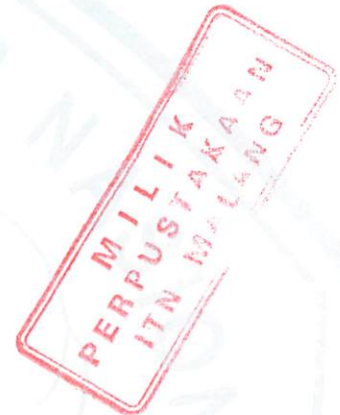


**INSTITUT TEKNOLOGI NASIONAL MALANG  
FAKULTAS TEKNOLOGI INDUSTRI  
JURUSAN TEKNIK ELEKTRO  
KONSENTRASI TEKNIK ENERGI LISTRIK (S-1)**



**ANALISIS PERBANDINGAN PEMBEBANAN EKONOMIS DENGAN  
MENGUNAKAN KOMBINASI *GENETIC ALGORITHM-  
EVOLUTIONARY PROGRAMMING-EVOLUTION STRATEGIES*  
DI PT. PEMBANGKITAN JAWA-BALI**

**SKRIPSI**

**Disusun oleh ;  
MAHRUS ALI  
00.12081**

**SEPTEMBER 2006**

DECLASSIFIED COPY

DO NOT  
ALTER OR  
REPRODUCE

SECRET

ON 11 JULY 1964 THE  
SECRETARY OF DEFENSE  
ORDERED THE DECLASSIFICATION  
OF ALL INFORMATION CONTAINED  
HEREIN EXCEPT WHERE SHOWN  
OTHERWISE

EXEMPT FROM AUTOMATIC  
DECLASSIFICATION  
EXCEPT WHERE SHOWN  
OTHERWISE

**LEMBAR PERSETUJUAN**

**ANALISIS PERBANDINGAN PEMBEBANAN EKONOMIS DENGAN  
MENGUNAKAN KOMBINASI *GENETIC ALGORITHM-  
EVOLUTIONARY PROGRAMMING-EVOLUTION STRATEGIES*  
DI PT. PEMBANGKITAN JAWA-BALI**

**SKRIPSI**

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

**Disusun Oleh :**

**MAHRUS ALI  
NIM. 00.12.081**



**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**

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

## ABSTRAKSI

### ANALISA PERBANDINGAN PEMBEBANAN EKONOMIS MENGUNAKAN KOMBINASI *GENETIC ALGORITHM- EVOLUTIONARY PROGRAMMING-EVOLUTIONARY STRATEGY* DI PT. PEMBANGKITAN JAWA-BALI

(Mahrus Ali, Nim. 00.12.081, Teknik Elektro/Teknik Energi Listrik S-1)  
(Dosen Pembimbing : Ir Yusuf Ismail Nakhoda, MT)

**Kata Kunci:** *Economic Load Dispatch, Genetic Algorithm, Evolutionary Programming Dan Evolutionary Strategy.*

Adanya persoalan dalam menghadapi kebutuhan daya listrik yang tidak tetap dari waktu ke waktu, sehingga menimbulkan permasalahan yaitu bagaimana mengoperasikan suatu sistem tenaga listrik yang selalu dapat memenuhi permintaan daya pada setiap saat dengan biaya bahan bakar yang murah. Oleh karena itu pada suatu operasi pada beban tertentu, perhitungan ekonomis harus tetap merupakan suatu prioritas atau nilai yang harus diperhitungkan disamping hal-hal lain, sehingga nantinya diperlukan suatu rencana operasi yang optimum dengan tetap memenuhi beberapa persyaratan pengoperasian sistem tenaga listrik yaitu antara lain : daya yang dibangkitkan cukup untuk memasok beban.

Skripsi ini menganalisis *Economic Load Dispatch* dengan menggunakan perbandingan kombinasi metode *Genetic Algorithm-Evolution strategies* (GA-ES) dengan kombinasi metode *Evolutionary Programming-Evolutionary Strategy* (EP-ES). Hasil dari analisa tersebut nantinya dapat digunakan sebagai salah satu acuan dalam operasi pembangkitan yang ekonomis dan optimal, terutama mengenai optimasi biaya pembangkitan. Masukan dari program ini adalah data pembangkitan dan pembebanan. Hasil akhir dari program ini yaitu hasil perhitungan biaya pembangkitan yang optimum.

Analisa dilakukan dengan bantuan program komputer dengan menggunakan bahasa pemrograman Delphi versi 7.0 dan dicoba pada data unit-unit pembangkit termal pada PT. PJB, dimana telah memperoleh selisih penghematan biaya pembangkitan.

## KATA PENGANTAR

Dengan rahmat Allah SWT, dan mengucapkan syukur kehadirat-Nya atas karunia yang dilimpahkan kepada saya sehingga dapat menyelesaikan skripsi yang berjudul **“ANALISIS PERBANDINGAN PEMBEBANAN EKONOMIS DENGAN MENGGUNAKAN METODE *GENETIC ALGORITHM-EVOLUTIONARY PROGRAMMING-EVOLUTIONARY STRATEGY* PADA PT. PEMBANGKITAN JAWA-BALI”**.

Skripsi ini bertujuan untuk memenuhi kurikulum akademik yang harus ditempuh oleh setiap mahasiswa ITN Malang dalam menempuh sekaligus mengakhiri pendidikan pada jenjang S-1 pada jurusan Teknik Elektro konsentrasi Energi Listrik.

Saya menyadari bahwa skripsi ini masih jauh dari kesempurnaan, karena itu saran dan kritik membangun sangat saya harapkan.

Atas segala bimbingan, pengarahan dan bantuan yang diberikan, sehingga tersusun skripsi ini, maka penulis menyampaikan terima kasih kepada:

1. Bapak Dr. Ir. Abraham Lomi, MSEE, selaku Rektor ITN Malang.
2. Bapak Ir. F Yudi Limpraptono, MT, selaku Kajar Teknik Elektro ITN Malang.
3. Bapak Ir. Yusuf Ismail Nakhoda, MT, selaku Dosen Pembimbing.
4. Bapak dan Ibu dosen Jurusan Teknik Elektro Energi Listrik S-1.
5. PT PLN ( PERSERO ).
6. Kedua Orang Tua dan seluruh keluargaku atas do'a restunya.
7. Teman-teman di Jurusan Teknik Elektro Institut Teknologi Nasional Malang, terutama angkatan 2000 yang telah membantu dalam penyelesaian skripsi ini.

8. Semua pihak yang tidak dapat disebutkan satu per satu, yang telah membantu dalam penyelesaian skripsi ini.

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

Akhirnya penyusun berharap semoga dalam skripsi ini dapat membantu serta bermanfaat khususnya bagi rekan-rekan mahasiswa pada Jurusan Teknik Elektro Energi Listrik..

Malang, September 2006

Penyusun

## DAFTAR ISI

<b>HALAMAN JUDUL .....</b>	<b>i</b>
<b>LEMBAR PERSETUJUAN .....</b>	<b>ii</b>
<b>ABSTRAKSI.....</b>	<b>iii</b>
<b>KATA PENGANTAR.....</b>	<b>iv</b>
<b>DAFTAR ISI .....</b>	<b>vi</b>
<b>DAFTAR GAMBAR.....</b>	<b>xii</b>
<b>DAFTAR TABEL.....</b>	<b>xiv</b>
<b>DAFTAR GRAFIK.....</b>	<b>xvi</b>
<b>BAB I PENDAHULUAN</b>	
1.1. Latar Belakang .....	1
1.2. Permasalahan.....	2
1.3. Tujuan .....	2
1.4. Batasan Masalah.....	3
1.5. Metodologi Penelitian .....	3
1.6. Sistematika Penulisan.....	4
1.7. Kontribusi.....	5
<b>BAB II TEORI DASAR</b>	
2.1. Operasi Sistem Tenaga Listrik .....	6
2.2. Karakteristik Pembangkit Tenaga Listrik .....	10
2.2.1. Karakteristik Masukan Keluaran ( <i>Input-ouput Characteristic</i> ).....	11
2.2.2. Karakteristik Hate Rate.....	12

2.3. Komitmen Unit.....	13
2.3.1. Kndala Pada Komitmen Unit.....	16
2.3.2. Biaya <i>Star-up</i> .....	18
2.3.3. Fungsi Biaya Bahan Bakar .....	18
2.4. Pembebanan Ekonomis Pembangkit Listrik .....	19
2.4.1. Penyelesaian <i>Economic Load Dispatch</i> dengan Metode Pengali Lagrange .....	19
2.4.2. Penyelesaian <i>Economic Load Dispatch</i> dengan Metode Iterasi Lamda .....	22

**BAB III ANALISA PEMBEBANAN EKONOMIS MENGGUNAKAN  
KOMBINASI *GENETIC ALGORITHM-EVOLUTIONARY  
PROGRAMMING-EVOLUTION STRATEGIES***

3.1. <i>Genetic Algorithm</i> .....	23
3.2. <i>Evolutionary Programming</i> .....	25
3.3. <i>Evolution Strategies</i> .....	26
3.4. <i>Genetic Algorithm-Evolution Stretegies (GA-ES)</i> .....	28
3.4.1. Parameter <i>Genetic Algorithm-Evolution Stretegies (GA-ES)</i> .....	28
3.4.1.1. Jumlah Generasi ( <i>Maxgen</i> ).....	28
3.4.1.2. Ukuran Populasi ( <i>Popsiz</i> e) .....	28
3.4.1.3. Probabilitas <i>Crossover</i> ( <i>Pc</i> ).....	29
3.4.1.4. Probabilitas <i>Mutasi</i> ( <i>Pm</i> ).....	29
3.4.1.5. Panjang <i>Kromosom</i> ( <i>NVAR</i> ).....	30



3.4.2. Mekanisme <i>Genetic Algorithm-Evolution Strategies</i> (GA-ES) .....	30
3.4.2.1. Pengkodean atau Representasi .....	30
3.4.2.2. Fungsi <i>Fitness</i> (Fungsi Evaluasi) .....	31
3.4.2.3. Seleksi.....	32
3.4.2.4. <i>Rank Seleksion</i> .....	32
3.4.2.5. <i>Crossover</i> .....	33
3.4.2.6. Mutasi ( <i>Mutation</i> ).....	34
3.5. <i>Evolutioanry Programming-Evolution Strategies</i> (EP-ES) .....	35
3.5.1. Parameter <i>Evolutioanry Programming-Evolution Strategies</i> (EP-ES).....	35
3.5.1.1. Jumlah Generasi ( <i>Maxgen</i> ).....	35
3.5.1.2. Ukuran Populasi ( <i>Popsiz</i> e) .....	36
3.5.1.3. Probabilitas Mutasi ( <i>Pm</i> ).....	36
3.5.1.4. Panjang <i>Kromosom</i> ( <i>NVAR</i> ).....	36
3.6. Mekanisme <i>Evolutioanry Programming-Evolution Strategies</i> (EP-ES).....	37
3.6.1. Pengkodean.....	37
3.6.2. Fungsi Evaluasi (Fungsi <i>Fitness</i> ).....	37
3.6.3. Seleksi ( <i>Selection</i> ).....	38
3.6.3.1. <i>Roulette Whell Selection</i> .....	38
3.6.3.2. <i>Rank Selection</i> .....	39
3.6.4. Rekombinasi ( <i>Recombination</i> ) .....	39
3.6.5. Mutasi ( <i>Mutation</i> ).....	40
3.6.6. Kompetisi ( <i>Competition</i> ) .....	41
3.7. Formulasi Masalah Pembebanan Ekonomis .....	42

3.8. Adaptasi <i>Evolutionary Programming</i> dan <i>Evolutionary Strategies</i> ke Masalah Pembebanan Ekonomis .....	43
3.8.1. Representasi Solusi .....	43
3.8.2. Inisialisasi ( <i>Initialization</i> ) .....	43
3.8.3. Statistik ( <i>Statistik</i> ) .....	44
3.8.4. Mutasi ( <i>Mutation</i> ) .....	44
3.8.4.1. Mutasi <i>Evolutionary Programming</i> .....	44
3.8.4.2. Mutasi <i>Evolution Strategies</i> .....	44
3.8.5. Kompetisi ( <i>Competition</i> ) .....	45
3.8.6. Konvergensi ( <i>Convergence</i> ) .....	46
3.9. Algoritma dan Flowchart .....	47
3.9.1. Algoritma Pembebanan Ekonomis menggunakan metode <i>Genetic Algorithm-Evolution Strategy</i> (GE-ES) .....	47
3.9.2. Flowchart Pembebanan Ekonomis Dengan Metode <i>Genetic Algorithms Dan Evolution Strategy</i> .....	49
3.9.3. Algoritma Pembebanan Ekonomis menggunakan metode <i>Evolutionary Programming-Evolution Strategy</i> (EP-ES) .....	50
3.9.4. Flowchart Pembebanan Ekonomis Dengan Metode <i>Evolutionary Programming -Evolution Strategy</i> .....	52

## **BAB IV ANALISA DAN HASIL PROGRAM**

4.1. Penggunaan Program Komputer Untuk Menyelesaikan Masalah <i>Economic Load Dispatch</i> di pembangkitan jawa-bali .....	53
4.2. Data Pembangkit Termal.....	53

4.3. Aplikasi Metode <i>Genetic Algorithm-Evolution Strategis</i> (GA-ES) dan Metode <i>Evolutionary Programming-evolution Strategis</i> (EP-ES) Pada PT. PLN PJB Untuk Memecahkan Permasalahan <i>Economic Load Dispatch</i> .....	56
4.4. Beban Sistem.....	57
4.5. Uji Validasi .....	58
4.5.1. Tampilan Program Uji Validasi Pembebanan Ekonomis dengan Metode <i>Genetic Algorithm</i> dan <i>Evolution Strategis</i> (GA-ES).....	60
4.5.2 Tampilan Program Uji Validasi Pembebanan Ekonomis dengan Metode <i>Evolutionary Programming</i> dan <i>Evolution Strategis</i> (GA-ES) .....	62
4.5.3. Hasil Uji Validasi Pembebanan Ekonomis dengan Metode <i>Evolutionary Proqramming</i> dan <i>Evolution Strategis</i> (EP-ES) .....	64
4.6. Hasil Perhitungan dan Analisa Data .....	65
4.6.1. Tampilan Program <i>Economic Load Dispatch</i> dengan Metode <i>Genetic Algorithm</i> dan <i>Evolution Strategis</i> (GA-ES) .....	66
4.6.2. Hasil Perhitungan Biaya Operasional PT. PLN PJB dan Biaya Operasional Menggunakan Metode GA-ES.....	70
4.6.2.1. Hari Sabtu, 25 Maret 2006 .....	70
4.6.2.2. Hari Minggu,26 Maret 2006.....	72
4.6.2.3. Hari Rabu, 29 Maret 2006 .....	74
4.6.3. Perbandingan Total Biaya PT. PLN PJB dengan Metode GA-ES ....	76
4.6.4. Tampilan Program <i>Economic Load Dispatch</i> dengan Metode <i>Evolutionary Proqramming</i> dan <i>Evolution Strategis</i> (GA-ES) .....	77

4.6.5. Hasil Perhitungan Biaya Operasional PT. PLN PJB dan Biaya Operasional Menggunakan Metode EP-ES.....	81
4.6.5.1. Hari Sabtu, 25 Maret 2006 .....	81
4.6.5.2. Hari Minggu, 26 Maret 2006.....	83
4.6.5.3. Hari Rabu 29 Maret 2006 .....	85
4.6.6. Perbandingan Total Biaya PT. PLN PJB dengan Metode EP-ES.....	87
4.6.7. Perbandingan Total Biaya Operasional Metode GA-ES Dan Metode EP-ES Terhadap PT. PLN PJB.....	88

## **BAB V KESIMPULAN DAN SARAN**

5.1 Kesimpulan .....	90
5.2 Saran .....	91

<b>DAFTAR PUSTAKA .....</b>	<b>92</b>
-----------------------------	-----------

## **LAMPIRAN**

## DAFTAR GAMBAR

Gambar 2.1	Elemen Pokok Sistem Distribusi.....	7
Gambar 2.2	Unit Boiler–Turbin–Generator.....	10
Gambar 2.3	Kurva karakteristik <i>Input-Output</i> Pembangkit Termal .....	12
Gambar 2.4	Kurva Karakteristik <i>Hate–Rate</i> Unit Pembangkit.....	13
Gambar 2.5	Sistem Interkoneksi .....	14
Gambar 2.6	N unit melayani beban $P_R$ .....	20
Gambar 3.1	Ilustrasi operator dengan <i>One Point Crossover</i> .....	33
Gambar 3.2	Ilustrasi operator dengan <i>Two Point Crossover</i> .....	34
Gambar 3.3	Ilustrasi operator <i>crossover</i> dengan <i>uniform crossover</i> .....	34
Gambar 3.4	Ilustrasi operator mutasi untuk representasi string biner .....	35
Gambar 3.5	<i>Intermediate recombination</i> dari Induk ( <i>parent</i> ) a) dan b) menjadi Anak ( <i>offspring</i> ) c) .....	40
Gambar 3.6	Mutasi Gaussian Dari Induk ( <i>parent</i> ) (a) Menghasilkan Anak ( <i>offspring</i> ) (b).....	41
Gambar 3.7	<i>Flowchart</i> Pembebanan Ekonomis Dengan Metode <i>Genetic Algorithms</i> Dan <i>Evolution Strategy</i> (GA-ES) .....	49
Gambar 3.8	<i>Flowchart</i> Economic Load Dispatch Dengan Metode Evolutionary Programming dan Evolution Strategy (EP-ES) .....	52
Gambar 4.1	Tampilan Menu Utama Uji Validasi GA-ES .....	60
Gambar 4.2	Tampilan Data dengan 10 Unit Pembangkit Selama 1 Jam Untuk Uji Validasi GA-ES.....	60
Gambar 4.3	Tampilan Parameter GA-ES Untuk Uji Validasi .....	61

