikan Skripsi : 22 September 2006
8. Dosen Pembimbing : Ir. Yusuf Ismail Nakhoda, MT
10. Telah dievaluasi dengan nilai : 88,00 (A)

Malang, 22 September 2006

Mengetahui,
Ketua Jurusan Teknik Elektro S-1

Ir. F. Yudi Limpraptono, MT
NIP. Y. 1039500274

Disetujui,
Dosen Pembimbing

Ir. Yusuf Ismail Nakhoda, MT
NIP. Y. 101 880 0189

ABSTRAKSI

KOMITMEN UNIT PEMBANGKIT TERMAL DENGAN METODE *INTEGER-CODED GENETIC ALGORITHM* PADA PT. PEMBANGKITAN JAWA-BALI

(Alief Catur Wibowo, Nim. 00.12.051, Teknik Elektro S-1/Teknik Energi Listrik)
(Dosen Pembimbing : Ir. Yusuf Ismail Nakhoda, MT)

Kata kunci : Penjadwalan pembangkit, *genetic algorithm*, komitmen unit.

Kebutuhan tenaga listrik yang terus meningkat menyebabkan biaya pembangkitan tenaga listrik juga bertambah besar. Operasi pusat-pusat pembangkit didalam sistem tenaga listrik harus selalu dikoordinasikan dalam pembagian pembebanan secara optimal dan ekonomis pada setiap perubahan beban dalam interval waktu tertentu. Oleh karena itu perlu adanya penjadwalan operasi unit-unit pembangkit dalam melayani beban sistem selama periode waktu tertentu, atau dikenal dengan istilah komitmen unit.

Skripsi ini menganalisis permasalahan komitmen unit atau penjadwalan unit-unit pembangkit dalam melayani beban sistem selama periode waktu tertentu dengan menggunakan metode *Integer-Coded Genetic Algorithm*. Hasil dari analisa tersebut nantinya dapat digunakan sebagai salah satu acuan dalam operasi pembangkitan dan penyaluran daya yang ekonomis dan optimal, terutama mengenai biaya pembangkitan. Input dari program ini adalah koefisien biaya bahan bakar (*Fuel cost*), daya maksimum dan daya minimum, dan data pembebanan.

Analisa dilakukan dengan bantuan program komputer dengan menggunakan bahasa pemrograman Delphi versi 7.0 dan telah sukses dicoba pada PT. Pembangkitan Jawa-Bali yang terdiri dari 37 unit pembangkit termal dimana telah berhasil dilakukan penghematan biaya pembangkitan untuk hari Sabtu, 01 Juli 2006 dengan metode *Integer-Coded Genetic Algorithm* sebesar Rp 11.909.383.546, sedangkan total biaya PT. PJB sebesar Rp 12.361.275.322 sehingga selisih biaya sebesar Rp 451.891.776 atau 3,65%; untuk hari Minggu, 02 Juli 2006 dengan metode *Integer-Coded Genetic Algorithm* sebesar Rp 10.238.241.824, pada PT. PJB sebesar Rp 10.587.998.085 sehingga selisih biaya sebesar Rp 349.756.261 atau 3,30%; hari Rabu, 05 Juli 2006 dengan metode *Integer-Coded Genetic Algorithm* sebesar Rp 12.265.172.192, sedangkan total biaya PT. PJB sebesar Rp 12.752.542.668 sehingga selisih biaya sebesar Rp 487.370.476 atau 3,82%.

KATA PENGANTAR

Dengan mengucapkan puji syukur kehadirat Allah SWT, atas limpahan Rahmat dan Hidayah-Nya, sehingga penyusun dapat menyelesaikan skripsi ini dengan judul :

**“KOMITMEN UNIT PEMBANGKIT TERMAL DENGAN METODE
INTEGER - CODED GENETIC ALGORITHM PADA
PT. PEMBANGKITAN JAWA-BALI”**

Skripsi ini disusun sebagai salah satu persyaratan dalam menyelesaikan program studi strata satu (S-1) Jurusan Teknik Elektro Konsentrasi Energi Listrik, Fakultas Teknologi Industri, Institut Teknologi Nasional Malang.

Sebelum dan selama penyusunan skripsi ini, penyusun telah banyak mendapatkan bantuan dan bimbingan dari berbagai pihak. Untuk itu pada kesempatan ini penyusun menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Bapak Prof. Dr. Ir. Abraham Lomi, MSEE, selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. Mochtar Asroni, MSME, selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang.
3. Bapak Ir. F. Yudi Limpraptono, MT, selaku Ketua Jurusan Teknik Elektro S-1 Fakultas Teknologi Industri Institut Teknologi Nasional Malang.
4. Bapak Ir. Yusuf Ismail Nakhoda, MT, selaku Dosen Pembimbing dalam penyusunan skripsi ini.

5. Bapak dan Ibu Dosen serta staf Jurusan Teknik Elektro Energi Listrik Institut Teknologi Nasional Malang.
6. PT PLN Pembangkitan Jawa Bali Kantor Pusat Surabaya.
7. Bapak, Ibuku Tercinta, Kakak-kakakku dan seluruh keluargaku atas do'a restunya.
8. Mahasiswa Jurusan Teknik Elektro Energi Listrik Institut Teknologi Nasional Malang, yang telah membantu dalam penyusunan skripsi ini.
9. Semua pihak yang tidak dapat disebutkan satu per satu, yang telah membantu penyusunan skripsi ini.

Penyusun menyadari sepenuhnya akan segala kekurangan yang ada pada penyusunan skripsi ini, maka dengan kerendahan hati penyusun mengharapkan kritik dan saran demi penyempurnaan skripsi ini.

Akhirnya penyusun berharap semoga skripsi ini dapat bermanfaat bagi semua pihak khususnya bagi rekan-rekan Mahasiswa pada Jurusan Teknik Elektro Energi Listrik.

Malang, September 2006

penyusun

DAFTAR ISI

| | |
|---|-------|
| HALAMAN JUDUL | i |
| LEMBAR PERSETUJUAN | ii |
| ABSTRAKSI | iii |
| KATA PENGANTAR | iv |
| DAFTAR ISI | vi |
| DAFTAR GAMBAR | ix |
| DAFTAR TABEL | xi |
| DAFTAR GRAFIK | x iii |
| BAB I PENDAHULUAN | 1 |
| 1.1. Latar Belakang | 1 |
| 1.2. Rumusan Masalah | 2 |
| 1.3. Tujuan | 2 |
| 1.4. Batasan Masalah | 2 |
| 1.5. Metodologi Penelitian | 3 |
| 1.6. Sistematika Penulisan | 4 |
| 1.7. Relevansi dan kontribusi | 4 |
| | |
| BAB II LANDASAN TEORI | |
| 2.1. Sistem Tenaga Listrik | 5 |
| 2.2. Karakteristik Pembangkit Tenaga Listrik | 9 |
| 2.2.1. Karakteristik Masukan Keluaran (<i>Input-ouput Characteristic</i>) | 10 |
| 2.2.2. Karakteristik Laju Tambahan Biaya Bahan Bakar | 12 |

| | |
|---|----|
| 2.3. <i>Economic Dispatch</i> | 13 |
| 2.3.1 Fungsi Biaya Bahan Bakar | 13 |
| 2.3.2 Penyelesaian <i>Economic Dispatch</i> dengan Metode <i>Pengali Lagrange</i> | 14 |
| 2.3.3 Penyelesaian <i>Economic Dispatch</i> dengan Metode <i>Iterasi Lamda</i> | 17 |
| 2.4. Komitmen Unit..... | 17 |
| 2.4.1 Kendala Pada Unit Komitmen..... | 18 |
| 2.4.2 Biaya <i>Start</i> | 20 |
| 2.5. Fungsi Obyektif Dan Kendala..... | 21 |

BAB III *INTEGER-CODED GENETIC ALGORITHM*

| | |
|--|----|
| 3.1. Konsep Dasar <i>Genetic Algorithm</i> | 23 |
| 3.1.1 Istilah-istilah Algoritma Genetika | 25 |
| 3.1.2 Parameter Algoritma Genetika | 26 |
| 3.1.3 Mekanisme Algoritma Genetika | 29 |
| 3.1.3.1 Pengkodean atau Representasi | 29 |
| 3.1.3.2 Fungsi <i>Fitness</i> | 29 |
| 3.1.3.3 Seleksi | 30 |
| 3.1.3.4 <i>Crossover</i> | 32 |
| 3.1.3.5 <i>Mutation</i> | 33 |
| 3.1.3.6 <i>Elitism</i> | 34 |
| 3.2. Solusi <i>Integer-Coded Genetic Algorithm</i> | 35 |
| 3.2.1 Definisi Kromosom | 35 |
| 3.2.2 Perhitungan Fungsi <i>Fitness</i> | 38 |
| 3.2.3 Evolusi <i>Genetic Algorithm</i> | 38 |

| | |
|---|----|
| 3.3. Algoritma Komitmen Unit Menggunakan ICGA..... | 40 |
| 3.4. Algoritma <i>Program Sub Routine Program Fitness</i> | 41 |
| 3.5. <i>Flowchart</i> Komitmen Unit Menggunakan ICGA | 42 |
| 3.6. <i>Flowchart Program Sub Routine Program Fitness</i> | 43 |

BAB IV KOMITMEN UNIT DENGAN METODE *INTEGER-CODED GENETIC ALGORITHM*.

| | |
|---|----|
| 4.1. Program Komputer Untuk Menyelesaikan Masalah Komitmen Unit..... | 44 |
| 4.2. Uji Validasi | 44 |
| 4.2.1 Hasil Validasi Dengan Menggunakan Metode ICGA..... | 46 |
| 4.3. Data Pembangkit Termal..... | 50 |
| 4.4. Aplikasi Metode ICGA pada PT. PJB Untuk Memecahkan Permasalahan Komitmen Unit | 53 |
| 4.5. Beban Sistem..... | 54 |
| 4.6. Hasil Perhitungan Dan Analisa Data..... | 55 |
| 4.6.1. Penjadwalan Unit Pembangkit PT. PJB | 57 |
| 4.6.2. Tampilan Program Komitmen Unit Dengan Metode ICGA | 63 |
| 4.6.3. Perbandingan Hasil Perhitungan PT. PJB Dengan Metode ICGA..... | 67 |
| 4.6.4. Perbandingan Total Biaya Operasional PT. PJB Dengan ICGA | 70 |

BAB V KESIMPULAN DAN SARAN

| | |
|----------------------|----|
| 5.1 Kesimpulan | 72 |
| 5.2 Saran | 73 |

| | |
|-----------------------------|----|
| DAFTAR PUSTAKA | 74 |
|-----------------------------|----|

LAMPIRAN

DAFTAR GAMBAR

| | | |
|-------------|---|----|
| Gambar 2.1 | Elemen Pokok Sistem Distribusi | 6 |
| Gambar 2.2 | Unit Boiler – Turbin – Generator | 10 |
| Gambar 2.3 | Kurva karakteristik <i>Input-Output</i> Pembangkit Termal | 11 |
| Gambar 2.4 | Kurva Karakteristik <i>Incremental Fuel Cost Rate</i> | 12 |
| Gambar 2.5 | N unit melayani beban P_R | 14 |
| Gambar 3.1 | Jumlah Generasi | 26 |
| Gambar 3.2 | Ukuran Populasi | 27 |
| Gambar 3.3 | Pengkodean atau Representasi | 29 |
| Gambar 3.4 | <i>Roulette-wheel</i> | 31 |
| Gambar 3.5 | Ilustrasi operator dengan <i>One Point Crossover</i> | 32 |
| Gambar 3.6 | Ilustrasi operator dengan <i>Two Point Crossover</i> | 33 |
| Gambar 3.7 | Ilustrasi operator dengan <i>Uniform Crossover</i> | 33 |
| Gambar 3.8 | Ilustrasi operator mutasi | 34 |
| Gambar 3.9 | Pembentukan <i>Next Generation</i> dalam Algoritma Genetika | 35 |
| Gambar 3.10 | Perhitungan waktu dari c^{th} yang <i>ON</i> pada perputaran T_i^c | 37 |
| Gambar 3.11 | <i>Flowchart</i> Komitmen Unit Menggunakan ICGA | 42 |
| Gambar 3.12 | Flowchart Program Sub Routine Program Fitness | 43 |
| Gambar 4.1 | Tampilan Data Dengan 10 Unit pembangkit | 46 |
| Gambar 4.2 | Tampilan Parameter ICGA Untuk Uji Validasi | 46 |
| Gambar 4.3 | Status <i>On-Off</i> Untuk uji validasi | 48 |
| Gambar 4.4 | Hasil perhitungan pembebanan untuk uji validasi | 48 |
| Gambar 4.5 | Hasil Uji Validasi | 49 |
| Gambar 4.6 | Tampilan Menu Utama | 63 |

| | | |
|-------------|---------------------------------|----|
| Gambar 4.7 | Tampilan Data Secara Umum | 63 |
| Gambar 4.8 | Tampilan Data Generator | 64 |
| Gambar 4.9 | Tampilan Data Pembebanan..... | 64 |
| Gambar 4.10 | Tampilan Data PLN..... | 65 |
| Gambar 4.11 | Tampilan Paramter ICGA..... | 65 |
| Gambar 4.12 | Tampilan biaya..... | 66 |

DAFTAR TABEL

| | |
|---|----|
| Tabel 3.1 Istilah yang digunakan dalam Algoritma Genetika | 25 |
| Tabel 4.1 Data Komitmen Unit 10 Unit Pembangkit Untuk Validasi | 45 |
| Tabel 4.2 Data beban sistem untuk validasi | 45 |
| Tabel 4.3 Perbandingan Hasil Data Referensi Jurnal Dengan Data Hasil Program..... | 49 |
| Tabel 4.4 Data Unit Termal Pada PT. Pembangkitan Jawa-Bali Agustus 2002 . | 50 |
| Tabel 4.5 Data Biaya dan Parameter unit Termal Pada PT. Pembangkitan Jawa-Bali Agustus 2002 | 51 |
| Tabel 4.6 Data Beban Unit Termal PT. Pembangkitan Jawa-Bali..... | 55 |
| Tabel 4.7 Unit Pembangkitan Termal Pada PT. PJB | 56 |
| Tabel 4.8 Penjadwalan Unit Pembangkit Pada PT. PJB Sabtu, 01 Juli 2006..... | 57 |
| Tabel 4.9 Penjadwalan Unit Pembangkit Pada PT. PJB Minggu, 02 Juli 2006 | 58 |
| Tabel 4.10 Penjadwalan Unit Pembangkit Pada PT. PJB Rabu, 05 Juli 2006..... | 59 |
| Tabel 4.11 Hasil Penjadwalan Unit Pembangkit Pada PT. PJB dengan metode ICGA Sabtu, 01 Juli 2006..... | 60 |
| Tabel 4.12 Hasil Penjadwalan Unit Pembangkit Pada PT. PJB dengan metode ICGA Minggu, 02 Juli 2006..... | 61 |
| Tabel 4.13 Hasil Penjadwalan Unit Pembangkit Pada PT. PJB dengan metode ICGA Rabu, 05 Juli 2006 | 62 |

| | |
|--|-----------|
| Tabel 4.14 Perbandingan Biaya Oprasional PT. PJB dengan Metode ICGA | |
| Sabtu, 01 Juli 2006 | 67 |
| Tabel 4.14 Perbandingan Biaya Oprasional PT. PJB dengan Metode ICGA | |
| Minggu, 02 Juli 2006 | 68 |
| Tabel 4.14 Perbandingan Biaya Oprasional PT. PJB dengan Metode ICGA | |
| Rabu, 05 Juli 2006..... | 69 |

DAFTAR GRAFIK

| | |
|---|----|
| Grafik 4.1 Perbandingan Biaya Operasional PT. PJB dan Metode ICGA Pada Tanggal 01, 02 dan 05 Juli 2006 | 70 |
| Grafik 4.2 Penghematan Biaya Operasional | 71 |

BAB I PENDAHULUAN

1.1. Latar Belakang

Bagi Indonesia sebagai negara yang berkembang, pemenuhan akan kebutuhan energi listrik merupakan salah satu masalah yang perlu diatasi. Untuk itu, diperlukan perencanaan penggunaan pembangkit listrik yang ada secara efisien dan seoptimal mungkin, dengan memperkirakan kemampuan dan keandalan tiap unit pembangkit baik dalam mensuplai beban nyata maupun beban cadangan. Mengingat hal tersebut diatas, maka PT. Pembangkitan Jawa-Bali sebagai salah satu produsen tenaga listrik di P. Jawa harus dapat menjadwalkan operasi secara optimal pembangkit-pembangkit yang dimilikinya, baik pembangkit hidro (PLTA) maupun pembangkit termal (PLTU, PLTG, PLTGU).

Sistem pembangkitan tenaga listrik yang terdiri dari sejumlah unit pembangkit tersebut terhubung secara interkoneksi. Penentuan penjadwalan pembangkit adalah faktor yang sangat penting dalam sistem tenaga listrik, karena hal ini berhubungan erat dengan biaya operasional dan perencanaan operasional pembangkit.

Dalam pengoptimalan operasional unit pembangkit listrik, diperlukan metode yang akurat dan cepat dalam proses perhitungannya. Banyak metode yang digunakan untuk memecahkan masalah komitmen unit, diantaranya yaitu : *Metode Skala Prioritas, Branch and Bound, Tabu Search, Lagrangian Relaxation, Dynamic Programming* dan lain-lain. Skripsi ini akan membahas sebuah pendekatan untuk menyelesaikan masalah komitmen unit dengan menggunakan metode *Integer-Coded Genetic Algorithm*.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan sebelumnya maka muncul permasalahan yaitu bagaimana penerapan *Integer-Coded Genetic Algorithm* bisa menentukan penjadwalan unit pembangkit termal dengan biaya operasional sistem seoptimal mungkin pada PT. Pembangkitan Jawa-Bali.

Sehubungan dengan permasalahan dan latar belakang tersebut di atas maka skripsi ini di beri judul :

**“ KOMITMEN UNIT PEMBANGKIT TERMAL DENGAN
METODE *INTEGER – CODED GENETIC ALGORITHM* PADA
PT. PEMBANGKITAN JAWA BALI “**

1.3. Tujuan

Tujuan yang ingin dicapai penulis dalam skripsi ini adalah untuk menentukan penjadwalan unit pembangkit termal yang akan melayani kebutuhan beban yang berubah tiap jam dan untuk mengoptimalkan biaya bahan bakar yang seekonomis mungkin dengan metode *Integer–Coded Genetic Algorithm*.

1.4. Batasan Masalah

Dalam skripsi ini akan dilakukan analisa tentang komitmen unit menggunakan metode *Integer–Coded Genetic Algorithm* dengan mengambil sistem pembangkit tenaga listrik termal yang dimiliki oleh PT. Pembangkitan Jawa-Bali sebagai obyek utama dalam penelitian. Pembatasan masalah dibatasi sebagai berikut :

- Penjadwalan dilakukan dalam waktu satu hari (24 jam dengan rentang tiap jam).
- Tidak membahas masalah rugi-rugi transmisi.

- Pembahasan dititikberatkan pada segi ekonomis, dan hanya menyangkut optimasi biaya bahan bakar tidak membahas segi teknis.
- *Combined Cycle* pada PLTGU, setiap unit pembangkit dianggap bekerja sendiri-sendiri, untuk ST (*Steam Turbin*) pada *Combined Cycle*, diambil data parameter dari pola PLTGU CC-3.3.1 yang beroperasi.
- Tidak membahas masalah biaya cadangan berputar (*spinning reserve*), hanya memperhatikan kendala batasan cadangan berputar.
- Analisa dilakukan dengan asumsi bahwa sistem dalam operasi normal.

1.5 Metodologi Penelitian

Metodologi yang digunakan dalam pembahasan dilaksanakan dengan langkah-langkah sebagai berikut :

- Studi lapangan untuk mendapatkan data parameter unit termal yang dibutuhkan dari objek penelitian yaitu pada PT. PJB dengan berpedoman pada teori yang diperoleh dan studi kepustakaan.
- Studi kepustakaan
 - Pengumpulan referensi-referensi yang mendukung dalam penyusunan proposal ini antara lain :
 - Teori tentang Komitmen Unit.
 - Teori tentang *Integer – Coded Genetic Algorithm* (ICGA).
- Menyusun program komputer dengan bahasa pemrograman Delphi
- Membuat analisa dan evaluasi, sehingga dapat disimpulkan apakah metode yang diterapkan lebih efisien atau ekonomis dibandingkan dengan yang digunakan pada PT. Pembangkitan Jawa-Bali.

1.6. Sistematika Penulisan

Penulisan skripsi ini terdiri atas bab-bab dengan susunan sebagai berikut:

BAB I PENDAHULUAN

Terdiri dari latar belakang penulisan, rumusan masalah, tujuan skripsi, ruang lingkup pembahasan, dan sistematika penulisan.

BAB II TEORI DASAR

Sistem tenaga listrik dan karakteristik pembangkit, komitmen unit, *Economic Dispatch*, fungsi biaya bahan bakar.

BAB III TEORI KOMITMEN UNIT DENGAN METODE *INTEGER – CODED GENETIC ALGORITHM* (ICGA).

BAB IV KOMITMEN UNIT DENGAN METODE *INTEGER – CODED GENETIC ALGORITHM*.

Berisi tentang data sistem pembangkitan termal pada PT. Pembangkitan Jawa-Bali dan Simulasi program untuk pemecahan masalah komitmen unit dengan menggunakan pembebanan ekonomis *Integer – Coded Genetic Algorithm* pada PT. Pembangkitan Jawa-Bali.

BAB V PENUTUP

Berisikan tentang kesimpulan dari pembahasan skripsi dan saran.

1.7. Relevansi dan kontribusi dari metode *Integer-Coded Genetic Algorithm*.

Adapun kontribusi dari skripsi ini adalah diharapkan langkah efisiensi bisa diambil oleh PT. PJB, sehingga dapat menambah keuntungan bagi PT. PJB sebagai perusahaan penyedia energi listrik di Indonesia.

BAB II

LANDASAN TEORI

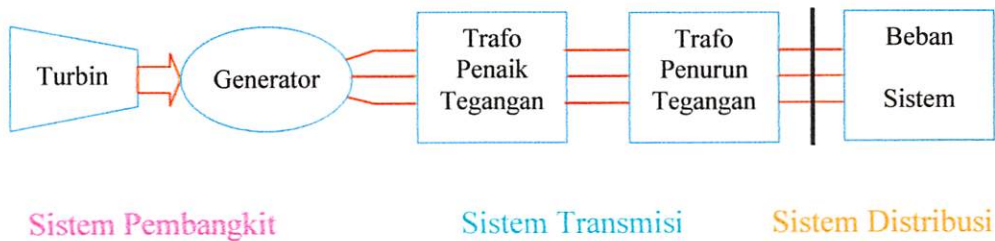
2.1 Sistem Tenaga Listrik

Untuk keperluan penyediaan tenaga listrik bagi para pelanggan, diperlukan berbagai peralatan listrik. Berbagai peralatan listrik ini dihubungkan satu sama lain sehingga mempunyai hubungan inter relasi dan secara keseluruhan membentuk suatu sistem tenaga listrik. Yang dimaksud dengan sistem tenaga listrik disini adalah sekumpulan pusat-pusat listrik dan gardu induk (pusat beban) yang satu sama lain dihubungkan oleh jaringan transmisi sehingga merupakan sebuah kesatuan yang terinterkoneksi.

Karena berbagai persoalan teknis, tenaga listrik hanya dapat dibangkitkan pada lokasi tertentu saja. Mengingat tenaga listrik atau pelanggan tenaga listrik tersebar diberbagai tempat, maka penyaluran tenaga listrik dari tempat dibangkitkan sampai ke tempat pelanggan memerlukan berbagai penanganan teknis.

Tenaga listrik dibangkitkan dari pusat-pusat pembangkit seperti : PLTA, PLTU, PLTD, PLTG dan PLTGU kemudian disalurkan melalui transmisi setelah tegangannya dinaikkan terlebih dahulu, oleh transformator penaik tegangan yang terdapat di pusat-pusat pembangkit listrik. Setelah tenaga listrik disalurkan melalui transmisi, maka sampailah tenaga listrik tersebut pada gardu induk (GI) yang kemudian tegangannya diturunkan oleh trafo penurun tegangan menjadi tegangan menengah atau rendah.

Jaringan setelah keluar dari gardu induk umumnya disebut jaringan distribusi dan jaringan antara pusat listrik dengan gardu induk disebut jaringan transmisi. Setelah disalurkan melalui jaringan distribusi primer maka tenaga listrik kemudian diturunkan tegangannya oleh gardu distribusi menjadi tegangan 380/220 volt atau 220/127 volt dan baru kemudian disalurkan ke pelanggan listrik.



Gambar 2.1^[1]
Elemen Pokok Sistem Distribusi

Dari uraian diatas dapat dimengerti bahwa besar kecilnya tegangan listrik ditentukan oleh konsumen, yaitu tergantung dari bagaimana konsumen memakai peralatan listriknya, kemudian pihak PLN akan mengimbangi kebutuhan tenaga listrik tersebut. PLN selalu menyesuaikan daya listrik yang dibangkitkan dengan permintaan tenaga listrik oleh pelanggan listrik.

Biaya operasi dari sistem tenaga listrik pada umumnya merupakan bagian biaya yang terbesar dari biaya operasi suatu sistem tenaga listrik, secara garis besar biaya operasi dari sistem tenaga listrik terdiri atas :

- Biaya pembelian tenaga listrik
- Biaya pegawai
- Biaya bahan bakar dan materi operasi
- Biaya lain-lain.

Dari keempat biaya tersebut, biaya bahan bakar pada umumnya adalah biaya yang terbesar. Untuk PLN biaya bahan bakar adalah kira-kira 60% dari biaya operasi secara keseluruhan.

Karena daya listrik yang dibangkitkan harus sama dengan tenaga listrik yang dibutuhkan oleh konsumen, maka manajemen operasi sistem tenaga listrik harus memperhatikan hal-hal sebagai berikut :

- a. Prakiraan beban.
- b. Syarat-syarat pemeliharaan peralatan.
- c. Keandalan yang diinginkan.
- d. Pengaturan dan penyaluran beban.
- e. Proses tenaga listrik yang ekonomis.

Dari kelima hal diatas masih harus sering kali dikaji ulang terhadap berbagai kendala seperti :

- a. Aliran beban dalam jaringan.
- b. Daya hubung singkat dan gangguan yang sering menimpa peralatan.
- c. Stabilitas sistem.
- d. Penyediaan suku cadang dan dana.

Dengan memperhatikan kendala-kendala diatas maka seringkali harus dilakukan pengaturan kembali terhadap rencana pemeliharaan dan alokasi beban. Makin besar sistem, maka makin banyak hal yang harus diamati dan dikoordinasi, sehingga diperlukan perencanaan, pelaksanaan, pengendalian dan evaluasi sistem yang cermat.

Dalam mengoperasikan sistem tenaga listrik ditemui berbagai persoalan. Hal ini antara lain disebabkan karena pemakaian tenaga listrik yang selalu berubah dari waktu ke waktu, biaya bahan bakar yang relatif tinggi serta kondisi alam dan lingkungan yang sering mengganggu jalannya operasi. Berbagai persoalan pokok yang dihadapi dalam mengoperasikan sistem tenaga listrik adalah :

a. Pengaturan frekwensi.

Sistem tenaga listrik harus dapat memenuhi kebutuhan akan tenaga listrik dari para konsumen dari waktu ke waktu. Untuk ini daya yang dibangkitkan dalam sistem tenaga listrik harus selalu sama dengan beban sistem, hal ini diamati melalui frekwensi sistem. Kalau daya yang dibangkitkan dalam sistem lebih kecil dari pada beban sistem maka frekwensi akan turun dan begitu pula sebaliknya, apabila daya yang dibangkitkan lebih besar dari pada beban maka frekwensi turun.

b. Pemeliharaan peralatan.

Peralatan yang beroperasi dalam sistem tenaga listrik perlu dipelihara secara periodik dan juga perlu segera diperbaiki apabila ada kerusakan.

c. Biaya operasi.

Biaya operasi khususnya biaya bahan bakar adalah biaya yang terbesar dari suatu perusahaan listrik sehingga perlu dipakai teknik-teknik optimasi untuk menekan biaya tersebut.

d. Perkembangan sistem.

Beban selalu berubah sepanjang waktu dan juga selalu berkembang seiring dengan perkembangan kegiatan masyarakat yang tidak dapat dirumuskan secara eksak, sehingga perlu diamati serta terus menerus agar pengembangan sistem yang harus dilakukan selalu dapat mengikuti perkembangan beban, sehingga tidak akan terjadi pemadaman tenaga listrik.

e. Gangguan dalam sistem.

Gangguan dalam sistem tenaga listrik adalah sesuatu yang tidak dapat sepenuhnya dihindarkan. penyebab gangguan yang paling besar adalah petir, hal ini sesuai dengan isokeraunik level yang tinggi di negara kita.

f. Tegangan dalam sistem.

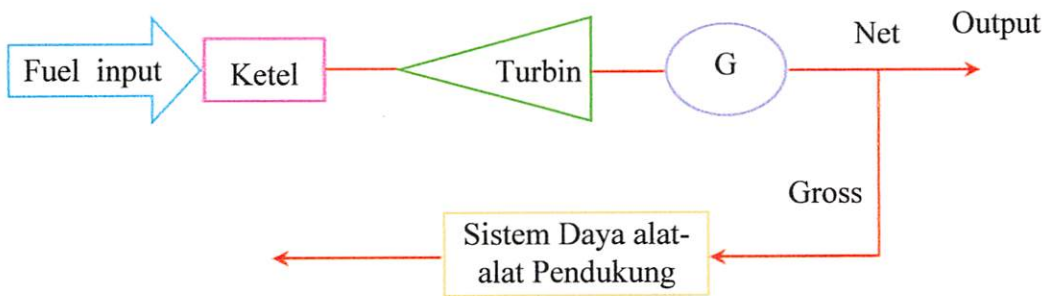
Tegangan merupakan salah satu unsur kualitas penyediaan tenaga listrik dalam sistem, oleh karena itu perlu diperhatikan dalam pengoperasian sistem.

2.2 Karakteristik Pembangkit Listrik

Hal yang paling mendasar dalam optimasi ekonomi adalah dari sebuah pembangkit listrik tenaga termal adalah dengan ditentukannya karakteristik masukan-keluaran (*input-output characteristic*) pusat listrik tersebut. Dalam mendefinisikan karakteristik masukan-keluaran, akan dibicarakan tentang *gross input* dan *net output* yang dihasilkan pusat listrik tersebut. *Gross input* pembangkit termal menyatakan jumlah keseluruhan bahan bakar yang diperlukan,

sedangkan *net output* adalah daya nyata (*real power*) yang dihasilkan pembangkit listrik (generator).

Model sebuah pembangkit termal tampak pada gambar 2.2 bagan tersebut terdiri dari sebuah ketel yang menghasilkan uap untuk menggerakkan turbin uap yang dikopel dengan sebuah generator listrik. Daya listrik dihasilkan tidak sepenuhnya disalurkan ke sistem tetapi sebagian kecil digunakan untuk mengoperasikan peralatan yang terdapat pada pusat pembangkit listrik tersebut seperti : ketel, pompa, kompresor dan sebagainya serta untuk mencatu peralatan kontrol, komunikasi, penerangan dan komputer.



Gambar 2.2^[2]
Unit Boiler – Turbin – Generator

2.2.1 Karakteristik Masukan-Keluaran

Masukan unit pembangkit termal umumnya dinyatakan sebagai banyaknya energi persatuan waktu dari bahan bakar yang diberikan ke ketel untuk menghasilkan daya listrik yang merupakan keluaran dari pusat listrik tersebut. Terdapat dua notasi umum yang digunakan, yaitu^[2] :

H dengan satuan $[Mbtu/h]$

F dengan satuan $[\$US/h]$

Dimana $F = H \times \text{\$/Btu}$, dan $\text{\$/Btu}$ menyatakan harga bahan bakar persatuan energi yang dikandung oleh bahan bakar tersebut.

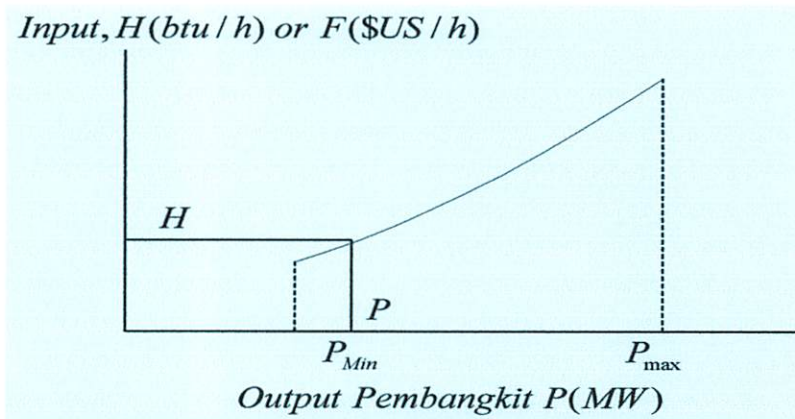
Sedangkan keluaran dari pembangkit termal adalah daya nyata yang dihasilkan oleh generator dikurangi daya nyata yang dipakai oleh pusat listrik tersebut. Notasi yang umum digunakan adalah :

P dengan satuan $[MW]$

Jika dapat disimpulkan bahwa masukan pusat listrik merupakan fungsi terhadap keluarannya, maka hubungan tersebut dapat ditulis sebagai berikut :

$$H = f(P) \quad [Mbtu/h] \quad \text{atau} \quad F = f(P) \quad [\text{\$/h}]$$

Pembahasan selanjutnya akan berpedoman atas dasar fungsi biaya bahan bakar ($F=f(P) \quad [\text{\$/h}]$), sehingga kurva dari karakteristik masukan-keluaran dari sebuah unit pembangkit termal yang telah diidealkan ditunjukkan pada gambar 2.3. masukan adalah sebuah ordinat yang berupa banyaknya energi yang diperlukan per satuan waktu $[Mbtu]$ atau juga merupakan biaya bahan bakar yang dikonsumsi per satuan waktu $[\text{\$/h}]$. sedangkan keluaran adalah daya listrik $[MW]$ yang dihasilkan blok tersebut untuk melayani beban sistem.



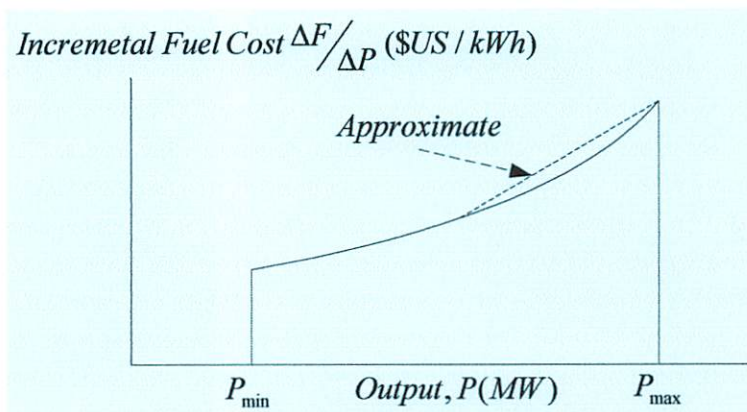
Gambar 2.3^[2]
Kurva Karakteristik *Input – Output* Pembangkit Termal

Data yang dibutuhkan untuk menggambarkan diagram fungsi karakteristik masukan – keluaran dapat diperoleh dari perhitungan pada saat perencanaan atau tes yang telah dilakukan terhadap unit pembangkit yang bersangkutan.

2.2.2 Karakteristik Laju Tambahan Biaya Bahan Bakar

Karakteristik laju tambahan biaya bahan bakar atau *incremental fuel cost characteristic* adalah turunan pertama dari fungsi biaya bahan bakar terhadap tingkat pembebanan $P[MW]$ dari pusat listrik yang bersangkutan. Fungsi ini menunjukkan besarnya kenaikan dan penurunan biaya bahan bakar untuk setiap satu satuan perubahan beban.

Secara luas, fungsi biaya bahan bakar akan digunakan untuk menentukan pembebanan ekonomis dari sebuah pembangkit listrik tenaga termal. Tampak pada gambar 2.4 adalah kurva laju tambahan biaya bahan bakar yang telah diidealkan dari sebuah pembangkit termal.



Gambar 2.4^[2]
Kurva karakteristik *Incremental Fuel Cost Rate*

2.3. *Economic Dispatch*^[2]

Yang dimaksud *Economic Dispatch* adalah pembagian pembebanan pada pembangkit-pembangkit yang ada dalam sistem, secara optimal ekonomi pada harga beban tertentu. Tujuan dari *economic dispatch* adalah untuk mendapatkan biaya harga bahan bakar semurah mungkin dalam suatu sistem pembangkit pada beban tertentu. Dengan dilakukan *Economic Dispatch* maka didapat harga bahan bakar daya yang paling murah dalam suatu sistem pembangkit. Oleh karena itu beban yang harus ditanggung oleh sistem pembangkit selalu berubah setiap periode waktu tertentu, maka perhitungan *Economic Dispatch* ini dilakukan untuk setiap harga beban tertentu.

2.3.1. Fungsi Biaya Bahan Bakar

Economic Dispatch yaitu pengoperasian secara ekonomis setiap tenaga listrik dengan pemodelan efisiensi pembangkit.

Biaya bahan bakar merupakan unsur biaya yang penting dalam operasi system pembangkit thermal. Fungsi biaya bahan bakar $F_i(P_i)$ untuk tiap unit pembangkit terhadap daya keluaran diekspresikan dalam bentuk fungsi kuadrat, yang dapat dinyatakan sebagai berikut^[2] :

$$F_i(P_{it}) = a_i P_{it}^2 + b_i P_{it} + c_i \dots\dots\dots (2.1)$$

Dimana:

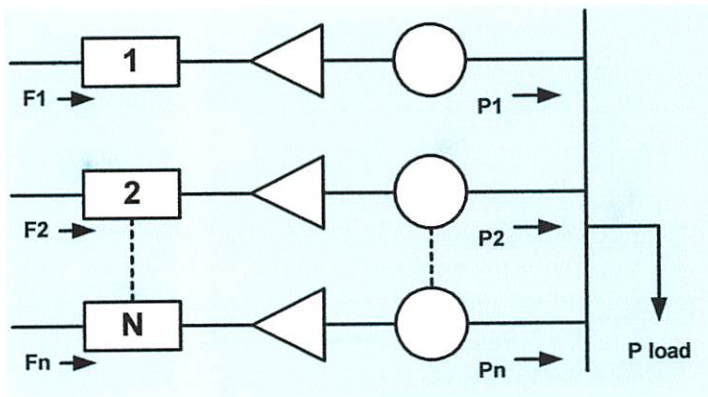
a_i, b_i, c_i = konstanta persamaan dari unit ke-i

P_{it} = daya keluaran dari unit ke-I pada jam t

2.3.2. Penyelesaian *Economic Dispatch* dengan metode *pengali Lagrange*

Sistem dengan mengabaikan rugi-rugi transmisi dapat dilihat pada gambar 2.5 system ini terdiri dari N unit generator thermal yang dihubungkan pada single busbar yang melayani beban P_R . Input dari masing-masing unit ditunjukkan oleh F_i yang mewakili biaya dari satu unit generator dan *output* dari masing-masing unit P_i adalah daya yang dihasilkan oleh satu unit generator.

Total biaya rata-rata yang ditanggung system adalah jumlah biaya dari masing-masing unit generator. Dan pembatas yang paling penting adalah jumlah dari *output* masing-masing unit generator sama dengan beban konsumen.



Gambar 2.5^[2]
 N unit melayani beban P_R

Yang menjadi permasalahan adalah meminimumkan total biaya F_T dengan memperhatikan pembatas ϕ bahwa daya yang dihasilkan generator sama dengan yang diterima beban. Secara matematika pernyataan tersebut diatas dapat dinyatakan dengan persamaan berikut^[2] :

$$F_T = F_1 + F_2 + F_3 + \dots + F_N$$

$$= \sum_{i=1}^N F_i(P_i) \dots\dots\dots (2.2)$$

$$\phi = 0 = P_D - \sum_{i=1}^N P_i \dots\dots\dots (2.3)$$

Persamaan ini adalah pembatas yang merupakan masalah dari optimasi dan ini dapat dipecahkan dengan metode kalkulus tingkat lanjut yang melibatkan fungsi *Lagrange*. Dimana fungsi *Lagrange* didapat dengan cara menambah pembatas ϕ yang telah dikalikan dengan factor *pengali Lagrange* ϕ pada fungsi F_T . Fungsi *Lagrange* dapat ditunjukkan dengan persamaan dibawah ini^[2] :

$$L = F_T + \lambda \phi \dots\dots\dots (2.4)$$

Dimana :

F_T = fungsi tujuan

λ = factor pengali

ϕ = fungsi pembatas (*constrain*)

Persamaan *Lagrange* diatas merupakan fungsi dari *output* pembangkit P_i dan factor *pengali Lagrange* λ . Keadaan dari optimasi dari fungsi tujuan F_T dapat diperoleh dengan operasi gradient dari persamaan *Lagrange* sama dengan nol.

$$\nabla L = 0 \dots\dots\dots (2.5)$$

$$\nabla F_T + \lambda \phi = 0 \dots\dots\dots (2.6)$$

$$\frac{\delta L}{\delta P} = \frac{\delta F_T}{\delta P_i} + \lambda \left[\frac{\delta P_R}{\delta P_i} - \frac{\delta P_i}{\delta P_i} \right] = 0 \text{ atau } \dots\dots\dots (2.7)$$

$$\frac{\delta F_i}{\delta P_i} + \lambda (0 - 1) = 0$$

$$\frac{\delta F_i}{\delta P_i} = \lambda \dots\dots\dots (2.8)$$

Persamaan terakhir ini menunjukkan bahwa bila digunakan biaya bahan bakar, F_T yang paling minimum maka *Incremental Cost* setiap unit generator pembangkit harus sama yaitu sebesar λ . Kondisi optimal ini tentunya dengan tetap memperhatikan pembatas yang ada yaitu bahwa daya dari setiap unit generator pembangkit harus lebih besar atau sama dengan daya *output* minimum atau sama dengan daya *output* maksimum yang diijinkan.

Dari N buah unit generator pembangkit dalam sistem tenaga yang telah dibahas dan beban sistem sebesar P_R , maka dapat diambil kesimpulan berikut^[2] :

$$\frac{\delta F_i}{\delta P_i} = \lambda \text{ ada N buah Persamaan}$$

$$P_{i_{\min}} \leq P_i \leq P_{i_{\max}} \text{ ada 2 N buah pertidaksamaan..} \dots\dots\dots (2.9)$$

$$\sum_{i=1}^N P_i = P_R \text{ ada 1 buah pembatas}$$

Dari batasan pertidaksamaan pembatasan diatas dapat diperluas menjadi :

$$\frac{\delta F_i}{\delta P_i} = \lambda \text{ untuk } P_{i_{\min}} \leq P_i \leq P_{i_{\max}}$$

$$\frac{\delta F_i}{\delta P_i} \leq \lambda \text{ untuk } P_i = P_{i_{\max}} \dots\dots\dots (2.10)$$

$$\frac{\delta F_i}{\delta P_i} \geq \lambda \text{ untuk } P_i = P_{i_{\min}} \dots\dots\dots (2.11)$$

Karena F_i hanya sebagai fungsi P_i maka $\frac{\delta F_i}{\delta P_i}$ dapat diganti dengan $\frac{dF_i}{dP_i}$

2.3.3. Penyelesaian *Economic Dispatch* dengan metode *iterasi Lamda*^[2]

Dalam metode *iterasi lamda*, kita menentukan sembarang λ . Dari λ yang telah ditentukan, kita menghitung harga *output* masing-masing pembangkit dengan menggunakan syarat optimum.

Dengan menggunakan *constraint* diperiksa apakah jumlah total dari *output* sama dengan beban sistem. Bila jumlah dari P_1 , P_2 , dan P_3 , lebih kecil dari P_R (beban system). Maka ditentukan kembali harga λ kedua yang lebih besar dari λ pertama. Bila sebaliknya maka ditentukan harga λ kedua yang lebih kecil dari λ pertama.

Dengan telah diperoleh dua hasil perhitungan diatas maka secara ekstrapolasi dapat ditentukan harga λ selanjutnya sampai dicapai harga yang dikehendaki dimana $P_1 + P_2 + P_3 = P_R$

2.4. Komitmen Unit^[2,3]

Dalam suatu siklus waktu, misalnya siklus harian yang terbagi dalam interval waktu 1 jam selama 24 jam, beban listrik dalam sistem tenaga harus selalu dikoordinasikan dalam pembagian pembebanan secara optimal pada setiap perubahan beban dalam interval waktu untuk siklus tertentu. Sehingga diperlukan penjadwalan operasi unit pembangkit dalam melayani beban sistem selama periode waktu tertentu atau dikenal dengan istilah komitmen unit.

Pada masalah komitmen unit diasumsikan ada sejumlah N unit pembangkit yang tersedia dan dioperasikan untuk memenuhi permintaan beban, sehingga dapat disimpulkan bahwa, bila tersedia N unit pembangkit yang

diharapkan mencukupi permintaan beban, maka masalahnya adalah unit-unit pembangkit mana yang seharusnya dioperasikan dengan biaya operasi seekonomis mungkin.

Dalam skripsi ini akan dititikberatkan pada pengoptimalan pusat pembangkit termal saja, Dalam pengoperasian pembangkit termal harus di capai biaya operasi dan bahan bakar yang paling minimum, untuk itu komitmen unit merupakan metode tepat untuk mengatasi permasalahan diatas guna mencari jadwal unit pembangkit yang harus beroperasi untuk periode tertentu agar dicapai biaya operasi seekonomis mungkin.

2.4.1. Kendala Pada Unit Komitmen^[2,3,4]

Dalam pengoperasian unit pembangkit untuk memenuhi kebutuhan beban terdapat berbagai kendala yang merupakan syarat pembatas (*constraint*). Kendala tersebut antara lain:

a. *Spining reserve*

Merupakan cadangan daya yang harus diperhitungkan dari unit-unit yang beroperasi, dimana apabila ada salah satu unit mengalami kegagalan operasi atau keluai dari sistem, harus ada cukup cadangan daya untuk mengganti atau menutupi berkurangnya suplay daya dalam periode waktu tertentu.

Spining reserve harus dialokasikan untuk menuruti aturan-aturan tertentu, biasanya aturan yang dipakai bahwa cadangan tersebut berupa sebuah persentase yang diberikan terhadap beban puncak yang

diperkirakan, atau bahwa cadangan tersebut harus mampu menutupi kehilangan daya dari unit yang paling besar yang dibebani penuh dalam suatu periode tertentu.

Selain itu cadangan tersebut tidak hanya harus mencukupi untuk memenuhi kegagalan pembangkit unit, tetapi cadangan-cadangan tersebut harus dialokasikan diantara unit-unit yang dapat bereaksi cepat dan unit-unit yang bereaksi lambat. Ini memungkinkan sistem kontrol pembangkitan otomatis untuk mengembalikan suplai daya yang cepat pada saat unit pembangkit keluar dari sistem.

Besar cadangan berputar tersebut harus ditentukan secara hati-hati. Sebab seringkali penentuan yang didasarkan untuk menjaga keandalan sistem berbenturan dengan biaya pengoperasian yang diusahakan seekonomis mungkin. Misalnya bila cadangan berputar kecil dan unit yang terbesar mengalami gangguan dan trip unit keluar dari sistem secara mendadak, maka cadangan berputar tidak cukup mengatasi kekurangan pembangkitan yang terjadi. Jika makin besar cadangan berputar dalam sistem maka makin handal sistem tersebut dalam menghadapi gangguan, tetapi makin besar biaya operasi terutama biaya bahan bakarnya, oleh karena itu perlu adanya suatu kesepakatan antara pemenuhan keandalan dan pengoptimalan biaya operasi.

b. Kendala unit termal

Sebuah unit pembangkit termal dapat mengalami perubahan temperatur setahap demi setahap, dan ini diterjemahkan kedalam sebuah

periode waktu dalam jam dimana dibutuhkan untuk membawa unit *on line*. Hal ini menyebabkan kendala-kendala antara lain:

1. *Minimum Up Time*

Minimum Up Time adalah interval waktu minimum dimana suatu unit yang dihidupkan (*ON*) tidak boleh dimatikan (*OFF*) kembali sebelum melewati batas waktunya (*Up Time*).

2. *Minimum Down Time*

Minimum down time adalah interval waktu minimum dimana suatu unit yang dimatikan (*OFF*) tidak boleh dihidupkan (*ON*) kembali sebelum melewati batas waktunya (*Down Time*).

2.4.2. Biaya *Start*²¹

Biaya *start* adalah biaya yang diperlukan oleh pembangkit untuk *start* dari keadaan tidak beroperasi sampai pembangkit beroperasi (terhubung pada sistem tenaga listrik). Ada dua macam biaya *start* yaitu:

- Biaya *start* pada kondisi dingin (*cold start*)

Kondisi ini terjadi pada saat pembangkit lepas dari sistem (tidak beroperasi), sedangkan temperatur boiler dibiarkan turun dari temperatur kerjanya. Sehingga pada saat beroperasi kembali perlu dilakukan pemanasan kembali.

- Biaya *start* pada kondisi panas (*hot start*)

Kondisi ini terjadi karena saat pembangkit dilepas dari sistem (tidak beroperasi), temperatur boiler tetap dijaga pada temperatur kerja.

2.5. Fungsi Obyektif Dan Kendala^[3]

Sasaran dari masalah unit komitmen adalah meminimalkan total biaya operasi dalam penjadwalan unit pembangkit. Oleh karena itu, fungsi obyektif dinyatakan sebagai jumlah dari fungsi biaya bahan bakar dan biaya *start up* dari unit pembangkit dapat dinyatakan dengan:

$$F(P_{it}, U_{it}) = \sum_{t=1}^T \sum_{i=1}^N [F_i(P_{it}) + ST_i(1 - U_{it-1})] U_{it} \dots\dots (2.12)$$

- Dimana :
- $F_i(P_{it})$ = fungsi biaya bahan bakar unit ke- i
 - ST_i = biaya start up dari unit ke- i
 - U_{it} = status *on* atau *off* dari unit ke- i pada jam t , $U_{it}=0$ ketika unit *off* dan $U_{it}=1$ ketika unit *on*
 - N = jumlah unit
 - T = jumlah jam

Sehubungan dengan minimalisasi dari total biaya operasi, terdapat berbagai *constraint* atau syarat pembatas, sebagai berikut:

a. Batasan pembebanan

$$\sum_{i=1}^N P_{it} U_{it} = P_{Dt} \text{ untuk } t=1,$$

Dimana : P_{Dt} =kebutuhan beban pada jam t

b. Batasan cadangan berputar

$$\sum_{i=1}^H U_{it} P_i^{maks} \geq P_{Dt} + R_t$$

Dimana : R_t =cadangan berputar pada jam t

c. Batasan pembangkitan

$$U_{it} P_i^{\min} \leq P_i \leq U_{it} P_i^{\max} \text{ untuk } i=1 \dots N \text{ dan } t=1 \dots T$$

d. *Minimum up time* (up_i) dan *minimum down time* ($down_i$).

$$U_{it} = 1 \text{ untuk } \sum_{x=t-up_i}^{t-1} U_{ix} < up_i, \text{ dan } U_{it} = 0 \text{ untuk } \sum_{x=t-down_i}^{t-1} (1-U_{ix}) < down_i$$

BAB III

INTEGER – CODED GENETIC ALGORITHM (ICGA)

3.1. Konsep Dasar *Genetic Algorithm*^[5]

Genetic Algorithm merupakan metode *adaptive* yang bisa digunakan untuk memecahkan suatu pencarian nilai dalam sebuah masalah optimasi. Algoritma ini didasarkan pada proses genetik yang ada dalam makhluk hidup yaitu perkembangan generasi dalam sebuah populasi yang alami, secara lambat laun mengikuti prinsip seleksi alam “siapa yang kuat, dia yang bertahan (*survive*)”. Dengan meniru proses ini, algoritma genetika dapat digunakan untuk mencari solusi permasalahan-permasalahan dalam dunia nyata.

Genetic Algorithm ditemukan oleh John Holland pada awal tahun 1970 yang dilandasi oleh sifat-sifat evolusi alam. Holland percaya bahwa ini sangat cocok digabungkan dalam sebuah algoritma komputer, menghasilkan sebuah teknik penyelesaian untuk permasalahan-permasalahan yang sulit dengan langkah alami yaitu melalui evolusi. John Holland mulai bekerja dengan algoritma yang dibentuk dari string-string biner 1 dan 0 yang disebut kromosom. Seperti halnya alam, algoritma ini menyelesaikan permasalahan-permasalahan dengan menemukan kromosom-kromosom yang baik dengan memanipulasi materi dan sifat (*gene*) kromosom-kromosom. Algoritma ini tidak mengetahui type permasalahan yang akan diselesaikan. Hanya informasi yang telah diberikan dari *evaluasi* berupa nilai *fitness* setiap kromosom dengan nilai *fitness* terbaik yang bertahan hidup dan selalu diproduksi.

Sebelum Algoritma Genetika dijalankan, maka sebuah kode yang sesuai (representasi) untuk persoalan harus dirancang. Titik solusi dalam ruang permasalahan dikodekan dalam bentuk kromosom/string yang terdiri dari komponen genetik terkecil yaitu gen. Pemakaian bilangan seperti *integer*, *floating point* dan abjad sebagai *allele* (nilai gen) memungkinkan penerapan operator genetika yaitu proses produksi (*reproduction*), pindah silang (*crossover*), mutasi (*mutation*) untuk menciptakan himpunan titik-titik solusi. Untuk memeriksa hasil optimasi, kita membutuhkan fungsi *fitness* yang menandakan gambaran hasil (*solution*) yang sudah dikodekan. Selama proses, induk harus digunakan untuk reproduksi, pindah silang dan mutasi untuk menciptakan keturunan (*offspring*). Jika Algoritma Genetika didesain dengan baik, populasi akan mengalami konvergensi dan akan mendapatkan sebuah solusi yang optimum.

Algoritma Genetika memiliki empat dasar kerja yaitu :

1. Bekerja dengan mengkodekan parameter-parameter permasalahan dan tidak bekerja secara langsung dengan parameter-parameter tersebut.
2. Mencari solusi masalah dari sejumlah populasi kandidat solusi, tidak hanya memproses satu solusi saja.
3. Hanya memperhitungkan fungsi *fitness* setiap kandidat solusi untuk mendapatkan hasil optimum global.
4. Menggunakan aturan transisi secara probabilistik bukan deterministik.

3.1.1. Istilah-Istilah Algoritma Genetika

Algoritma Genetika menggunakan mekanisme genetika yang ada pada proses alami dan sistem buatan. Istilah-istilah yang digunakan adalah gabungan dari dua disiplin ilmu, yaitu ilmu Biologi dan ilmu komputer. Mitsuo Gen dan Runwei Cheng (1997) menjelaskan istilah-istilah yang digunakan dalam Algoritma Genetika sebagai berikut :

Tabel 3.1^[5]
Istilah yang digunakan dalam Algoritma Genetika

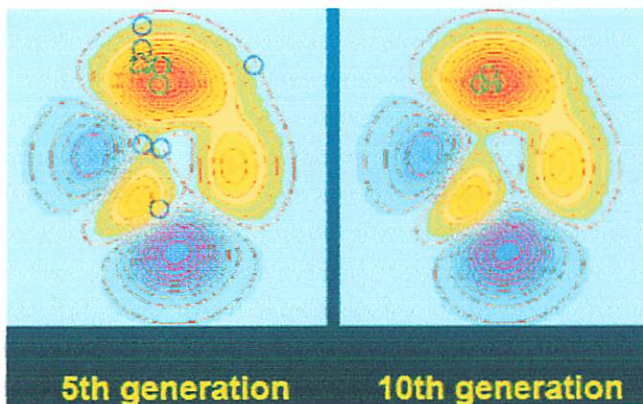
| Istilah | Keterangan |
|-----------|---|
| Kromosom | Individu berupa segmen string yang sudah ditentukan |
| Gen | Bagian dari string |
| Loci | Posisi dari gen |
| Allele | Nilai yang dimasukkan dalam gen |
| Phenotype | String yang merupakan solusi terakhir |
| Genotype | Sejumlah string hasil perkawinan yang berpotensi sebagai solusi |

3.1.2. Parameter Algoritma Genetika

Terdapat beberapa parameter yang digunakan dalam GA. Parameter tersebut digunakan untuk melihat kompleksitas dari GA. Parameter yang digunakan tersebut adalah :

- Jumlah Generasi (*MAXGEN*)

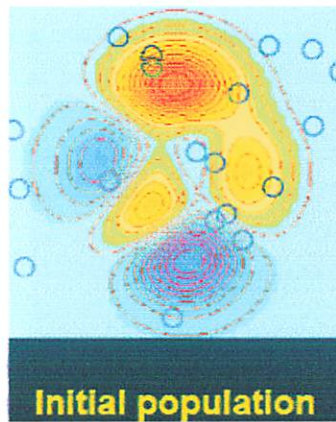
Merupakan jumlah perulangan (iterasi) dilakukannya rekombinasi dan seleksi. Jumlah generasi ini mempengaruhi kestabilan output dan lama iterasi (waktu proses Algoritma Genetika). Jumlah generasi yang besar dapat mengarahkan kearah solusi yang optimal, namun akan membutuhkan waktu yang lama. Sedangkan jika jumlah generasinya terlalu sedikit maka solusi akan terjebak pada *local optimum*.



Gambar 3.1^[6]
Jumlah Generasi

- Ukuran Populasi (*POPSIZE*)

Ukuran populasi mempengaruhi kinerja dan efektifitas dari Algoritma Genetika. Jika ukuran populasi kecil maka populasi tidak menyediakan cukup materi untuk mencakup ruang permasalahan, sehingga pada umumnya kinerja Algoritma Genetika menjadi buruk. Dalam hal ini dibutuhkan ruang yang lebih besar untuk mempersentasikan keseluruhan ruang permasalahan. Selain itu penggunaan populasi yang besar dapat mencegah terjasinya konvergensi pada wilayah *local*.