Gambar 4.4 Hasil Uji Validasi GA-ES .....	61
Gambar 4.5 Tampilan Menu Uji Validasi EP-ES .....	62
Gambar 4.6 Tampilan Data dengan 10 Unit Pembangkit Selama 1 Jam Untuk Uji Validasi EP-ES.....	62
Gambar 4.7 Tampilan Parameter EP-ES Untuk Uji Validasi .....	63
Gambar 4.8 Hasil Uji Validasi EP-ES .....	63
Gambar 4.9 Tampilan Menu Utama program GA-ES .....	66
Gambar 4.10 Tampilan Secara Umum GA-ES .....	66
Gambar 4.11 Tampilan Data Generaor GA-ES .....	67
Gambar 4.12 Tampilan Data Pembebanan GA-ES.....	67
Gambar 4.13 Tampilan Data PLN GA-ES.....	68
Gambar 4.14 Tampilan Parameter GA-ES.....	68
Gambar 4.15 Tampilan Daya dari Unit yang Beroperasi Serta Hasil Biaya GA-ES .....	69
Gambar 4.16 Tampilan Menu Utama program GA-ES .....	77
Gambar 4.17 Tampilan Secara Umum EP-ES .....	77
Gambar 4.18 Tampilan Data Generaor EP-ES .....	78
Gambar 4.19 Tampilan Data Pembebanan EP-ES .....	78
Gambar 4.20 Tampilan Data PLN EP-ES.....	79
Gambar 4.21 Tampilan Parameter EP-ES.....	79
Gambar 4.22 Tampilan Daya dari Unit yang Beroperasi Serta Hasil Biaya EP-ES .....	80

## DAFTAR TABEL

Tabel 4.1	Data Unit Termal Pada PT. Pembangkitan Jawa-Bali Agustus 2002 .....	54
Tabel 4.2	Data Biaya dan Parameter unit Termal Pada PT. Pembangkitan Jawa-Bali Agustus 2002 .....	55
Tabel 4.3	Parameter <i>Genetic Algorithm, Evolutionary Programming</i> dan <i>Evolution Strategies</i> .....	58
Tabel 4.4	Data <i>Economicload Load Dispatch</i> Dengan 10 Unit Pembangkit Untuk Pengujian Validasi Program .....	59
Tabel 4.5	Data Beban Masing-Masing Unit dengan Total 2400MW dan Cadangan Berputar 240MW .....	59
Tabel 4.6	Hasil Uji Validasi .....	64
Tabel 4.7	37 Unit Pembangkit Pada PT. PLN PJB .....	65
Tabel 4.8	Hasil Perhitungan Biaya Operasional PT. PLN PJB dan Biaya Operasional Menggunakan Metode GA-ES Pada Hari: Sabtu, 25 Maret 2006 .....	70
Tabel 4.9	Hasil Perhitungan Biaya Operasional PT. PLN PJB dan Biaya Operasional Menggunakan Metode GA-ES Pada Hari: Minggu, 26 Maret 2006.....	72
Tabel 4.10	Hasil Perhitungan Biaya Operasional PT. PLN PJB dan Biaya Operasional Menggunakan Metode GA-ES Pada Hari: Rabu, 29 Maret 2006 .....	74

Tabel 4.11 Perbandingan Total Biaya Oprasional PT. PLN PJB dengan Metode GAES.....	76
Tabel 4.12 Hasil Perhitungan Biaya Operasional PT. PLN PJB dan Biaya Operasional Menggunakan Metode EP-ES Pada Hari: Sabtu, 25 Maret 2006 .....	81
Tabel 4.13 Hasil Perhitungan Biaya Operasional PT. PLN PJB dan Biaya Operasional Menggunakan Metode EP-ES Pada Hari: Minggu, 26 Maret 2006.....	83
Tabel 4.14 Hasil Perhitungan Biaya Operasional PT. PLN PJB dan Biaya Operasional Menggunakan Metode EP-ES Pada Hari: Rabu, 29 Maret 2006 .....	85
Tabel 4.15 Perbandingan Total Biaya Oprasional PT. PLN PJB dengan Metode EP-ES .....	87
Tabel 4.16 Perbandingan Total Biaya Operasional Metode GA-ES Dan Metode EP-ES Terhadap PT. PLN PJB.....	88



## DAFTAR GRAFIK

Grafik 4.1 Perbandingan Biaya Operasional P.T. PLN PJB dengan Biaya Operasional Menggunakan Metode GA-ES Pada Hari: Sabtu, 25 Maret 2006 .....	71
Grafik 4.2 Perbandingan Biaya Operasional P.T. PLN PJB dengan Biaya Operasional Menggunakan Metode GA-ES Pada Hari: Minggu, 26 Maret 2006.....	73
Grafik 4.3 Perbandingan Biaya Operasional P.T. PLN PJB dengan Biaya Operasional Menggunakan Metode GA-ES Pada Hari: Minggu, 26 Maret 2006.....	75
Grafik 4.4 Selisih Perbandingan Total Biaya Operasional PT. PLN PJB dengan Total Biaya Operasional Metode GA-ES Pada Tanggal 25, 26 dan 29 Maret 2006.....	76
Grafik 4.5 Perbandingan Biaya Operasional P.T. PLN PJB dengan Biaya Operasional Menggunakan Metode EP-ES Pada Hari: Sabtu, 25 Maret 2006 .....	82
Grafik 4.6 Perbandingan Biaya Operasional P.T. PLN PJB dengan Biaya Operasional Menggunakan Metode EP -ES Pada Hari: Minggu, 26 Maret 2006.....	84
Grafik 4.7 Perbandingan Biaya Operasional P.T. PLN PJB dengan Biaya Operasional Menggunakan Metode EP -ES Pada Hari: Minggu, 26 Maret 2006.....	86
Grafik 4.8 Selisih Perbandingan Total Biaya Operasional PT. PLN PJB	

dengan Total Biaya Operasional Metode GA-ES  
Pada Tanggal 25, 26 dan 29 Maret 2006..... 87

**Grafik 4.9 Perbandingan Total Biaya Operasional**

**Metode GA-ES dan Metode EP-ES Terhadap PT. PLN PJB**  
Pada Tanggal 25, 26 dan 29 Maret 2006..... 88

**Grafik4.10 Selisih Perbandingan Total Biaya Operasional PT. PLN PJB**

**dengan Total Biaya Operasional Metode GA-ES**  
Pada Tanggal 25, 26 dan 29 Maret 2006..... 89

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Pembangkit tenaga listrik merupakan bagian dari permasalahan energi dan lingkungan yang dihadapi oleh Indonesia sebagai negara berkembang. Secara garis besar, suatu sistem tenaga listrik dapat dibagi menjadi 3 bagian, yaitu : sisi pembangkit tenaga listrik, saluran transmisi dan jaringan distribusi atau beban. Untuk suatu operasi pada beban tertentu, perhitungan ekonomis harus tetap merupakan suatu prioritas atau nilai yang harus diperhitungkan disamping hal-hal yang lain, sehingga nantinya diperlukan suatu rencana operasi yang optimum dengan tetap memenuhi beberapa persyaratan pengoperasian sistem tenaga listrik, yaitu antara lain : daya yang dibangkitkan cukup untuk memasok beban, tegangan bus harus sesuai dengan ratingnya serta tidak adanya pembebanan lebih unit-unit pembangkit yang beroperasi.

Dalam pembangkitan tenaga listrik, dilakukan usaha agar dihasilkan biaya pembangkitan semurah mungkin. Usaha untuk mengoptimalkan biaya operasi ini, salah satunya dilakukan dengan penerapan pembebanan ekonomis yang dihitung dengan kombinasi metode *Genetic Algorithm-Evolution Strategies* (GA-ES) dan kombinasi metode *Evolutionary Programming-Evolutionary Strategies* (EP-ES).

. Di dalam operasi sistem tenaga listrik, Pembebanan ekonomis adalah hal yang sangat perlu diperhatikan untuk mendapatkan biaya bahan bakar yang sangat ekonomis dalam suatu sistem pembangkit.

Koordinasi antara unit-unit pembangkit yang ada pada sistem tenaga listrik sangat diperlukan untuk mencapai biaya operasi yang seoptimum mungkin, dalam

hal ini yang dimaksud adalah optimum secara ekonomis dengan tetap memperhatikan besar beban yang ada, sehingga dengan demikian nantinya akan didapatkan nilai keuntungan pada PT. PLN (Persero) sebagai perusahaan penyedia energi listrik di Indonesia saat ini.

## **1.2. Permasalahan**

Berdasarkan latar belakang, dijelaskan bahwa biaya pembangkitan sangat berhubungan dan berpengaruh terhadap koordinasi antara unit-unit pembangkit yang menyalurkan tenaga listrik pada beban yang berubah-ubah. Maka muncul permasalahan, yaitu bagaimana mengoptimalkan biaya pembangkitan yang dihasilkan oleh pembangkit tenaga listrik thermal. Sehubungan dengan permasalahan tersebut, maka skripsi ini diberi judul:

**“ANALISA PERBANDINGAN PEMBEBANAN EKONOMIS  
MENGUNAKAN KOMBINASI *GENETIC ALGORITHM-  
EVOLUTIONARY PROGRAMMING-EVOLUTION STRATEGIES*  
DI P.T. PEMBANGKITAN JAWA-BALI”**

## **1.3. Tujuan**

Tujuan dari skripsi ini adalah memberikan analisis penerapan pembebanan ekonomis dengan menggunakan perbandingan metode *Genetic Algorithm-Evolution Strategis* (GA-ES) dan metode *Evolutionary Programming-Evolutionary Strategies* (EP-ES) untuk mengoptimalkan biaya bahan bakar pada suatu sistem

tenaga listrik, dimana analisa dalam skripsi ini hanya pada sistim tenaga listrik di PT. Pembangkitan Jawa Bali.

#### **1.4. Batasan Masalah**

Untuk membatasi apa saja yang dibahas, maka ditentukan pada skripsi ini :

- ❖ Tidak membahas masalah rugi-rugi transmisi.
- ❖ Pembahasan dititikberatkan pada segi ekonomis, dan hanya menyangkut optimasi biaya bahan bakar tidak membahas segi teknis.
- ❖ Unit pembangkit yang dihitung hanya pembangkit thermal yang termasuk dalam area pembangkit Jawa Bali.
- ❖ Tidak membahas *combined cycle* pada PLTGU.
- ❖ Tidak membahas masalah biaya cadangan berputar (*spinning reserve*), hanya memperhatikan kendala batasan cadangan berputar.
- ❖ Analisa dilakukan dengan asumsi bahwa sistem dalam operasi normal.
- ❖ Metode yang digunakan adalah perbandingan kombinasi metode antara *Genetic Algorithm-Evolution Strategis (GA-ES)* dan kombinasi metode *Evolutionary Programming-Evolutionary Strategies (EP-ES)*.

#### **1.5. Metode Penelitian**

Metodologi yang dibahas dalam skripsi ini dilakukan dengan langkah-langkah sebagai berikut:

- 1 Studi kepustakaan mengenai hal-hal yang berhubungan dengan pembahasan masalah.

- 2 Studi lapangan untuk mendapatkan data parameter unit thermal yang dibutuhkan dari obyek penelitian yaitu di PT. PJB yang diperlukan, berpedoman pada teori yang diperoleh dari studi kepustakaan.
- 3 Pehitungan Pembebanan ekonomis dengan menggunakan perbandingan kombinasi metode *Genetic Algorith-Evolution Strategis (GA-ES)* dan kombinasi metode *Evolutionary Progamming-Evolutionary Strategies (EP-ES)*.
- 4 Membuat evaluasi, sehingga dapat disimpulkan dari perhitungan antara sebelum dan sesudah optimasi.

#### **1.6. Sistematika Penulisan**

Adapun sistimatika pembahasan pada skripsi ini adalah sebagai berikut :

##### **BAB I : PENDAHULUAN**

Menguraikan latar belakang, rumusan masalah, tujuan, batasan masalah, metode penulisan, sistematika penulisan dan kontribusi penelitian.

##### **BAB II : TEORI DASAR**

Teori sistem tenaga listrik, pembebanan ekonomis, *genetic algorith, evolutionary programming* dan *evolutionary Strategies*.

### **BAB III: ANALISA PEMBEBANAN EKONOMIS**

#### **MENGGUNAKAN KOMBINASI *GENETIC ALGORITHM-EVOLUTIONARY PROGRAMMING-EVOLUTION STRATEGIES***

Menguraikan teori dasar dari *economic load dispatch*, teori serta adaptasi *genetic algorithm-evolution Strategies* dan adaptasi *evolutionary programming-evolutionary Strategies* ke permasalahan pembebanan secara ekonomis.

### **BAB IV : ANALISA DAN HASIL PROGRAM**

Menguraikan alur program, hasil validasi, serta hasil perhitungan pembebanan ekonomis menggunakan perbandingan kombinasi metode *genetic algorithm-evolution Strategies* (GA-ES) dan kombinasi metode *evolutionary programming-evolutionary Strategies* (EP-ES).

### **BAB V : KESIMPULAN DAN SARAN**

Memuat intisari dan hasil pembahasan, yang berisikan kesimpulan dan saran yang dapat digunakan sebagai pertimbangan untuk pengembangan selanjutnya.

#### **1.6. Kontribusi**

Adapun kontribusi dari tugas akhir ini adalah diharapkan langkah efisiensi bisa diambil oleh PT. PJB, sehingga nantinya nilai kerugian yang diderita oleh PT. PJB bisa berkurang, dan pada akhirnya bisa menambah keuntungan bagi PT. PJB sebagai perusahaan penyedia energi listrik di Indonesia.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **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.

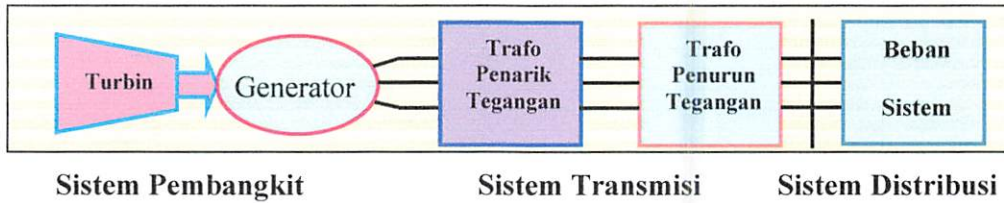
Karena berbagai persoalan teknis, tenaga listrik hanya dapat dibangkitkan pada lokasi tertentu saja. Mengingat tenaga listrik atau pelanggan tenaga listrik tersebar diberbagai tempat, maka penyaluran tenaga listrik dari tempat dibangkitkan sampai ke tempat pelanggan memerlukan berbagai penanganan teknis.

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<sup>[2]</sup>  
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. Syarat-syarat pemeliharaan peralatan.
- b. Keandalan yang diinginkan.
- c. Pengaturan dan penyaluran beban.
- d. Proses tenaga listrik yang ekonomis.

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 mengopersikan 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.

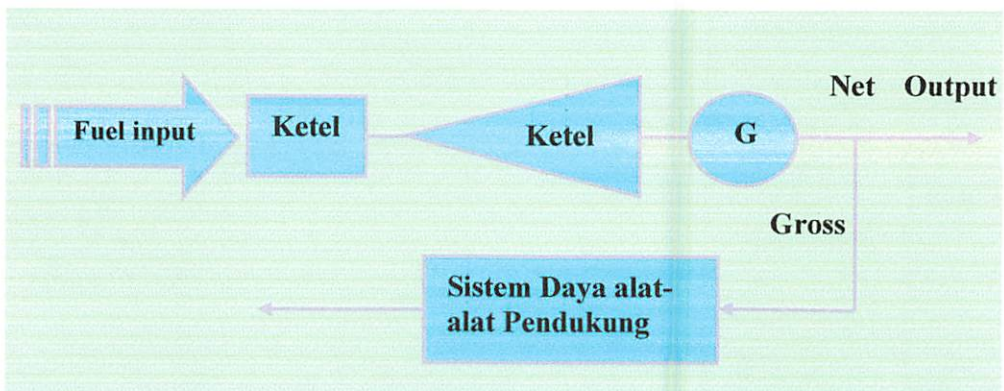
**f. Tegangan dalam sistem.**

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<sup>[2]</sup>  
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<sup>[8]</sup> :

$H$  dengan satuan  $[Mbtu/h]$

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

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

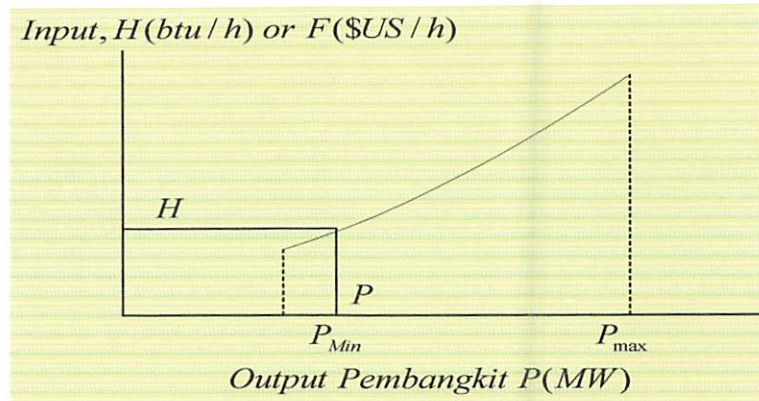
Sedangkan keluaran dari pembangkit termal adalah daya nyata yang dihasilkan oleh generator dikurangi daya nyata yang dipakai oleh pusat listrik tersebut. Notasi yang umum digunakan adalah<sup>[8]</sup>:

$P$  dengan satuan  $[MW]$

Jika dapat disimpulkan bahwa masukan pusat listrik merupakan fungsi terhadap keluarannya, maka hubungan tersebut dapat ditulis sebagai berikut<sup>[8]</sup>:

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

Pembahasan selanjutnya akan berpedoman atas dasar fungsi biaya bahan bakar ( $F=f(P) \quad [\$US/h]$ ), 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  $[\$US/h]$ . sedangkan keluaran adalah daya listrik  $[MW]$  yang dihasilkan blok tersebut untuk melayani beban sistem.