Gambar 3.2^[6]
Ukuran Populasi

- Probabilitas *Crossover* (P_c)

Probabilitas *crossover* ini digunakan untuk mengendalikan frekuensi operator *crossover*. Dalam ha ini, dalam populasi terdapat $P_c \times POPSIZE$ struktur (individu) yang melakukan pindah silang. Semakin besar nilai probabilitas *crossover* maka semakin cepat struktur baru yang diperkenalkan dalam populasi. Namun jika probabilitas crossover terlalu besar maka struktur dengan nilai fungsi

obyektif yang baik dapat hilang dengan lebih cepat dari seleksi. Sebaliknya jika probabilitas terlalu kecil akan menghalangi proses pencarian dalam proses Algoritma Genetika.

- Probabilitas Mutasi (P_m)

Mutasi digunakan untuk meningkatkan variasi populasi digunakan untuk menentukan tingkat mutasi yang terjadi, karena frekuensi terjadinya mutasi tersebut menjadi $P_m \times \text{POPSIZE} \times N$, dimana N adalah panjang struktur / gen dalam satu individu. Probabilitas mutasi yang rendah akan menyebabkan gen-gen yang berpotensi tidak dicoba. Dan sebaliknya, tingkat mutasi yang tinggi akan menyebabkan keturunan akan semakin mirip dengan induknya. Dalam Algoritma Genetika, mutasi menjalankan aturan penting yaitu :

1. Mengganti gen-gen yang hilang sama proses seleksi.
2. Menyediakan gen-gen yang tidak muncul pada saat inialisasi awal populasi.

- Panjang Kromosom.

Panjang kromosom berbeda-beda sesuai dengan model permasalahan. Titik solusi dalam ruang permasalahan dikodekan dalam bentuk kromosom/string yang terdiri dari komponen genetik terkecil yaitu gen. Pengkodean dapat memakai bilangan seperti *string biner*, *integer*, *floating point* dan *abjad*.

3.1.3. Mekanisme Algoritma Genetika^[6]

3.1.3.1. Pengkodean atau Representasi

Langkah pertama kali yang dilakukan dalam penggunaan GA adalah melakukan pengkodean atau representasi terhadap permasalahan yang akan dilakukan.

Secara umum GA dibentuk oleh serangkaian kromosom yang ditandai dengan x_i ($i = 1, 2 \dots N$). Setiap elemen dalam kromosom ini adalah variabel string yang disebut gen, berisi nilai-nilai atau allele. Variabel-variabel ini dapat dinyatakan dalam bentuk bilangan real (*floating point*).

Selanjutnya beberapa kromosom dibentuk dan berkumpul membentuk populasi. Populasi inilah populasi awal bagi GA untuk awal melakukan pencarian.



Gambar 3.3^[6]
Pengkodean Representasi

3.1.3.2. Fungsi *Fitness* (Fungsi Evaluasi)

Dalam GA, sebuah fungsi *fitness* $f(x)$ harus dirancang untuk masing-masing permasalahan yang akan diselesaikan. Dengan menggunakan kromosom tertentu, fungsi obyektif atau fungsi evaluasi akan mengevaluasi status masing-

masing kromosom. Setiap gen x_i ($i = 1, 2, \dots, N$) dipergunakan untuk menghitung $f_k(x)$ ($k = 1, 2, \dots, \text{POPSIZE}$).

Pada permulaan optimasi, biasanya nilai *fitness* masing-masing individu masih mempunyai rentang yang lebar. Seiring dengan bertambah besar generasi, beberapa kromosom mendominasi populasi dan mengakibatkan rentang nilai *fitness* semakin kecil.

3.1.3.3. Seleksi

Masalah yang paling mendasar pada proses ini adalah bagaimana proses penyeleksiannya. Menurut teori Darwin proses seleksi individu adalah : “*individu terbaik akan tetap hidup dan menghasilkan keturunan*”. Pada proses seleksi ini dapat menggunakan banyak metode seperti *rouleete wheel selection*, *rank selection*, *elitism* dan lain sebagainya.

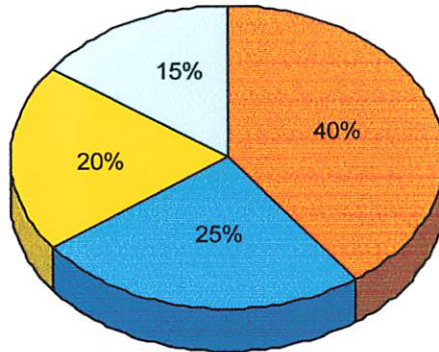
- ***Roulette Wheel Selection***

Dimana setiap individual memiliki harga *fitness* sehingga didapatkan probabilitas individual $(f(t)/\sum f(t))$ tersebut dikopikan pada populasi yang baru. Untuk individual yang memiliki probabilitas 20% untuk jumlah populasi 10 maka kemungkinan individual tersebut dapat terpilih sebanyak dua kali.

Adapun algoritma dari *roulette-wheel* adalah sebagai berikut :

1. Menjumlahkan *fitness* dari seluruh anggota populasi.
2. Membangkitkan nilai k , suatu nilai random antara 0 dan total *fitness*nya.

- Menjumlahkan *fitness* dari kromosom-kromosom dari populasi mulai 0 hingga total *fitness* lebih besar atau sama dengan nilai k lalu ambil kromosom tersebut



Gambar 3.4^[6]
roulette-wheel

- Rank Selection**

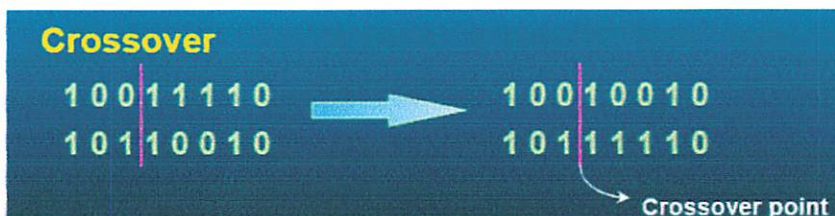
Apabila *fitness* yang dimiliki oleh suatu kromosom dalam populasi berbeda terlalu jauh dari kromosom lainnya maka hal ini dapat menjadi permasalahan. Misalnya bila kromosom terbaik mempunyai *fitness* yang menyebabkan besarnya tempat yang dimilikinya dalam *roulette wheel* sebesar 90% maka kromosom-kromosom yang lain akan mempunyai peluang yang terlalu kecil untuk diseleksi.

Rank selection pertama kali merangking populasi dan kemudian setiap kromosom diberi nilai *fitness* baru berdasarkan hasil rangking tersebut. Yang pertama akan mempunyai *fitness* 1, yang kedua akan mempunyai *fitness* 2 dan seterusnya sampai yang terakhir akan mempunyai *fitness* N. Dengan demikian semua kromosom akan mempunyai peluang untuk diseleksi..

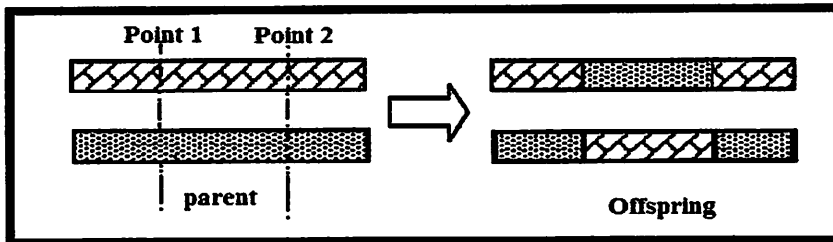
3.1.3.4. Crossover (Pindah Silang)

Fungsi dari *crossover* adalah menghasilkan kromosom anak dari kombinasi materi-materi gen dua kromosom induk. Cara kerjanya dengan membangkitkan sebuah nilai random r_k dimana $k = 1, 2, \dots, \text{POPSIZE}$. Probabilitas *crossover* (P_c) ditentukan dan digunakan untuk mengendalikan frekuensi operator *crossover*. Apabila nilai $r_k < P_c$ maka kromosom ke- k terpilih untuk mengalami *crossover*. *Crossover* yang paling sederhana adalah *one point crossover*. Posisi titik persilangan (point) ditentukan secara random pada range satu sampai panjang kromosom. Kemudian nilai *offspring* diambil dari dua *parent* tersebut dengan batas titik persilangan tersebut. Ilustrasi kerja operator ini digambarkan seperti pada gambar 3.5.

Kemudian ditingkatkan lagi dengan menggunakan *two point crossover*. Penentuan posisi titik persilangan sama seperti sama seperti *one point crossover* sebelumnya. Pemilihan secara random dilakukan 2 kali. Kemudian nilai *offspring* diambil dari dua *parent* tersebut dengan batas dua titik persilangan tersebut. Ilustrasi kerja operator ini digambarkan seperti pada gambar 3.6.

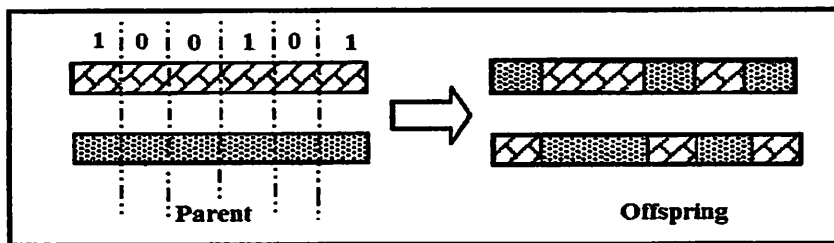


Gambar 3.5^[6]
Ilustrasi operator dengan *One Point Crossover*



Gambar 3.6^[6]
 Ilustrasi operator dengan *Two Point Crossover*

Untuk *crossover* uniform dibangkitkan suatu nilai random 0 dan 1 sepanjang jumlah kromosom untuk nilai loci. Jika nilai yang dibangkitkan mempunyai nilai 1 maka *allele parent 2* dan *offspring 2* untuk loci tersebut diambil dari *allele parent 1* dan *offspring 2* untuk loci tersebut diambil dari *allele parent 2*. Ilustrasi kerja operator ini digambarkan seperti pada gambar 3.7.



Gambar 3.7^[6]
 Ilustrasi operator *crossover* dengan *uniform crossover*

3.1.3.5. *Mutation* (Mutasi)

Operator mutasi digunakan untuk memodifikasi satu atau lebih nilai gen dalam satu individu. Cara kerjanya dengan membangkitkan sebuah nilai random r_k dimana $k = 1, 2, \dots, NVAR$ (panjang kromosom). Probabilitas mutasi (P_m) ditentukan dan digunakan untuk mengendalikan frekuensi operator mutasi. Apabila nilai random $r_k < P_m$ maka gen ke- k kromosom tersebut terpilih untuk

mengalami mutasi. Proses mutasi dalam Algoritma Genetika sama dengan Algoritma Genetika yaitu menggunakan operator *Gaussian mutation*, dimana setiap individu akan terpilih secara acak untuk mengalami mutasi berdasarkan nomor acak Gaussian untuk menciptakan individu baru (*offspring*).



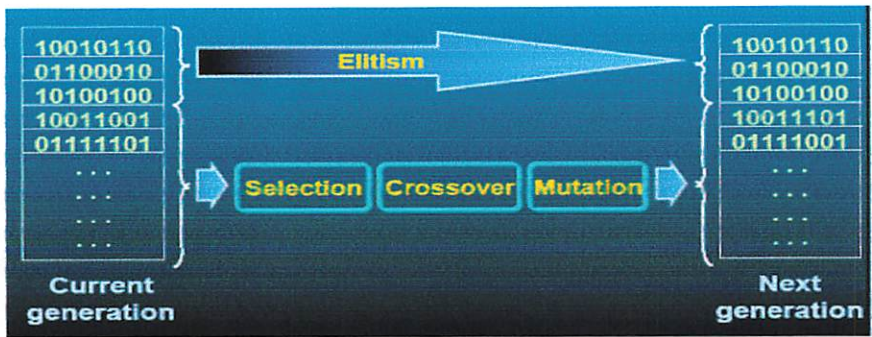
Gambar 3.8^[6]
Ilustrasi operasi mutasi

Fungsi dari operator mutasi adalah untuk menghindari agar solusi masalah yang diperoleh bukan merupakan solusi optimum lokal. Tipe dan implementasi dari operator mutasi bergantung pada jenis pengkodean dan permasalahan yang dihadapi. Seberapa sering mutasi dilakukan dinyatakan dengan suatu probabilitas mutasi, P_m . Posisi elemen pada kromosom yang akan mutasi ditentukan secara random. Mutasi dikerjakan dengan cara melakukan perubahan pada elemen tersebut.

3.1.3.6. *Elitism*

Selama membuat populasi baru dengan crossover dan mutasi, kemungkinan akan terjadi kehilangan kromosom terbaik (*best / few best*). *Elitism* adalah metode yang pertama kali meng-copy-kan kromosom terbaik (*best / few best*) kedalam populasi baru. Sisanya dikerjakan dengan cara biasa, yaitu melalui seleksi, *crossover* dan mutasi. *Elitism* dapat secara cepat meningkatkan performansidari

Algoritma Genetika karena *elitism* menghindarkan hilangnya hilangnya solusi terbaik (*best / few best*) yang telah ditemukan. Ilustrasi kerja operator ini dapat digambarkan seperti pada gambar 3.9.



Gambar 3.9^[6]
Pembentukan *Next Generation* dalam Algoritma Genetika

3.2. Solusi *Integer – Coded Genetic Algorithm*^[3]

3.2.1. Definisi Kromosom

Suatu cara untuk mendefinisikan kromosom Algoritma Genetika yang akan digunakan untuk memecahkan permasalahan komitmen unit adalah menggunakan urutan abjad biner dan menggunakan digit biner untuk status operasi *ON/OFF* dari unit yang akan beroperasi selama selang waktu tertentu. Pada metode ini, kromosom GA terdiri dari $N \cdot T$ bit. Pada solusi dari pemecahan masalah unit komitmen ini, kromosom terdiri dari beberapa bilangan integer yang merepresentasikan durasi perputaran *ON/OFF* unit generator selama proses komitmen unit. Panjang kromosom merepresentasikan sejumlah unit yang *ON/OFF* selama masa penjadwalan komitmen unit.

Jumlah beban puncak selama penjadwalan komitmen unit dan penjumlahan *minimum up* dan *down time* dari unit yang beroperasi dapat dirumuskan dengan persamaan berikut^[3] :

$$\bar{C}_i = \min \left((2.L_{pk} + 1) \left(\left[\frac{2.T}{MU_i + MD_i} \right] + 1 \right) \right) \quad i = 1, \dots, N \dots\dots\dots(2.13)$$

Kondisi awal pada parental yang luas dari persamaan (2.13) mengurangi panjang dari kromosom GA, yang didasarkan pada percobaan kondisi puncak unit yang beroperasi.

Implementasi GA memungkinkan penempatan dari panjang kromosom untuk memilih unit yang beroperasi sehingga dapat menghasilkan perputaran yang lebih baik.

Bagian dari kromosom merepresentasikan penjadwalan operasi dari suatu unit selama proses komitmen unit terdiri dari beberapa bilangan *integer* $\{ T_i^c, c = 1, \dots, \bar{C}_i \}$. Nilai penjumlahan mutlak dari tanda *integer* ini harus sama dengan penjadwalan komitmen unit yang ditunjukkan dengan persamaan berikut :

$$\sum_{c=1}^{\bar{C}_i} |T_i^c| = T \dots\dots\dots(2.14)$$

Inisial populasi dari GA ini menandakan tanda *integer* yang merepresentasikan dari penjadwalan operasi unit i. Durasi dari perputaran pertama dari unit i yang beroperasi (T_i^c) diinisialisasikan sehingga unit tersebut dapat menjalankan mode operasi (*ON/OFF*) dari perputaran terakhir pada hari sebelumnya untuk waktu yang diperlukan sehingga batasan *minimum up/down*

time-nya menjadi (T_i^0 adalah durasi dari perputaran terakhir pada penjadwalan hari sebelumnya) :

$$T_i^1 = \begin{cases} + RAND(\max(0, MU_i - T_i^0), T), & \text{if } T_i^0 > 0 \\ - RAND(\max(0, MD_i + T_i^0), T), & \text{if } T_i^0 < 0 \end{cases} \dots\dots\dots (2.15)$$

Untuk $c < \bar{C}_i$, lama waktu perputaran ke-c dari unit i yang beroperasi, T_i^c dihitung hingga mencapai nilai batasan *minimum up* dan *down time* komitmen unit. Dan, (c-1) waktu perputaran dari unit yang beroperasi.

Jika $T_i^{c-1} < 0$, perputaran c direpresentasikan dengan status ON dengan

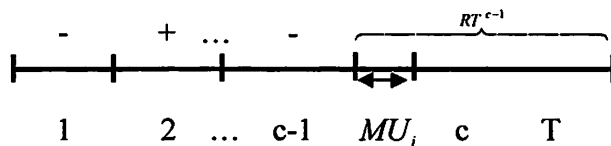
$$\text{durasi } T_i^c = \begin{cases} + RAND(MU_i, RT_i^{c-1}), & \text{jika } (RT_i^{c-1} > MU_i) \\ + RT_i^{c-1} & , \text{jika tidak} \end{cases} \dots\dots\dots (2.16)$$

Jika $T_i^{c-1} > 0$, perputaran c direpresentasikan dengan status OFF dengan

$$\text{durasi } T_i^c = \begin{cases} - RAND(MD_i, RT_i^{c-1}), & \text{jika } (RT_i^{c-1} > MD_i) \\ - RT_i^{c-1} & , \text{jika tidak} \end{cases} \dots\dots\dots(2.17)$$

dimana RT_i^{c-1} merepresentasikan waktu penjadwalan setelah penempatan dari perputaran pertama c – 1

$$RT_i^{c-1} = T - \sum_{p=1}^{c-1} |T_i^p|$$



Gambar 3.10^[3]
Perhitungan waktu dari cth yang ON pada perputaran T_i^c

Waktu perputaran terakhir dari ($c = \bar{C}_i$) dideterminasikan dengan waktu dari perputaran sebelumnya $\bar{C}_i - 1$: $T_i^{\bar{C}_i-1} = RT_i^{\bar{C}_i-1}$. Jika dipilih generasi secara acak pada waktu perputaran, periode penjadwalan ditutupi dengan perputaran operasi unit pertama $c < \bar{C}_i$, perputaran selanjutnya ($c+1, \dots, \bar{C}_i$) dipilih berdasarkan posisi terakhir dari bagian kromosom yang merepresentasikan penjadwalan operasi dari unit i dan dinotasikan dengan 0.

3.2.2. Perhitungan Fungsi *Fitness*

Fungsi *fitness* dari GA dirumuskan dengan^[3] :

$$fitness = \frac{A}{F_T + \Pi} \dots\dots\dots(2.18)$$

dimana $\Pi = \Pi_{SpRes} + \Pi_{ExcCap}$ dan $A = 10^8$ adalah sistem sendiri yang konstan. Nilai konstan A digunakan untuk menjaga supaya fungsi *fitness* mendapatkan nilai yang kecil, dan biaya operasi maksimum dari suatu sistem yang dijadwalkan pada suatu periode dirumuskan dengan^[3]:

$$\bar{F}_T = T \sum_{i=1}^N F_i(P_i^{\max}) \dots\dots\dots (2.19)$$

3.2.3. Evolusi *Genetic Algorithm*

1. *Crossover* (Pindah Silang)

Crossover adalah operator genetika yang utama. Operator ini bekerja dengan mengambil 2 individu dan memotong string kromosom mereka pada posisi yang terpilih secara random. Sebagai contoh adalah jika kita

mengambil induk yang dipresentasikan dengan 5 dimensi vektor $(a_1, b_1, c_1, d_1, e_1)$ dan $(a_2, b_2, c_2, d_2, e_2)$ kemudian dilakukan *crossing* pada posisi ketiga kromosom-kromosomnya sehingga didapat keturunan $(a_1, b_1, c_2, d_2, e_2)$ dan $(a_2, b_2, c_1, d_1, e_1)$. Operasi ini tidak selalu dilakukan pada semua individu yang ada. Individu dipilih secara random untuk dilakukan secara *crossing* dengan P_c antara 0,6 sampai dengan 1,0. Jika *crossover* tidak dilakukan, maka nilai dari induk akan diturunkan kepada keturunan.

2. Mutasi

Operator mutasi yang digunakan untuk melakukan modifikasi satu atau lebih nilai gen dalam individu yang sama. Mutasi memastikan bahwa probabilitas untuk pencarian pada daerah tertentu dalam persoalan tidak akan pernah nol dan mencegah kehilangan total materi genetika setelah pemilihan dan penghapusan. Mutasi ini bukanlah operator genetika yang utama, yang dilakukan secara random pada gen dengan kemungkinan yang kecil (P_m sekitar 0,1).

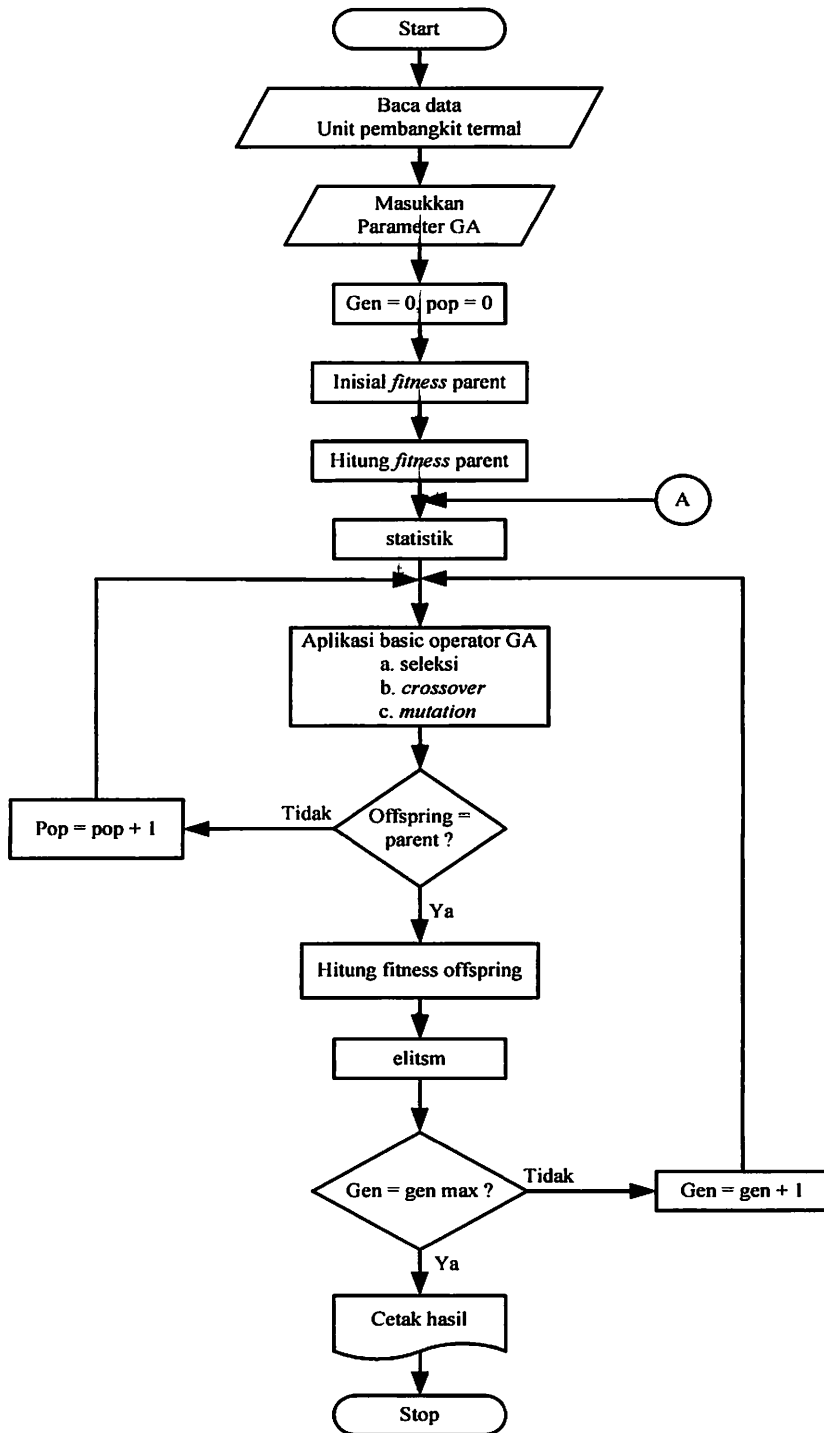
3.3. Algoritma Komitmen Unit Menggunakan ICGA

1. Memasukkan inputan data unit pembangkit termal.
2. Menentukan parameter inputan algoritma genetika yang meliputi jumlah pop, maksimum generasi, nilai kemungkinan *crossover*, nilai kemungkinan mutasi dan panjang kromosom tiap-tiap individu.
3. Generasi = 0 , populasi = 0.
4. Inisial *fitnes parent* dengan cara melakukan kombinasi pada kromosom secara *random*.
5. Melakukan perhitungan *fitness* dari kromosom tiap-tiap individu.
6. Di statistik hasil dari perhitungan *fitness*.
7. Menerapkan *basic operators* algoritma genetika yang meliputi :
 - Seleksi
 - *Crossover*
 - mutasi
8. Mengulang proses 7, sampai *offspring* sama dengan *parent*.
9. Menghitung *fitness* dari *offspring*.
10. Melakukan proses *elitism*.
11. Apakah generasi yang diinginkan sudah terpenuhi?
12. Jika “ Tidak “ maka generasi = gen + 1, kembali ke langkah 7.
13. Jika “ Ya “ maka selesai.

3.4. Algoritma *Program Sub Routine Program Fitness*

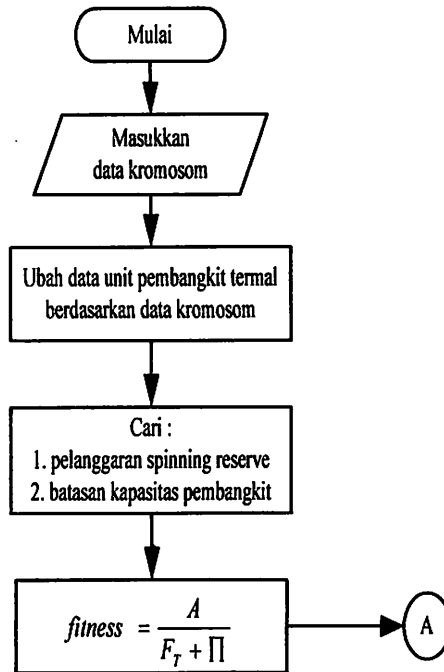
1. Memasukkan data unit pembangkit termal.
2. Memasukkan input kromosom yang dilakukan secara *random* (acak) menggunakan metode *Integer – Coded Genetic Algorithm*.
3. Melakukan proses *Encoding* atau mengubah data pembangkit termal berdasarkan data kromosom.
4. Kemudian periksa apakah ada pelanggaran cadangan berputar dan pelanggaran kapasitas pembangkitan.
5. Menghitung $fitness = \frac{A}{F_r + \Pi}$

3.5. Flowchart Komitmen Unit Menggunakan ICGA



Gambar 3.11
Flowchart Komitmen Unit Menggunakan ICGA

3.6. Flowchart Program Sub Routine Program Fitness.



Gambar 3.12
Flowchart Program Sub Routine Program Fitness

B A B IV
KOMITMEN UNIT DENGAN METODE
INTEGER – CODED GENETIC ALGORITHM

**4.1. Program Komputer Untuk Menyelesaikan Komitmen Unit Pada
PT. PJB**

Untuk pemecahan masalah Komitmen Unit digunakan bantuan program komputer. Program komputer ini sangat berguna untuk mempercepat proses perhitungan yang membutuhkan ketelitian tinggi dan sering melibatkan iterasi yang memerlukan waktu yang lama bila dikerjakan secara manual.

Program komputer ini menggunakan bahasa pemrograman Borland Delphi versi 7.0 dan diaplikasikan pada komputer pentium IV dengan prosesor 2,4 GHz, Memori 512 Mb.

4.2. Uji validasi

Sebelum dilakukan uji validasi program untuk melihat kelayakan dari program tersebut terlebih dahulu dilakukan pemilihan parameter metode ICGA yang tepat untuk menghasilkan nilai solusi yang optimal Selanjutnya untuk pengujian validasi berpedoman pada data jurnal S. A. Kazarlis, A. G. Bakirtzis, and V. Petridis, “*A genetic algorithm solution to the unit commitment problem,*” IEEE Trans. Power Syst., vol. 11, pp. 83-92, Feb 1996. Hasil pengujian dapat dilihat pada gambar 4.1. kemudian dilakukan perhitungan biaya total dengan metode *Integer – Coded Genetic Algorithm*.