Gambar 2.3 <sup>[8]</sup>

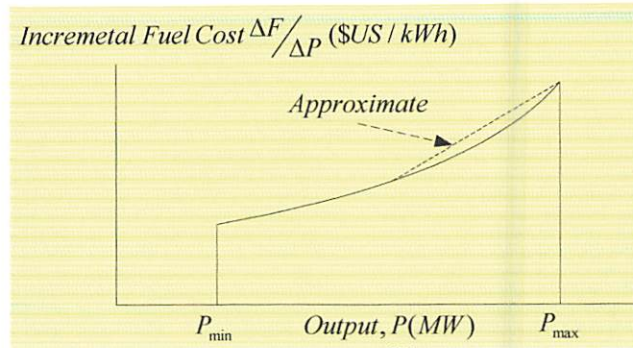
Kurva Karakteristik *Input – Output* Pembangkit Termal

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 <sup>[8]</sup>

Kurva Karakteristik Laju Tambahan Biaya Bahan Bakar

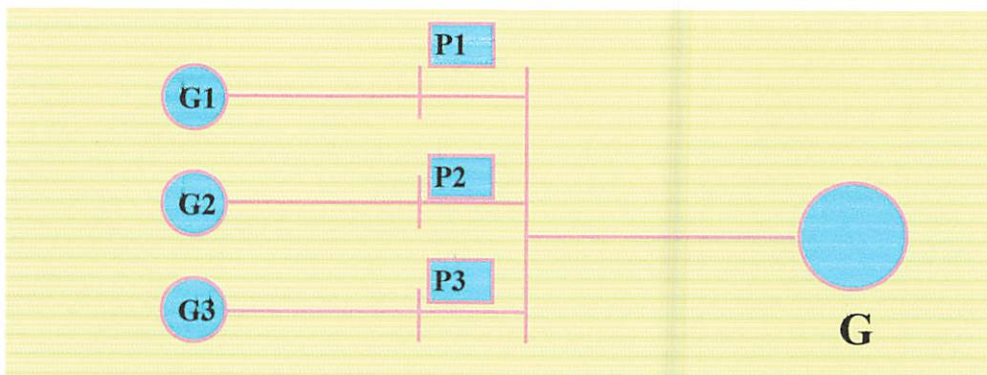
### 2.3 Komitmen Unit.

Kebutuhan akan energi listrik sudah mengalami pergeseran yang cukup signifikan dari masa lalu hingga sekarang, dimana pada masa lalu listrik hanya digunakan sebagai sarana penerangan saja, sedangkan pada masa sekarang listrik hampir sepenuhnya menjadi denyut nadi manusia seiring dengan perkembangan teknologi yang ada, dimana saat ini hampir semua peralatan yang dipakai manusia menggunakan listrik sebagai sumber energinya. Apalagi dengan perkembangan sektor industri dimana energi listrik menjadi bagian yang sangat vital untuk kelangsungan proses produksinya. Misalnya dalam siklus harian yang terbagi dalam interval waktu 1 jam selama 24 jam, beban listrik dalam sistem tenaga selalu berubah. Operasi pusat-pusat pembangkit di dalam sistem tenaga harus selalu dikoordinasikan dalam pembagian pembebanan secara optimal atau seekonomis mungkin pada setiap perubahan beban dalam interval waktu untuk siklus waktu tertentu, atau dikenal dengan istilah *Unit Commitment*.

Guna memenuhi kebutuhan akan energi listrik ini, maka diperlukan suatu pengaturan operasi sistem tenaga listrik yang sangat ekonomis dimana bukan hanya faktor pemenuhan kebutuhan beban yang harus dipenuhi tetapi juga faktor

kualitas, keandalan dan nilai ekonomis menjadi syarat mutlak yang harus diperhatikan.

Untuk mengatasi masalah diatas, maka sistem tenaga listrik yang ada sekarang ini menggunakan sistem interkoneksi, dimana prinsip dari sistem interkoneksi ini adalah menggabungkan beberapa pusat pembangkit yang tersebar diberbagai lokasi, baik unit *hidro* maupun termal secara paralel melalui suatu jaringan transmisi bertegangan tinggi untuk menyuplai beban gabungan (*infinite bus*). Hal ini bearti bahwa seluruh unit pembangkit yang berada dalam satu wilayah menjadi satu kesatuan yang terpadu. Untuk sistem interkoneksi yang besar, yang terdiri dari banyak unit pembangkit dan banyak pusat beban (gardu induk), sarana pengendalian operasi dengan menggunakan sarana komunikasi saja tidak cukup, tetapi harus ditambah dengan peralatan telemetering dan alat-alat pengolah data elektronis, seperti komputer. Disamping itu, pada pengoperasian sistem yang terinterkoneksi PLN berkewajiban menyediakan energi listrik dengan rating tegangan yang berada dalam batas-batas tertentu.



Gambar 2.5 <sup>[8]</sup>  
Sistem Interkoneksi



Tetapi pada kenyataannya, pengoperasian secara interkoneksi menimbulkan masalah teknis yang cukup rumit dan kompleks. Hal ini dikarenakan terbenturnya dua kepentingan PLN dalam menjalankan misinya. Misi utama PLN adalah sebagai perusahaan jasa teknik kelistrikan adalah mencari laba, dan untuk itu suatu langkah ekonomis harus diambil untuk memperoleh keuntungan yang maksimal atas modal yang ditanamkan, yaitu dengan mengoptimalkan pengoperasiannya. Mengoptimalkan pengoperasian berarti harus dicapai biaya operasi yang seminimal mungkin, khususnya biaya bahan bakar, mengingat bahwa biaya bahan bakar merupakan unsur terbesar dalam total biaya operasi. Kemudian misi dari PLN lainnya adalah mengolah dan menyediakan energi listrik bagi masyarakat dengan kualitas dan keandalan yang terbaik. Melihat kedua misi ini, maka diperlukan suatu perencanaan penyaluran dan penyediaan energi listrik yang memenuhi faktor kualitas dan keandalan dengan biaya yang seekonomis mungkin.

Komitmen Unit merupakan suatu metode solusi yang tepat untuk mengatasi permasalahan diatas guna mencari jadwal unit pembangkit yang harus beroperasi untuk periode waktu tertentu agar dicapai biaya operasi yang seekonomis mungkin. Pada masalah komitmen unit diasumsikan bahwa ada sejumlah  $N$  unit pembangkit yang tersedia dan harus dioperasikan untuk memenuhi permintaan beban.

### 2.3.1 Kendala Pada Komitmen Unit.<sup>[2][3]</sup>

Dalam pengoperasian pembangkit untuk memenuhi permintaan beban terdapat berbagai kendala yang merupakan syarat pembatas. Kendala tersebut antara lain :

- **Cadangan berputar (*Spinning Reserve*)**

Cadangan berputar adalah kata yang digunakan untuk menggambarkan jumlah total daya yang mungkin dibangkitkan dari semua unit yang tersinkronisasi (dihubungkan paralel) pada sistem dikurangi dengan beban saat itu dan rugi-rugi penyaluran yang terjadi. Cadangan berputar harus ada untuk menghindari penurunan frekwensi sistem yang terlalu besar bila satu atau lebih unit harus keluar dari sistem atau dengan kata lain harus ada cukup cadangan pada unit-unit lain untuk menutupi kekurangan suplai daya yang hilang dalam periode waktu tertentu. Cadangan berputar harus dialokasikan untuk menaati aturan-aturan tertentu, biasanya aturan yang dipakai adalah 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 harus mencukupi untuk menutupi untuk menutupi kegagalan pembangkit unit, tetapi cadangan-cadangan tersebut harus dialokasikan diantara unit-unit yang bereaksi lambat. Ini memungkinkan sistem kontrol pembangkit otomatis untuk mengembalikan frekwensi dan perputaran yang cepat pada saat unit pembangkit keluar dari sistem.

Besar cadangan berputar tersebut harus ditentukan secara hati-hati, sebab sering kali penentuan yang didasarkan untuk menjaga keandalan sistem berbenturan dengan biaya pengoperasian yang diusahakan seekonomis mungkin, misalnya jika cadangan berputar kecil dan unit pembangkit terbesar mengalami gangguan dan trip sehingga unit tersebut keluar secara mendadak dari sistem, maka cadangan berputar tersebut tidak cukup untuk mengatasi kekurangan pembangkitan yang terjadi dan untuk menghindari sistem *collaps* maka perlu dilakukan pelepasan beban dan ini mengakibatkan keandalan sistem menurun. Sehingga makin besar cadangan berputar dalam sistem, maka makin handal pula sistem tersebut dalam menghindari gangguan, Oleh karena itu perlu adanya suatu kesepakatan antara pemenuhan keandalan dan pengoptimalan biaya operasi.

- **Kendala Unit Termal**

Unit termal biasanya memerlukan “*crew*” untuk mengoperasikannya, terutama ketika dinyalakan ataupun pada saat dimatikan. Sebuah unit termal hanya dapat dijalankan dibawah perubahan temperatur yang gradual, dan ini diterjemahkan kedalam sebuah periode waktu dalam jam yang dibutuhkan untuk membawa unit tersebut *on-line*. Hal ini menyebabkan kendala-kendala antara lain:

- a. ***Minimum Up Time* (MUT).**

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

**b. *Minimum Down Time (MDT).***

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

**2.3.2 Biaya *Start-up*<sup>[2][8]</sup>**

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

**a. Biaya *star-up* pada kondisi dingin (*cooling*).**

Kondisi ini terjadi karena saat pembangkit dilepas dari sistem (tidak beroperasi), temperatur boiler dibiarkan turun dari temperatur dari temperatur kerjanya, sehingga pada saat beroperasi kembali perlu dilakukan pemanasan kembali.

**b. Biaya *start-up* dalam kondisi panas (*banking*).**

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

**2.3.3 Fungsi Biaya Bahan Bakar.**

Biaya bahan bakar merupakan unsur biaya yang paling penting dalam operasi sistem pembangkit termal. 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<sup>[3]</sup> :

$$F_i(P'_i) = a_i + b_i P'_i + c_i (P'_i)^2 \dots\dots\dots(2.1)$$

Dimana :

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

$P'_i$  = keluaran daya unit pembangkit pada jam  $t$

#### 2.4. Pembebanan Ekonomis Pembangkit Listrik.<sup>[2][8]</sup>

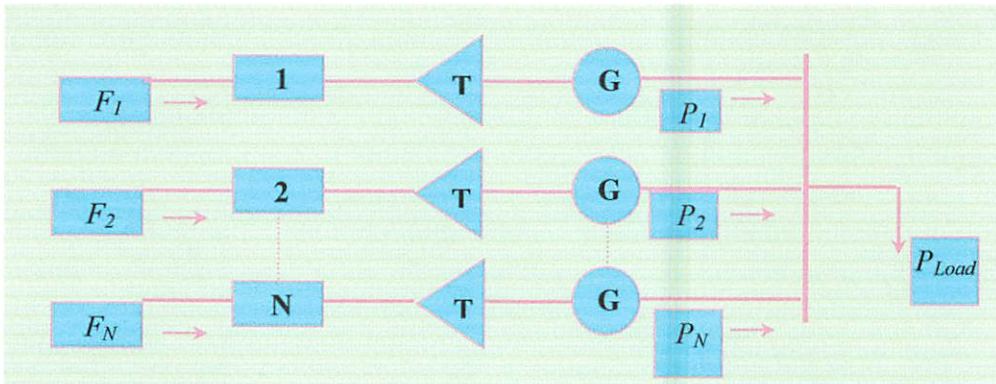
Pembebanan ekonomis atau *economic dispatch* adalah pembagian pembebanan pada pembangkit-pembangkit yang ada dalam sistem tenaga listrik, secara optimal dan ekonomis pada harga beban tertentu. Komponen terbesar dari biaya pembangkitan adalah bahan bakar. Oleh sebab itu dilakukannya *economic dispatch* berarti pula didapatkan biaya bahan bakar pembangkitan yang paling murah. Oleh karena beban yang ditanggung oleh sistem pembangkit selalu berubah setiap periode waktu tertentu, maka perhitungan *economic dispatch* ini dilakukan untuk setiap harga beban tertentu. Penyelesaian *economic dispatch* dapat dilakukan dengan beberapa cara yang akan dibahas pada sub bab dibawah ini.

##### 2.4.1 Penyelesaian *Economic Dispatch* dengan Metode Pengali *Lagrange*.

Sistem tenaga listrik dengan mengabaikan rugi-rugi transmisi dapat dilihat pada gambar 2.6 sistem ini memperlihatkan pembangkit termal yang terdiri dari  $N$  unit pembangkit yang dihubungkan pada sebuah bus bar untuk melayani total beban sebesar  $P_d$ . Masukan untuk setiap unit ke- $i$  adalah  $F_i$  yang menyatakan

tingkat biaya dari masing-masing unit, dan keluaran dari masing-masing unit  $P_i$  adalah daya listrik yang dibangkitkan oleh tiap-tiap unit.

Biaya total  $F_t$  yang ditanggung sistem adalah jumlah biaya dari tiap-tiap unit pembangkit. Dan batasan yang paling penting dari pengoperasian pembangkit termal adalah daya listrik yang dihasilkan harus sama dengan besarnya daya yang dibutuhkan oleh konsumen (beban listrik).



Gambar 2.6 <sup>[2]</sup>  
N Unit Pembangkit Termal Melayani Beban

Yang jadi permasalahan adalah meminimumkan biaya total  $F_t$  dengan memperhatikan kendala  $\phi$  bahwa daya yang dihasilkan unit pembangkit sama dengan yang diterima beban. Secara matematis pernyataan tersebut diatas dapat dinyatakan persamaan sebagai berikut <sup>[2]</sup>:

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

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

dan daya listrik yang dihasilkan oleh setiap unit untuk melayani beban total adalah

$$P_{Load} = \sum_{i=1}^N P_i \dots \dots \dots (2.3)$$

$$P_{Load} = \sum_{i=1}^N P_i = \phi = 0 \dots \dots \dots (2.4)$$

dimana :  $P_{Load}$  = kebutuhan beban, dan

$P_i$  = jumlah daya yang dihasilkan

penyelesaian permasalahan optimasi seperti ini dapat diselesaikan dengan metode yang menyangkut fungsi *lagrange* [2]:

$$\mathcal{L} = F_T + \lambda \varphi$$

atau

$$\mathcal{L} = \sum_{i=1}^N F_i(P_i) + \lambda \left( P_{Load} - \sum_{i=1}^N P_i \right) \dots\dots\dots(2.5)$$

dimana :  $\lambda$  adalah *lagrange multipliers*.

Untuk mencari harga optimal dari fungsi *lagrange* terhadap  $P_i$ , dapat diperoleh dengan operasi gradient dari persamaan *lagrange* sama dengan nol.

$$\Delta L = 0$$

$$\frac{\partial L}{\partial P_i} = \frac{\partial F_T}{\partial P_i} + \left( \frac{\partial P_D}{\partial P_i} - \frac{\partial P_i}{\partial P_i} \right) = 0$$

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

$$\frac{\partial F_i}{\partial P_i} = \lambda \dots\dots\dots(2.6)$$

Persamaan diatas menunjukkan bahwa distribusi beban yang optimal terjadi apabila semua unit pembangkit beroperasi pada tingkat laju tambahan biaya bahan bakar yang sama, yang ternyata sama dengan nilai  $\lambda$ . Kondisi optimal ini tentunya memerlukan persamaan-persamaan pembatas (*constrains*) agar keluaran dari setiap unit pembangkit harus lebih besar atau sama dengan keluaran minimum yang diijinkan.