Tabel 4.1^[7]
Data Komitmen Unit 10 Unit Pembangkit Untuk Validasi

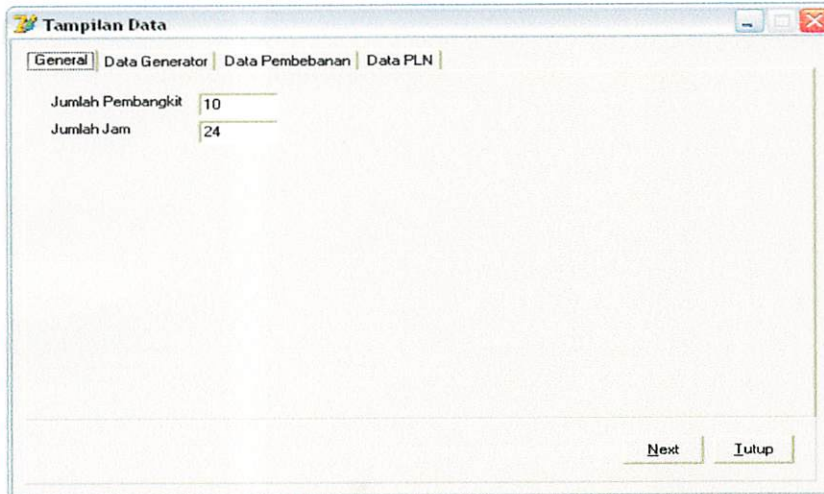
| No Unit | P_{\max} (MW) | P_{\min} (MW) | a | b | c | Min up (h) | Min down (h) | Hot start cost (\$) | cold start cost (\$) | Cold Start Hrs (h) | initial status (h) |
|---------|-----------------|-----------------|------|-------|---------|------------|--------------|---------------------|----------------------|--------------------|--------------------|
| 1 | 455 | 150 | 1000 | 16.19 | 0.00048 | 8 | 8 | 4500 | 9000 | 5 | 8 |
| 2 | 455 | 150 | 970 | 17.26 | 0.00031 | 8 | 8 | 5000 | 10000 | 5 | 8 |
| 3 | 130 | 20 | 700 | 16.60 | 0.00200 | 5 | 5 | 550 | 110 | 4 | -5 |
| 4 | 130 | 20 | 680 | 16.50 | 0.00211 | 5 | 5 | 560 | 1120 | 4 | -5 |
| 5 | 162 | 25 | 450 | 19.70 | 0.00398 | 6 | 6 | 900 | 1800 | 4 | -6 |
| 6 | 80 | 20 | 370 | 22.26 | 0.00712 | 3 | 3 | 170 | 340 | 2 | -3 |
| 7 | 85 | 20 | 480 | 27.74 | 0.00079 | 3 | 3 | 260 | 520 | 2 | -3 |
| 8 | 55 | 25 | 660 | 25.92 | 0.00413 | 1 | 1 | 30 | 60 | 0 | -1 |
| 9 | 55 | 10 | 665 | 27.27 | 0.00222 | 1 | 1 | 30 | 60 | 0 | -1 |
| 10 | 55 | 10 | 670 | 27.79 | 0.00173 | 1 | 1 | 30 | 60 | 0 | -1 |

Tabel 4.2^[7]
Data beban sistem untuk validasi

| Jam | Beban Sistem (MW) | cadangan berputar (MW) |
|-------|-------------------|------------------------|
| 01.00 | 700 | 70 |
| 02.00 | 750 | 75 |
| 03.00 | 850 | 85 |
| 04.00 | 950 | 95 |
| 05.00 | 1000 | 100 |
| 06.00 | 1100 | 110 |
| 07.00 | 1150 | 115 |
| 08.00 | 1200 | 120 |
| 09.00 | 1300 | 130 |
| 10.00 | 1400 | 140 |
| 11.00 | 1450 | 145 |
| 12.00 | 1500 | 150 |
| 13.00 | 1400 | 140 |
| 14.00 | 1300 | 130 |
| 15.00 | 1200 | 120 |
| 16.00 | 1050 | 105 |
| 17.00 | 1000 | 100 |
| 18.00 | 1100 | 110 |
| 19.00 | 1200 | 120 |
| 20.00 | 1400 | 140 |
| 21.00 | 1300 | 130 |
| 22.00 | 1100 | 110 |
| 23.00 | 900 | 90 |
| 24.00 | 800 | 80 |

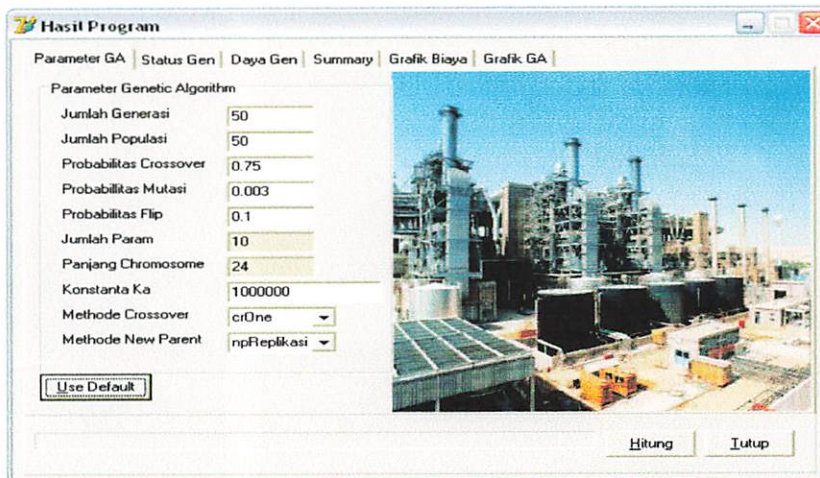
4.2.1. Hasil Validasi Dengan Menggunakan Metode *INTEGER – CODED GENETIC ALGORITHM*

1. Tampilan Data



Gambar 4.1
Tampilan Data Dengan 10 Unit pembangkit

2. Tampilan Parameter Validasi



Gambar 4.2
Tampilan Parameter *Integer – Coded Genetic Algorithm* Untuk Uji Validasi

Keterangan :

- Jumlah generasi merupakan suatu nilai yang menunjukkan jumlah pengulangan maksimum yang mungkin, jika pada nilai iterasi tersebut belum memperoleh nilai yang optimal maka dapat dirubah untuk nilai yang lebih besar untuk mendapatkan hasil yang optimal.
- Jumlah populasi merupakan jumlah individu dalam suatu kelompok yang menampilkan calon solusi dari suatu masalah.
- Jumlah probabilitas crossover menunjukkan rasio dari anak yang dihasilkan dalam setiap generasi dengan ukuran populasi.
- Jumlah probabilitas mutasi menunjukkan prosentasi jumlah total gen pada populasi yang akan mengalami mutasi.
- Jumlah probabilitas flip menunjukkan inisial chromosome parent.
- Jumlah parameter dan Panjang chromosom digunakan untuk membentuk suatu ruang pencarian yang mewakili banyaknya pembangkit dan jumlah jam.
- Konstanta K_a menunjukkan besarnya penalty yang dikenakan pada unit pembangkit yang pada suatu periode jam tertentu tidak dapat memenuhi kebutuhan beban sistem.

3. Tampilan Status *On-Off* Uji Validasi.

| Parameter GA | Status Gen | Daya Gen | Summary | Grafik Biaya | Grafik GA | | | | |
|--------------|------------|----------|---------|--------------|-----------|-------|-------|-------|-------|
| | Jam 1 | Jam 2 | Jam 3 | Jam 4 | Jam 5 | Jam 6 | Jam 7 | Jam 8 | Jam 9 |
| Unit 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Unit 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Unit 3 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| Unit 4 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Unit 5 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| Unit 6 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| Unit 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Unit 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| Unit 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Unit 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

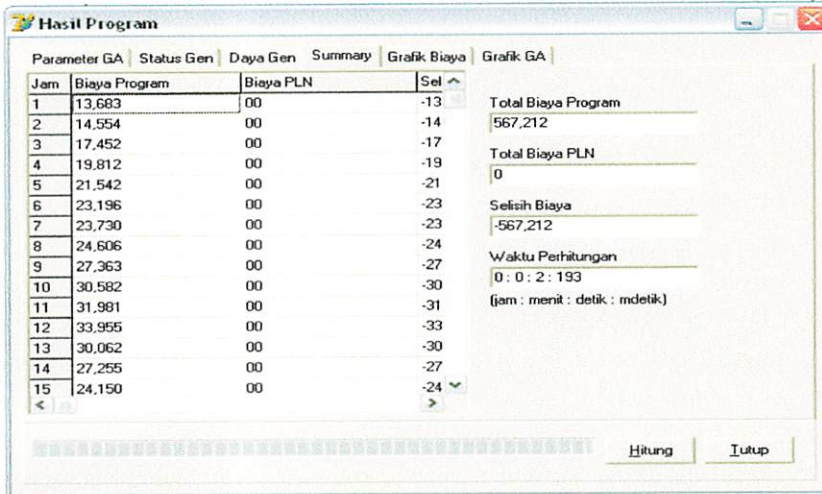
Gambar 4.3
Status *On-Off* untuk uji validasi

4. Tampilan Hasil Perhitungan Pembebanan Uji Validasi

| Parameter GA | Status Gen | Daya Gen | Summary | Grafik Biaya | Grafik GA | | | | |
|--------------|------------|----------|---------|--------------|-----------|-------|-------|-------|-------|
| | Jam 1 | Jam 2 | Jam 3 | Jam 4 | Jam 5 | Jam 6 | Jam 7 | Jam 8 | Jam 9 |
| Unit 1 | 455 | 455 | 455 | 455 | 455 | 455 | 455 | 455 | 455 |
| Unit 2 | 245 | 295 | 265 | 235 | 260 | 340 | 390 | 440 | 455 |
| Unit 3 | 0 | 0 | 0 | 130 | 130 | 130 | 130 | 130 | 130 |
| Unit 4 | 0 | 0 | 130 | 130 | 130 | 130 | 130 | 130 | 130 |
| Unit 5 | 0 | 0 | 0 | 0 | 25 | 25 | 25 | 25 | 100 |
| Unit 6 | 0 | 0 | 0 | 0 | 0 | 20 | 20 | 20 | 20 |
| Unit 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Unit 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 10 |
| Unit 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Unit 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Gambar 4.4
Hasil perhitungan pembebanan untuk uji validasi

5. Tampilan Hasil Program Uji Validasi



Gambar 4.5
Hasil Uji Validasi

Tabel 4.3
Perbandingan Hasil Data Referensi Jurnal Dengan Data Hasil Program

| Unit Generator | Data Referensi | Hasil Program |
|----------------|--------------------|--------------------|
| | Biaya Pembangkitan | Biaya Pembangkitan |
| | (\$) | (\$) |
| 10 | 566.404 | 567.212 |

Dari hasil pengujian disini dapat dilihat bahwa program tersebut layak untuk digunakan karena hasil perhitungan program tersebut mendekati hasil yang ada pada jurnal. Pada tampilan program diatas dapat dilihat bahwa hasil perhitungan menghasilkan biaya \$567.212 yang mendekati perhitungan dengan Damousis, I. G, "A Solution to the Unit-Commitment Problem Using Integer-Coded Genetic Algorithm", IEEE Trans. On Power Systems, May, 2004 yaitu \$566.404.

Sehingga tingkat kesalahan (*error*) dapat di hitung yaitu :

$$\begin{aligned}
 (\text{error}) &= \frac{\text{Jumlah biaya referensi} - \text{Jumlah biaya program}}{\text{jumlah biaya referensi}} \times 100\% \\
 &= \frac{566,404 - 567,212}{566,404} \times 100\% = 0,14\%
 \end{aligned}$$

dengan demikian *persentase error* sebesar 0,14%, dengan demikian program ini *valid* untuk digunakan.

4.3. Data Pembangkit Termal

Pembangkitan termal yang berada pada kawasan PT. Pembangkitan Jawa-Bali berjumlah 37 unit pembangkit yang terdiri dari 11 unit Pembangkit Listrik Tenaga Uap (PLTU), 4 unit Pembangkit Listrik Tenaga Gas (PLTG), dan 22 unit Pembangkit Listrik Tenaga Gas dan Uap (PLTGU) yang tersebar di seluruh Jawa dan Bali. Adapun data-data lebih lengkapnya dapat dilihat pada table 3.1 dan 3.2, untuk harga bahan bakar berdasarkan statistik PLN tahun 2002 dimana dipakai nilai tukar Rp. 9.000 per dolar Amerika.

Tabel 4.4
Data Unit Termal Pada PT. Pembangkitan Jawa-Bali
Agustus 2002

| No | Nama Pembangkit | bahan bakar | kapasitas (MW) | | Lama Waktu | | | |
|----|---------------------|-------------|----------------|-----|------------|-----|------------|-----------|
| | | | Min | Max | MUT | MDT | Cold start | Hot start |
| 1 | PLTU Paiton 1 | coal | 225 | 370 | 72 | 48 | 17 | 4 |
| 2 | PLTU Paiton 2 | coal | 225 | 370 | 72 | 48 | 17 | 4 |
| 3 | PLTGU Gresik GT 1.1 | gas | 53 | 102 | 36 | 10 | 1 | 0 |
| 4 | PLTGU Gresik GT 1.2 | gas | 53 | 102 | 36 | 10 | 1 | 0 |
| 5 | PLTGU Gresik GT 1.3 | gas | 53 | 102 | 36 | 10 | 1 | 0 |
| 6 | PLTGU Gresik ST 1.0 | gas | 250 | 480 | 36 | 10 | 3 | 1 |
| 7 | PLTGU Gresik GT 2.1 | gas | 53 | 102 | 36 | 10 | 1 | 0 |
| 8 | PLTGU Gresik GT 2.2 | gas | 53 | 102 | 36 | 10 | 1 | 0 |
| 9 | PLTGU Gresik GT 2.3 | gas | 53 | 102 | 36 | 10 | 1 | 0 |
| 10 | PLTGU Gresik ST 2.0 | gas | 250 | 480 | 36 | 10 | 3 | 1 |

| | | | | | | | | |
|----|------------------------|-----|-----|-----|----|----|----|---|
| 11 | PLTGU Gresik GT 3.1 | gas | 53 | 102 | 36 | 10 | 1 | 0 |
| 12 | PLTGU Gresik GT 3.2 | gas | 53 | 102 | 36 | 10 | 1 | 0 |
| 13 | PLTGU Gresik GT 3.3 | gas | 53 | 102 | 36 | 10 | 1 | 0 |
| 14 | PLTGU Gresik ST 3.0 | gas | 250 | 480 | 36 | 10 | 3 | 1 |
| 15 | PLTU Gresik 1 | gas | 43 | 85 | 48 | 10 | 9 | 1 |
| 16 | PLTU Gresik 2 | gas | 43 | 85 | 48 | 10 | 9 | 1 |
| 17 | PLTU Gresik 3 | gas | 90 | 175 | 48 | 10 | 9 | 2 |
| 18 | PLTU Gresik 4 | gas | 90 | 175 | 48 | 10 | 9 | 2 |
| 19 | PLTG Gresik 1 | gas | 5 | 16 | 3 | 1 | 1 | 0 |
| 20 | PLTG Gresik 2 | gas | 5 | 16 | 3 | 1 | 1 | 0 |
| 21 | PLTG Gilitimur 1 | HSD | 5 | 16 | 3 | 1 | 1 | 0 |
| 22 | PLTG Gilitimur 2 | HSD | 5 | 16 | 3 | 1 | 1 | 0 |
| 23 | PLTGU M. Karang GT 1.1 | gas | 50 | 95 | 36 | 10 | 1 | 0 |
| 24 | PLTGU M. Karang GT 1.2 | gas | 50 | 95 | 36 | 10 | 1 | 0 |
| 25 | PLTGU M. Karang GT 1.3 | gas | 50 | 95 | 36 | 10 | 1 | 0 |
| 26 | PLTGU M. Karang ST 1.0 | HSD | 300 | 465 | 36 | 10 | 3 | 1 |
| 27 | PLTGU M. Tawar GT 1.1 | HSD | 72 | 138 | 36 | 10 | 0 | 0 |
| 28 | PLTGU M. Tawar GT 1.2 | HSD | 72 | 138 | 36 | 10 | 0 | 0 |
| 29 | PLTGU M. Tawar GT 1.3 | HSD | 72 | 138 | 36 | 10 | 0 | 0 |
| 30 | PLTGU M. Tawar GT 2.1 | HSD | 72 | 138 | 36 | 10 | 0 | 0 |
| 31 | PLTGU M. Tawar GT 2.2 | HSD | 72 | 138 | 36 | 10 | 0 | 0 |
| 32 | PLTGU M. Tawar ST 1.0 | HSD | 162 | 202 | 36 | 10 | 3 | 1 |
| 33 | PLTU M. Karang 1 | MFO | 44 | 85 | 48 | 10 | 6 | 1 |
| 34 | PLTU M. Karang 2 | MFO | 44 | 85 | 48 | 10 | 6 | 1 |
| 35 | PLTU M. Karang 3 | MFO | 44 | 85 | 48 | 10 | 6 | 1 |
| 36 | PLTU M. Karang 4 | gas | 90 | 165 | 48 | 10 | 11 | 2 |
| 37 | PLTU M. Karang 5 | gas | 90 | 165 | 48 | 10 | 11 | 2 |

Keterangan: MUD = *Minimum up time*
MDT = *minimum down time*

Tabel 4.5
Data Biaya dan Parameter unit Termal Pada PT. Pembangkitan Jawa-Bali
Agustus 2002

| No | Nama Pembangkit | Biaya Start-up (Juta Rp) | | Koefisien biaya | | |
|----|---------------------|--------------------------|--------------|-----------------|------------|---------|
| | | Cold start-up | Hot start-up | A | B | C |
| 1 | PLTU Paiton 1 | 682.98 | 149.68 | 3244978 | 111712.15 | 10.2971 |
| 2 | PLTU Paiton 2 | 682.98 | 149.68 | 3244978 | 111712.15 | 10.2971 |
| 3 | PLTGU Gresik GT 1.1 | 7.82 | 0 | 5467532.4 | 217963.548 | 334.155 |
| 4 | PLTGU Gresik GT 1.2 | 7.82 | 0 | 5467532.4 | 217963.548 | 334.155 |
| 5 | PLTGU Gresik GT 1.3 | 7.82 | 0 | 5467532.4 | 217963.548 | 334.155 |
| 6 | PLTGU Gresik ST 1.0 | 57.68 | 31.46 | 10936203.3 | 72527.004 | 368.875 |
| 7 | PLTGU Gresik GT 2.1 | 7.82 | 0 | 5467532.4 | 217963.548 | 334.155 |
| 8 | PLTGU Gresik GT 2.2 | 7.82 | 0 | 5467532.4 | 217963.548 | 334.155 |
| 9 | PLTGU Gresik GT 2.3 | 7.82 | 0 | 5467532.4 | 217963.548 | 334.155 |

| | | | | | | |
|----|------------------------|--------|-------|-------------|------------|-----------|
| 10 | PLTGU Gresik ST 2.0 | 57.68 | 31.46 | 10936203.3 | 72527.004 | 368.875 |
| 11 | PLTGU Gresik GT 3.1 | 7.82 | 0 | 5467532.4 | 217963.548 | 334.155 |
| 12 | PLTGU Gresik GT 3.2 | 7.82 | 0 | 5467532.4 | 217963.548 | 334.155 |
| 13 | PLTGU Gresik GT 3.3 | 7.82 | 0 | 5467532.4 | 217963.548 | 334.155 |
| 14 | PLTGU Gresik ST 3.0 | 57.68 | 31.46 | 10936203.3 | 72527.004 | 368.875 |
| 15 | PLTU Gresik 1 | 143.74 | 40.59 | 1327126.68 | 217378.359 | 132.066 |
| 16 | PLTU Gresik 2 | 143.74 | 40.59 | 1327126.68 | 217378.359 | 132.066 |
| 17 | PLTU Gresik 3 | 229.5 | 95.52 | 5017369.5 | 169242.579 | 193.545 |
| 18 | PLTU Gresik 4 | 229.5 | 95.52 | 5017369.5 | 169242.579 | 193.545 |
| 19 | PLTG Gresik 1 | 6.13 | 0 | 352707.3 | 350680.77 | 903.969 |
| 20 | PLTG Gresik 2 | 6.13 | 0 | 352707.3 | 350680.77 | 903.969 |
| 21 | PLTG Gilitimur 1 | 6.13 | 0 | 687181.85 | 683240.965 | 1762.3893 |
| 22 | PLTG Gilitimur 2 | 6.13 | 0 | 687181.85 | 683240.965 | 1762.3893 |
| 23 | PLTGU M. Karang GT 1.1 | 7.35 | 0 | 5730795 | 202052.97 | 108.045 |
| 24 | PLTGU M. Karang GT 1.2 | 7.35 | 0 | 5730795 | 202052.97 | 108.045 |
| 25 | PLTGU M. Karang GT 1.3 | 7.35 | 0 | 5730795 | 202052.97 | 108.045 |
| 26 | PLTGU M. Karang ST 1.0 | 54.22 | 29.67 | 11560815 | 53685.135 | 460.845 |
| 27 | PLTGU M. Tawar GT 1.1 | 0 | 0 | 14706521.25 | 433337.8 | 49.4605 |
| 28 | PLTGU M. Tawar GT 1.2 | 0 | 0 | 14706521.25 | 433337.8 | 49.4605 |
| 29 | PLTGU M. Tawar GT 1.3 | 0 | 0 | 14706521.25 | 433337.8 | 49.4605 |
| 30 | PLTGU M. Tawar GT 2.1 | 0 | 0 | 14706521.25 | 433337.8 | 49.4605 |
| 31 | PLTGU M. Tawar GT 2.2 | 0 | 0 | 14706521.25 | 433337.8 | 49.4605 |
| 32 | PLTGU M. Tawar ST 1.0 | 118.08 | 64.4 | 672630 | 144191.717 | 519.1757 |
| 33 | PLTU M. Karang 1 | 122.58 | 31.08 | 2417820.7 | 473895.41 | 120.77935 |
| 34 | PLTU M. Karang 2 | 122.58 | 31.08 | 2417820.7 | 473895.41 | 120.77935 |
| 35 | PLTU M. Karang 3 | 122.58 | 31.08 | 2417820.7 | 473895.41 | 120.77935 |
| 36 | PLTU M. Karang 4 | 215.34 | 89.29 | 2949187.5 | 205217.145 | 83.79 |
| 37 | PLTU M. Karang 5 | 215.34 | 89.29 | 2949187.5 | 205217.145 | 83.79 |

Catatan:

Harga batubara : 255 Rp/Kg
 Harga MFO : 1595,5 Rp/Liter
 Harga HSD : 1595,5 Rp/Liter
 Harga gas UP. Gresik : 2.53 US\$/MMBTU
 Harga gas UP. M . Karang : 2,45 US\$/MMBTU
 Nilai Tukar : 9000 Rp/US\$

4.4. Aplikasi Metode *Integer – Coded Genetic Algorithm* Pada PT. PJB Untuk Memecahkan Permasalahan Komitmen Unit

Perhitungan dan analisa ini dilakukan pada kebutuhan daya yang ditanggung PT. Pembangkitan Jawa-Bali tanggal 01, 02 dan 05 Juli 2006. analisa data dilakukan untuk ketiga hari tersebut, karena ketiga hari tersebut mewakili karakteristik kurva beban yang berlainan.

Berdasarkan data unit yang terdapat dalam PT. Pembangkitan Jawa-Bali pada sistem tenaga pada tabel 3.1, ternyata pada saat dilakukan pengambilan data, semua unit pembangkit dalam kondisi siap beroperasi. Maka dapat disusun *input* data unit pembangkit termal yang siap operasi pada tanggal 01 Juli 2006 sampai dengan 05 Juli 2006, yaitu sebanyak 37 unit pembangkit.

Dalam data beban harian sistem yang diperoleh dari PT. Pembangkitan Jawa-Bali, terdapat data hasil perhitungan mengenai jumlah total pembangkitan, beban total dan cadangan berputar pada setiap jam dalam setiap area. Data-data ini tidak dipakai dalam skripsi ini karena data tersebut menyangkut sistem secara keseluruhan dalam suatu area. Dalam suatu area biasanya terdapat lebih dari satu perusahaan penyedia energi listrik. Misalnya pada area 4, terdapat tiga perusahaan penyedia energi listrik, yaitu: PT. Pembangkitan Jawa-Bali, PT. Indonesia Power dan perusahaan milik swasta.

Model yang digunakan dalam melakukan perhitungan optimalisasi penjadwalan PLTGU, PLTG, maupun PLTU menggunakan karakteristik tiap unit termal, meskipun PLTGU mempunyai karakteristik tiap blok yang saling tergantung antara unit gas (GT) dan unit uap (ST) atau yang sering disebut *combined cycle*. Untuk memudahkan perhitungan dilakukan dengan pendekatan

per unit termis, dimana parameter unit tiap GT kita ambil dari parameter unit pembangkit sendiri sedangkan untuk unit ST diambil dari parameter kombinasi CC.3.3.1.

PT. Pembangkitan Jawa-Bali tidak mempunyai dasar yang pasti untuk menentukan nilai dari cadangan berputar (*spinning reserve*) tiap periode jam, tetapi PT. Pembangkitan Jawa-Bali menggunakan asumsi bahwa nilai cadangan berputar diambil dari daya terpasang terbesar dari unit pembangkit yang beroperasi. Dalam hal ini PT. Pembangkitan Jawa-Bali menggunakan daya terpasang dari unit pembangkit PLTU Paiton yaitu sebesar 400 MW sebagai nilai cadangan berputar tiap periode jam.

4.5. Beban sistem

Dalam wilayah Jawa-Bali, pembangkit-pembangkit yang ada dikoordinasi oleh PT. Pembangkitan Jawa-Bali. Proses penjadwalan pembangkit dengan metode *Integer – Coded Genetic Algorithm* bertujuan untuk membuat rencana penjadwalan pembangkit dalam sistem tenaga listrik yang dapat memenuhi kebutuhan beban dengan biaya operasi yang seekonomis mungkin.

Untuk mengetahui seberapa besar efisiensi dari metode ini, maka dilakukan evaluasi dengan mengambil data unit pembangkit termal dan beban yang ditanggung oleh PT. Pembangkitan Jawa-Bali sebagai bahan perbandingan. Sedangkan kombinasi jadwal dan daya *output* pembangkit tenaga listrik dalam sistem PT. Pembangkitan Jawa Bali tanggal 01, 02, dan 05 Juli 2006, terdapat pada lampiran. Untuk beban sistem terdapat pada tabel 4.6 (beban sistem yang ditanggung oleh pembangkit termal saja).

Tabel 4.6
Data Beban Unit Termal PT. Pembangkitan Jawa-Bali

| Jam | Sabtu 01 Juli 2006 | | Minggu 02 Juli 2006 | | Rabu 05 Juli 2006 | |
|-------|-----------------------|------------------------|------------------------|------------------------|----------------------|------------------------|
| | Beban Sistem (MW) | Cadangan Berputar (MW) | Beban Sistem (MW) | Cadangan Berputar (MW) | Beban Sistem (MW) | Cadangan Berputar (MW) |
| 01.00 | 2562 | 400 | 2568 | 400 | 2663 | 400 |
| 02.00 | 2658 | 400 | 2564 | 400 | 2694 | 400 |
| 03.00 | 2424 | 400 | 2443 | 400 | 2669 | 400 |
| 04.00 | 2451 | 400 | 2444 | 400 | 2739 | 400 |
| 05.00 | 2658 | 400 | 2448 | 400 | 2746 | 400 |
| 06.00 | 2551 | 400 | 2267 | 400 | 2483 | 400 |
| 07.00 | 2449 | 400 | 2015 | 400 | 2430 | 400 |
| 08.00 | 2604 | 400 | 2031 | 400 | 2583 | 400 |
| 09.00 | 2693 | 400 | 2028 | 400 | 2823 | 400 |
| 10.00 | 2682 | 400 | 2089 | 400 | 2723 | 400 |
| 11.00 | 2604 | 400 | 2114 | 400 | 2764 | 400 |
| 12.00 | 2613 | 400 | 2039 | 400 | 2687 | 400 |
| 13.00 | 2579 | 400 | 2088 | 400 | 2733 | 400 |
| 14.00 | 2647 | 400 | 2031 | 400 | 2757 | 400 |
| 15.00 | 2657 | 400 | 2054 | 400 | 2731 | 400 |
| 16.00 | 2604 | 400 | 2075 | 400 | 2706 | 400 |
| 17.00 | 2678 | 400 | 2388 | 400 | 2858 | 400 |
| 18.00 | 3238 | 400 | 3106 | 400 | 3302 | 400 |
| 19.00 | 3202 | 400 | 3179 | 400 | 3294 | 400 |
| 20.00 | 3157 | 400 | 3081 | 400 | 3274 | 400 |
| 21.00 | 2970 | 400 | 2849 | 400 | 3119 | 400 |
| 22.00 | 2883 | 400 | 2698 | 400 | 2916 | 400 |
| 23.00 | 2789 | 400 | 2499 | 400 | 2893 | 400 |
| 24.00 | 2931 | 400 | 2401 | 400 | 2816 | 400 |

4.6. Hasil Perhitungan dan Analisa Data

Program optimasi pembebanan unit pembangkit termal pada sistem PT.PJB dengan metode *Integer – Coded Genetic Algorithm* Untuk uji validasi, terdiri dari:

1. Tahap input data dengan inisialisasi data karakteristik tiap unit dan beban tiap jam
2. Input parameter metode *Integer – Coded Genetic Algorithm*.
3. Melakukan pencarian nilai yang paling minimum dari kombinasi fungsi keanggotaan yang masing-masing untuk biaya, beban sistem dan cadangan

berputar sehingga didapatkan kombinasi penjadwalan unit-unit pembangkit yang paling optimal untuk melayani beban sistem.