Dari  $N$  buah unit pembangkit yang telah dibahas, maka dapat diambil kesimpulan sebagai berikut<sup>[2]</sup>:

$$\frac{\partial F_i}{\partial P_i} = \lambda \quad \text{ada } N \text{ buah persamaan}$$

$$P_{i,\min} \leq P_i \leq P_{i,\max} \quad \text{ada } 2N \text{ buah persamaan}$$

$$\sum_{i=1}^N P_i = P_{Load} \quad \text{ada } 1 \text{ buah persamaan}$$

batasan-batasan terdahulu dapat diperluas menjadi<sup>[2]</sup> :

$$\frac{\partial F_i}{\partial P_i} = \lambda \quad \text{untuk } P_{i,\min} \leq P_i \leq P_{i,\max}$$

$$\frac{\partial F_i}{\partial P_i} \leq \lambda \quad \text{untuk } P_i = P_{i,\max}$$

$$\frac{\partial F_i}{\partial P_i} \geq \lambda \quad \text{untuk } P_i = P_{i,\min}$$

#### 2.4.2 Penyelesaian Economic Dispatch dengan metode Iterasi Lamda.<sup>[2][8]</sup>

Pada metode iterasi lamda ( $\lambda$ ), lamda ditentukan dahulu kemudian dihitung harga keluaran untuk tiap-tiap unit pembangkit. Bila jumlah dari daya keluaran tiap-tiap unit pembangkit tidak sama dengan daya beban maka dilakukan perhitungan kembali terhadap harga lamda berikutnya. Bila lamda masukan untuk lamda yang pertama ditentukan menghasilkan daya keluaran yang lebih rendah dari daya beban dan harga lamda yang ditentukan kemudian menghasilkan daya keluaran lebih dari daya beban, maka dapat dilakukan interpolasi. Atau dapat pula dilakukan pada lamda pertama, penambahan untuk mendapatkan lamda berikutnya yang mendekati solusi sebenarnya, setelah beberapa kali iterasi.



## BAB III

### ANALISA PEMBEBANAN EKONOMIS MENGGUNAKAN KOMBINASI *GENETIC ALGORITHM- EVOLUTIONARY PROGRAMMING- EVOLUTION STRATEGIES*

#### 3.1. *Genetic Algorithm*<sup>[3][4][6]</sup>

*Genetic Algorithm* merupakan metode adaptif 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, *Genetic Algorithm* dapat digunakan untuk mencari solusi permasalahan-permasalahan dalam dunia nyata.

Algoritma Genetika 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 tipe 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 *Genetic Algorithm* dijalankan, maka sebuah kode yang sesuai (representasi) untuk persoalan harus dirancang. Titik solusi dalam ruang permasalahan dikodekan dalam bentuk kromosom atau *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 *Genetic Algorithm* 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
4. Menggunakan aturan transisi secara probabilistik bukan deterministik.

*Genetic Algorithm* 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.

Istilah-istilah yang digunakan dalam perumusan Algoritma Genetika sebagai berikut :

Tabel 3-1<sup>[3]</sup>

Istilah yang digunakan dalam *Genetic Algorithm*

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

### 3.2. *Evolutionary Programming*<sup>[3][4][6]</sup>

*Evolutionary programming* merupakan metode yang biasa digunakan untuk memecahkan suatu pencarian nilai dalam sebuah masalah optimasi. Metode ini didasarkan pada proses evolusi yang ada dalam makhluk hidup yaitu perkembangan generasi dalam sebuah populasi yang alami, secara lambat laun mengikuti mekanisme alam dimana individu yang lebih kuat memiliki kemungkinan untuk menjadi pemenang dan mempunyai kesempatan hidup yang lebih besar di dalam lingkungan yang kompetitif. Dengan meniru proses ini

*evolutionary programming* dapat digunakan untuk mencari solusi permasalahan-permasalahan dalam dunia nyata.

*Evolutionary programming* ditemukan oleh Lawrence.J. Fogel pada tahun 1960 yang dilandasi oleh sifat - sifat evolusi alam. Fogel 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. Fogel mulai bekerja dengan algoritma yang dibentuk dari string-string bilangan real yang disebut *kromosom*.

Sebelum *evolutionary programming* 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 real (*floating point*) sebagai nilai gen (*allele*) memungkinkan penerapan operator *Evolutionary programming* yaitu inisialisasi (*initialization*), statistik (*statistics*), mutasi (*mutation*), dan kompetisi (*competition*) 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, mutasi dan kompetisi untuk menciptakan keturunan (*offspring*).

### 3.3. *Evolutionary Strategies*<sup>[3][4][6]</sup>

*Evolutionary Strategies* merupakan metode yang biasa digunakan untuk memecahkan suatu pencarian nilai dalam sebuah masalah optimasi. Metode ini

didasarkan pada proses evolusi yang ada dalam makhluk hidup yaitu perkembangan generasi dalam sebuah populasi yang alami, secara lambat laun mengikuti mekanisme alam, dimana individu yang lebih kuat memiliki kemungkinan untuk menjadi pemenang dan mempunyai kesempatan hidup yang lebih besar di dalam lingkungan yang kompetitif. Dengan meniru proses ini *evolutionary Strategies* dapat digunakan untuk mencari solusi permasalahan-permasalahan dalam dunia nyata.

*Evolutionary Strategies* ditemukan oleh Rechenberg pada tahun 1973 dengan seleksi (*selection*), mutasi (*mutation*), populasi (*population*) kemudian dikembangkan oleh Schwefel tahun 1981 yang memperkenalkan *recombination* dan populasi (*population*) individu lebih dari satu. Mereka kemudian menyajikan suatu perbandingan *evolutionary Strategies* dengan teknik optimasi yang lebih tradisional.

Sebelum *evolutionary Strategies* 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 real (*floating point*) sebagai *allele* (nilai gen) memungkinkan penerapan operator *evolutionary Strategies* yaitu proses seleksi (*selection*), *recombination*, mutasi (*mutation*), dan kompetisi (*competition*) 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, mutasi dan kompetisi untuk menciptakan keturunan (*offspring*).

### **3.4. Genetic Algorithm-Evolution Strategies (GA-ES)<sup>[4]</sup>**

#### **3.4.1. Parameter Genetic Algorithm-Evolution Strategies (GA-ES)**

Terdapat beberapa parameter yang digunakan dalam *Genetic Algorithm-Evolution Strategies* (GA-ES). Parameter yang digunakan tersebut adalah :

##### **3.4.1.1. Jumlah Generasi (*MAXGEN*)**

Merupakan jumlah perulangan (iterasi) dilakukannya rekombinasi dan seleksi. Jumlah generasi ini mempengaruhi kestabilan *output* dan lama iterasi (waktu proses *Genetic Algorithm*). 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 lokal optimum.

##### **3.4.1.2. Ukuran Populasi (*POPSIZE*)**

Ukuran populasi mempengaruhi kinerja dan efektifitas dari *Genetic Algorithm*. Jika ukuran populasi kecil maka populasi tidak menyediakan cukup materi untuk mencakup ruang permasalahan, sehingga pada umumnya kinerja *Genetic Algorithm* menjadi buruk. Dalam hal ini dibutuhkan ruang yang lebih besar untuk mempersentasikan keseluruhan ruang permasalahan. Selain itu

penggunaan populasi yang besar dapat mencegah terjadinya konvergensi pada wilayah lokal.

#### **3.4.1.3. Probabilitas *Crossover* ( $P_c$ )**

Probabilitas *crossover* ini digunakan untuk mengendalikan frekuensi operator *crossover*. Dalam hal 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 *Genetic Algorithm*.

#### **3.4.1.4. Probabilitas Mutasi ( $P_m$ )**

Mutasi digunakan untuk meningkatkan variasi populasi digunakan untuk menentukan tingkat mutasi yang terjadi, karena frekuensi terjadinya mutasi tersebut menjadi  $P_m \times POPSIZE \times N$ , 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 *Genetic Algorithm*, 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.

#### **3.4.1.5. Panjang *Kromosom* (*NVAR*)**

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

#### **3.4.2. Mekanisme *Genetic Algorithm-Evolution Strategies* (GA-ES) <sup>[4]</sup>**

Sangat perlu untuk mengetahui mekanisme *Genetic Algorithm-Evolution Strategies* (GA-ES). Dibawah ini akan diuraikan mengenai hal itu, dimana uraian ini merupakan penjabaran dari Metode *Genetic Algorithm-Evolution Strategies* (GA-ES).

##### **3.4.2.1. Pengkodean atau Representasi**

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

Secara umum Algoritma Genetika 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 *Genetic Algorithm* untuk awal melakukan pencarian.

#### 3.4.2.2. Fungsi *Fitness* (Fungsi Evaluasi)

Dalam Algoritma Genetika, 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. Hal ini dapat mengakibatkan konvergensi dini (*premature convergence*).

Permasalahan klasik dalam *Genetic Algorithm* adalah beberapa kromosom dengan nilai *fitness* yang tinggi (tetap bukan nilai optimum) mendominasi populasi dan mengakibatkan *Genetic Algorithm* konvergen pada lokal optimum. Ketika mencapai konvergen, kemampuan Algoritma Genetika untuk mencari solusi yang lebih baik menghilang. Tukar silang antara kromosom induk yang

hampir identik. Dalam hal ini hanya operasi mutasi yang mampu menghasilkan kromosom yang relatif baru dan merupakan cara untuk menghindari kromosom tertentu mendominasi populasi.

#### **3.4.2.3. 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*, *elitesm* dan lain sebagainya.

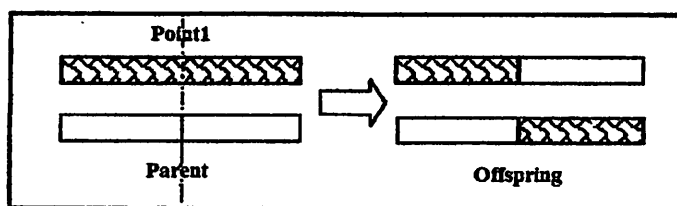
#### **3.4.2.4. Rank Selection**

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 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..

### 3.4.2.5. Crossover (Pindah Silang)<sup>[4]</sup>

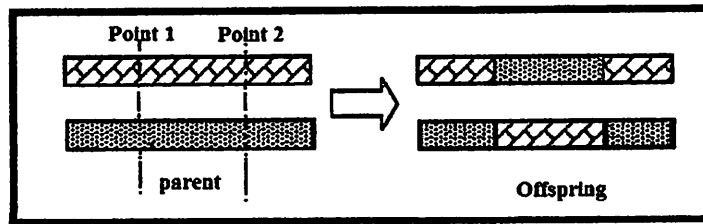
Fungsi dari *crossover* adalah menghasilkan kromosom anak dari kombinasi materi-materi gen dua kromosom induk. Cara kerjanya dengan membangkitkan sebuah nilai random  $r_k$  dimana  $k = 1, 2, \dots, \text{POPSIZE}$ . Probabilitas *crossover* ( $P_c$ ) ditentukan dan digunakan untuk mengendalikan frekuensi 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.

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 3.2



Gambar 3.1.<sup>[4]</sup>

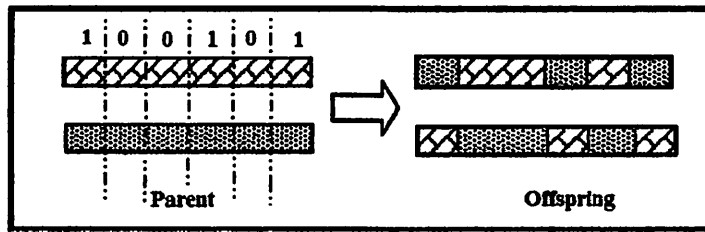
Ilustrasi operator dengan *One Point Crossover*



Gambar 3.2.<sup>[4]</sup>

Illustrasi 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*. Illustrasi kerja operator ini digambarkan seperti pada gambar 3.3



Gambar 3.3.<sup>[4][2]</sup>

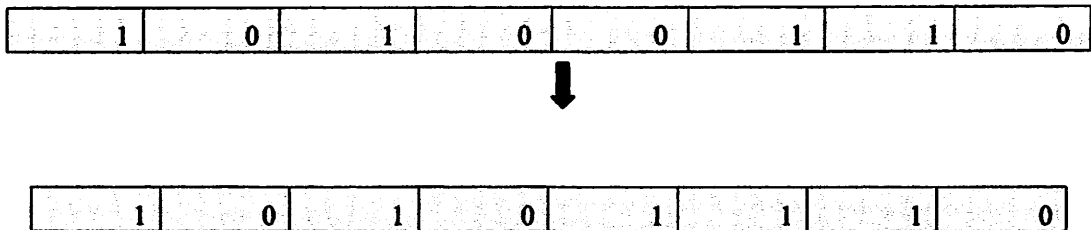
Illustrasi operator *crossover* dengan *uniform crossover*

### 3.4.2.6. Mutasi (*Mutation*)<sup>[4][2]</sup>

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. Mutasi dengan mengganti gen 0 dengan 1 atau sebaliknya gen 1 dengan 0. Biasanya disebut *flip* yaitu membalik nilai ke 1 atau 1 ke 0.



Gambar 3-4<sup>[4]</sup>

Ilustrasi operator mutasi untuk representasi string biner

### 3.5. *Evolutionary Programming-Evolution Strategies (EP+ES)*<sup>[4]</sup>

#### 3.5.1. Parameter *Evolutionary Programming* dan *Evolutionary Strategies*

Terdapat beberapa parameter yang digunakan dalam *evolutionary programming* dan *evolutionary Strategies*. Parameter tersebut digunakan untuk melihat kompleksitas dari *evolutionary programming* dan *evolutionary Strategies*. Parameter yang digunakan tersebut adalah :

##### 3.5.1.1. Jumlah Generasi (*MAXGEN*)

Merupakan jumlah perulangan (iterasi) dilakukannya rekombinasi dan seleksi. Jumlah generasi ini mempengaruhi kestabilan output dan lama iterasi (waktu proses *evolutionary programming* dan *evolutionary Strategies*). 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 solution*.

### **3.5.1.2. Ukuran Populasi (*POPSIZE*)**

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

### **3.5.1.3. Probabilitas Mutasi (*Pm*)**

Mutasi digunakan untuk meningkatkan variasi populasi digunakan untuk menentukan tingkat mutasi yang terjadi, karena frekuensi terjadinya mutasi tersebut menjadi  $Pm \times POPSIZE \times N$ , dimana  $N$  adalah panjang struktur/gen dalam satu individu. Probabilitas mutasi yang rendah akan menyebabkan gen-gen yang berpotensi tidak dicoba. Sebaliknya tingkat mutasi yang tinggi akan menyebabkan keturunan akan semakin mirip dengan induknya.

### **3.5.1.4. Panjang Kromosom**

Panjang kromosom berbeda-beda sesuai dengan pemodelan permasalahan. Titik solusi dalam ruang permasalahan dikodekan dalam bentuk kromosom/string yang terdiri dari komponen genetik terkecil yaitu gen. Pengkodean memakai *string* bilangan *real*.

### **3.6. Mekanisme *Evolutionary Programming* dan *Evolutionary Strategies*<sup>[4]</sup>**

#### **3.6.1. Pengkodean (*initialization*)**

Langkah pertama kali yang dilakukan dalam penggunaan *evolutionary programming* dan *evolutionary Strategies* adalah melakukan pengkodean atau representasi terhadap permasalahan yang akan dilakukan.

Secara umum *evolutionary programming* dan *evolutionary Strategies* 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 *evolutionary programming* dan *evolutionary Strategies* untuk awal melakukan pencarian

#### **3.6.2. Fungsi Evaluasi (Fungsi *Fitness*)**

Dalam *evolutionary programming* dan *evolutionary Strategies* 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.

### **3.6.3. Seleksi (*selection*)**

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 *roulette wheel selection, rank selection, elitesm* dan lain sebagainya.

#### **3.6.3.1. 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.



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

### **3.6.3.2. Rank Selection**

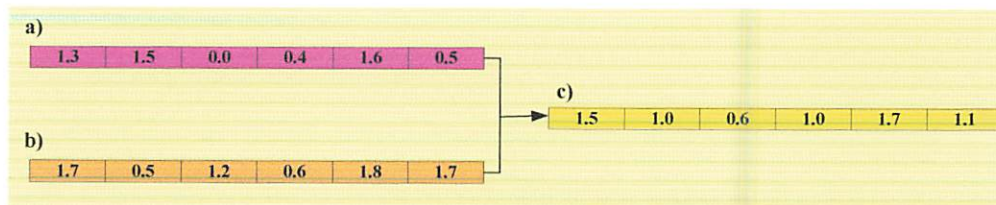
Apabila fitness yang dimiliki oleh suatu kromosom dalam populasi berbeda terlalu jauh dari kromosom lainnya maka hal ini dapat menjadi permasalahan. Misalnya bila kromosom terbaik mempunyai *fitness* yang menyebabkan besarnya tempat yang dimilikinya dalam *roulette wheel* sebesar 90% maka kromosom-kromosom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

### **3.6.4. Rekombinasi (*Recombination*)**

Khusus dalam *evolutionary Strategies* setelah melakukan pengkodean dan evaluasi, individual dipilih secara acak menjadi *parents*. Setelah melakukan seleksi *parent* kemudian melakukan *Intermediate recombination*, dimana vektor

dari dua orang tua (*parent*) dirata-ratakan bersama-sama, unsur demi unsur, untuk membentuk suatu keturunan yang baru.



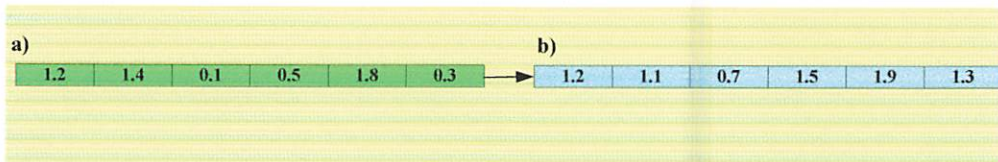
Gambar 3.5<sup>[4]</sup>  
*Intermediate recombination* dari Induk (*parent*) a) dan b) menjadi Anak (*offspring*) c)

Seleksi orang tua (*parent*) menjadi turunan (*offspring*) menjadi lebih mudah, sebagai contoh dalam kaitannya dengan sifat alami penyajian, adalah sangat mudah merata-ratakan dari banyak individu menjadi turunan tunggal. Didalam *Evolutionary Strategies*,  $N$  *parent* yang terpilih secara acak (tidak didasarkan pada fitness terbaik), lebih dari  $N$  *offspring* dihasilkan melalui penggunaan *intermediate recombination* dan kemudian  $N$  *parent* yang selamat terpilih secara deterministik

### 3.6.5. Mutasi (*Mutation*)

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 *evolutionary programming* dan

*evolutionary Strategies* menggunakan operator *Gaussian mutation*, dimana setiap individu akan terpilih secara acak untuk mengalami mutasi berdasarkan nomor acak *Gaussian* untuk menciptakan individu baru (*offspring*).



Gambar 3.2.<sup>[4]</sup>  
Mutasi Gaussian Dari Induk (*parent*) (a) Menghasilkan Anak (*offspring*) (b)

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 dimutasi ditentukan secara random. Mutasi dikerjakan dengan cara melakukan perubahan pada elemen tersebut.

### 3.6.6. Kompetisi (*Competition*)

Dalam tahap kompetisi, mekanisme seleksi dipakai untuk menghasilkan populasi baru dari populasi yang ada. Melalui penggunaan skema kompetisi setiap individu dalam populasi rang tua (*parent*) maupun anak (*offspring*) akan dikompetisi/bersaing satu dengan yang lainnya. Kompetisi setiap individu dengan lawannya didasarkan pada nilai *fitness* dari setiap individu tersebut. Agar optimal, solusi yang lebih pas atau lebih optimal seharusnya memiliki peluang seleksi yang

lebih besar. Individu yang memenangkan dari kompetisi akan digunakan sebagai individu yang baru bagi pembangkitan selanjutnya.

### 3.7. Formulasi Masalah Pembebanan Ekonomis

Permasalahan perencanaan pembebanan ekonomis adalah untuk meminimalkan biaya operasional dan biaya instalasi pembangkitan daya, Mempunyai fungsi Objektif sebagai berikut<sup>[4]</sup>:

$$F = \sum_{i=1}^N F_i(P_i) \dots\dots\dots(3.1)$$

$$F_i(P_i) = \begin{cases} a_{i1} + b_{i1}P_i + c_{i1}P_i^2 & \text{jika } P_i^{\text{MIN}} \leq P_i < P_{i1} \\ a_{i2} + b_{i2}P_i + c_{i2}P_i^2 & \text{jika } P_{i1} \leq P_i < P_{i2} \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ a_{im} + b_{im}P_i + c_{im}P_i^2 & \text{jika } P_{im-1} \leq P_i < P_i^{\text{MAX}} \end{cases}$$

Dimana:

- F : Total biaya
- $F_i(P_i)$  : Biaya dari generator
- $a_{ij}, b_{ij}, c_{ij}$  : Koefisien biaya dari generator i untuk level daya ke j
- $P_i$  : Daya generator i (MW)
- N : Jumlah generator

### **3.8. Adaptasi *Evolutionary Programming* dan *Evolutionary Strategies* ke Masalah Optimasi Daya Reaktif<sup>[8]</sup>**

*Evolutionary programming* dan *evolutionary Strategies* adalah suatu pada mekanisme seleksi alam. Individu dari sebuah populasi dikodekan secara real, Populasi pertama dibangkitkan secara random. Untuk metode *evolutionary programming* generasi baru dibuat dengan mengaplikasikan operator-operator berikut terhadap sebuah populasi yaitu ; inisialisasi, mutasi dan kompetisi. Sedangkan untuk metode *evolutionary Strategies* generasi baru dibuat dengan mengaplikasikan operator berikut terhadap sebuah populasi yaitu ; inisialisasi, rekombinasi, mutasi dan kompetisi dimana dari kedua metode ini tingkat pengoptimalan setiap calon solusi atau individu diukur dengan ketepatannya tergantung dari fungsi tujuan (*objective function*) permasalahan.

#### **3.8.1. Representasi Solusi**

Individu dalam populasi menunjukkan calon pada solusi optimasi daya reaktif secara ekonomis. Elemen solusi terdiri dari daya pembangkit yang ditetapkan pada semua bus generator (PV) selain dari bus slack, magnitude tegangan yang ditetapkan pada semua bus generator (PV).

#### **3.8.2. Inisialisasi (*Initialization*)**

Populasi awal diambil secara acak dari daerah variabel kontrol dari batasan atas sampai batasan bawah. Nilai *fitness*  $f_i$  diperoleh menurut fungsi obyektif .

### 3.8.3. Statistik (*Statistics*)

Fitnes maksimum ( $f_{\max}$ ), fitness minimum ( $f_{\min}$ ), Jumlah fitnes  $\sum f$ , dan fitnes rata-rata ( $f_{\text{avg}}$ ) dari generasi dihitung.

### 3.8.4. Mutasi (*Mutation*)<sup>[1]</sup>

#### 3.8.4.1. Mutasi *Evolutionary Programming*

Di dalam proses mutasi, setiap terpilihnya orang tua, misalnya  $P_i$  dimutasikan dan ditambahkan ke populasinya, sesuai persamaan berikut<sup>[4]</sup>:

$$P_{i+m,j} = P_{i,j} + N(0, \beta (\bar{\chi}_j - \underline{\chi}_j) \frac{f_i}{f_{\max}}), j = 1, 2, \dots, n \dots \dots \dots (3.2.)$$

Dimana :

- n = Banyaknya variabel keputusan di dalam setiap individu.
- $P_{i,j}$  = Elemen  $j^{\text{th}}$  unsur dari  $I^{\text{th}}$
- $N(\mu, \sigma^2)$  = Variabel acak dengan menggunakan  $\mu$  dan  $\sigma^2$
- $f_{\max}$  = Nilai *fitness* terbesar dari statistik,  $\bar{\chi}_j$  dan  $\underline{\chi}_j$
- $\beta$  = Skala mutasi,  $0 < \beta \leq 1$

#### 3.8.4.2. Mutasi *Evolutionary Strategies*

Di dalam proses mutasi, setiap terpilihnya orang tua, misalnya  $P_i$  dimutasikan dan ditambahkan ke populasinya, sesuai persamaan berikut<sup>[4]</sup>:

$$P_{i+m,j} = P_{i,j} + N(0, \beta \nabla_{dev}), \quad j = 1, 2, \dots, n, \dots \dots \dots (3.3.)$$

Dimana:

$P_{i,j}$  = Elemen ke  $j$  dari individu ke  $i$

$\beta$  = skala mutasi ( $0 < \beta < 1$ )

$\nabla_{dev}$  = Variabel deviasi

### 3.8.5. Kompetisi (*Competition*)

Di dalam proses kompetisi, fitness individu dari populasi yang berukuran  $2m$  disortir didalam order turunan. Dan  $m$  individu yang pertama dijaga sebagai orang tua untuk proses mutasi selanjutnya. Nilai berat  $W_i$  dari individu yang ke  $i$  dihitung menurut kompetisi berikut ini<sup>[4]</sup>:

$$W_i = \sum_{i=1}^N W_{i,t} \dots \dots \dots (3.4.)$$

Dimana  $N$  adalah nomor kompetisi generasi secara acak.  $W_{i,t}$  untuk yang manapun, 0 untuk kalah atau 1 untuk menang sebagai individu ke  $i$  yang bersaing dengan individu ke  $r$  yang dipilih secara acak di dalam populasi yang dikombinasikan itu. Nilai  $W_{i,t}$  disampaikan sesuai persamaan berikut<sup>[4]</sup>:

$$W_{i,t} = \begin{cases} 1 & \text{jika } U_t < \frac{f_r}{f_r + f_i} \\ 0 & \text{jika sebaliknya} \end{cases} \dots \dots \dots (3.5.)$$

Dimana:

$f_r$  = Fitness dari individu ke  $r$  yang dipilih secara acak

$f_i$  = Fitness dari individu ke  $i$

$U_t$  = Set yang terpilih secara acak,  $U(0,1)$

Ketika semua  $2m$  individu mendapat nilai  $W_i$  kompetisinya, mereka akan diatur dalam order turunan menurut nilai  $W_i$  yang bersesuaian,  $m$  individu yang pertama dipilih dengan fitness  $f_i$  menjadi dasar untuk generasi selanjutnya. Kemudian fitness maksimum, minimum, dan fitness rata-rata serta jumlah fitness dari generasi ini dihitung dalam statistik.

### 3.8.6. Konvergensi (*Convergence*)

Jika kondisi konvergen tidak dijumpai, proses mutasi dan kompetisi akan berjalan lagi. Nomor generasi maksimum dapat digunakan untuk kondisi konvergen. Kriteria lain, seperti rasio dari *fitness* rata-rata dan maksimum dari populasi dihitung berulang-ulang sampai<sup>[4]</sup>:

$$\{f_{avg} / f_{max}\} \geq \delta \dots\dots\dots (3.6.)$$

dimana,  $\delta$  harus mendekati 1. Jika penelitian telah mencapai ketelitian, suatu solusi optimal telah ditemukan untuk optimasi masalah.



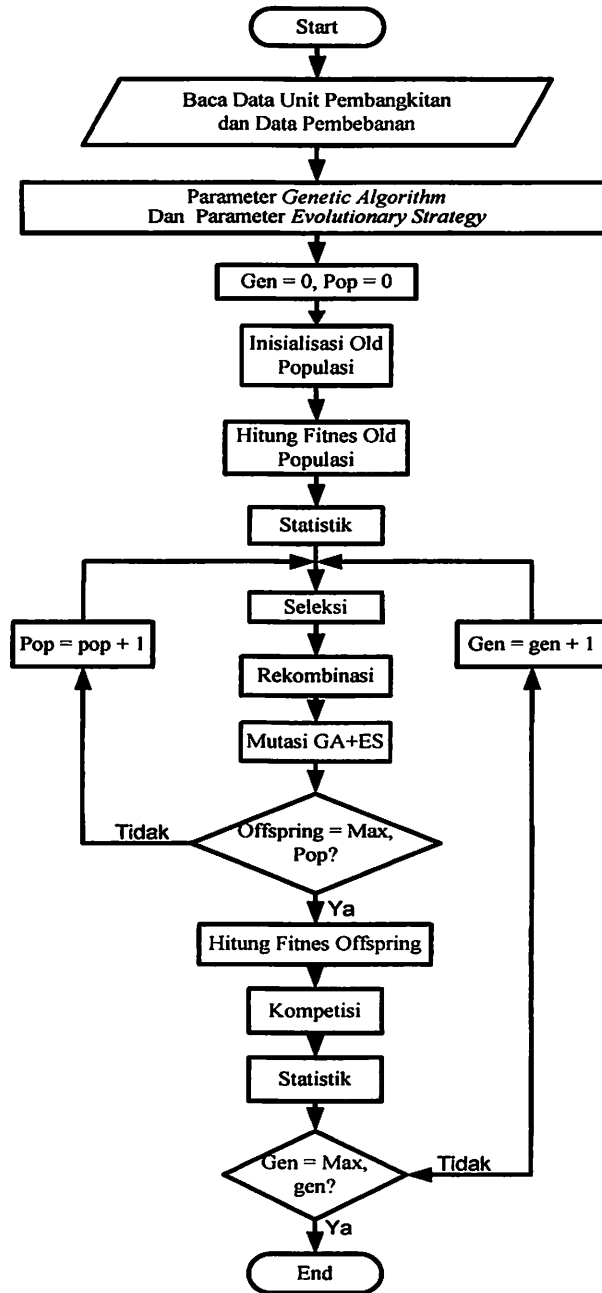
### 3.9. Algoritma dan Flowchart

#### 3.9.1. Algoritma Pembebanan Ekonomis menggunakan metode *Genetic Algorithm-Evolution Strategy (GE+ES)*

1. Memasukan inputan dan pembangkitan dan data pembebanan.
2. Menentukan Parameter inputan *Genetic Algorithm-Evolutionary Strategies* yang meliputi Jumlah populasi, maksimum generasi, nilai kemungkinan mutasi, dan panjang kromosom tiap-tiap individu.
3. Generasi = 0, Populasi = 0.
4. Melakukan inialisasi calon solusi.
5. Menghitung *fitness* dari kromosom tiap-tiap individu.
6. Melakukan proses statistik.
7. Melakukan Proses mutasi.
8. Apakah *offspring* yang diinginkan sudah terpenuhi (maxPop)
9. Jika “Tidak” maka Populasi = pop+1, kembali ke 7.
10. Jika “Ya” maka perhitungan dilakukan ke langkah 11.
11. Menghitung *fitness* dan *Offspring*.
12. Melakukan proses kompetensi.
13. Melakukan Proses statistik
14. Apakah Generasi yang diinginkan sudah terpenuhi (max Gen).
15. Jika “Tidak” maka generasi = gen + 1. kembali ke langkah 7.
16. Jika “Ya” maka perhitungan berhenti.
17. Memasukan inputan dan pembangkitan dan data pembebanan.

18. Menentukan Parameter inputan *Genetic Algorithm-Evolutionary Strategies* yang meliputi Jumlah populasi, maksimum generasi, nilai kemungkinan mutasi, dan panjang kromosom tiap-tiap individu.
19. Generasi = 0, Populasi = 0.
20. Melakukan inisialisasi calon solusi.
21. Menghitung *fitness* dari kromosom tiap-tiap individu.
22. Melakukan proses statistik.
23. Melakukan Proses mutasi.
24. Apakah *offspring* yang diinginkan sudah terpenuhi (maxPop)
25. Jika “Tidak” maka Populasi = pop+1, kembali ke 7.
26. Jika “Ya” maka perhitungan dilakukan ke langkah 11.
27. Menghitung *fitness* dan *Offspring*.
28. Melakukan proses kompetensi.
29. Melakukan Proses statistik
30. Apakah Generasi yang diinginkan sudah terpenuhi (max Gen).
31. Jika “Tidak” maka generasi = gen + 1. kembali ke langkah 7.
32. Jika “Ya” maka perhitungan berhenti.

**3.9.2. Flowchart Pembebanan Ekonomis Dengan Metode *Genetic Algorithms* Dan *Evolution Strategy***



Gambar 3.7  
Flowchart Pembebanan Ekonomis Dengan Metode *Genetic Algorithms* Dan *Evolution Strategy* (GA-ES)

### 3.9.3. Algoritma Pembebanan Ekonomis menggunakan metode *Evolutionary Programming -Evolution Strategy (EP+ES)*

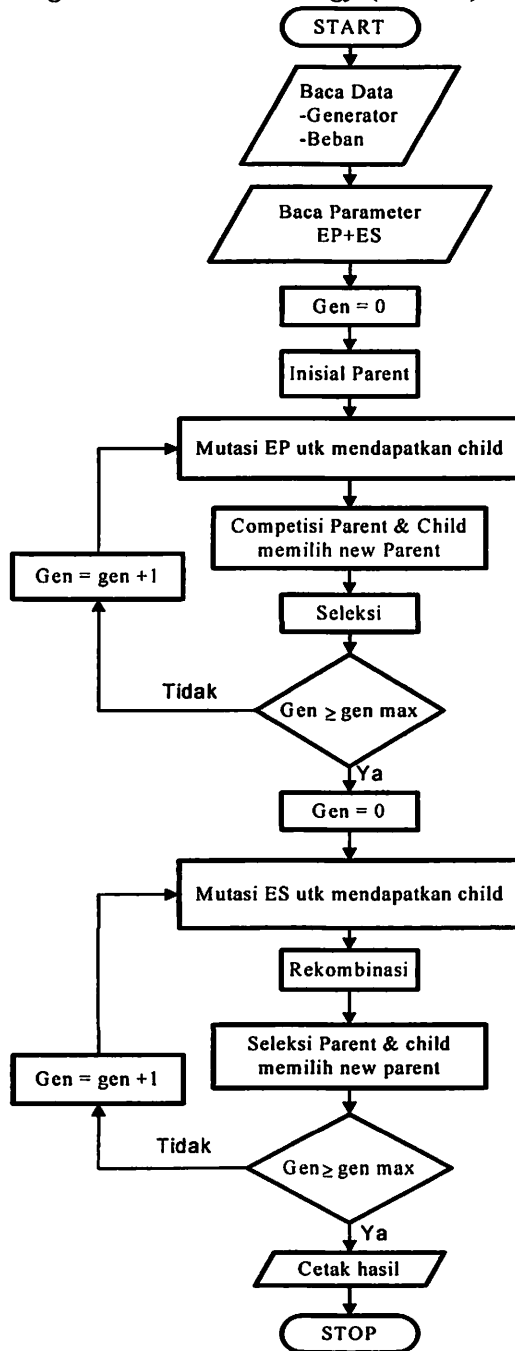
Adapun Algoritma *Economic Dispatch* menggunakan metode *evolutionary programming* dan *evolutionary strategy* adalah seperti berikut:

1. Masukkan inputan data pembangkitan  $\{P_{\text{mak}}, P_{\text{min}}, \text{koefisien biaya bahan bakar}\}$ , data pembebanan.
2. Menentukan parameter inputan *evolutionary programming* dan *evolutionary strategy* yang meliputi *population size*, maksimum generasi EP dan ES, *Betha*, dan  $\nabla$  ES.
3. Gen = 0.
4. Melakukan inisialisasi *parent*.
5. Melakukan proses mutasi *evolutionary programming* untuk mendapatkan *child*.
6. Melakukan kompetisi antara *parent* dan *child* untuk memilih *new parent*.
7. Melakukan proses seleksi.
8. Apakah  $\text{Gen} \geq \text{gen max}$ .
9. Jika “Tidak” maka Gen = gen +1 dan kembali ke langkah 5.
10. Jika “Ya” maka lanjutkan ke langkah berikutnya.
11. Gen = 0.
12. Melakukan inisial *parent*.
13. Melakukan proses mutasi *evolutionary strategy* pada setiap *parent* untuk mendapatkan *child*.