Seluruh unit termal yang siap beroperasi dalam PT. PJB terdiri dari 37 unit pembangkit, yaitu sebagai berikut:

Tabel 4.7
Unit Pembangkitan Termal Pada PT. PJB

| No | Nama Pembangkit | P Max (MW) |
|----|------------------------|------------|
| 1 | PLTU Paiton 1 | 370 |
| 2 | PLTU Paiton 2 | 370 |
| 3 | PLTGU Gresik GT 1.1 | 102 |
| 4 | PLTGU Gresik GT 1.2 | 102 |
| 5 | PLTGU Gresik GT 1.3 | 102 |
| 6 | PLTGU Gresik ST 1.0 | 480 |
| 7 | PLTGU Gresik GT 2.1 | 102 |
| 8 | PLTGU Gresik GT 2.2 | 102 |
| 9 | PLTGU Gresik GT 2.3 | 102 |
| 10 | PLTGU Gresik ST 2.0 | 480 |
| 11 | PLTGU Gresik GT 3.1 | 102 |
| 12 | PLTGU Gresik GT 3.2 | 102 |
| 13 | PLTGU Gresik GT 3.3 | 102 |
| 14 | PLTGU Gresik ST 3.0 | 480 |
| 15 | PLTU Gresik 1 | 85 |
| 16 | PLTU Gresik 2 | 85 |
| 17 | PLTU Gresik 3 | 175 |
| 18 | PLTU Gresik 4 | 175 |
| 19 | PLTG Gresik 1 | 16 |
| 20 | PLTG Gresik 2 | 16 |
| 21 | PLTG Gilitimur 1 | 16 |
| 22 | PLTG Gilitimur 2 | 16 |
| 23 | PLTGU M. Karang GT 1.1 | 95 |
| 24 | PLTGU M. Karang GT 1.2 | 95 |
| 25 | PLTGU M. Karang GT 1.3 | 95 |
| 26 | PLTGU M. Karang ST 1.0 | 465 |
| 27 | PLTGU M. Tawar GT 1.1 | 138 |
| 28 | PLTGU M. Tawar GT 1.2 | 138 |
| 29 | PLTGU M. Tawar GT 1.3 | 138 |
| 30 | PLTGU M. Tawar GT 2.1 | 138 |
| 31 | PLTGU M. Tawar GT 2.2 | 138 |
| 32 | PLTGU M. Tawar ST 1.0 | 202 |
| 33 | PLTU M. Karang 1 | 85 |
| 34 | PLTU M. Karang 2 | 85 |
| 35 | PLTU M. Karang 3 | 85 |
| 36 | PLTU M. Karang 4 | 165 |
| 37 | PLTU M. Karang 5 | 165 |

Dan penjadwalan dari seluruh unit pembangkit dipergunakan sebagai bahan perbandingan dengan hasil penjadwalan metode *Integer – Coded Genetic Algorithm*. Adapun penjadwalan pada PT. PJB sebagai berikut :

4.6.1. Penjadwalan Unit Pembangkit PT. PJB

Tabel 4.8
 Penjadwalan Unit Pembangkit Pada PT. PJB
 Sabtu, 01 Juli 2006

| Jam | Status on/off unit Pembangkit | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | Beban System (MW) | | |
|-----|-------------------------------|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------------------|------|------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | | | |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 2562 | | |
| 2 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 2658 | |
| 3 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 2424 | |
| 4 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 2451 | |
| 5 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 2658 | |
| 6 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 2551 | |
| 7 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 2449 | |
| 8 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 2604 | |
| 9 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 2693 | |
| 10 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 2682 | |
| 11 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 2604 | |
| 12 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 2613 | |
| 13 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 2579 | |
| 14 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 2647 | |
| 15 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 2657 | |
| 16 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 2604 | |
| 17 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 2678 |
| 18 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 3238 |
| 19 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 3202 |
| 20 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 3157 |
| 21 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 2970 |
| 22 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 2883 |
| 23 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 2789 |
| 24 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 2931 |

Keterangan : 1 = ON dan 0 = OFF

Tabel 4.9
 Penjadwalan Unit Pembangkit Termal Pada PT. PJB
 Minggu, 02 Juli 2006

| m | Status on/off Unit Pembangkit | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | Beban System (MW) | |
|---|-------------------------------|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------------------|------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | | |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 2568 | |
| 2 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 2564 | |
| 3 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 2443 | |
| 4 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 2444 | |
| 5 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 2448 | |
| 6 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 2267 | |
| 7 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 2015 | |
| 8 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 2031 | |
| 9 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 2028 | |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 2089 | |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 2114 | |
| 2 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 2039 |
| 3 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 2088 |
| 4 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 2031 |
| 5 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 2054 |
| 6 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 2075 |
| 7 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 2388 |
| 8 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 3106 |
| 9 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 3179 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 3081 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 2849 |
| 2 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 2698 |
| 3 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 2499 |
| 4 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 2401 |

Keterangan : 1 = ON dan 0 = OFF

Tabel 4.10
 Penjadwalan Unit Pembangkit Termal Pada PT. PJB
 Rabu, 05 Juli 2006

| m | Status on/off Unit Pembangkit | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | Beban System | | | | | |
|---|-------------------------------|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------------|------|---|------|------|------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | (MW) | | | | | |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 2663 | | | | |
| 2 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 2694 | | |
| 3 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 2669 | | |
| 4 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 2739 | | |
| 5 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 2746 | |
| 6 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 2483 | |
| 7 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 2430 | |
| 8 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 2583 | |
| 9 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 2823 | |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 2723 | |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 2764 | |
| 2 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 2687 |
| 3 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 2733 | |
| 4 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 2757 |
| 5 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 2731 |
| 6 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 2706 |
| 7 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 2858 |
| 8 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 3302 |
| 9 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 3294 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 3274 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 3119 |
| 2 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 2916 |
| 3 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 2893 |
| 4 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 2816 |

Keterangan : 1 = ON dan 0 = OFF

Mengenai hasil penjadwalan setelah optimasi unit pembangkit termal dengan metode *integer – coded genetic algorithm* adalah sebagai berikut:

Tabel 4.11
 Penjadwalan Unit Pembangkit Pada PT. PJB
 Dengan Metode *Integer – Coded Genetic Algorithm*
 Sabtu, 01 Juli 2006

| Jam | Status on/off Unit Pembangkit | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | Beban System | |
|-----|-------------------------------|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------------|------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | (MW) | |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2562 | |
| 2 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2658 |
| 3 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2424 |
| 4 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2451 |
| 5 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2658 |
| 6 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2551 |
| 7 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2449 |
| 8 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2604 |
| 9 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2693 |
| 10 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2682 |
| 11 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2604 |
| 12 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2613 |
| 13 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2579 |
| 14 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2647 |
| 15 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2657 |
| 16 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2604 |
| 17 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2678 |
| 18 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 3238 |
| 19 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 3202 |
| 20 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 3157 |
| 21 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2970 |
| 22 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2883 |
| 23 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2789 |
| 24 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2931 |

Keterangan : 1 = ON dan 0 = OFF

Tabel 4.12
 Penjadwalan Unit Pembangkit Pada PT. PJB
 Dengan Metode *Integer – Coded Genetic Algorithm*
 Minggu, 02 Juli 2006

| m | Status on/off Unit Pembangkit | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | Beban System | | | | | | |
|---|-------------------------------|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------------|---|---|---|------|------|------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | (MW) | | | | | | |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 2568 | | |
| 2 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2564 | |
| 3 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2443 | |
| 4 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2444 | |
| 5 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2448 | |
| 6 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2267 | |
| 7 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2015 |
| 8 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2031 | |
| 9 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2028 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2089 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2114 |
| 2 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2039 |
| 3 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2088 |
| 4 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2031 |
| 5 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2054 |
| 6 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2075 |
| 7 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2388 |
| 8 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 3106 |
| 9 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 3179 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 3081 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2849 |
| 2 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2698 |
| 3 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2499 |
| 4 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2401 |

Keterangan : 1 = ON dan 0 = OFF

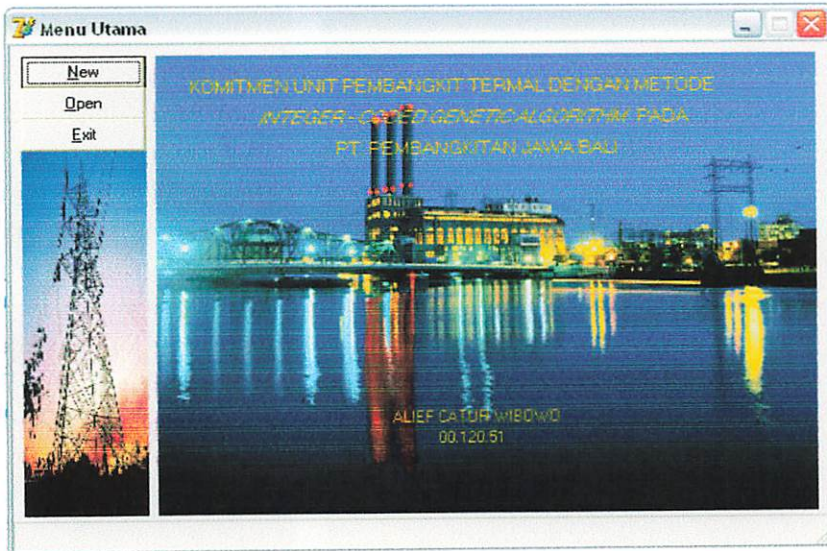
Tabel 4.13
 Penjadwalan Unit Pembangkit Pada PT. PJB
 Dengan Metode *Integer – Coded Genetic Algorithm*
 Rabu, 05 Juli 2006

| Jam | Status on/off Unit Pembangkit | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | Beban System |
|-----|-------------------------------|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | (MW) |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2663 |
| 2 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2694 |
| 3 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2669 |
| 4 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2739 |
| 5 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2746 |
| 6 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2483 |
| 7 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2430 |
| 8 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2583 |
| 9 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2823 |
| 10 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2723 |
| 11 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2764 |
| 12 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2687 |
| 13 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2733 |
| 14 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2757 |
| 15 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2731 |
| 16 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2706 |
| 17 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2858 |
| 18 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 3302 |
| 19 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 3294 |
| 20 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 3274 |
| 21 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 3119 |
| 22 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2916 |
| 23 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2893 |
| 24 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2816 |

Keterangan : 1 = ON dan 0 = OFF

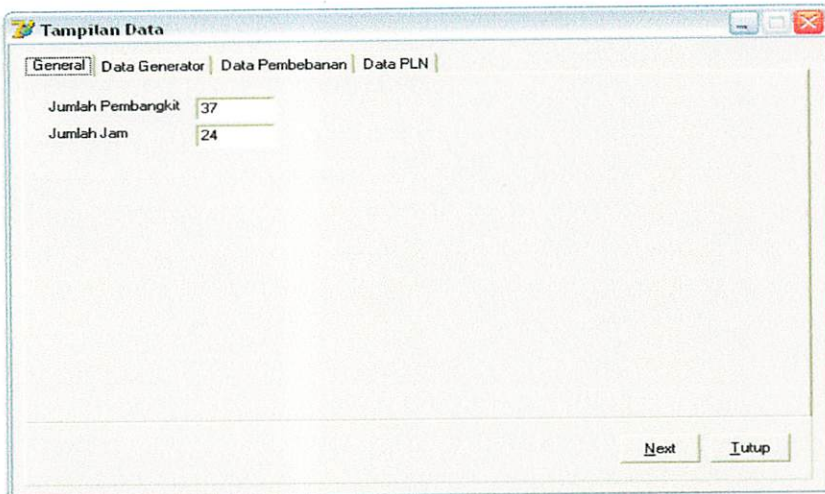
4.6.2. Tampilan Program Komitmen Unit Dengan Metode *Integer – Coded Genetic Algorithm*.

1. Tampilan Menu Utama



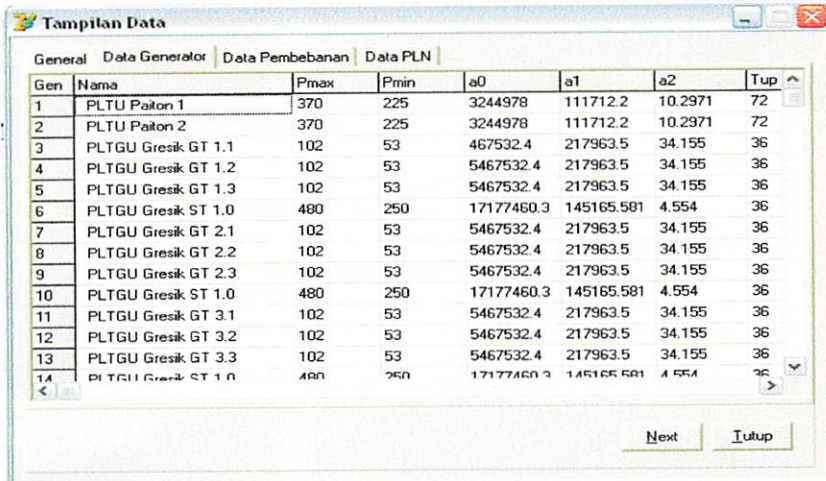
Gambar 4.6
Tampilan Menu Utama

2. Tekan tombol *Open File*, kemudian akan muncul tampilan data sebagai berikut



Gambar 4.7
Tampilan Data Secara Umum

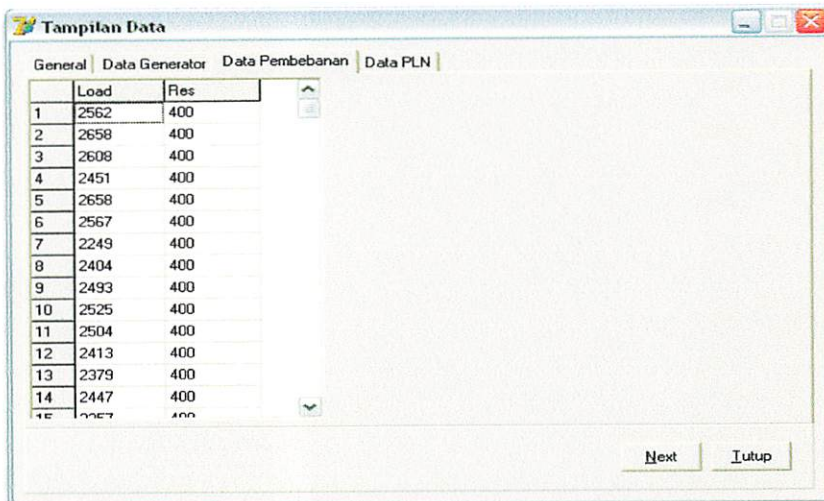
3. Tekan tombol Data Generator untuk menampilkan unit – unit pembangkit



| Gen | Nama | Pmax | Pmin | a0 | a1 | a2 | Tup |
|-----|---------------------|------|------|------------|------------|---------|-----|
| 1 | PLTU Paton 1 | 370 | 225 | 3244978 | 111712.2 | 10.2971 | 72 |
| 2 | PLTU Paton 2 | 370 | 225 | 3244978 | 111712.2 | 10.2971 | 72 |
| 3 | PLTGU Gresik GT 1.1 | 102 | 53 | 5467532.4 | 217963.5 | 34.155 | 36 |
| 4 | PLTGU Gresik GT 1.2 | 102 | 53 | 5467532.4 | 217963.5 | 34.155 | 36 |
| 5 | PLTGU Gresik GT 1.3 | 102 | 53 | 5467532.4 | 217963.5 | 34.155 | 36 |
| 6 | PLTGU Gresik ST 1.0 | 480 | 250 | 17177460.3 | 145165.581 | 4.554 | 36 |
| 7 | PLTGU Gresik GT 2.1 | 102 | 53 | 5467532.4 | 217963.5 | 34.155 | 36 |
| 8 | PLTGU Gresik GT 2.2 | 102 | 53 | 5467532.4 | 217963.5 | 34.155 | 36 |
| 9 | PLTGU Gresik GT 2.3 | 102 | 53 | 5467532.4 | 217963.5 | 34.155 | 36 |
| 10 | PLTGU Gresik ST 1.0 | 480 | 250 | 17177460.3 | 145165.581 | 4.554 | 36 |
| 11 | PLTGU Gresik GT 3.1 | 102 | 53 | 5467532.4 | 217963.5 | 34.155 | 36 |
| 12 | PLTGU Gresik GT 3.2 | 102 | 53 | 5467532.4 | 217963.5 | 34.155 | 36 |
| 13 | PLTGU Gresik GT 3.3 | 102 | 53 | 5467532.4 | 217963.5 | 34.155 | 36 |
| 14 | PLTGU Gresik ST 1.0 | 480 | 250 | 17177460.3 | 145165.581 | 4.554 | 36 |

Gambar 4.8
Tampilan Data Generator

4. Tekan tombol Data Pembebanan untuk menampilkan beban sistem dan cadangan berputar



| | Load | Res |
|----|------|-----|
| 1 | 2562 | 400 |
| 2 | 2658 | 400 |
| 3 | 2608 | 400 |
| 4 | 2451 | 400 |
| 5 | 2658 | 400 |
| 6 | 2567 | 400 |
| 7 | 2249 | 400 |
| 8 | 2404 | 400 |
| 9 | 2493 | 400 |
| 10 | 2525 | 400 |
| 11 | 2504 | 400 |
| 12 | 2413 | 400 |
| 13 | 2379 | 400 |
| 14 | 2447 | 400 |
| 15 | 2257 | 400 |


Gambar 4.9
Tampilan Data Pembebanan

5. Tekan tombol Data PLN untuk mengetahui daya yang dioperasikan oleh tiap unit pembangkit termal selama 24 jam.

| General | Data Generator | Data Pembebanan | Data PLN | | | | | |
|---------|----------------|-----------------|----------|-------|-------|-------|-------|----|
| | Jam 1 | Jam 2 | Jam 3 | Jam 4 | Jam 5 | Jam 6 | Jam 7 | Je |
| Gen 1 | 360 | 360 | 360 | 360 | 360 | 350 | 300 | 30 |
| Gen 2 | 360 | 360 | 360 | 360 | 360 | 300 | 250 | 25 |
| Gen 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Gen 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Gen 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Gen 6 | 400 | 325 | 300 | 300 | 400 | 325 | 250 | 25 |
| Gen 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Gen 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Gen 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Gen 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Gen 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Gen 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Gen 13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Gen 14 | 350 | 300 | 300 | 300 | 400 | 325 | 300 | 30 |

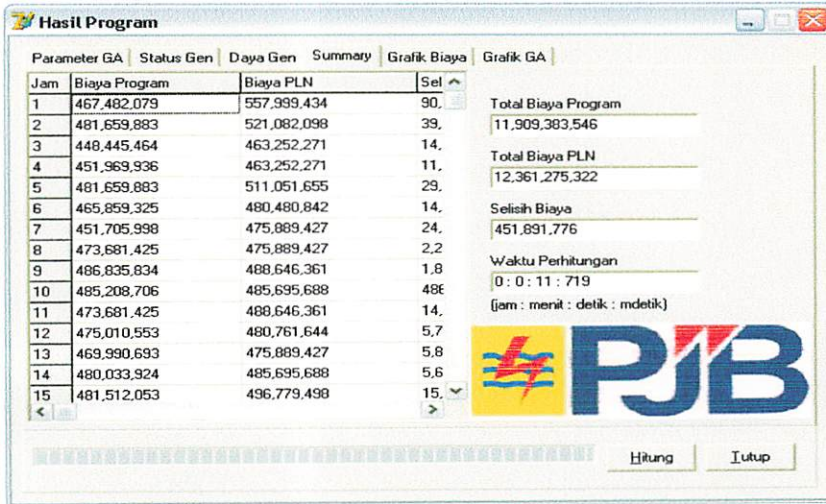
Gambar 4.10
Tampilan Data PLN

6. Kemudian tekan menu “ Next “, dan akan terlihat tampilan parameter ICGA, kemudian tekan “ Use Default “ untuk memasukkan nilai dari parameter ICGA.

| Parameter GA | Status Gen | Daya Gen | Summary | Grafik: Biaya | Grafik: GA |
|--|-------------|----------|---------|---------------|------------|
| Parameter Genetic Algorithm | | | | | |
| Jumlah Generasi | 50 | | | | |
| Jumlah Populasi | 50 | | | | |
| Probabilitas Crossover | 0.75 | | | | |
| Probabilitas Mutasi | 0.003 | | | | |
| Probabilitas Flip | 0.1 | | | | |
| Jumlah Param | 37 | | | | |
| Panjang Chromosome | 24 | | | | |
| Konstanta Ka | 1000000 | | | | |
| Methode Crossover | crOne | | | | |
| Methode New Parent | npReplikasi | | | | |
| <input type="button" value="Use Default"/> | | | | | |
|  | | | | | |
| <input type="button" value="Hitung"/> <input type="button" value="Tutup"/> | | | | | |

Gambar 4.11
Tampilan Parameter ICGA

7. Kemudian tekan tombol “ Hitung “ untuk menghitung biaya dari unit pada jam tertentu.



The screenshot shows a software window titled "Hasil Program" with several tabs: "Parameter GA", "Status Gen", "Daya Gen", "Summary", "Grafik Biaya", and "Grafik GA". The "Summary" tab is active, displaying a table with four columns: "Jam", "Biaya Program", "Biaya PLN", and "Sel". The table contains 15 rows of data. To the right of the table is a summary panel with the following information:

- Total Biaya Program: 11,909,383,546
- Total Biaya PLN: 12,361,275,322
- Selish Biaya: 451,891,776
- Waktu Perhitungan: 0 : 0 : 11 : 719 (jam : menit : detik : mdetik)

At the bottom of the window, there is a "Hitung" button and an "Iutup" button. The PJTB logo is visible in the bottom right corner of the window.

| Jam | Biaya Program | Biaya PLN | Sel |
|-----|---------------|-------------|-----|
| 1 | 467,482,079 | 557,999,434 | 90, |
| 2 | 481,659,883 | 521,082,098 | 39, |
| 3 | 448,445,464 | 463,252,271 | 14, |
| 4 | 451,969,936 | 463,252,271 | 11, |
| 5 | 481,659,883 | 511,051,655 | 29, |
| 6 | 465,859,325 | 480,480,842 | 14, |
| 7 | 451,705,998 | 475,889,427 | 24, |
| 8 | 473,681,425 | 475,889,427 | 2,2 |
| 9 | 486,835,834 | 488,646,361 | 1,8 |
| 10 | 485,208,706 | 485,695,688 | 48, |
| 11 | 473,681,425 | 488,646,361 | 14, |
| 12 | 475,010,553 | 480,761,644 | 5,7 |
| 13 | 469,990,693 | 475,889,427 | 5,8 |
| 14 | 480,033,924 | 485,695,688 | 5,6 |
| 15 | 481,512,053 | 496,779,498 | 15, |

Gambar 4.12
Tampilan biaya

4.6.3. Perbandingan Hasil Perhitungan PT. PLN PJB dengan Metode *Integer*

– *Coded Genetic Algorithm.*

Tabel 4.14
Perbandingan Biaya Operasional PT. PJB dengan Metode ICGA
Sabtu, 01 Juli 2006

| Jam | PT. PJB | ICGA |
|-----|-------------|-------------|
| | (Rp) | (Rp) |
| 1 | 557.999.434 | 467.482.079 |
| 2 | 521.082.098 | 481.659.883 |
| 3 | 463.252.271 | 448.445.464 |
| 4 | 463.252.271 | 451.969.936 |
| 5 | 511.051.655 | 481.659.883 |
| 6 | 480.480.842 | 465.859.325 |
| 7 | 475.889.427 | 451.705.998 |
| 8 | 475.889.427 | 473.681.425 |
| 9 | 488.646.361 | 486.835.834 |
| 10 | 485.695.688 | 485.208.706 |
| 11 | 488.646.361 | 473.681.425 |
| 12 | 480.761.644 | 475.010.553 |
| 13 | 475.889.427 | 469.990.693 |
| 14 | 485.695.688 | 480.033.924 |
| 15 | 496.779.498 | 481.512.053 |
| 16 | 475.889.427 | 473.681.425 |
| 17 | 509.496.798 | 484.617.114 |
| 18 | 611.659.312 | 585.537.006 |
| 19 | 611.659.312 | 580.168.434 |
| 20 | 614.043.893 | 573.463.363 |
| 21 | 557.999.434 | 545.665.927 |
| 22 | 535.757.812 | 532.769.616 |
| 23 | 535.757.812 | 518.861.498 |
| 24 | 557.999.434 | 539.881.981 |

Tabel 4.15
Perbandingan Biaya Oporasional PT. PJB dengan Metode ICGA
Minggu, 02 Juli 2006

| Jam | PT. PJB | ICGA |
|-----|-------------|-------------|
| | (Rp) | (Rp) |
| 1 | 483.272.941 | 459.562.943 |
| 2 | 459.055.753 | 450.190.713 |
| 3 | 438.501.567 | 432.293.772 |
| 4 | 438.501.567 | 432.441.498 |
| 5 | 438.501.567 | 433.032.434 |
| 6 | 427.417.757 | 397.783.737 |
| 7 | 370.992.261 | 354.434.982 |
| 8 | 365.065.255 | 356.423.575 |
| 9 | 372.448.770 | 356.048.478 |
| 10 | 395.629.036 | 363.878.283 |
| 11 | 395.629.036 | 367.210.479 |
| 12 | 370.992.261 | 357.428.878 |
| 13 | 370.992.261 | 363.746.486 |
| 14 | 388.245.522 | 356.423.575 |
| 15 | 388.245.522 | 359.333.601 |
| 16 | 370.992.261 | 362.043.555 |
| 17 | 427.417.757 | 415.649.179 |
| 18 | 572.596.363 | 557.211.227 |
| 19 | 572.596.363 | 568.090.870 |
| 20 | 572.596.363 | 553.488.567 |
| 21 | 521.082.098 | 519.036.070 |
| 22 | 509.496.786 | 496.700.036 |
| 23 | 478.673.619 | 468.870.708 |
| 24 | 459.055.664 | 456.918.177 |

Tabel 4.16
Perbandingan Biaya Oprasional PT. PJB dengan Metode ICGA
Rabu, 05 Juli 2006

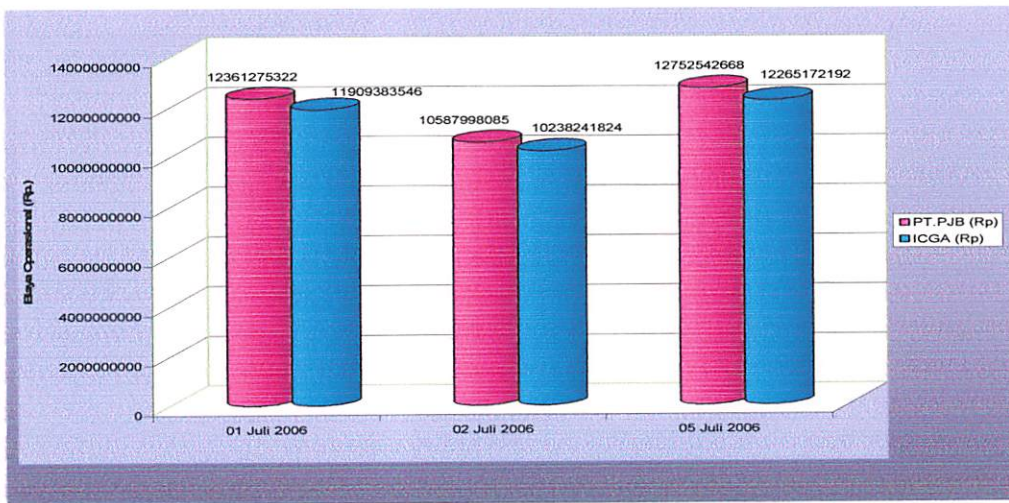
| Jam | PT. PJB | ICGA |
|-----|-------------|-------------|
| | (Rp) | (Rp) |
| 1 | 490.355.878 | 482.399.077 |
| 2 | 508.162.051 | 486.983.773 |
| 3 | 490.355.967 | 483.286.210 |
| 4 | 509.496.786 | 493.644.169 |
| 5 | 508.162.051 | 494.680.783 |
| 6 | 457.161.055 | 456.255.327 |
| 7 | 450.703.396 | 449.221.457 |
| 8 | 490.355.878 | 470.581.083 |
| 9 | 509.496.798 | 506.093.357 |
| 10 | 551.612.104 | 491.275.324 |
| 11 | 551.612.015 | 497.347.045 |
| 12 | 490.355.878 | 485.948.264 |
| 13 | 509.496.798 | 492.755.761 |
| 14 | 509.496.786 | 496.310.049 |
| 15 | 509.496.786 | 492.459.649 |
| 16 | 508.162.051 | 488.759.278 |
| 17 | 589.448.395 | 511.286.841 |
| 18 | 611.659.312 | 595.090.690 |
| 19 | 611.659.312 | 593.895.711 |
| 20 | 619.395.348 | 599.552.270 |
| 21 | 614.043.893 | 576.472.550 |
| 22 | 557.999.434 | 546.355.889 |
| 23 | 557.999.434 | 542.951.547 |
| 24 | 545.855.172 | 531.566.091 |