14. Melakukan Rekombinasi.
15. Melakukan proses seleksi dari *parent* dan *child* untuk memilih *new parent*.
16. Apakah  $Gen \geq gen \text{ max}$ .
17. Jika “Tidak” maka  $Gen = gen + 1$  dan kembali ke langkah 14.
18. Jika “Ya “ maka cetak hasil dan perhitungan selesai.

### 3.9.4. Flowchart *Economic Load Dispatch* Dengan Metode *Evolutionary*

#### *Progammig dan Evolution Strategy (EP+ES)*



Gambar 3.8

Flowchart *Economic Load Dispatch* Dengan Metode *Evolutionary Progammig dan Evolution Strategy (EP-ES)*

## **BAB IV**

### **ANALISA DAN HASIL PROGRAM**

#### **4.1. Penggunaan Program Komputer Untuk Menyelesaikan *Economic Load Dispatch* Di PT. Pembangkitan Jawa-Bali**

Dalam pemecahan masalah *economic load dispatch* penggunaan program komputer dibutuhkan untuk tujuan mempercepat proses perhitungan dengan ketelitian yang tinggi. Program perhitungan yang digunakan adalah bahasa pemrograman Borland Delphi versi 7.0. dan diaplikasikan pada komputer pentium IV 2,4GHz, DDR256Mb/2700 dengan sistem operasi windowsXP-service pack 2.

#### **4.2. Data Pembangkitan Termal**

Pembangkitan termal yang berada pada kawasan P.T. Pembangkitan Jawa-Bali berjumlah 37 unit pembangkit yang terdiri dari 11 Pembangkit Tenaga Uap (PLTU), 4 unit Pembangkit Listrik Tenaga Gas (PLTG), dan 22 unit Pembangkit Listrik Tenaga Gas dan Uap (PLTGU) yang tersebar diseluruh Jawa dan Bali. Adapun data-data lebih lengkapnya dapat dilihat pada table 4.1 dan table 4.2, untuk harga bahan bakar berdasarkan statistik PLN tahun 2002 nilai tukar Rp. 9.000,00 per dolar Amerika.

Tabel 4.1  
Data Unit Termal Pada P.T. PLN Pembangkitan Jawa-Bali  
Agustus 2002

No	Nama Pembangkit	Bahan Bakar	Kapasitas (MW)		Lama Waktu			
			Min	Max	Mut	Mdt	Cold start	Hot Star
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



Tabel 4.2  
Data Biaya dan Parameter unit Termal Pada PT. PLN  
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	160.1	96.42	43043399	288609.995	7.6584
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 gas UP. Gresik	: 2.53 US\$/MMBTU
Harga MFO	: 1595,5 Rp/Liter	Harga gas UP. M. Karang	: 2,45 US\$/MMBTU
Harga HSD	: 1595,5 Rp/Liter	Nilai Tukar	: 9000 Rp/US\$

#### **4.3. Aplikasi Metode *Genetic Algorithm-Evolution Strategis (GA-ES)* dan Metode *Evolutionary Programming-evolution Strategis (EP-ES)* Pada PT. PLN PJB Untuk Memecahkan Permasalahan *Economic Load Dispatch***

Perhitungan dan analisa ini dilakukan pada kebutuhan daya yang ditanggung PT. Pembangkitan Jawa-Bali. analisa data dilakukan untuk tiga hari, yaitu :

- sabtu, tanggal 25 Maret 2006
- Minggu, tanggal 26 Maret 2006
- Rabu, tanggal 29 Maret 2006

Berdasarkan data unit yang terdapat dalam PT. Pembangkitan Jawa-Bali pada sistem tenaga pada tabel 4.1, semua unit pembangkit diasumsikan dalam kondisi siap beroperasi. Maka dapat disusun *input* data unit pembangkit termal yang siap operasi pada tanggal 25, 26, dan 29 maret 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.

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.

#### **4.4. Beban sistem**

Dalam wilayah Jawa-Bali, pembangkit-pembangkit yang ada dikoordinasi oleh PT. Pembangkitan Jawa-Bali. Proses penjadwalan pembangkit dengan metode *Genetic Algorithm-Evolution Strategis* (GA-ES) dan metode *Evolutionary Programming-evolution Strategis* (EP-ES) 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 25, 26, dan 29 maret 2006, terdapat pada tabel 4.3 (beban sistem yang ditanggung oleh pembangkit termal saja).

#### 4.5. Uji Validasi

Untuk melakukan uji validasi pada program bantu perhitungan sehingga bisa dilihat kelayakan dari program tersebut, terlebih dahulu dilakukan pemilihan parameter *Genetic Algorithm*, *Evolutionary Programming*, dan *Evolution Strategies* yang sesuai sehingga menghasilkan solusi yang optimal. Berdasarkan jurnal June Ho Park\*, Sung-Oh Yang\*, Hwa Seok Lee \*, Young Moon Park\* “*Economic Load Dispatch Using Evolutionary Algorithms*” IEEE 1996 dengan nilai parameter adalah :

Tabel 4.3<sup>[1]</sup>  
Parameter *Genetic Algorithm*, *Evolutionary Programming* dan *Evolution Strategies*

Parameter	GA	ES	EP	GA+ES	EP+ES
A	1	1	1	1	1
B	1	1.5	1.5	1.5	1.5
Panjang String	140	10	10	40 (GA) 10 (ES)	10
Populasi	100	100	100	100	100
Mutasi	0.95	x	x	0.01 (GA)	X
Crossover	0.95	x	x	0.95 (GA)	X
Standart Deviasi	x	2	2	1 (ES)	4 (EP) 1 (ES)
Generasi	600	600	600	200	200

Selanjutnya untuk pengujian validasi berpedoman pada data jurnal yang sama. Hasil pengujian dapat dilihat pada gambar dibawah. kemudian dilakukan perbandingan perhitungan biaya total dengan metode *Genetic Algorithm-Evolution Strategis* (GA-ES) dan metode *Evolutionary Programming-evolution Strategis* (EP-ES)

Tabel 4.4<sup>[1]</sup>  
Data *Economicload Load Dispatch* Dengan 10 Unit Pembangkit Untuk Pengujian Validasi Program

Unit	$P_{max}$ (MW)	$P_{min}$ (MW)	a	b	c
1	196	100	26.97	-0.3975	0.002176
2	230	157	118.4	-1.269	0.004194
3	332	200	39.79	-0.3116	0.001457
4	265	200	226.8	-2.338	0.005935
5	338	190	13.92	-0.0873	0.001066
6	265	200	226.8	-2.338	0.005935
7	331	200	18.39	-0.1325	0.001107
8	265	200	226.8	-2.338	0.005935
9	370	213	88.53	-0.5675	0.001554
10	362	200	13.97	-0.0994	0.001102

Tabel 4.5<sup>[1]</sup>  
Data Beban Masing-Masing Unit dengan Total 2400MW dan Cadangan Berputar 240MW

Unit	Beban Sistem (MW)
1	188.29
2	202.12
3	255.68
4	233.80
5	240.88
6	233.14
7	254.29
8	232.81
9	319.66
10	239.32

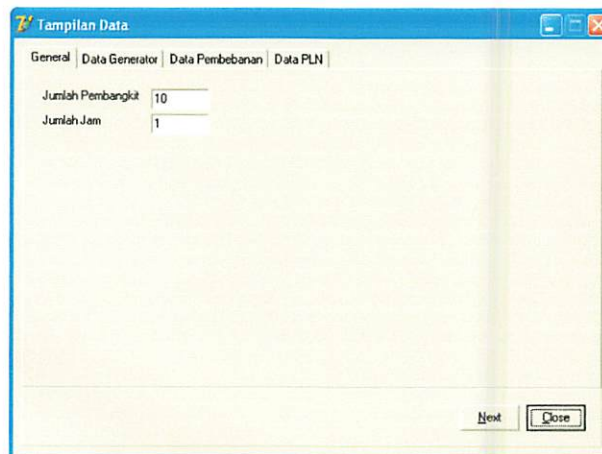
#### 4.5.1. Tampilan Program Uji Validasi Pembebanan Ekonomis dengan Metode *Genetic Algorithm* dan *Evolution Strategies* (GA-ES).

1. Tampilan Menu Utama :



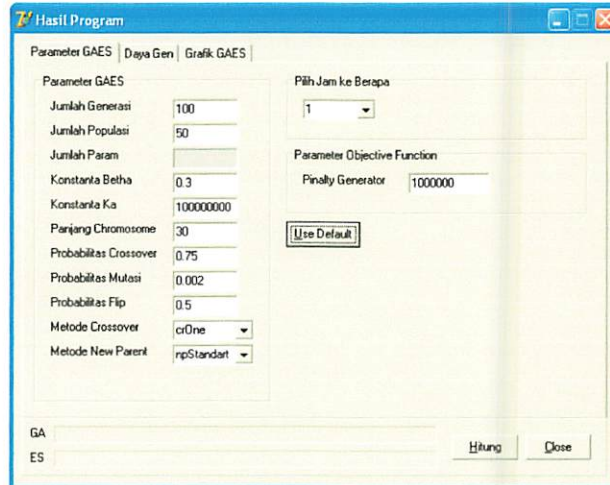
Gambar 4.1.  
Tampilan Menu Utama

2. Pilih Open File, kemudian akan muncul tampilan sebagai berikut :



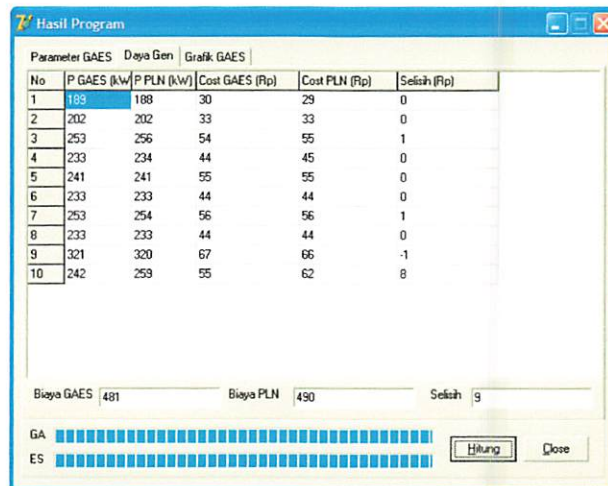
Gambar 4.2.  
Tampilan Data dengan 10 Unit Pembangkit Selama 1 Jam  
Untuk Uji Validasi

3. Kemudian tekan menu “Next”, dan akan terlihat tampilan parameter GA-ES, kemudian tekan “Use Default” untuk memasukan nilai dari parameter GA-ES.



Gambar 4.3.  
Tampilan Parameter GA-ES Untuk Uji Validasi

4. Kemudian tekan tombol “Hitung” untuk menghitung biaya dari unit.



Gambar 4.4.  
Hasil Uji Validasi GA-ES

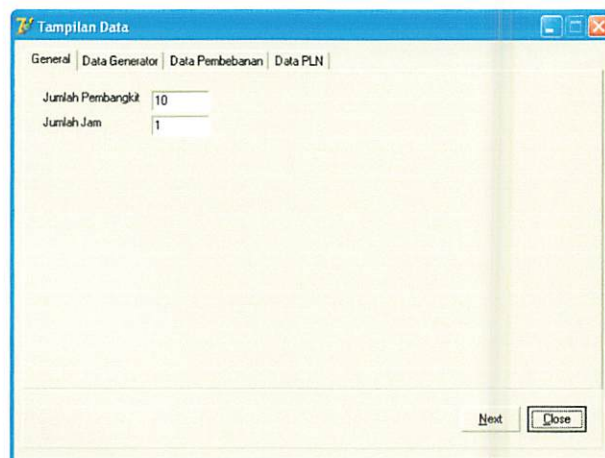
#### 4.5.2. Tampilan Program Uji Validasi Pembebanan Ekonomis dengan Metode *Evolutionary Programming* dan *Evolution Strategies* (EP-ES).

1. Tampilan Menu Utama :



Gambar 4.5.  
Tampilan Menu Utama

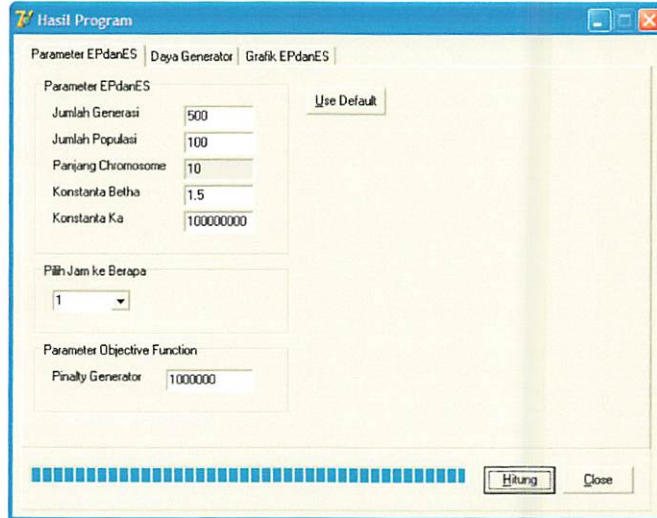
2. Pilih Open File, kemudian akan muncul tampilan sebagai berikut :



Gambar 4.6.  
Tampilan Data dengan 10 Unit Pembangkit Selama 1 Jam Untuk Uji Validasi



- Kemudian tekan menu “Next”, dan akan terlihat tampilan parameter GA-ES, kemudian tekan “Use Default” untuk memasukan nilai dari parameter EP-ES.



Gambar 4.7.  
Tampilan Parameter EP-ES Untuk Uji Validasi

- Kemudian tekan tombol “Hitung” untuk menghitung biaya dari unit.

No	P EPdanES (MW)	P Jurnal (MW)	Cost EPdanES (\$)	Cost Jurnal (\$)	Selisih (\$)
1	190	188	30	29	-1
2	202	202	33	33	0
3	253	256	54	55	1
4	233	234	44	45	0
5	243	241	56	55	-1
6	233	233	44	44	0
7	252	254	55	56	1
8	233	233	44	44	0
9	320	350	66	80	14
10	240	239	54	53	0

Biaya EPES 481      Biaya Jurnal 495      Selisih 14

Gambar 4.8.  
Hasil Uji Validasi EP-ES

**4.5.3. Hasil Uji Validasi Pembebanan Ekonomis dengan Metode *Evolutionary Programming* dan *Evolution Strategies* (EP-ES).**

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 481 \$/H yang mendekati perhitungan dengan jurnal June Ho Park\*, Sung-Oh Yang\*, Hwa Seok Lee\*, Young Moon Park\*\* “*Economic Load Dispatch Using Evolutionary Algorithms*” IEEE 1996 yaitu :

- ✚ Untuk kombinasi metode *Genetic Algorithm-Evolution Strategies* (GA-ES), dari hasil perhitungan program didapat selisih untuk validasi sebesar 9 \$/H dengan error 0.0187%, sehingga program ini valid digunakan.
- ✚ Untuk kombinasi metode *Evolutionary Programming-Evolution Strategies* (EP-ES), dari hasil perhitungan program didapat selisih untuk validasi sebesar 14 \$/H dengan error 0.0291%, sehingga program ini valid digunakan.

Tabel 4.6  
Hasil Uji Validasi

Jurnal		Program		Error (%)	
GA-ES	EP-ES	GA-ES	EP-ES	GA-ES	EP-ES
481	481	490	495	0.0187 %	0.0291 %

#### 4.6. Hasil Perhitungan dan Analisa Data

Program optimasi *economic load dispatch* unit pembangkit termal pada sistem PT. PJB dengan perbandingan metode *Genetic Algorithm-Evolution Strategis* (GA-ES) dan metode *Evolutionary Programming-evolution Strategis* (EP-ES), terdiri dari:

1. Tahap input data dengan inisialisasi data karakteristik tiap unit dan beban tiap jam.
2. Input parameter metode *Genetic Algorithm-Evolution Strategis* dan metode *Evolutionary Programming-Evolution Strategis*.
3. Melakukan pencarian nilai yang paling minimum dari fungsi keanggotaan yang masing-masing untuk biaya, beban sistem dan cadangan berputar sehingga didapatkan biaya operasional 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  
37 Unit Pembangkit Pada PT. PLN PJB

NAMA PEMBANGKIT			
1	PLTU Paiton 1	20	PLTG Gresik 2
2	PLTU Paiton 2	21	PLTG Gilitimur 1
3	PLTGU Gresik GT 1.1	22	PLTG Gilitimur 2
4	PLTGU Gresik GT 1.2	23	PLTGU M. Karang GT 1.1
5	PLTGU Gresik GT 1.3	24	PLTGU M. Karang GT 1.2
6	PLTGU Gresik ST 1.0	25	PLTGU M. Karang GT 1.3
7	PLTGU Gresik GT 2.1	26	PLTGU M. Karang ST 1.0
8	PLTGU Gresik GT 2.2	27	PLTGU M. Tawar GT 1.1
9	PLTGU Gresik GT 2.3	28	PLTGU M. Tawar GT 1.2
10	PLTGU Gresik ST 2.0	29	PLTGU M. Tawar GT 1.3
11	PLTGU Gresik GT 3.1	30	PLTGU M. Tawar GT 2.1
12	PLTGU Gresik GT 3.2	31	PLTGU M. Tawar GT 2.2
13	PLTGU Gresik GT 3.3	32	PLTGU M. Tawar ST 1.0

14	PLTGU Gresik ST 3.0	33	PLTU M. Karang 1
15	PLTU Gresik 1	34	PLTU M. Karang 2
16	PLTU Gresik 2	35	PLTU M. Karang 3
17	PLTU Gresik 3	36	PLTU M. Karang 4
18	PLTU Gresik 4	37	PLTU M. Karang 5
19	PLTG Gresik 1		

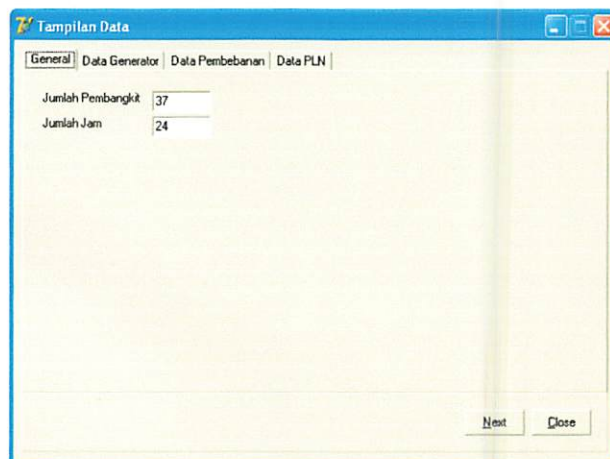
#### 4.6.1. Tampilan Program *Economic Load Dispatch* dengan Metode *Genetic Algorithm* dan *Evolution Strategies* (GA-ES)

##### 1. Tampilan Menu Utama



Gambar 4.9.  
Tampilan Menu Utama

##### 2. Pilih Open File, kemudian akan muncul tampilan sebagai berikut



Gambar 4.10  
Tampilan Secara Umum

- Pilih Data Generator untuk menampilkan unit – unit pembangkit

Gen	Nama	Pmax	Pmin	a0	a1	a2	Tap
1	PLTU Pakon 1	370	225	3244978	111712.2	10.2971	72
2	PLTU Pakon 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 4.11  
Tampilan Data Generaor

- Pilih Data Pembebanan untuk menampilkan beban sistem dan cadangan berputar

	Load	Res
1	165	90
2	2417	400
3	2471	400
4	2393	400
5	2322	400
6	2616	400
7	2342	400
8	2077	400
9	2199	400
10	2405	400
11	2489	400
12	2467	400
13	2368	400
14	2284	400
15	...	...

Gambar 4.12  
Tampilan Data Pembebanan

- Pilih Data PLN untuk mengetahui daya yang dioperasikan oleh tiap unit pembangkit thermal selama 24 jam

General	Data PLN							Je
	Jam 1	Jam 2	Jam 3	Jam 4	Jam 5	Jam 6	Jam 7	
Gen 1	3455	400	370	370	370	370	370	32
Gen 2	370	370	370	370	370	325	325	37
Gen 3	0	0	0	0	0	0	0	0
Gen 4	0	0	0	0	0	0	0	0
Gen 5	87	85	78	76	85	103	101	84
Gen 6	285	276	268	258	250	250	250	30
Gen 7	0	0	0	0	0	0	0	0
Gen 8	0	0	0	0	0	0	0	0
Gen 9	87	85	91	92	89	84	85	86
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	89	87	86	91	95	94	97	92
Gen 14	285	286	285	287	280	280	280	280

Gambar 4.13  
Tampilan Data PLN

- Kemudian tekan menu "Next", dan akan terlihat tampilan parameter GA-ES, kemudian tekan "Use Default" untuk memasukkan nilai dari parameter GA-ES, kemudian pilih jam ke berapa yang akan dihitung

Parameter GAES		Pilih Jam ke Berapa
Jumlah Generasi	100	1
Jumlah Populasi	50	
Jumlah Param	17	
Konstanta Beta	0.3	
Konstanta Ka	100000000	
Parang Chromosome	30	
Probabilitas Crossover	0.75	
Probabilitas Mutasi	0.002	
Probabilitas Flip	0.5	
Metode Crossover	crOne	
Metode New Parent	rpStandart	
Parameter Objective Function		
Penalty Generator		1000000
Use Default		

GA  ES

Hitung Close

Gambar 4.14  
Tampilan Parameter GA-ES

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

No	P GAES (kW)	P PLN (kW)	Cost GAES (Rp)	Cost PLN (Rp)	Selisih (Rp)
1	362	370	45,091,807	45,988,165	896,358
2	53	85	12,118,023	19,330,713	7,212,690
3	53	78	17,115,539	22,743,475	5,627,936
4	64	78	19,513,510	22,676,484	3,162,974
5	300	142	61,128,997	37,882,800	-23,246,197
6	53	74	17,120,101	21,828,469	4,708,369
7	67	74	20,315,384	21,828,469	1,513,085
8	67	75	20,154,432	22,006,917	1,852,485
9	463	127	85,328,374	35,686,941	-49,641,433
10	53	52	17,115,539	16,783,240	-332,299
11	60	51	18,619,713	16,761,092	-1,858,621
12	324	65	64,643,216	26,661,615	-37,981,601
13	43	41	10,918,588	10,461,644	-456,944
14	99	83	23,719,451	20,317,319	-3,402,132
15	90	83	21,816,918	20,397,837	-1,419,081

Biaya GAES 619,821,061      Biaya PLN 638,377,914      Selisih 18,556,853

GA [Progress Bar]      ES [Progress Bar]      [Hitung]      [Close]

Gambar 4.15  
Tampilan Daya dari Unit yang Beroperasi Serta Hasil Biaya

**4.6.2. Hasil Perhitungan Biaya Operasional PT. PLN PJB dan Biaya Operasional Menggunakan Metode GA-ES**

**4.6.2.1. Hari Sabtu, 25 Maret 2006**

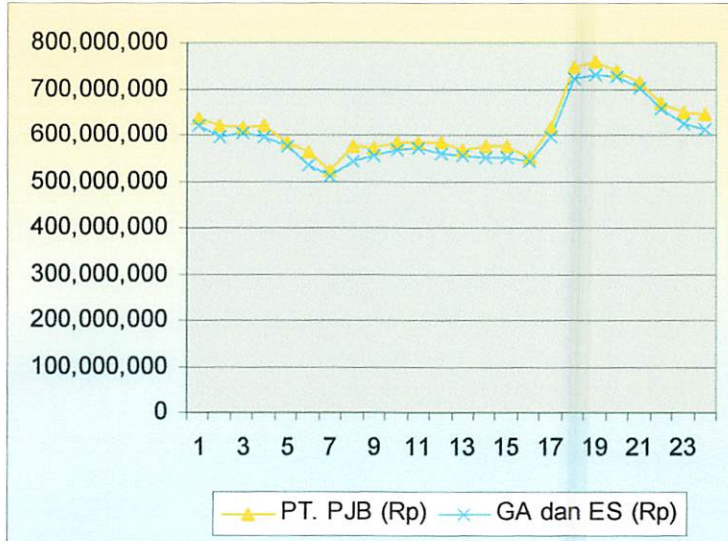
Tabel 4.8  
 Hasil Perhitungan Biaya Operasional PT. PLN PJB dan Biaya Operasional Menggunakan Metode GA-ES  
 Pada Hari: Sabtu, 25 Maret 2006

Jam	Beban Sistem (MW)	PT. PJB (Rp)	GA-ES (Rp)
1	2605	638,377,914	619,821,061
2	2465	619,608,327	596,059,273
3	2519	616,846,694	605,610,466
4	2464	621,946,317	596,663,711
5	2570	584,252,229	576,117,500
6	2293	564,239,693	534,318,225
7	2119	523,659,498	512,008,190
8	2336	575,064,682	541,491,412
9	2433	570,643,341	556,301,651
10	2528	582,161,576	569,108,010
11	2544	583,815,067	573,046,368
12	2449	583,781,450	558,685,611
13	2436	567,639,245	555,165,516
14	2395	577,148,404	550,545,694
15	2409	577,288,473	552,814,969
16	2331	551,301,502	541,480,698
17	2412	616,924,209	595,148,853
18	3157	748,717,715	721,638,032
19	3202	760,958,497	731,438,114
20	3175	740,204,811	725,287,902
21	3073	714,679,747	703,020,188
22	2803	669,298,622	657,622,518
23	2650	648,765,742	624,123,001
24	2583	643,182,522	613,183,525

Total biaya selama 1 hari yang diperoleh PT. Pembangkitan Jawa-Bali sebesar Rp. 14,880,506,277. Sedangkan total biaya yang diperoleh dari hasil perhitungan metode GA-ES sebesar Rp.14,410,700,488, sehingga didapat selisih sebesar Rp.469,805,789



Grafik 4.1  
 Perbandingan Biaya Operasional P.T. PLN PJB dengan Biaya  
 Operasional Menggunakan Metode GA-ES  
 Pada Hari: Sabtu, 25 Maret 2006



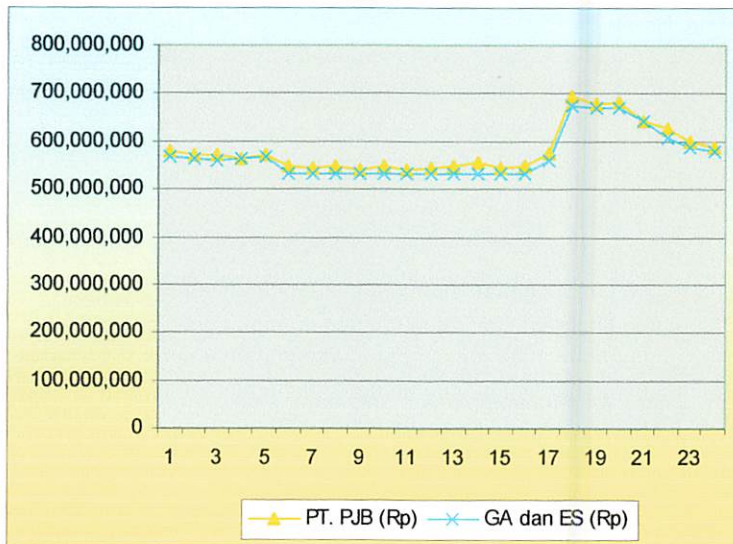
#### 4.6.2.2. Hari Minggu, 26 Maret 2006

Tabel 4.9  
Hasil Perhitungan Biaya Operasional PT. PLN PJB dan Biaya Operasional  
Menggunakan Metode GA-ES  
Pada Hari: Minggu, 25 Maret 2006

Jam	Beban Sistem (MW)	PT. PJB (Rp)	GA-ES (Rp)
1	2459	578,851,581	569,274,868
2	2427	570,084,120	563,258,208
3	2401	570,032,692	558,344,690
4	2409	563,330,312	561,893,168
5	2465	573,241,791	569,383,737
6	2168	549,125,663	530,522,165
7	2024	544,406,319	530,330,805
8	2045	546,301,664	530,391,630
9	2181	539,345,905	530,799,099
10	2149	547,455,316	530,097,720
11	2132	541,021,776	530,493,626
12	2118	542,401,689	530,483,913
13	2114	547,098,639	530,350,412
14	2201	555,707,793	533,563,824
15	2117	543,035,146	530,217,168
16	2149	547,322,850	530,474,706
17	2325	574,694,387	559,229,679
18	3130	691,694,001	675,455,373
19	3091	677,987,866	668,666,548
20	3086	683,302,822	670,731,914
21	2912	644,148,701	643,600,174
22	2678	626,790,732	606,274,469
23	2573	598,244,982	585,473,890
24	2527	588,857,033	577,835,069

Total biaya selama 1 hari yang diperoleh PT. Pembangkitan Jawa-Bali sebesar Rp. 13,944,483,780. Sedangkan total biaya yang diperoleh dari hasil perhitungan metode GA-ES sebesar Rp. 13,647,146,855, sehingga didapat selisih sebesar Rp. 297,336,925

Grafik 4.2  
 Perbandingan Biaya Operasional P.T. PLN PJB dengan Biaya  
 Operasional Menggunakan Metode GA-ES  
 Pada Hari: Minggu, 26 Maret 2006



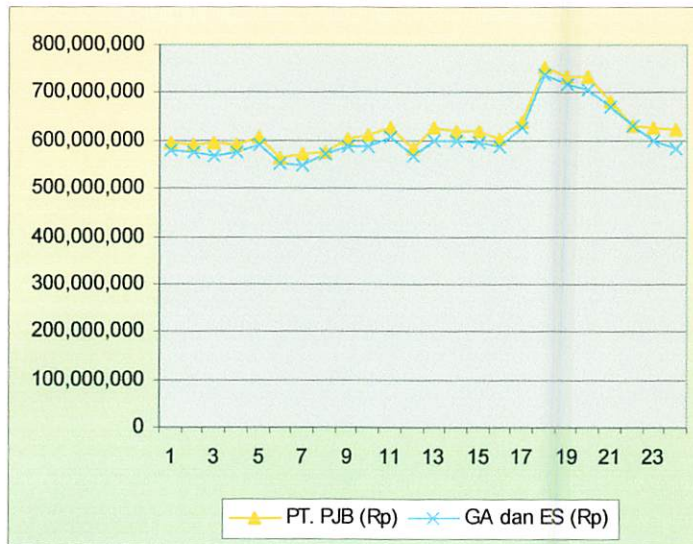
#### 4.6.2.3. Hari Rabu, 29 Maret 2006

Tabel 4.10  
Hasil Perhitungan Biaya Operasional PT. PLN PJB dan Biaya Operasional  
Menggunakan Metode GA-ES  
Pada Hari: Rabu, 29 Maret 2006

Jam	Beban Sistem (MW)	PT. PJB (Rp)	GA-ES (Rp)
1	2439	595,906,486	578,721,213
2	2429	590,022,240	573,845,825
3	2397	593,116,281	569,344,353
4	2397	593,000,980	573,571,621
5	2547	606,044,949	591,450,520
6	2254	563,642,905	551,967,006
7	2237	572,849,311	548,896,495
8	2396	576,473,705	570,452,270
9	2529	604,727,886	588,529,249
10	2513	611,923,732	587,037,313
11	2653	627,519,194	605,471,835
12	2380	585,180,222	569,039,734
13	2588	627,323,187	599,258,148
14	2608	620,118,493	599,478,917
15	2577	618,598,732	596,352,869
16	2507	602,558,953	587,052,590
17	2735	639,298,127	627,232,058
18	3342	754,279,170	736,168,565
19	3254	734,925,126	717,586,133
20	3205	732,454,236	706,762,136
21	3012	681,477,414	669,849,173
22	2777	631,879,841	629,555,135
23	2587	626,329,006	598,758,118
24	2478	621,387,397	581,499,729

Total biaya selama 1 hari yang diperoleh PT. Pembangkitan Jawa-Bali sebesar Rp. 15,011,037,573. Sedangkan total biaya yang diperoleh dari hasil perhitungan metode GA-ES sebesar Rp. 14,557,881,005, sehingga didapat selisih sebesar Rp. 453,156,568

Grafik 4.3  
Perbandingan Biaya Operasional P.T. PLN PJB dengan Biaya  
Operasional Menggunakan Metode GA-ES  
Pada Hari: Rabu, 29 Maret 2006



#### 4.6.3. Perbandingan Total Biaya PT. PLN PJB dengan Metode GA-ES

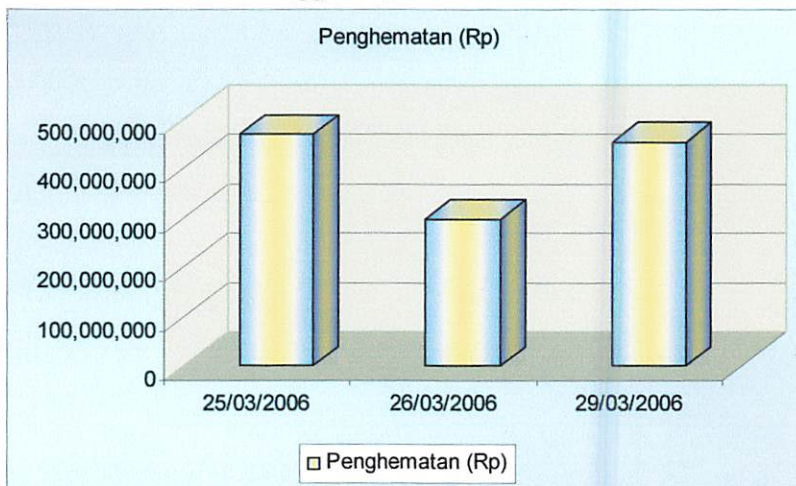
Tabel 4.11

Perbandingan Total Biaya Oprasional PT. PLN PJB dengan Metode GA-ES

TANGGAL	PT.PJB (Rp)	GA-ES (Rp)	Penghematan (Rp)	Persentase (%)
25/03/2006	14,880,506,277	14,410,700,488	469,805,789	3.157
26/03/2006	13,944,483,780	13,647,146,855	297,336,925	2.132
29/03/2006	15,011,037,573	14,557,881,005	453,156,568	3.018

Grafik 4.4

Selisih Perbandingan Total Biaya Operasional PT. PLN PJB dengan Total Biaya Operasional Metode GA-ES Pada Tanggal 25, 26 dan 29 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 GA-ES. Bila dilihat pada tabel diatas, tampak apabila dihitung dengan kedua cara, yaitu antara biaya yang dikeluarkan oleh PT. PJB dengan biaya hasil optimasi dengan menggunakan metode GA-ES terdapat perbedaan, ini berarti hasil yang diperoleh dari optimasi dengan menggunakan metode GA-ES menghasilkan penghematan yang relatif besar, sehingga tujuan utama dari optimasi sudah terpenuhi.