4.6.4. Perbandingan Total Biaya Operasional PT. PJB dengan ICGA

Tabel 4.17
Perbandingan Total Biaya Operasional PT. PJB dengan Metode ICGA

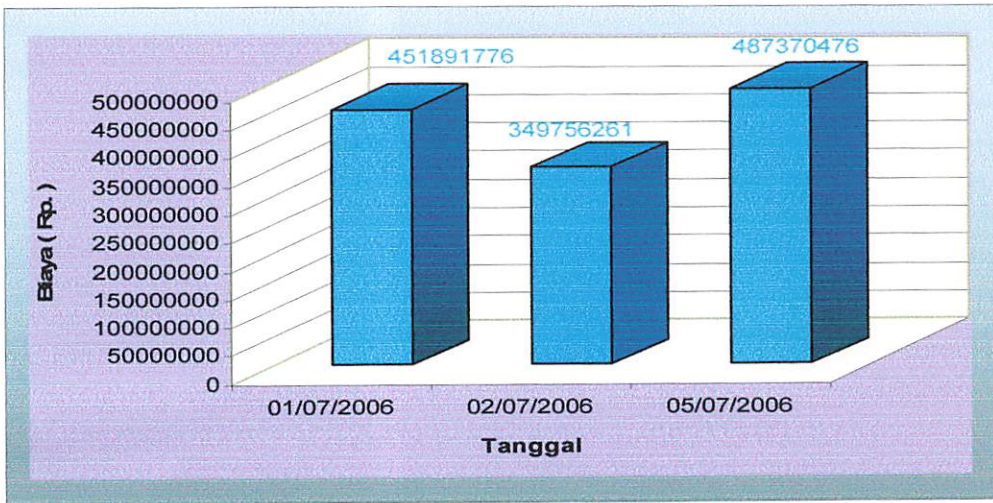
| Hari | Tanggal | PT.PJB | ICGA | Penghematan | Persentase |
|--------|------------|----------------|----------------|-------------|------------|
| | | (Rp) | (Rp) | (Rp) | (%) |
| Sabtu | 01/07/2006 | 12.361.275.322 | 11.909.383.546 | 451.891.776 | 3,65 |
| Minggu | 02/07/2006 | 10.587.998.085 | 10.238.241.824 | 349.756.261 | 3,30 |
| Rabu | 05/07/2006 | 12.752.542.668 | 12.265.172.192 | 487.370.476 | 3,82 |

Grafik 4.1
Perbandingan Biaya Operasional PT. PJB dan Metode ICGA
Pada Tanggal 01, 02 dan 05 Juli 2006



Dari tabel perbandingan biaya operasional untuk tiap jam selama 24 jam, dapat dibuat perbandingan biaya total operasional untuk periode 24 jam (satu hari) antara biaya total operasional selama 24 jam milik PT. PJB dengan biaya operasional selama 24 jam hasil optimasi metode ICGA. Bila dilihat pada grafik di atas maka di dapatkan penghematan biaya operasional, yang ditunjukkan pada grafik 4.2 sebagai berikut :

Grafik 4.2
Penghematan Biaya Operasional



Bila dilihat pada grafik di atas, tampak apabila dihitung antara biaya yang dikeluarkan oleh PT. PJB dengan biaya hasil optimasi dengan menggunakan metode ICGA akan didapatkan penghematan sebagai berikut:

Pada hari Sabtu, tanggal 01 Juli 2006 biaya yang dikeluarkan oleh PT. PJB sebesar Rp 12.361.275.322, optimasi dengan menggunakan metode ICGA sebesar Rp 11.909.383.546, dengan penghematan sebesar Rp 451.891.776 atau 3,65 %. Pada hari Minggu, tanggal 02 Juli 2006 biaya yang dikeluarkan oleh PT. PJB sebesar Rp 10.587.998.085, optimasi dengan menggunakan metode ICGA sebesar Rp 10.238.241.824, dengan penghematan sebesar Rp 349.756.261 atau 3,30 %. Pada hari Rabu, tanggal 05 Juli 2006 biaya yang dikeluarkan oleh PT. PJB sebesar Rp 12.752.542.668, optimasi dengan menggunakan metode ICGA sebesar Rp 12.265.172.192, dengan penghematan sebesar Rp 487.370.476 atau 3,82 %.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil analisa program dan hasil perhitungan terhadap penggunaan metode *Integer – Coded Genetic Algorithm* pada komitmen unit atau penjadwalan unit pembangkit terhadap beban yang ditanggung PT.PJB pada tanggal 01, 02 dan 05 Juli 2006, maka dapat diambil kesimpulan:

1. Proses metode *Integer – Coded Genetic Algorithm* memberikan sebuah analisis penyelesaian yang efektif dalam mengoptimalkan pembebanan dan sekaligus menghasilkan penghematan biaya total operasional PT. PJB. Adapun hasil perhitungan biaya total pembangkitan pada metode *Integer – Coded Genetic Algorithm* pada tanggal 01 Juli 2006 sebesar Rp 11.909.383.546, dengan penghematan sebesar Rp 451.891.776 atau 3,65% dengan waktu eksekusi 17 detik 734 mdetik. Pada tanggal 02 Juli 2006 sebesar Rp 10.238.241.824 dengan penghematan sebesar Rp 349.756.261 atau 3,30% dengan waktu eksekusi 15 detik 500 mdetik. Pada tanggal 05 Juli 2006 sebesar Rp 12.265.172.192 dengan penghematan sebesar Rp 487.370.476 atau 3,82% dengan waktu eksekusi 18 detik 250 mdetik
2. Kombinasi Metode *Integer – Coded Genetic Algorithm* layak diterapkan pada PT. PJB karena biaya total yang dihasilkan lebih ekonomis dibandingkan dengan biaya total pada PT. PJB.

5.2 Saran

Berdasarkan kesimpulan diatas, diajukan saran yang berhubungan dengan skripsi ini :

Untuk studi lebih lanjut, metode *Integer – Coded Genetic Algorithm* ini dapat dikembangkan dengan menambahkan masalah kendala rugi-rugi transmisi, sistem pembangkitan hidro dan memasukkan proses *combined cycle* pada PLTGU secara lengkap.

DAFTAR PUSTAKA

- [1] Djiteng Marsudi, Ir, "*Operasi Sistem Tenaga Listrik*", Balai Penerbit dan Humas ISTN, 1990.
- [2] Wood, A.J. "*Power Generations, Operation, and Control 2nd ED*", New York : Willey, 1996.
- [3] Damousis, I. G, "*A Solution to the Unit-Commitment Problem Using Integer-Coded Genetic Algorithm* ", IEEF. Trans. On Power Systems, May, 2004.
- [4]. Stevenson Jr., William, "*Analisa Sistem Tenaga* ", Erlangga Edisi ke-4, 1996.
- [5] Dewi, Sri Kusuma, "*Artificial Intelligence (Teknik dan Aplikasinya)*", Graha Ilmu, Yogyakarta, 2003.
- [6]. *Reference book: Jang et al "Neuro Neuro-Fuzzy and Soft Computing Fuzzy and Soft Computing" Prentice Hall 1997. Prentice Hall 1997.*
- [7] S. A. Kazarlis, A. G. Bakirtzis, and V. Petridis, "*A genetic algorithm solution to the unit commitment problem,*" IEEE Trans. Power Syst.,vol. 11, pp. 83-92, Feb 1996.



- FORM
- DATA
- LISTING PROGRAM





**LEMBAR PENGAJUAN JUDUL SKRIPSI
JURUSAN TEKNIK ELEKTRO S-1**

Konsentrasi : Teknik Energi Listrik/Teknik Elektronika *)

| | | |
|--|--|---|
| 1. | Nama Mahasiswa : Alief Catur Wibowo | Nim : 00.12.051 |
| 2. | Waktu Pengajuan : | Tanggal : |
| | | Bulan : 12 |
| | | Tahun : 2005 |
| Spesifikasi Judul (berilah tanda silang) | | |
| 3. | <input checked="" type="checkbox"/> a. Sistem Tenaga Elektrik <input type="checkbox"/> b. Energi & Konversi Energi <input type="checkbox"/> c. Tegangan Tinggi & Pengukuran <input type="checkbox"/> d. Sistem Kendali Industri | <input type="checkbox"/> e. Elektronika & Komponen <input type="checkbox"/> f. Elektronika Digital & Komputer <input type="checkbox"/> g. Elektronika Komunikasi <input type="checkbox"/> h. Lainnya |
| | Konsultasikan judul sesuai materi bidang ilmu kepada Dosen *) : | |
| 4. | <i>Dr. Yusuf Ismail ke. dit</i> | Mengetahui, Ketua Jurusan Ir. F Yudi Limpraptono, MT Nip. 1039500274 |
| 5. | Judul yang diajukan mahasiswa | KOMITMEN UNIT PEMBANGKIT TERMAL DENGAN METODE <i>INTEGER – CODED</i> ALGORITMA GENETIKA PADA PT. PEMBANGKITAN JAWA BALI |
| 6. | Perubahan Judul yang disetujui Dosen sesuai materi bidang ilmu | KOMITMEN UNIT PEMBANGKIT TERMAL DENGAN METODE <i>INTEGER – CODED GENETIC ALGORITHM</i> PADA PT. PEMBANGKITAN JAWA BALI |
| 7. | Catatan : <i>Dilengkapi data untuk uji validasi (dari jurnal referensi) dan survei data update PJB</i> | |
| 8. | Persetujuan Judul Skripsi yang dikonsultasikan kepada Kelompok Dosen materi bidang ilmu | Disetujui, ... <i>30.12</i> ..., 2005 Dosen Ir. Yusuf Ismail H. MT |

Perhatian :

1. Formulir Pengajuan ini harap dikembalikan ke Jurusan paling lambat satu minggu setelah disetujui kelompok dosen keahlian dengan dilampirkan proposal skripsi beserta persyaratan skripsi sesuai Form. S-1.

2. Keterangan : *) coret yang tidak perlu
 **) dilingkari a, b, c, ... atau g, sesuai bidang keahlian.

Lampiran : 1 (satu) berkas
Pembimbing Skripsi

Kepada : Yth. Bpk Ir. Yusuf Ismail Nahkoda, MT
Dosen Institut Teknologi Nasional
MALANG

Yang bertanda tangan dibawah ini :

Nama : Alief Catur Wibowo
Nim : 00.12.051
Jurusan : Teknik Elektro S-1
Konsentrasi : Energi Listrik

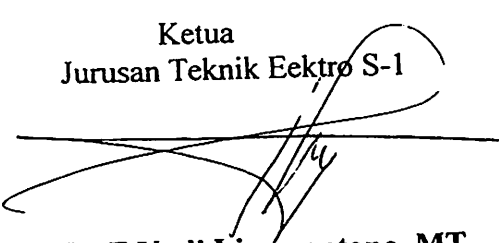
Dengan ini mengajukan permohonan, kiranya Bapak/Ibu bersedia menjadi Dosen Pembimbing Utama/ Pendamping dari 1/2 dosen pembimbing *), untuk penyusunan Skripsi dengan judul (proposal terlampir):

**KOMITMEN UNIT PEMBANGKIT TERMAL DENGAN
METODE *INTEGER – CODED GENETIC ALGORITHM* PADA
PT. PEMBANGKITAN JAWA BALI**

Adapun tugas tersebut sebagai salah satu syarat untuk menempuh Ujian Akhir Sarjana Teknik
Demikian permohonan kami dan atas kesediaan Bapak/Ibu kami ucapkan terimakasih.

Malang, 31 – 12 - 2005

Ketua
Jurusan Teknik Eektro S-1


Ir. F Yudi Limpraptono, MT
NIP:Y. 1039500274

Hormat kami,


Alief Catur Wibowo

*) Coret yang tidak perlu

Form.S-3a

PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai Permohonan dari mahasiswa :

Nama : Alief Catur Wibowo

Nim : 00.12.051

Semester : XI

Jurusan : Teknik Elektro S-1

Konsentrasi : Energi Listrik

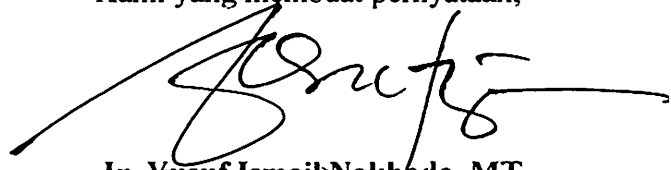
Dengan ini Menyatakan bersedia / ~~tidak bersedia~~ *) Membimbing Skripsi dari mahasiswa tersebut, dengan judul :

KOMITMEN UNIT PEMBANGKIT TERMAL DENGAN
METODE *INTEGER – CODED GENETIC ALGORITHM* PADA
PT. PEMBANGKITAN JAWA BALI

Demikian surat Pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Malang, 04 – 01 - 2006

Kami yang membuat pernyataan,



Ir. Yusuf Ismail Nakhoda, MT
NIP. P. 101 880 0189

Catatan :

Setelah disetujui agar formulir ini
Diserahkan mahasiswa yang bersangkutan
Kepada Jurusan untuk diproses lebih lanjut

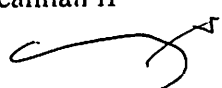
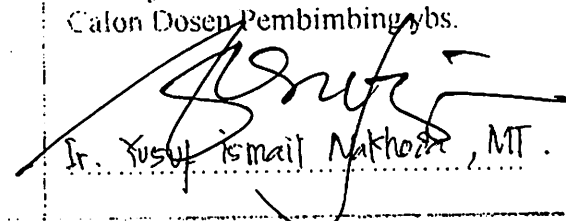
*) Coret yang tidak perlu

Form S-3b



**BERITA ACARA SEMINAR PROPOSAL SKRIPSI
JURUSAN TEKNIK ELEKTRO S-1**

Konsentrasi : Teknik Energi Listrik/Teknik Elektronika^{*)}

| | | | | |
|---|---|---|-------|----------|
| 1 | Nama Mahasiswa : | Alif Cah W | Nim : | 00.12051 |
| 2 | Keterangan | Tanggal | Waktu | Tempat |
| | Pelaksanaan | 18 | | Ruang : |
| 3 | Spesifikasi judul ^{**)} : | | | |
| | a. Sistem Tenaga Elektrik b. Energi & Konversi Energi c. Tegangan Tinggi & Pengukuran d. Sistem Kendali Industri | c. Elektronika & Komponen f. Elektronika Digital & Komputer g. Elektronika Komunikasi h. lainnya | | |
| 4 | Judul Proposal yang diseminarkan Mahasiswa | Komitmen Unit Pembangkit Termal dgn metode integer-codes Genetic algorithm pada PT. Pembangkitan Jawa Bali | | |
| 5 | Perubahan Judul yang diusulkan oleh Kelompok Dosen Keahlian | | | |
| 6 | Catatan : | | | |
| 7 | Persetujuan Judul Skripsi : | | | |
| | Disetujui, Dosen Keahlian I | Disetujui, Dosen Keahlian II  | | |
| | Mengetahui, Ketua Jurusan. Ir. F. Yudi Limpraptono, MT Nip. Y. 1039500274 | Disetujui, Calon Dosen Pembimbing/bs.  Ir. Yusuf Ismail Nakhoda, MT. | | |

Perhatian :

- *) coret yang tidak perlu
- **) dilingkari a, b, c, atau g. sesuai bidang keahlian.



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
 FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
 PROGRAM PASCASARJANA MAGISTER TEKNIK

RSERO) MALANG
 IAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting) Fax. (0341) 553015 Malang 65145
 Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 12 Juni 2006

Nomor : ITN-1419/I.TA/2/'06
 Lampiran : satu lembar
 Perihal : **BIMBINGAN SKRIPSI**

Kepada : Yth. Sdr. **Ir. YUSUF ISMAIL NAKHODA, MT**
 Dosen Institut Teknologi Nasional
 di –
 Malang

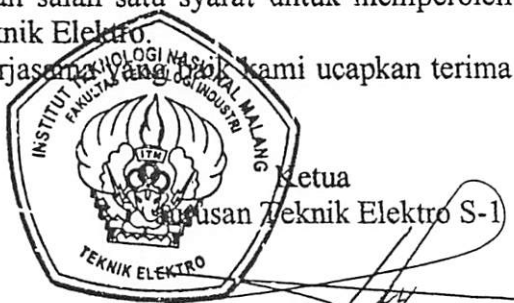
Dengan Hormat,
 Sesuai dengan permohonan dan persetujuan dalam proposal skripsi melalui seminar proposal yang telah dilakukan untuk mahasiswa :

Nama : ALIEF CATUR W
 Nim : 0012051
 Fakultas : Teknologi Industri
 Jurusan : Teknik Elektro
 Konsentrasi : T. Energi Listrik (S-1)

Dengan ini pembimbingan skripsi tersebut kami serahkan sepenuhnya kepada saudara/I selama masa waktu **6 (enam) bulan** terhitung mulai tanggal:

18 Feb. 2006 s/d 18 Agt. 2006

Adapun tugas tersebut merupakan salah satu syarat untuk memperoleh gelar Sarjana Teknik, Jurusan Teknik Elektro.
 Demikian atas perhatian serta kerjasannya kami ucapkan terima kasih



Ir. F. Yudi Limpraptono, MT
 Nip. Y. 1039500274

- Tindakan :**
1. Mahasiswa yang bersangkutan
 2. Arsip.

Form. S-4a



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI ENERGI LISTRIK

FORMULIR BIMBINGAN SKRIPSI

Nama : Alief Catur Wibowo
Nim : 00.12.051
Masa Bimbingan : 18 Februari 2006 – 18 Agustus 2006
Judul Skripsi : **Komitmen Unit Pembangkit Termal Dengan Metode Integer – Coded Genetic Algorithm Pada PT. Pembangkitan Jawa Bali**

| No. | Tanggal | Uraian | Paraf Pembimbing |
|-----|------------|--|------------------|
| 1. | 05-06-2006 | Bab I : - Perbaiki Sistematika Penulisan - Perbaiki Rumusan Masalah dan Tujuan | |
| 2. | 27-07-2006 | Bab II : - Perbaiki Teori Dasar - Pelajari Dasar Komitmen Unit | |
| 3. | 10-08-2006 | Bab III : - Perbaiki Tabel Status <i>On/Off</i> Unit Pembangkit - Perbaiki <i>Flowchart</i> | |
| 4. | 15-08-2006 | Bab IV : - Tampilkan Tabel Hasil Uji Validasi - Tambahkan Tabel Pmax Unit Pembangkit - Buat Tabel Perbandingan PT. PJB dengan Metode ICGA - Buat Grafik Penghematan Biaya Operasional | |
| 5. | 01-09-2006 | Bab V : - Perbaiki Kesimpulan, Tambahkan Waktu Eksekusi Program - Perbaiki Daftar Pustaka | |
| 6. | 05-09-2006 | Buat Makalah Seminar Hasil | |
| 7. | 06-09-2006 | Acc Seminar Hasil | |
| 8. | 21-09-2006 | Acc Kompre | |

Malang, September 2006
Dosen Pembimbing,

Ir. Yusuf Ismail Nakhoda, MT
NIP. P. 101 880 0189

DATA PENAWARAN
PT PLN PEMBANGKITAN JAWA BALI
AGUSTUS 2002

| No. | NAMA PEMBANGKIT | KAPASITAS | | | LAMA WAKTU (JAM) | | | | BIAYA START UP (JUTA Rp) | | KOEFSIEN BIAYA BAHAN BAKAR | | |
|-----|--------------------------------|---------------------|----------|----------|------------------|---------------|---------------|--------------|--------------------------|--------------|----------------------------|------------|-----------|
| | | Daya Terpasang (MW) | MIN (MW) | MAX (MW) | MIN UP TIME | MIN DOWN TIME | COLD START UP | HOT START UP | COLD START UP | HOT START UP | a0 | a1 | a2 |
| 1 | UP. PASTOH PLTU #1/2 (COAL) | 2 x 400 | 225 | 370 | 72 | 48 | 17 | 4 | 682.98 | 149.68 | 3244978 | 111712.15 | 10.2971 |
| 2 | UP. GRESIK | | | | | | | | | | | | |
| | GT 1-9 OC (GAS) | 9 x 112 | 53 | 102 | 36 | 10 | 1 | 0 | 7.82 | 0 | 5487532.4 | 217963.548 | 34.155 |
| | CC - 1.1.1 (GAS) | | 115 | 143 | 36 | 10 | 3 | 1 | 57.68 | 31.48 | 10936203.3 | 72527.004 | 368.874 |
| | CC - 2.2.1 (GAS) | | 164 | 314 | 36 | 10 | 3 | 2 | 65.5 | 39.28 | 11795770.8 | 152515.737 | 6.831 |
| | CC - 3.3.1 (GAS) | 3 x 526 | 250 | 480 | 36 | 10 | 3 | 2 | 73.32 | 47.1 | 17177460.3 | 145185.581 | 4.554 |
| | PLTU # 1/2 (GAS) | 100 | 43 | 85 | 48 | 10 | 9 | 1 | 143.74 | 40.59 | 1327128.68 | 217378.359 | 132.066 |
| | PLTU # 3/4 (GAS) | 200 | 90 | 175 | 48 | 10 | 9 | 2 | 229.5 | 92.52 | 5017369.5 | 169242.579 | 193.545 |
| | PLTG GRESIK 1-3 (GAS) | 3 x 20 | 5 | 18 | 3 | 1 | 1 | 0 | 6.13 | 0 | 352707.3 | 350680.77 | 903.969 |
| | PLTG GILTIMUR 1-2 (HSD) | 2 x 20 | 5 | 16 | 3 | 1 | 1 | 0 | 6.33 | 0 | 687181.85 | 683240.965 | 1762.3893 |
| 3 | UP. MUARA KARANG | | | | | | | | | | | | |
| | GT 1/2/3 - OC | 3 x 107 | 50 | 95 | 36 | 10 | 1 | 0 | 7.35 | 0 | 5730795 | 202052.97 | 108.045 |
| | CC - 1.1.1 (GAS) | 153 | 110 | 150 | 36 | 10 | 3 | 1 | 54.22 | 29.67 | 11560015 | 53685.135 | 460.845 |
| | CC - 2.2.1 (GAS) | 317 | 200 | 300 | 36 | 10 | 3 | 2 | 61.57 | 38.92 | 16010084 | 127208.655 | 35.28 |
| | CC - 3.3.1 (GAS) | 508 | 300 | 485 | 36 | 10 | 3 | 2 | 68.92 | 44.27 | 31017735 | 87825.15 | 57.33 |
| | MTW GT 1/2 - OC (HSD) | 2 x 140 | 72 | 138 | 36 | 10 | 0 | 0 | 0 | 0 | 14706521.25 | 433337.8 | 49.4605 |
| | MTW CC - 1.1.1 (HSD) | 200 | 162 | 202 | 36 | 10 | 3 | 1 | 118.08 | 64.4 | 672630 | 144191.717 | 519.1757 |
| | MTW CC - 2.2.1 (HSD) | 420 | 210 | 403 | 36 | 10 | 3 | 2 | 134.1 | 80.42 | 30123040 | 303208.82 | 11.64715 |
| | MTW CC - 3.3.1 (HSD) | 640 | 315 | 605 | 36 | 10 | 3 | 2 | 160.1 | 96.42 | 43043399 | 288609.995 | 7.6584 |
| | PLTU # 1/2/3 (MFO) | 3 x 100 | 44 | 85 | 48 | 10 | 6 | 1 | 122.58 | 31.08 | 2417820.7 | 473895.41 | 120.77935 |
| | PLTU # 4/5 (Gas) | 2 x 200 | 90 | 165 | 48 | 10 | 11 | 2 | 215.34 | 89.29 | 2949187.5 | 205217.145 | 83.79 |

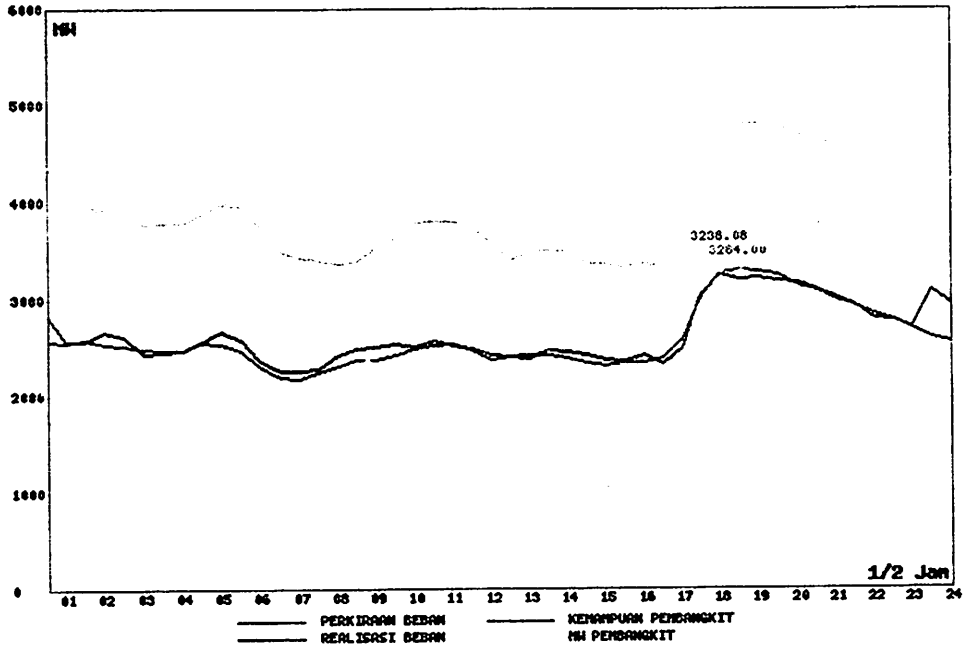
Catatan :
 Harga Batubara
 Harga MFO
 Harga HSD
 Harga Gas UP, Gresik
 Harga Gas UP, M. Karang
 Nilai Tukar

253 Rp/Kg
 1595.5 Rp/liter
 1595.5 Rp/liter
 2.53 US\$/MMBTU
 2.45 US\$/MMBTU
 9000 Rp/US\$

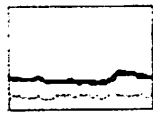
DATA BEBAN
SABTU, 01/07/2006
Sumber : PT. PJB REGION JAWA TIMUR DAN BALI

Kondisi Sistem: Sabtu, 1 Juli 2006
 Region Jawa Timur dan Bali

Grafik Hari Lainnya < 1/1 >



Region Jakarta dan Banten



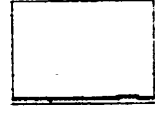
Region Jawa Barat



Region Jawa Tengah dan DIY



Region Jawa Timur dan Bali



Sub Region Bali

border="1"><AREA title="Pkl. 00.30 - Realisasi Beban: 2819 MW"
shape=CIRCLE coords=27,211,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 01.00 - Realisasi Beban: 2562 MW"
shape=CIRCLE coords=38,227,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 01.30 - Realisasi Beban: 2558 MW"
shape=CIRCLE coords=50,227,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 02.00 - Realisasi Beban: 2658 MW"
shape=CIRCLE coords=61,221,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 02.30 - Realisasi Beban: 2608 MW"
shape=CIRCLE coords=73,224,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 03.00 - Realisasi Beban: 2424 MW"
shape=CIRCLE coords=85,235,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 03.30 - Realisasi Beban: 2448 MW"
shape=CIRCLE coords=96,234,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 04.00 - Realisasi Beban: 2451 MW"
shape=CIRCLE coords=108,233,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 04.30 - Realisasi Beban: 2544 MW"
shape=CIRCLE coords=119,228,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 05.00 - Realisasi Beban: 2658 MW"
shape=CIRCLE coords=131,221,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 05.30 - Realisasi Beban: 2567 MW"
shape=CIRCLE coords=143,226,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 06.00 - Realisasi Beban: 2551 MW"
shape=CIRCLE coords=154,239,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 06.30 - Realisasi Beban: 2344 MW"
shape=CIRCLE coords=166,246,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 07.00 - Realisasi Beban: 2449 MW"
shape=CIRCLE coords=178,246,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 07.30 - Realisasi Beban: 2359 MW"
shape=CIRCLE coords=189,245,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 08.00 - Realisasi Beban: 2604 MW"
shape=CIRCLE coords=201,236,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 08.30 - Realisasi Beban: 2474 MW"
shape=CIRCLE coords=212,232,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 09.00 - Realisasi Beban: 2693 MW"
shape=CIRCLE coords=224,231,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 09.30 - Realisasi Beban: 2525 MW"
shape=CIRCLE coords=236,229,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 10.00 - Realisasi Beban: 2682 MW"
shape=CIRCLE coords=247,231,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 10.30 - Realisasi Beban: 2546 MW"
shape=CIRCLE coords=259,227,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 11.00 - Realisasi Beban: 2604 MW"
shape=CIRCLE coords=270,230,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 11.30 - Realisasi Beban: 2579 MW"
shape=CIRCLE coords=282,232,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 12.00 - Realisasi Beban: 2613 MW"
shape=CIRCLE coords=294,236,5 href="javascript:void(0)"

border="1"><AREA title="Pkl. 12.30 - Realisasi Beban: 2493 MW"
shape=CIRCLE coords=305,237,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 13.00 - Realisasi Beban: 2579 MW"
shape=CIRCLE coords=317,238,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 13.30 - Realisasi Beban: 2460 MW"
shape=CIRCLE coords=329,233,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 14.00 - Realisasi Beban: 2647 MW"
shape=CIRCLE coords=340,234,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 14.30 - Realisasi Beban: 2405 MW"
shape=CIRCLE coords=352,236,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 15.00 - Realisasi Beban: 2657 MW"
shape=CIRCLE coords=363,239,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 15.30 - Realisasi Beban: 2340 MW"
shape=CIRCLE coords=375,240,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 16.00 - Realisasi Beban: 2604 MW"
shape=CIRCLE coords=387,236,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 16.30 - Realisasi Beban: 2310 MW"
shape=CIRCLE coords=398,242,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 17.00 - Realisasi Beban: 2678 MW"
shape=CIRCLE coords=410,232,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 17.30 - Realisasi Beban: 3018 MW"
shape=CIRCLE coords=421,198,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 18.00 - Realisasi Beban: 3238 MW"
shape=CIRCLE coords=433,185,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 18.30 - Realisasi Beban: 3192 MW"
shape=CIRCLE coords=445,188,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 19.00 - Realisasi Beban: 3202 MW"
shape=CIRCLE coords=456,187,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 19.30 - Realisasi Beban: 3170 MW"
shape=CIRCLE coords=468,189,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 20.00 - Realisasi Beban: 3157 MW"
shape=CIRCLE coords=480,190,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 20.30 - Realisasi Beban: 3091 MW"
shape=CIRCLE coords=491,194,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 21.00 - Realisasi Beban: 2970 MW"
shape=CIRCLE coords=503,201,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 21.30 - Realisasi Beban: 2918 MW"
shape=CIRCLE coords=514,204,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 22.00 - Realisasi Beban: 2883 MW"
shape=CIRCLE coords=526,213,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 22.30 - Realisasi Beban: 2770 MW"
shape=CIRCLE coords=538,214,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 23.00 - Realisasi Beban: 2789 MW"
shape=CIRCLE coords=549,219,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 23.30 - Realisasi Beban: 3070 MW"
shape=CIRCLE coords=561,195,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 24.00 - Realisasi Beban: 2931 MW"

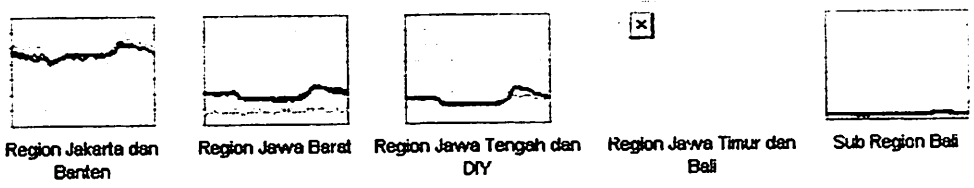
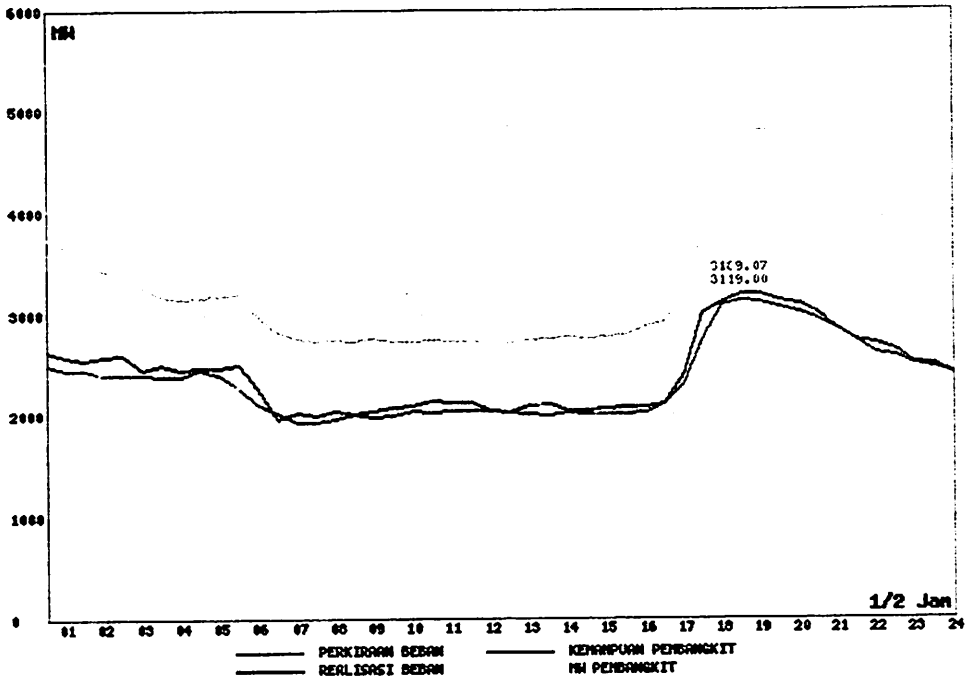
DATA BEBAN

MINGGU, 02/07/2006

Sumber : PT. PJB REGION JAWA TIMUR DAN BALI

Kondisi Sistem: Minggu, 2 Juli 2006
 Region Jawa Timur dan Bali

Grafik Hari Lainnya < >



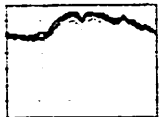
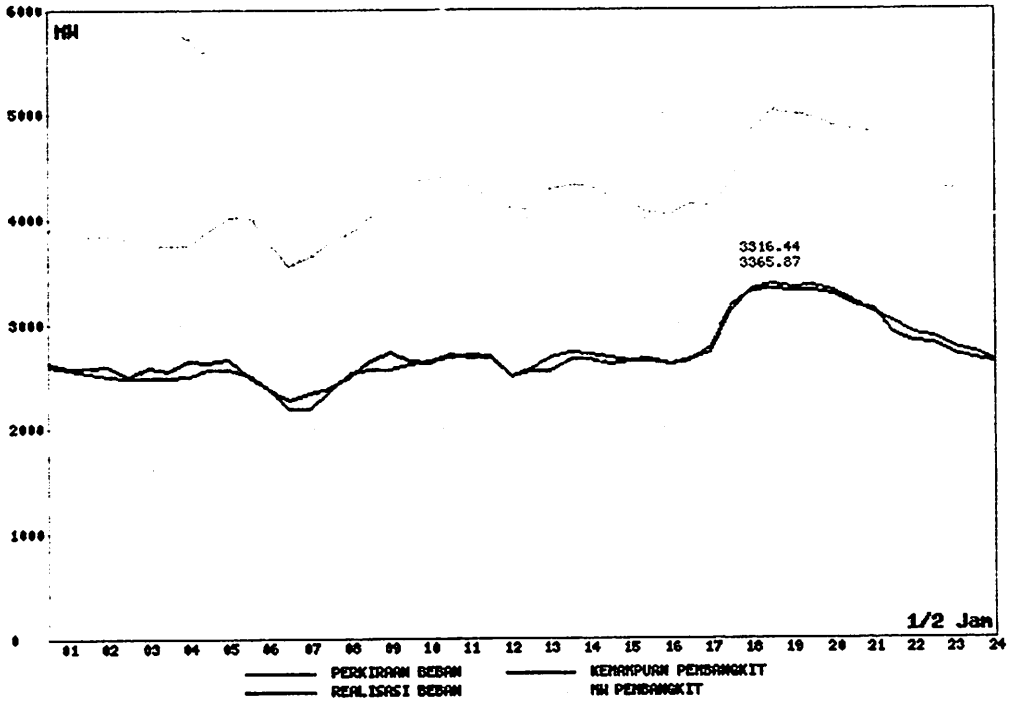
border="1"><AREA title="Pkl. 00.30 - Realisasi Beban: 2615 MW"
shape=CIRCLE coords=27,223,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 01.00 - Realisasi Beban: 2568 MW"
shape=CIRCLE coords=38,226,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 01.30 - Realisasi Beban: 2545 MW"
shape=CIRCLE coords=50,228,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 02.00 - Realisasi Beban: 2564 MW"
shape=CIRCLE coords=61,226,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 02.30 - Realisasi Beban: 2581 MW"
shape=CIRCLE coords=73,225,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 03.00 - Realisasi Beban: 2443 MW"
shape=CIRCLE coords=85,234,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 03.30 - Realisasi Beban: 2486 MW"
shape=CIRCLE coords=96,231,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 04.00 - Realisasi Beban: 2444 MW"
shape=CIRCLE coords=108,234,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 04.30 - Realisasi Beban: 2452 MW"
shape=CIRCLE coords=119,233,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 05.00 - Realisasi Beban: 2448 MW"
shape=CIRCLE coords=131,233,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 05.30 - Realisasi Beban: 2486 MW"
shape=CIRCLE coords=143,231,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 06.00 - Realisasi Beban: 2267 MW"
shape=CIRCLE coords=154,245,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 06.30 - Realisasi Beban: 1960 MW"
shape=CIRCLE coords=166,264,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 07.00 - Realisasi Beban: 2015 MW"
shape=CIRCLE coords=178,260,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 07.30 - Realisasi Beban: 1982 MW"
shape=CIRCLE coords=189,262,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 08.00 - Realisasi Beban: 2031 MW"
shape=CIRCLE coords=201,259,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 08.30 - Realisasi Beban: 2000 MW"
shape=CIRCLE coords=212,261,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 09.00 - Realisasi Beban: 2028 MW"
shape=CIRCLE coords=224,259,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 09.30 - Realisasi Beban: 2066 MW"
shape=CIRCLE coords=236,257,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 10.00 - Realisasi Beban: 2089 MW"
shape=CIRCLE coords=247,256,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 10.30 - Realisasi Beban: 2133 MW"
shape=CIRCLE coords=259,253,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 11.00 - Realisasi Beban: 2114 MW"
shape=CIRCLE coords=270,254,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 11.30 - Realisasi Beban: 2119 MW"
shape=CIRCLE coords=282,254,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 12.00 - Realisasi Beban: 2039 MW"
shape=CIRCLE coords=294,259,5 href="javascript:void(0)"

border="1"><AREA title="Pkl. 12.30 - Realisasi Beban: 2022 MW"
shape=CIRCLE coords=305,260,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 13.00 - Rcalisasi Beban: 2088 MW"
shape=CIRCLE coords=317,256,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 13.30 - Realisasi Beban: 2099 MW"
shape=CIRCLE coords=329,255,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 14.00 - Realisasi Beban: 2031 MW"
shape=CIRCLE coords=340,259,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 14.30 - Rcalisasi Beban: 2028 MW"
shape=CIRCLE coords=352,259,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 15.00 - Realisasi Beban: 2054 MW"
shape=CIRCLE coords=363,258,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 15.30 - Realisasi Beban: 2069 MW"
shape=CIRCLE coords=375,257,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 16.00 - Realisasi Beban: 2075 MW"
shape=CIRCLE coords=387,257,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 16.30 - Realisasi Beban: 2101 MW"
shape=CIRCLE coords=398,255,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 17.00 - Realisasi Beban: 2388 MW"
shape=CIRCLE coords=410,237,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 17.30 - Rcalisasi Beban: 2989 MW"
shape=CIRCLE coords=421,200,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 18.00 - Realisasi Beban: 3106 MW"
shape=CIRCLE coords=433,193,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 18.30 - Realisasi Beban: 3189 MW"
shape=CIRCLE coords=445,188,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 19.00 - Realisasi Beban: 3179 MW"
shape=CIRCLE coords=456,188,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 19.30 - Realisasi Beban: 3122 MW"
shape=CIRCLE coords=468,192,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 20.00 - Realisasi Beban: 3081 MW"
shape=CIRCLE coords=480,194,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 20.30 - Rcalisasi Beban: 3007 MW"
shape=CIRCLE coords=491,199,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 21.00 - Realisasi Beban: 2849 MW"
shape=CIRCLE coords=503,209,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 21.30 - Realisasi Beban: 2732 MW"
shape=CIRCLE coords=514,216,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 22.00 - Realisasi Beban: 2698 MW"
shape=CIRCLE coords=526,218,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 22.30 - Realisasi Beban: 2637 MW"
shape=CIRCLE coords=538,222,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 23.00 - Realisasi Beban: 2499 MW"
shape=CIRCLE coords=549,230,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 23.30 - Realisasi Beban: 2493 MW"
shape=CIRCLE coords=561,231,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 24.00 - Realisasi Beban: 2401 MW"

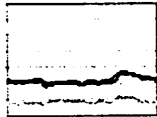
DATA BEBAN
RABU, 05/07/2006
Sumber : PT. PJB REGION JAWA TIMUR DAN BALI

Kondisi Sistem: Rabu, 5 Juli 2006
 Region Jawa Timur dan Bali

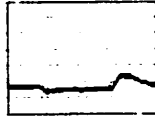
Grafik Hari Lainnya < [icon] >



Region Jakarta dan Banten



Region Jawa Barat



Region Jawa Tengah dan DIY



Region Jawa Timur dan Bali



Sub Region Bali

border="1"><AREA title="Pkl. 00.30 - Realisasi Beban: 2615 MW"
shape=CIRCLE coords=27,223,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 01.00 - Realisasi Beban: 2663 MW"
shape=CIRCLE coords=38,226,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 01.30 - Realisasi Beban: 2577 MW"
shape=CIRCLE coords=50,226,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 02.00 - Realisasi Beban: 2694 MW"
shape=CIRCLE coords=61,225,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 02.30 - Realisasi Beban: 2590 MW"
shape=CIRCLE coords=73,231,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 03.00 - Realisasi Beban: 2669 MW"
shape=CIRCLE coords=85,226,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 03.30 - Realisasi Beban: 2542 MW"
shape=CIRCLE coords=96,228,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 04.00 - Realisasi Beban: 2739 MW"
shape=CIRCLE coords=108,222,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 04.30 - Realisasi Beban: 2622 MW"
shape=CIRCLE coords=119,223,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 05.00 - Realisasi Beban: 2746 MW"
shape=CIRCLE coords=131,221,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 05.30 - Realisasi Beban: 2495 MW"
shape=CIRCLE coords=143,231,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 06.00 - Realisasi Beban: 2483 MW"
shape=CIRCLE coords=154,238,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 06.30 - Realisasi Beban: 2357 MW"
shape=CIRCLE coords=166,245,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 07.00 - Realisasi Beban: 2430 MW"
shape=CIRCLE coords=178,241,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 07.30 - Realisasi Beban: 2369 MW"
shape=CIRCLE coords=189,238,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 08.00 - Realisasi Beban: 2583 MW"
shape=CIRCLE coords=201,231,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 08.30 - Realisasi Beban: 2639 MW"
shape=CIRCLE coords=212,222,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 09.00 - Realisasi Beban: 2823 MW"
shape=CIRCLE coords=224,217,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 09.30 - Realisasi Beban: 2632 MW"
shape=CIRCLE coords=236,222,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 10.00 - Realisasi Beban: 2723 MW"
shape=CIRCLE coords=247,223,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 10.30 - Realisasi Beban: 2700 MW"
shape=CIRCLE coords=259,218,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 11.00 - Realisasi Beban: 2764 MW"
shape=CIRCLE coords=270,220,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 11.30 - Realisasi Beban: 2662 MW"
shape=CIRCLE coords=282,220,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 12.00 - Realisasi Beban: 2687 MW"
shape=CIRCLE coords=294,231,5 href="javascript:void(0)"

border="1"><AREA title="Pkl. 12.30 - Realisasi Beban: 2542 MW"
shape=CIRCLE coords=305,228,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 13.00 - Realisasi Beban: 2733 MW"
shape=CIRCLE coords=317,228,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 13.30 - Realisasi Beban: 2651 MW"
shape=CIRCLE coords=329,221,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 14.00 - Realisasi Beban: 2757 MW"
shape=CIRCLE coords=340,221,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 14.30 - Realisasi Beban: 2603 MW"
shape=CIRCLE coords=352,224,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 15.00 - Realisasi Beban: 2731 MW"
shape=CIRCLE coords=363,222,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 15.30 - Realisasi Beban: 2644 MW"
shape=CIRCLE coords=375,221,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 16.00 - Realisasi Beban: 2706 MW"
shape=CIRCLE coords=387,224,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 16.30 - Realisasi Beban: 2636 MW"
shape=CIRCLE coords=398,222,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 17.00 - Realisasi Beban: 2858 MW"
shape=CIRCLE coords=410,214,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 17.30 - Realisasi Beban: 3149 MW"
shape=CIRCLE coords=421,190,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 18.00 - Realisasi Beban: 3302 MW"
shape=CIRCLE coords=433,181,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 18.30 - Realisasi Beban: 3316 MW"
shape=CIRCLE coords=445,180,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 19.00 - Realisasi Beban: 3294 MW"
shape=CIRCLE coords=456,181,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 19.30 - Realisasi Beban: 3293 MW"
shape=CIRCLE coords=468,181,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 20.00 - Realisasi Beban: 3274 MW"
shape=CIRCLE coords=480,183,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 20.30 - Realisasi Beban: 3176 MW"
shape=CIRCLE coords=491,189,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 21.00 - Realisasi Beban: 3119 MW"
shape=CIRCLE coords=503,192,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 21.30 - Realisasi Beban: 2901 MW"
shape=CIRCLE coords=514,206,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 22.00 - Realisasi Beban: 2916 MW"
shape=CIRCLE coords=526,211,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 22.30 - Realisasi Beban: 2800 MW"
shape=CIRCLE coords=538,212,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 23.00 - Realisasi Beban: 2893 MW"
shape=CIRCLE coords=549,218,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 23.30 - Realisasi Beban: 2746 MW"
shape=CIRCLE coords=561,221,5 href="javascript:void(0)"
border="1"><AREA title="Pkl. 24.00 - Realisasi Beban: 2816 MW"

program UCGAInt;

uses

Forms,
uAbout in 'uAbout.pas' {frmAbout},
uHasil in 'uHasil.pas' {frmHasil},
uInputGenChild in 'uInputGenChild.pas',
uMenu in 'uMenu.pas' {frmMenu},
uGenetic in 'Genetic\uGenetic.pas',
uGenVar in 'Genetic\uGenVar.pas';

{\$R *.res}

begin

Application.Initialize;
Application.CreateForm(TfrmMenu, frmMenu);
Application.CreateForm(TfrmHasil, frmHasil);
Application.CreateForm(TfrmAbout, frmAbout);
Application.Run;
end.

unit uGenetic;

interface

uses uUtils,uObjFuncLR,uRandom,uGenVar,uHasil;

type

TBaseGenetic=class

private

FMaxGen,FPopSize,FLength:integer;

function getMin:dArr1;

function getAvg:dArr1;

function getMax:dArr1;

protected

FMin,FAvg,FMax:dArr1;

FRandom:TRandomu;

public

constructor Create(const rMaxGen,rPopSize,rLength:integer);

destructor Destroy;override;

property MaxGen:integer read FMaxGen write FMaxGen;

property PopSize:integer read FPopSize write FPopSize;

property Length:integer read FLength write FLength;

property Min:dArr1 read getMin;

property Avg:dArr1 read getAvg;

property Max:dArr1 read getMax;

end;

TGenetic=class(TBaseGenetic)

private

FPCross,FPMutasi,FKa:double;

FCrossType:TCrossType;

FNewParent:TNewParent;

public

constructor Create(const rMaxGen,rPopSize,rLength:integer;

const rPCross,rPMutasi,rKa:double;

const rCrossType:TCrossType;

const rNewParent:TNewParent);

property PCross:double read FPCross write FPCross;

property PMutasi:double read FPMutasi write FPMutasi;

property Ka:double read FKa write FKa;

property CrossType:TCrossType read FCrossType write FCrossType;

property NewParent:TNewParent read FNewParent write FNewParent;

end;

TGenBin=class(TGenetic)

private

```

FPflip:double;
public
  constructor Create(const rMaxGen,rPopSize,rLength:integer;
                    const rPCross,rPMutasi,rKa,rPflip:double;
                    const rCrossType:TCrossType;
                    const rNewParent:TNewParent);
  function Mutasi(const rAllele:boolean):boolean;
  property Pflip:double read FPflip write FPflip;
end;

```

```

TGenBin2=class(TGenBin)

```

```

private
  FParam:integer;
  FParent,FChild:TPopBin2;
  FBestIndi:TIndiBin2;
  FMin1,FAvg1,FMax1,FSumFitness:double;
  function getIndividu(const rIndi:TIndiBin2):TIndiBin2;
  function FindParentMax:TIndiBin2;
  procedure SwapIndi(var rIndi1,rIndi2:TIndiBin2);
  procedure InitParent;
  procedure Statistik;
  function Seleksi:integer;
  procedure Crossover(const rParent1,rParent2:bArr2;
                    var rChild1,rChild2:bArr2);
  procedure Generasi;
  procedure FindNewParent;
  procedure doHitung;
  function getBestChrom:bArr2;
public
  constructor Create(const rMaxGen,rPopSize,rLength,rParam:integer;
                    const rPCross,rPMutasi,rKa,rPflip:double;
                    const rCrossType:TCrossType;
                    const rNewParent:TNewParent);
  property Param:integer read FParam write FParam;
  property BestChrom:bArr2 read getBestChrom;
end;

```

```

TGenInt2=class(TGenBin)

```

```

private
  FParam:integer;
  FParent,FChild:TPopInt2;
  FBestIndi:TIndiInt2;
  FMin1,FAvg1,FMax1,FSumFitness:double;
  function DecodeIntToBin(const rChrom:iArr2):bArr2;
  function DecodeBinToInt(const rChrom:bArr2):iArr2;
  function getIndividu(const rIndi:TIndiInt2):TIndiInt2;

```

```

function FindParentMax:TIndiInt2;
procedure SwapIndi(var rIndi1,rIndi2:TIndiInt2);
procedure InitParent;
procedure Statistik;
function Seleksi:integer;
procedure Crossover(const rParent1,rParent2:bArr2;
    var rChild1,rChild2:bArr2);
procedure Generasi;
procedure FindNewParent;
procedure doHitung;
function getBestChrom:bArr2;
public
    constructor Create(const rMaxGen,rPopSize,rLength,rParam:integer;
        const rPCross,rPMutasi,rKa,rPflip:double;
        const rCrossType:TCrossType;
        const rNewParent:TNewParent);
    property Param:integer read FParam write FParam;
    property BestChrom:bArr2 read getBestChrom;
end;

```

implementation

```
{ TBaseGenetic }
```

```

//constructor
constructor TBaseGenetic.Create(const rMaxGen,rPopSize,rLength:integer);
begin
    inherited Create;
    FMaxGen:=rMaxGen;
    FPopSize:=rPopSize;
    FLength:=rLength;
    FRandom:=TRandomu.Create;
    SetLength(FMin,FMaxGen);
    SetLength(FAvg,FMaxGen);
    SetLength(FMax,FMaxGen);
end;

```

```

//destructor
destructor TBaseGenetic.Destroy;
begin
    try
        FRandom.Free;
    finally
        inherited Destroy;
    end;
end;

```



```

//data accessing
function TBaseGenetic.getAvg:dArr1;
var i:integer;
begin
  SetLength(result,FMaxGen);
  for i:=0 to FMaxGen-1 do
    begin
      result[i]:=FAvg[i];
    end;
  end;

function TBaseGenetic.getMax:dArr1;
var i:integer;
begin
  SetLength(result,FMaxGen);
  for i:=0 to FMaxGen-1 do
    begin
      result[i]:=FMax[i];
    end;
  end;

function TBaseGenetic.getMin:dArr1;
var i:integer;
begin
  SetLength(result,FMaxGen);
  for i:=0 to FMaxGen-1 do
    begin
      result[i]:=FMin[i];
    end;
  end;

{ TGenetic }

//constructor
constructor TGenetic.Create(const rMaxGen,rPopSize,rLength:integer;
  const rPCross,rPMutasi,rKa: double;
  const rCrossType:TCrossType;
  const rNewParent:TNewParent);
begin
  inherited Create(rMaxGen,rPopSize,rLength);
  FPCross:=rPCross;
  FPMutasi:=rPMutasi;
  FKa:=rKa;
  FCrossType:=rCrossType;
  FNewParent:=rNewParent;
end;

```

```

{ TGenBin }

//constructor
constructor TGenBin.Create(const rMaxGen,rPopSize,rLength:integer;
    const rPCross,rPMutasi,rKa,rPflip:double;
    const rCrossType:TCrossType;
    const rNewParent:TNewParent);
begin
    inherited Create(rMaxGen,rPopSize,rLength,rPCross,rPMutasi,rKa,
        rCrossType,rNewParent);
    FPFlip:=rPFlip;
end;

function TGenBin.Mutasi(const rAllele:boolean):boolean;
begin
    if FRandom.NextBoolean(PMutasi)=true then
        begin
            result:=not rAllele;
        end
    else
        begin
            result:=rAllele;
        end;
end;

{ TGenBin2 }

//constructor
constructor TGenBin2.Create(const rMaxGen,rPopSize,rLength,rParam:integer;
    const rPCross,rPMutasi,rKa,rPFlip:double;
    const rCrossType:TCrossType;
    const rNewParent:TNewParent);
begin
    inherited Create(rMaxGen,rPopSize,rLength,rPCross,rPMutasi,rKa,rPflip,
        rCrossType,rNewParent);
    FParam:=rParam;
end;

//data processing
function TGenBin2.getIndividu(const rIndi:TIndiBin2):TIndiBin2;
var i,j:integer;
begin
    SetLength(result.chrom,FParam+1,Length+1);
    for i:=1 to FParam do
        begin
            for j:=1 to Length do

```

```

begin
  result.chrom[i,j]:=rIndi.chrom[i,j];
end;
end;
result.fitness:=rIndi.fitness;
end;

```

```

function TGenBin2.FindParentMax:TIndiBin2;
var i:integer;
begin
  result:=getIndividu(FParent[0]);
  for i:=1 to PopSize-1 do
  begin
    if result.fitness<FParent[i].fitness then
    begin
      result:=getIndividu(FParent[i]);
    end;
  end;
end;

```

```

procedure TGenBin2.SwapIndi(var rIndi1,rIndi2:TIndiBin2);
var tmp:TIndiBin2;
begin
  tmp:=getIndividu(rIndi1);
  rIndi1:=getIndividu(rIndi2);
  rIndi2:=getIndividu(tmp);
end;

```

```

procedure TGenBin2.InitParent;
var i:integer;
    cost:double;
begin
  SetLength(FParent,PopSize);
  SetLength(FChild,PopSize);
  for i:=0 to PopSize-1 do
  begin
    SetLength(FParent[i].chrom,Param+1,Length+1);
    SetLength(FChild[i].chrom,Param+1,Length+1);
    FParent[i].chrom:=gObjFunc.getRandomChrom(Pflip);
    gObjFunc.doHitungChrom(FParent[i].chrom,cost);
    FParent[i].fitness:=Ka/cost;
  end;
end;

```

```

procedure TGenBin2.Statistik;
var i:integer;

```

```

begin
  FMin1:=FParent[0].fitness;
  FMax1:=FParent[0].fitness;
  FSumFitness:=FParent[0].fitness;
  for i:=0 to PopSize-1 do
    begin
      if FMax1<FParent[i].fitness then FMax1:=FParent[i].fitness;
      if FMin1>FParent[i].fitness then FMin1:=FParent[i].fitness;
      FSumFitness:=FSumFitness+FParent[i].fitness;
    end;
  FAvg1:=FSumFitness/PopSize;
end;

```

```

function TGenBin2.Seleksi:integer;
var randsum,partsum:double;
    i:integer;
begin
  partsum:=0;
  i:=0;
  randsum:=FRandom.NextDouble*FSumFitness;
  repeat
    i:=i+1;
    partsum:=partsum+FParent[i-1].fitness;
  until (partsum>randsum) or (i=PopSize);
  Result:=i-1;
end;

```

```

procedure TGenBin2.Crossover(const rParent1,rParent2:bArr2;
  var rChild1,rChild2:bArr2);
var i,j,point1,point2,count:integer;
begin
  SetLength(rChild1,Param+1,Length+1);
  SetLength(rChild2,Param+1,Length+1);
  if FRandom.NextBoolean(Pcross)=true then
    begin
      if FCrossType=crOne then
        begin
          point1:=FRandom.NextInt(1,Param*Length-1);
          count:=0;
          for i:=1 to Param do
            begin
              for j:=1 to Length do
                begin
                  inc(count);
                  if count<=point1 then
                    begin

```

```

    rChild1[i,j]:=Mutasi(rParent1[i,j]);
    rChild2[i,j]:=Mutasi(rParent2[i,j]);
end
else
begin
    rChild1[i,j]:=Mutasi(rParent2[i,j]);
    rChild2[i,j]:=Mutasi(rParent1[i,j]);
end;
end;
end;
end
else if FCrossType=crTwo then
begin
    point1:=FRandom.NextInt(1,Param*Length-1);
    repeat
        point2:=FRandom.NextInt(1,Param*Length-1);
    until point2<>point1;
    if point2<point1 then
    begin
        Swap(point1,point2);
    end;
    count:=0;
    for i:=1 to Param do
    begin
        for j:=1 to Length do
        begin
            inc(count);
            if count<=point1 then
            begin
                rChild1[i,j]:=Mutasi(rParent1[i,j]);
                rChild2[i,j]:=Mutasi(rParent2[i,j]);
            end
            else if (count>point1) and (count<=point2) then
            begin
                rChild1[i,j]:=Mutasi(rParent2[i,j]);
                rChild2[i,j]:=Mutasi(rParent1[i,j]);
            end
            else if count>point2 then
            begin
                rChild1[i,j]:=Mutasi(rParent1[i,j]);
                rChild2[i,j]:=Mutasi(rParent2[i,j]);
            end;
        end;
    end;
end;
end
else if FCrossType=crMulti then

```

```

begin
  for i:=1 to Param do
    begin
      for j:=1 to Length do
        begin
          if GetFlip(0.5)=true then
            begin
              rChild1[i,j]:=Mutasi(rParent2[i,j]);
              rChild2[i,j]:=Mutasi(rParent1[i,j]);
            end
          else
            begin
              rChild1[i,j]:=Mutasi(rParent1[i,j]);
              rChild2[i,j]:=Mutasi(rParent2[i,j]);
            end;
          end;
        end;
      end;
    end;
  end
else
  begin
    for i:=1 to Param do
      begin
        for j:=1 to Length do
          begin
            rChild1[i,j]:=Mutasi(rParent1[i,j]);
            rChild2[i,j]:=Mutasi(rParent2[i,j]);
          end;
        end;
      end;
    end;
  end;

procedure TGenBin2.Generasi;
var i,mate1,mate2:integer;
    cost:double;
begin
  i:=0;
  repeat
    mate1:=Seleksi;
    mate2:=Seleksi;
    Crossover(FParent[mate1].chrom,FParent[mate2].chrom,
    FChild[i].chrom,FChild[i+1].chrom);
    gObjFunc.doHitungChrom(FChild[i].chrom,cost);
    FChild[i].fitness:=Ka/cost;
    gObjFunc.doHitungChrom(FChild[i+1].chrom,cost);
    FChild[i+1].fitness:=Ka/cost;
  until

```

```
    i:=i+2;
    until i>=PopSize;
end;
```

```
procedure TGenBin2.FindNewParent;
var i,j,pos:integer;
    tmp:TPopBin2;
begin
    if NewParent=npStandart then
        begin
            for i:=0 to PopSize-1 do
                begin
                    FParent[i]:=getIndividu(FChild[i]);
                end;
            end
        else if NewParent=npReplikasi then
            begin
                SetLength(tmp,PopSize);
                for i:=0 to PopSize-1 do
                    begin
                        repeat
                            pos:=FRandom.NextInt(0,(PopSize-1));
                        until pos<>i;
                        if FChild[i].fitness>FParent[pos].fitness then
                            begin
                                tmp[i]:=getIndividu(FChild[i]);
                            end
                        else
                            begin
                                tmp[i]:=getIndividu(FParent[pos]);
                            end;
                        end;
                    end;
                for i:=0 to PopSize-1 do
                    begin
                        FParent[i]:=getIndividu(tmp[i]);
                    end;
                end
            else if NewParent=npElitism then
                begin
                    SetLength(tmp,2*PopSize);
                    for i:=0 to PopSize-1 do
                        begin
                            tmp[i]:=getIndividu(FParent[i]);
                            tmp[PopSize+i]:=getIndividu(FChild[i]);
                        end;
                    end;
                    for i:=0 to 2*PopSize-2 do
```

```

begin
  for j:=i to 2*PopSize-1 do
    begin
      if tmp[i].fitness<tmp[j].fitness then
        begin
          SwapIndi(tmp[i],tmp[j]);
        end;
      end;
    end;
  for i:=0 to PopSize-1 do
    begin
      FParent[i]:=getIndividu(tmp[i]);
    end;
  end;
end;

procedure TGenBin2.doHitung;
var gen:integer;
    tmpIndi:TIndiBin2;
begin
  InitParent;
  Statistik;
  FBestIndi:=FindParentMax;
  gen:=0;
  repeat
    Generasi;
    FindNewParent;
    Statistik;
    tmpIndi:=FindParentMax;
    if FBestIndi.fitness<tmpIndi.fitness then
      begin
        FBestIndi:=getIndividu(tmpIndi);
      end;
    FMin[gen]:=FMin1;
    FAvg[gen]:=FAvg1;
    FMax[gen]:=FMax1;
    frmHasil.pbIterasi.StepBy(1);
    gen:=gen+1;
  until gen>=MaxGen;
end;

//data output
function TGenBin2.getBestChrom:bArr2;
var i,j:integer;
begin
  doHitung;

```



```

SetLength(result,FParam+1,Length+1);
for i:=1 to FParam do
begin
  for j:=1 to Length do
  begin
    result[i,j]:=FBestIndi.chrom[i,j];
  end;
end;
end;

```

```

//-----NEW CODE HERE-----

```

```

//constructor
constructor TGenInt2.Create(const rMaxGen,rPopSize,rLength,rParam:integer;
  const rPCross,rPMutasi,rKa,rPflip:double;
  const rCrossType:TCrossType;
  const rNewParent:TNewParent);
begin
  inherited Create(rMaxGen,rPopSize,rLength,rPCross,rPMutasi,rKa,rPflip,
    rCrossType,rNewParent);
  FParam:=rParam;
end;

```

```

//data processing
function TGenInt2.DecodeIntToBin(const rChrom:iArr2):bArr2;
var i,j,k,st:integer;
begin
  SetLength(result,FParam+1,Length+1);
  for i:=1 to FParam do
  begin
    st:=1;
    for j:=1 to Length do
    begin
      if rChrom[i,j]<>0 then
      begin
        for k:=st to st+abs(rChrom[i,j])-1 do
        begin
          if rChrom[i,j]>0 then
          begin
            result[i,k]:=true;
          end
          else
          begin
            result[i,k]:=false;
          end;
        end;
      end;
      st:=st+abs(rChrom[i,j]);
    end;
  end;
end;

```

```
end;  
end;  
end;  
end;
```

```
function TGenInt2.DecodeBinToInt(const rChrom:bArr2):iArr2;  
var i,j,st,ab:integer;  
begin  
  SetLength(result,FParam+1,Length+1);  
  for i:=1 to FParam do  
    begin  
      for j:=1 to Length do  
        begin  
          result[i,j]:=0;  
        end;  
      end;  
    for i:=1 to FParam do  
      begin  
        ab:=1;  
        st:=0;  
        for j:=1 to Length do  
          begin  
            if rChrom[i,j]=true then  
              begin  
                if j=1 then  
                  begin  
                    st:=1;  
                  end  
                else if j=Length then  
                  begin  
                    if st<0 then  
                      begin  
                        result[i,ab]:=st;  
                        ab:=ab+1;  
                        st:=1;  
                      end  
                    else  
                      begin  
                        st:=st+1;  
                      end;  
                    result[i,ab]:=st;  
                  end  
                else  
                  begin  
                    if rChrom[i,j-1]=true then  
                      begin
```

```

    st:=st+1;
end
else
begin
    result[i,ab]:=st;
    st:=1;
    ab:=ab+1;
end;
end;
end
else
begin
    if j=1 then
    begin
        st:=-1;
    end
    else if j=Length then
    begin
        if st<0 then
        begin
            st:=st-1;
        end
        else
        begin
            result[i,ab]:=st;
            ab:=ab+1;
            st:=-1;
        end;
        result[i,ab]:=st;
    end
    else
    begin
        if rChrom[i,j-1]=true then
        begin
            result[i,ab]:=st;
            st:=-1;
            ab:=ab+1;
        end
        else
        begin
            st:=st-1;
        end;
    end;
end;
end;
end;
end;
end;
end;

```

end;

```
function TGenInt2.getIndividu(const rIndi:TIndiInt2):TIndiInt2;
var i,j:integer;
begin
  SetLength(result.chrom,FParam+1,Length+1);
  for i:=1 to FParam do
    begin
      for j:=1 to Length do
        begin
          result.chrom[i,j]:=rIndi.chrom[i,j];
        end;
      end;
    result.fitness:=rIndi.fitness;
  end;
end;
```

```
function TGenInt2.FindParentMax:TIndiInt2;
var i:integer;
begin
  result:=getIndividu(FParent[0]);
  for i:=1 to PopSize-1 do
    begin
      if result.fitness<FParent[i].fitness then
        begin
          result:=getIndividu(FParent[i]);
        end;
      end;
    end;
end;
```

```
procedure TGenInt2.SwapIndi(var rIndi1,rIndi2:TIndiInt2);
var tmp:TIndiInt2;
begin
  tmp:=getIndividu(rIndi1);
  rIndi1:=getIndividu(rIndi2);
  rIndi2:=getIndividu(tmp);
end;
```

```
procedure TGenInt2.InitParent;
var i:integer;
    cost:double;
    chrom:bArr2;
begin
  SetLength(FParent,PopSize);
  SetLength(FChild,PopSize);
  for i:=0 to PopSize-1 do
    begin
```

```

    SetLength(FParent[i].chrom,Param+1,Length+1);
    SetLength(FChild[i].chrom,Param+1,Length+1);
    chrom:=gObjFunc.getRandomChrom(Pflip);
    gObjFunc.doHitungChrom(chrom,cost);
    FParent[i].chrom:=DecodeBinToInt(chrom);
    FParent[i].fitness:=Ka/cost;
end;
end;

procedure TGenInt2.Statistik;
var i:integer;
begin
    FMin1:=FParent[0].fitness;
    FMax1:=FParent[0].fitness;
    FSumFitness:=FParent[0].fitness;
    for i:=0 to PopSize-1 do
        begin
            if FMax1<FParent[i].fitness then FMax1:=FParent[i].fitness;
            if FMin1>FParent[i].fitness then FMin1:=FParent[i].fitness;
            FSumFitness:=FSumFitness+FParent[i].fitness;
        end;
    FAvg1:=FSumFitness/PopSize;
end;

function TGenInt2.Seleksi:i:integer;
var randsum,partsum:double;
    i:integer;
begin
    partsum:=0;
    i:=0;
    randsum:=FRandom.NextDouble*FSumFitness;
    repeat
        i:=i+1;
        partsum:=partsum+FParent[i-1].fitness;
    until (partsum>randsum) or (i=PopSize);
    Result:=i-1;
end;

procedure TGenInt2.Crossover(const rParent1,rParent2:bArr2;
    var rChild1,rChild2:bArr2);
var i,j,point1,point2,count:integer;
begin
    SetLength(rChild1,Param+1,Length+1);
    SetLength(rChild2,Param+1,Length+1);
    if FRandom.NextBoolean(Pcross)=true then
        begin

```

```

if FCrossType=crOne then
begin
point1:=FRandom.NextInt(1,Param*Length-1);
count:=0;
for i:=1 to Param do
begin
for j:=1 to Length do
begin
inc(count);
if count<=point1 then
begin
rChild1[i,j]:=Mutasi(rParent1[i,j]);
rChild2[i,j]:=Mutasi(rParent2[i,j]);
end
else
begin
rChild1[i,j]:=Mutasi(rParent2[i,j]);
rChild2[i,j]:=Mutasi(rParent1[i,j]);
end;
end;
end;
end
end
else if FCrossType=crTwo then
begin
point1:=FRandom.NextInt(1,Param*Length-1);
repeat
point2:=FRandom.NextInt(1,Param*Length-1);
until point2<>point1;
if point2<point1 then
begin
Swap(point1,point2);
end;
count:=0;
for i:=1 to Param do
begin
for j:=1 to Length do
begin
inc(count);
if count<=point1 then
begin
rChild1[i,j]:=Mutasi(rParent1[i,j]);
rChild2[i,j]:=Mutasi(rParent2[i,j]);
end
else if (count>point1) and (count<=point2) then
begin
rChild1[i,j]:=Mutasi(rParent2[i,j]);

```

```

    rChild2[i,j]:=Mutasi(rParent1[i,j]);
end
else if count>point2 then
begin
    rChild1[i,j]:=Mutasi(rParent1[i,j]);
    rChild2[i,j]:=Mutasi(rParent2[i,j]);
end;
end;
end;
end
else if FCrossType=crMulti then
begin
    for i:=1 to Param do
    begin
        for j:=1 to Length do
        begin
            if GetFlip(0.5)=true then
            begin
                rChild1[i,j]:=Mutasi(rParent2[i,j]);
                rChild2[i,j]:=Mutasi(rParent1[i,j]);
            end
            else
            begin
                rChild1[i,j]:=Mutasi(rParent1[i,j]);
                rChild2[i,j]:=Mutasi(rParent2[i,j]);
            end;
        end;
    end;
end;
end;
end
else
begin
    for i:=1 to Param do
    begin
        for j:=1 to Length do
        begin
            rChild1[i,j]:=Mutasi(rParent1[i,j]);
            rChild2[i,j]:=Mutasi(rParent2[i,j]);
        end;
    end;
end;
end;
end;

procedure TGenInt2.Generasi;
var i,mate1,mate2:integer;
    cost:double;

```

```

    chromP1,chromP2,chromC1,chromC2:bArr2;
begin
    i:=0;
    repeat
        mate1:=Seleksi;
        mate2:=Seleksi;
        chromP1:=DecodeIntToBin(FParent[mate1].chrom);
        chromP2:=DecodeIntToBin(FParent[mate2].chrom);
        Crossover(chromP1,chromP2,chromC1,chromC2);
        gObjFunc.doHitungChrom(chromC1,cost);
        FChild[i].chrom:=DecodeBinToInt(chromC1);
        FChild[i].fitness:=Ka/cost;
        gObjFunc.doHitungChrom(chromC2,cost);
        FChild[i+1].chrom:=DecodeBinToInt(chromC2);
        FChild[i+1].fitness:=Ka/cost;
        i:=i+2;
    until i>=PopSize;
end;

```

```

procedure TGenInt2.FindNewParent;
var i,j,pos:integer;
    tmp:TPopInt2;
begin
    if NewParent=npStandart then
        begin
            for i:=0 to PopSize-1 do
                begin
                    FParent[i]:=getIndividu(FChild[i]);
                end;
            end
        else if NewParent=npReplikasi then
            begin
                SetLength(tmp,PopSize);
                for i:=0 to PopSize-1 do
                    begin
                        repeat
                            pos:=FRandom.NextInt(0,(PopSize-1));
                        until pos<>i;
                        if FChild[i].fitness>FParent[pos].fitness then
                            begin
                                tmp[i]:=getIndividu(FChild[i]);
                            end
                        else
                            begin
                                tmp[i]:=getIndividu(FParent[pos]);
                            end;
                    end;
                end;
            end;

```



```

end;
for i:=0 to PopSize-1 do
begin
  FParent[i]:=getIndividu(tmp[i]);
end;
end
else if NewParent=npElitism then
begin
  SetLength(tmp,2*PopSize);
  for i:=0 to PopSize-1 do
  begin
    tmp[i]:=getIndividu(FParent[i]);
    tmp[PopSize+i]:=getIndividu(FChild[i]);
  end;
  for i:=0 to 2*PopSize-2 do
  begin
    for j:=i to 2*PopSize-1 do
    begin
      if tmp[i].fitness<tmp[j].fitness then
      begin
        SwapIndi(tmp[i],tmp[j]);
      end;
    end;
  end;
  for i:=0 to PopSize-1 do
  begin
    FParent[i]:=getIndividu(tmp[i]);
  end;
end;
end;

```

```

procedure TGenInt2.doHitung;
var gen:integer;
    tmpIndi:TIndiInt2;
begin
  InitParent;
  Statistik;
  FBestIndi:=FindParentMax;
  gen:=0;
  repeat
    Generasi;
    FindNewParent;
    Statistik;
    tmpIndi:=FindParentMax;
    if FBestIndi.fitness<tmpIndi.fitness then
    begin

```

```
    FBestIndi:=getIndividu(tmpIndi);
end;
FMin[gen]:=FMin1;
FAvg[gen]:=FAvg1;
FMax[gen]:=FMax1;
frmHasil.pbIterasi.StepBy(1);
gen:=gen+1;
until gen>=MaxGen;
end;

//data output
function TGenInt2.getBestChrom:bArr2;
begin
    doHitung;
    result:=DecodeIntToBin(FBestIndi.chrom);
end;

end.
```

unit uGenVar;

interface

uses uUtils;

type

TIndiBin1=record
chrom:bArr1;
fitness:double;
end;

TPopBin1=array of TIndiBin1;

TIndiBin2=record
chrom:bArr2;
fitness:double;
end;

TPopBin2=array of TIndiBin2;

TIndiFloat1=record
chrom:dArr1;
fitness:double;
end;

TPopFloat1=array of TIndiFloat1;

TIndiFloat2=record
chrom:dArr2;
fitness:double;
end;

TPopFloat2=array of TIndiFloat2;

TIndiInt1=record
chrom:iArr1;
fitness:double;
end;

TPopInt1=array of TIndiInt1;

TIndiInt2=record
chrom:iArr2;
fitness:double;
end;

TPopInt2=array of TIndiInt2;

TCrossType=(crOne,crTwo,crMulti);

TNewParent=(npStandart,npReplikasi,npElitism);

implementation

end.

unit uHasil;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, ExtCtrls, TeEngine, Series, TeeProcs, Chart, Grids, ComCtrls,
StdCtrls, jpeg;

type

TfrmHasil = class(TForm)
 TabSheet5: TTabSheet;
 TabSheet6: TTabSheet;
 TabSheet7: TTabSheet;
 Panel1: TPanel;
 btnClose: TButton;
 btnHitungEP: TButton;
 TabSheet1: TTabSheet;
 TabSheet2: TTabSheet;
 TabSheet4: TPageControl;
 GroupBox2: TGroupBox;
 Label2: TLabel;
 Label3: TLabel;
 Label4: TLabel;
 lblJmlParam: TLabel;
 Label12: TLabel;
 Label7: TLabel;
 Label10: TLabel;
 Label13: TLabel;
 edtMaxGen: TEdit;
 edtPopSize: TEdit;
 edtPCross: TEdit;
 edtParam: TEdit;
 edtLength: TEdit;
 edtPMutasi: TEdit;
 edtKa: TEdit;
 edtPflip: TEdit;
 pbIterasi: TProgressBar;
 fgDaya: TStringGrid;
 fgCostPerJam: TStringGrid;
 Label8: TLabel;
 edtTotalProgram: TEdit;
 Label9: TLabel;
 edtTotalPLN: TEdit;
 Label1: TLabel;
 edtSelisih: TEdit;

```

Label5: TLabel;
edtTime: TEdit;
Label6: TLabel;
fgStatus: TStringGrid;
Chart1: TChart;
Series1: TLineSeries;
Series2: TLineSeries;
cmbCrossType: TComboBox;
cmbNewParent: TComboBox;
Label11: TLabel;
Label14: TLabel;
TabSheet3: TTabSheet;
Chart2: TChart;
Series3: TLineSeries;
Series4: TLineSeries;
Series5: TLineSeries;
btnUseDefault: TButton;
Image1: TImage;
Image2: TImage;
procedure btnCloseClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure btnHitungEPClick(Sender: TObject);
procedure btnUseDefaultClick(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  frmHasil: TfrmHasil;

implementation

uses uObjFuncLR,uUtils, uGenVar, uGenetic;

{$R *.dfm}

procedure TfrmHasil.btnCloseClick(Sender: TObject);
begin
  Close;
end;

procedure TfrmHasil.FormCreate(Sender: TObject);
begin
  fgCostPerJam.Cells[0,0]:='Jam';

```

```
fgCostPerJam.Cells[1,0]:= 'Biaya Program';
fgCostPerJam.Cells[2,0]:= 'Biaya PLN';
fgCostPerJam.Cells[3,0]:= 'Selisih Biaya';
cmbCrossType.Text:= 'crOne';
cmbNewParent.Text:= 'npReplikasi';
end;
```

```
procedure TfrmHasil.btnHitungEPClick(Sender: TObject);
var i,j,MaxGen,PopSize,Param,Length:integer;
    CostTotal,CostPLN,PCross,PMutasi,Pflip,Ka:double;
    chrom:bArr2;
    PL:dArr2;
    CostPerJam,CostPerJamPLN,Min,Avg,Max:dArr1;
    mulai,selesai,selang:TDateTime;
    jam,menit,detik,mdetik:word;
    CrossType:TCrossType;
    NewParent:TNewParent;
    gas:TGenInt2;
begin
    MaxGen:=StrToInt(edtMaxGen.Text);
    pbIterasi.Max:=MaxGen;
    PopSize:=StrToInt(edtPopSize.Text);
    PCross:=StrToFloat(edtPCross.Text);
    PMutasi:=StrToFloat(edtPMutasi.Text);
    PFlip:=StrToFloat(edtPFlip.Text);
    Ka:=StrToFloat(edtKa.Text);
    Param:=StrToInt(edtParam.Text);
    Length:=StrToInt(edtLength.Text);
    CrossType:=crOne;
    if cmbCrossType.Text='crOne' then
    begin
        CrossType:=crOne;
    end
    else if cmbCrossType.Text='crTwo' then
    begin
        CrossType:=crTwo;
    end
    else if cmbCrossType.Text='crMulti' then
    begin
        CrossType:=crMulti;
    end;
    NewParent:=npStandart;
    if cmbNewParent.Text='npStandart' then
    begin
        NewParent:=npStandart;
    end
end
```

```

else if cmbNewParent.Text='npReplikasi' then
begin
  NewParent:=npReplikasi;
end
else if cmbNewParent.Text='npElitism' then
begin
  NewParent:=npElitism;
end;
mulai:=time;
gas:=TGenInt2.Create(MaxGen,PopSize,Length,Param,PCross,
  PMutasi,Ka,Pflip,CrossType,NewParent);
chrom:=gas.BestChrom;
Min:=gas.Min;
Avg:=gas.Avg;
Max:=gas.Max;
gas.Free;
gObjFunc.doHitungChrom(chrom,PL,CostPerJam,CostTotal);
//gObjFunc.doExecute(chrom,PL,CostPerJam,CostTotal);
selesai:=time;
selang:=selesai-mulai;
DecodeTime(selang,jam,menit,detik,mdetik);
edtTime.Text:=IntToStr(jam)+' : '+IntToStr(menit)+' : '+
  IntToStr(detik)+' : '+IntToStr(mdetik);
for i:=1 to high(chrom) do
begin
  for j:=1 to high(chrom[0]) do
  begin
    fgDaya.Cells[j,i]:=RealToStr(PL[i,j],0);
    if chrom[i,j]=true then
    begin
      fgStatus.Cells[j,i]:='1';
    end
    else
    begin
      fgStatus.Cells[j,i]:='0';
    end;
  end;
end;
edtTotalProgram.Text:=FormatFloat('#,##0',CostTotal);
gObjFunc.doHitungPLN(CostPerJamPLN,CostPLN);
Series1.Clear;
Series2.Clear;
for i:=1 to high(CostPerJam) do
begin
  fgCostPerJam.Cells[0,i]:=IntToStr(i);
  fgCostPerJam.Cells[1,i]:=FormatFloat('#,##00',CostPerJam[i]);

```



```

fgCostPerJam.Cells[2,i]:=FormatFloat('#,#00',CostPerJamPLN[i]);
fgCostPerJam.Cells[3,i]:=FormatFloat('#,#00',
    (CostPerJamPLN[i]-CostPerJam[i]));
Series1.Add(CostPerJam[i],IntToStr(i));
Series2.Add(CostPerJamPLN[i],IntToStr(i));
end;
edtTotalPLN.Text:=FormatFloat('#,##0',CostPLN);
edtSelisih.Text:=FormatFloat('#,##0',(CostPLN-CostTotal));
Series3.Clear;
Series4.Clear;
Series5.Clear;
for i:=0 to high(Min) do
begin
    Series3.Add(Min[i],IntToStr(i));
    Series4.Add(Avg[i],IntToStr(i));
    Series5.Add(Max[i],IntToStr(i));
end;
end;

```

```

procedure TfrmHasil.btnUseDefaultClick(Sender: TObject);
begin
    edtMaxGen.Text:='50';
    edtPopSize.Text:='50';
    edtPCross.Text:='0.75';
    edtPMutasi.Text:='0.003';
    edtPFlip.Text:='0.1';
    edtParam.Text:=IntToStr(gObjFunc.Ngen);
    edtLength.Text:=IntToStr(gObjFunc.Njam);
    edtKa.Text:='1000000';
    btnHitungEP.Enabled:=true;
end;

end.

```

unit uInputGenChild;

interface

uses uInputGen,Forms,uHasil,uObjFuncLR,SysUtils;

type

TfrmInputGenChild=class(TfrmInputGen)

protected

procedure ShowHasil;override;

end;

var frmInput:TfrmInputGenChild;

implementation

procedure TfrmInputGenChild.ShowHasil;

var i:integer;

begin

try

if frmHasil=nil then

begin

frmHasil:=TfrmHasil.Create(Application);

end;

frmHasil.fgStatus.RowCount:=gObjFunc.Ngen+1;

frmHasil.fgStatus.ColCount:=gObjFunc.Njam+1;

frmHasil.fgDaya.RowCount:=gObjFunc.Ngen+1;

frmHasil.fgDaya.ColCount:=gObjFunc.Njam+1;

frmHasil.fgCostPerJam.RowCount:=gObjFunc.Njam+1;

for i:=1 to gObjFunc.Ngen do

begin

frmHasil.fgStatus.Cells[0,i]:='Unit '+IntToStr(i);

frmHasil.fgDaya.Cells[0,i]:='Unit '+IntToStr(i);

end;

for i:=1 to gObjFunc.Njam do

begin

frmHasil.fgStatus.Cells[i,0]:='Jam '+IntToStr(i);

frmHasil.fgDaya.Cells[i,0]:='Jam '+IntToStr(i);

end;

frmHasil.ShowModal;

finally

frmHasil.Free;

end;

end;

end.

unit uMenu;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, ComCtrls, StdCtrls, ExtCtrls, jpeg;

type

TfrmMenu = class(TForm)

Panel1: TPanel;

btnNew: TButton;

btnOpen: TButton;

btnExit: TButton;

StatusBar1: TStatusBar;

Panel2: TPanel;

OpenDialog1: TOpenDialog;

Image1: TImage;

Image2: TImage;

Label1: TLabel;

Label5: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label6: TLabel;

procedure btnNewClick(Sender: TObject);

procedure btnOpenClick(Sender: TObject);

procedure btnExitClick(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

frmMenu: TfrmMenu;

implementation

uses uComplex,uUtils,uInputGenChild,uObjFuncLR,uGenerator;

{\$R *.dfm}

procedure TfrmMenu.btnNewClick(Sender: TObject);

begin

try

```

if frmInput=nil then
begin
  frmInput:=TfrmInputGenChild.Create(Application);
end;
frmInput.Caption:='Input Data';
frmInput.btnNext.Caption:='&Save';
frmInput.ShowModal;
finally
  frmInput.Free;
end;
end;

procedure TfrmMenu.btnOpenClick(Sender: TObject);
var NamaFile,Nama:string;
    output:TextFile;
    Pmin,Pmax,a2,a1,a0,Sh,Sc,Ramp,Load,Res:double;
    i,j,Ngen,Njam,Tup,Tdown,Tcold,InitSt:integer;
    aLoad,aRes:dArr1;
    aPLN:dArr2;
    aGen:TGenArr;
begin
try
if OpenFileDialog1.Execute then
begin
  NamaFile:=OpenDialog1.FileName;
  AssignFile(output,NamaFile);
  Reset(output);
  Readln(output,Ngen);
  Readln(output,Njam);
  try
  if frmInput=nil then
  begin
    frmInput:=TfrmInputGenChild.Create(Application);
  end;
  frmInput.edtNGen.Text:=IntToStr(Ngen);
  frmInput.edtNjam.Text:=IntToStr(Njam);
  frmInput.fgGen.RowCount:=Ngen+1;
  SetLength(aGen,Ngen+1);
  for i:=1 to Ngen do
  begin
    Readln(output,Pmax,Pmin,a0,a1,a2,Tup,Tdown,Sh,Sc,tcold,InitSt,
    Ramp,Nama);
    aGen[i]:=TPembangkit.Create(Nama,Pmin,Pmax,a2,a1,a0,Sh,Sc,Ramp,
    Tup,Tdown,Tcold,InitSt);
    frmInput.fgGen.Cells[1,i]:=Nama;
    frmInput.fgGen.Cells[2,i]:=FloatToStr(Pmax);
  end;
  end;
end;
end;

```

```

frmInput.fgGen.Cells[3,i]:=FloatToStr(Pmin);
frmInput.fgGen.Cells[4,i]:=FloatToStr(a0);
frmInput.fgGen.Cells[5,i]:=FloatToStr(a1);
frmInput.fgGen.Cells[6,i]:=FloatToStr(a2);
frmInput.fgGen.Cells[7,i]:=IntToStr(Tup);
frmInput.fgGen.Cells[8,i]:=IntToStr(Tdown);
frmInput.fgGen.Cells[9,i]:=FloatToStr(Sh);
frmInput.fgGen.Cells[10,i]:=FloatToStr(Sc);
frmInput.fgGen.Cells[11,i]:=IntToStr(Tcold);
frmInput.fgGen.Cells[12,i]:=IntToStr(InitSt);
frmInput.fgGen.Cells[13,i]:=FloatToStr(Ramp);
end;
frmInput.fgLoad.RowCount:=Njam+1;
SetLength(aLoad,Njam+1);
SetLength(aRes,Njam+1);
for i:=1 to Njam do
begin
  Readln(output,Load,Res);
  aLoad[i]:=Load;
  aRes[i]:=Res;
  frmInput.fgLoad.Cells[1,i]:=FloatToStr(Load);
  frmInput.fgLoad.Cells[2,i]:=FloatToStr(Res);
end;
frmInput.fgPLN.RowCount:=Ngen+1;
frmInput.fgPLN.ColCount:=Njam+1;
SetLength(aPLN,Ngen+1,Njam+1);
for i:=1 to Ngen do
begin
  for j:=1 to Njam do
  begin
    Read(output,Load);
    aPLN[i,j]:=Load;
    frmInput.fgPLN.Cells[j,i]:=FloatToStr(Load);
  end;
  Readln(output);
end;
CloseFile(output);
gObjFunc:=TObjFuncLR.Create(aLoad,aRes,aPLN,aGen,0);
for i:=1 to Ngen do
begin
  aGen[i].Free;
end;
frmInput.Caption:='Tampilan Data';
frmInput.btnNext.Caption:='&Next';
frmInput.ShowModal;
finally

```

```
    frmInput.Free;
  end;
end;
except
  MessageDlg('File Corrupt atau Error Program!',mtWarning,[mbOK],0);
end;
end;
```

```
procedure TfrmMenu.btnExitClick(Sender: TObject);
begin
  try
    gObjFunc.Free;
  finally
    Application.Terminate;
  end;
end;

end.
```