#### 4.6.4. Tampilan Program *Economic Load Dispatch* dengan Metode *Genetic Algorithm* dan *Evolution Strategies (EP-ES)*

##### 1. Tampilan Menu Utama



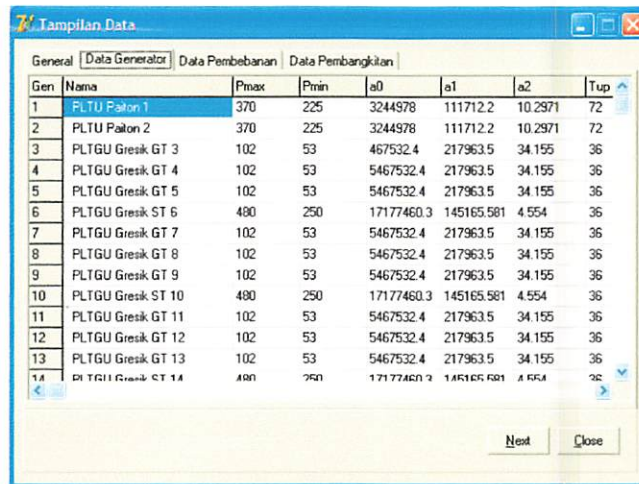
Gambar 4.16  
Tampilan Menu Utama

##### 2. Pilih Open File, kemudian akan muncul tampilan sebagai berikut



Gambar 4.17  
Tampilan Secara Umum

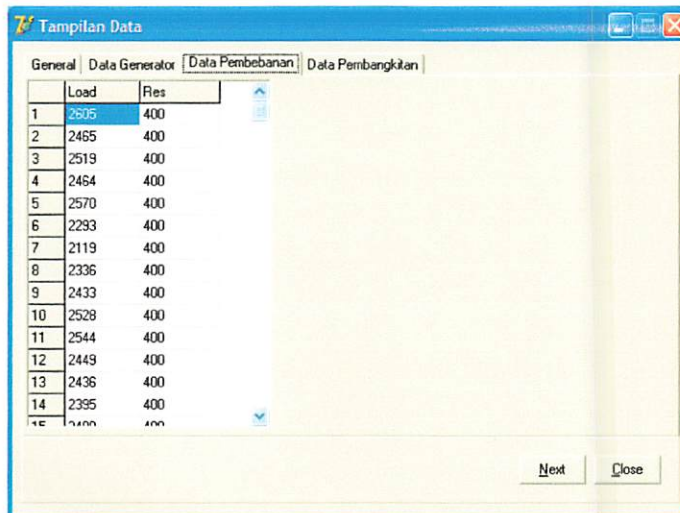
3. Pilih Data Generator untuk menampilkan unit – unit pembangkit



Gen	Nama	Pmax	Pmin	a0	a1	a2	Tup
1	PLTU Palton 1	370	225	3244978	111712.2	10.2971	72
2	PLTU Palton 2	370	225	3244978	111712.2	10.2971	72
3	PLTGU Gresik GT 3	102	53	5467532.4	217963.5	34.155	36
4	PLTGU Gresik GT 4	102	53	5467532.4	217963.5	34.155	36
5	PLTGU Gresik GT 5	102	53	5467532.4	217963.5	34.155	36
6	PLTGU Gresik ST 6	480	250	17177460.3	145165.581	4.554	36
7	PLTGU Gresik GT 7	102	53	5467532.4	217963.5	34.155	36
8	PLTGU Gresik GT 8	102	53	5467532.4	217963.5	34.155	36
9	PLTGU Gresik GT 9	102	53	5467532.4	217963.5	34.155	36
10	PLTGU Gresik ST 10	480	250	17177460.3	145165.581	4.554	36
11	PLTGU Gresik GT 11	102	53	5467532.4	217963.5	34.155	36
12	PLTGU Gresik GT 12	102	53	5467532.4	217963.5	34.155	36
13	PLTGU Gresik GT 13	102	53	5467532.4	217963.5	34.155	36
14	PLTGU Gresik ST 14	480	250	17177460.3	145165.581	4.554	36

Gambar 4.18  
Tampilan Data Generaor

4. Pilih Data Pembebanan untuk menampilkan beban sistem dan cadangan berputar



Load	Res	
1	2605	400
2	2485	400
3	2519	400
4	2464	400
5	2570	400
6	2293	400
7	2119	400
8	2336	400
9	2433	400
10	2528	400
11	2544	400
12	2449	400
13	2436	400
14	2395	400
15	2400	400

Gambar 4.19  
Tampilan Data Pembebanan



- Pilih Data PLN untuk mengetahui daya yang dioperasikan oleh tiap unit pembangkit thermal selama 24 jam

General	Data Generator	Data Pembebanan	Data Pembangkitan					Je
	Jam 1	Jam 2	Jam 3	Jam 4	Jam 5	Jam 6	Jam 7	Je
Gen 1	370	370	370	370	368	370	370	370
Gen 2	0	0	0	0	0	0	0	0
Gen 3	85.4	85.6	85.4	85	54.7	79.4	79.1	50
Gen 4	78.3	78.3	76.7	80	79.4	76.7	76.7	81
Gen 5	78	78	77.3	80.1	80.8	76.3	76.3	81
Gen 6	142	142	146.2	146.2	140	146.2	146.3	140
Gen 7	74.2	74	51.2	56.1	82.3	51.2	40.3	50
Gen 8	74.2	74	51.1	56.3	84.5	51.1	40.4	50
Gen 9	75	75	51	57	85.4	51	40.7	50
Gen 10	127	127	117.8	117.5	131.3	117.8	83.2	90
Gen 11	51.5	51	50.9	51.9	51.2	49.2	52.8	50
Gen 12	51.4	51	51.6	51.5	50.4	48.6	52.4	50
Gen 13	0	0	0	0	0	0	0	0
Gen 14	65.2	65	64.7	64.2	64.2	62.7	66.2	65

Gambar 4.20  
Tampilan Data PLN

- Kemudian tekan menu "Next", dan akan terlihat tampilan parameter EP-ES, kemudian tekan "Use Default" untuk memasukkan nilai dari parameter EP-ES, kemudian pilih jam ke berapa yang akan dihitung

Parameter EPdanES | Daya Generator | Grafik EPdanES

Parameter EPdanES

Jumlah Generasi: 500

Jumlah Populasi: 100

Panjang Chromosome: 37

Konstanta Beta: 1.5

Konstanta Ka: 100000000

Pilih Jam ke Berapa: 1

Parameter Objective Function

Penalty Generator: 1000000

Use Default

Hitung Close

Gambar 4.21  
Tampilan Parameter EP-ES

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

No	P EPdanES (MW)	P Jurnal (MW)	Cost EPdanES (\$)	Cost Jurnal (\$)	Selisih (\$)
1	368	370	45,797,519	45,980,165	190,646
2	54	85	12,253,845	19,330,713	7,076,868
3	54	78	17,261,701	22,743,475	5,481,774
4	53	78	17,153,483	22,676,484	5,523,001
5	475	142	87,146,675	37,882,800	-49,263,875
6	53	74	17,115,539	21,828,469	4,712,930
7	54	74	17,264,960	21,828,469	4,563,510
8	53	75	17,115,539	22,006,917	4,891,377
9	428	127	80,182,419	35,686,941	-44,495,479
10	53	52	17,115,539	16,783,240	-332,299
11	53	51	17,115,539	16,761,092	-354,447
12	281	65	58,287,568	26,661,615	-31,625,953
13	43	41	10,918,588	10,461,644	-456,944
14	90	83	21,816,918	20,317,319	-1,499,599
15	90	83	21,816,918	20,397,837	-1,419,081

Biaya EPES: 623,327,017      Biaya Jurnal: 638,377,914      Selisih: 15,050,897

Buttons: Hitung, Close

Gambar 4.22  
Tampilan Daya dari Unit yang Beroperasi Serta Hasil Biaya

**4.6.5. Hasil Perhitungan Biaya Operasional PT. PLN PJB dan Biaya Operasional Menggunakan Metode EP-ES**

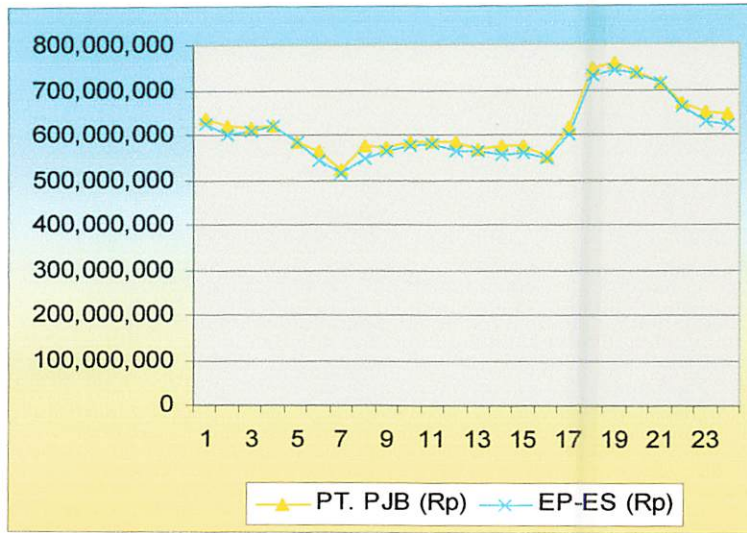
**4.6.5.1. Hari Sabtu, 25 Maret 2006**

Tabel 4.12  
 Hasil Perhitungan Biaya Operasional PT. PLN PJB dan Biaya Operasional Menggunakan Metode EP-ES  
 Pada Hari: Sabtu, 25 Maret 2006

Jam	Beban Sistem (MW)	PT. PJB (Rp)	EP-ES (Rp)
1	2605	638,377,914	623,327,017
2	2465	619,608,327	600,642,336
3	2519	616,846,694	610,050,107
4	2464	621,946,317	621,946,317
5	2570	584,252,229	582,620,499
6	2293	564,239,693	541,356,448
7	2119	523,659,498	515,230,213
8	2336	575,064,682	547,338,904
9	2433	570,643,341	561,918,843
10	2528	582,161,576	575,883,135
11	2544	583,815,067	578,616,675
12	2449	583,781,450	564,336,948
13	2436	567,639,245	562,227,329
14	2395	577,148,404	556,164,797
15	2409	577,288,473	558,494,664
16	2331	551,301,502	546,761,518
17	2412	616,924,209	601,791,021
18	3157	748,717,715	732,128,022
19	3202	760,958,497	742,174,836
20	3175	740,204,811	736,167,766
21	3073	714,679,747	713,733,877
22	2803	669,298,622	661,083,472
23	2650	648,765,742	629,697,225
24	2583	643,182,522	619,687,447

Total biaya selama 1 hari yang diperoleh PT. Pembangunan Jawa-Bali sebesar Rp. 14,880,506,277. Sedangkan total biaya yang diperoleh dari hasil perhitungan metode EP-ES sebesar Rp. 14,583,379,416, sehingga didapat selisih sebesar Rp. 297,126,861.

Grafik 4.5  
Perbandingan Biaya Operasional P.T. PLN PJB dengan Biaya  
Operasional Menggunakan Metode EP-ES  
Pada Hari: Sabtu, 25 Maret 2006



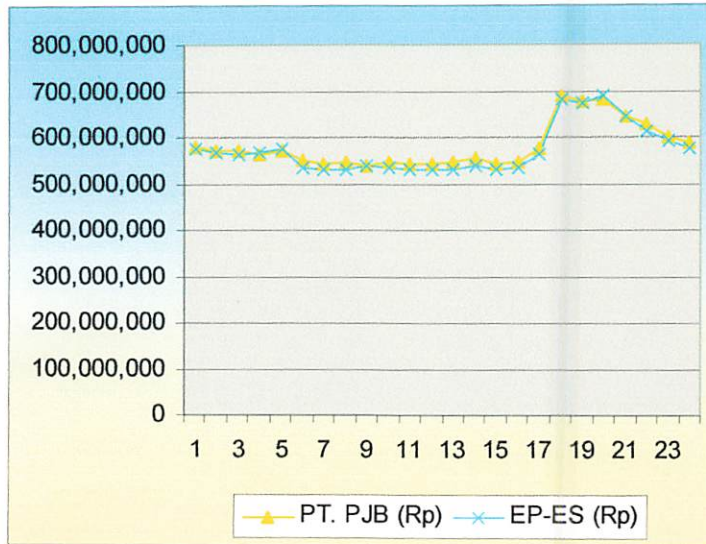
#### 4.6.5.2. Hari Minggu, 26 Maret 2006

Tabel 4.13  
Hasil Perhitungan Biaya Operasional PT. PLN PJB dan Biaya Operasional  
Menggunakan Metode EP-ES  
Pada Hari: Minggu, 25 Maret 2006

Jam	Beban Sistem (MW)	PT. PJB (Rp)	EP-ES (Rp)
1	2459	578,851,581	573,492,388
2	2427	570,084,120	568,237,471
3	2401	570,032,692	565,015,044
4	2409	563,330,312	566,073,509
5	2465	573,241,791	573,519,646
6	2168	549,125,663	535,262,937
7	2024	544,406,319	530,275,250
8	2045	546,301,664	530,228,233
9	2181	539,345,905	537,034,378
10	2149	547,455,316	532,960,188
11	2132	541,021,776	530,875,982
12	2118	542,401,689	530,195,785
13	2114	547,098,639	530,253,324
14	2201	555,707,793	539,062,769
15	2117	543,035,146	530,244,455
16	2149	547,322,850	533,035,967
17	2325	574,694,387	563,235,869
18	3130	691,694,001	679,887,812
19	3091	677,987,866	673,623,180
20	3086	683,302,822	688,489,556
21	2912	644,148,701	646,805,392
22	2678	626,790,732	613,180,980
23	2573	598,244,982	590,667,428
24	2527	588,857,033	576,477,751

Total biaya selama 1 hari yang diperoleh PT. Pembangkitan Jawa-Bali sebesar Rp. 13,944,483,780. Sedangkan total biaya yang diperoleh dari hasil perhitungan metode EP-ES sebesar Rp. 13,738,135,294, sehingga didapat selisih sebesar Rp. 206,348,486

Grafik 4.6  
 Perbandingan Biaya Operasional P.T. PLN PJB dengan Biaya  
 Operasional Menggunakan Metode EP-ES  
 Pada Hari: Minggu, 26 Maret 2006



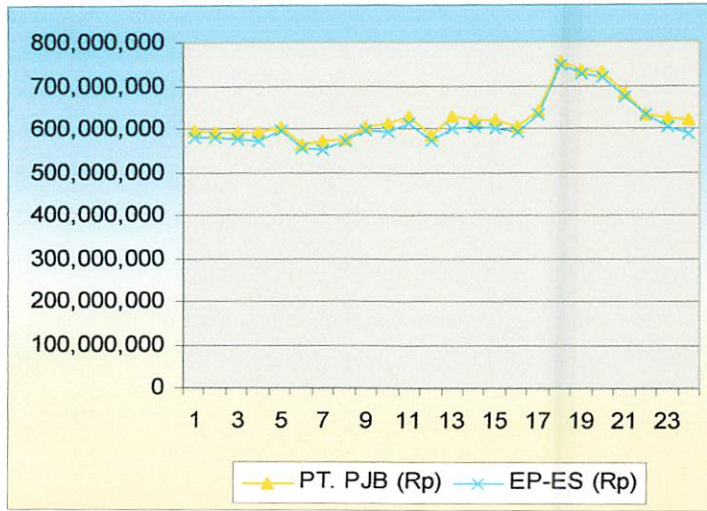
#### 4.6.5.3. Hari Rabu, 29 Maret 2006

Tabel 4.14  
Hasil Perhitungan Biaya Operasional PT. PLN PJB dan Biaya Operasional  
Menggunakan Metode EP-ES  
Pada Hari: Rabu, 29 Maret 2006

Jam	Beban Sistem (MW)	PT. PJB (Rp)	EP-ES (Rp)
1	2439	595,906,486	580,214,249
2	2429	590,022,240	578,941,827
3	2397	593,116,281	574,091,904
4	2397	593,000,980	572,614,608
5	2547	606,044,949	596,333,515
6	2254	563,642,905	555,956,546
7	2237	572,849,311	551,237,099
8	2396	576,473,705	572,320,429
9	2529	604,727,886	594,005,316
10	2513	611,923,732	591,397,711
11	2653	627,519,194	612,473,077
12	2380	585,180,222	571,629,251
13	2588	627,323,187	600,974,998
14	2608	620,118,493	605,770,131
15	2577	618,598,732	600,963,704
16	2507	602,558,953	590,662,692
17	2735	639,298,127	632,039,406
18	3342	754,279,170	747,202,295
19	3254	734,925,126	727,818,974
20	3205	732,454,236	716,711,621
21	3012	681,477,414	675,270,344
22	2777	631,879,841	633,900,588
23	2587	626,329,006	602,124,025
24	2478	621,387,397	586,189,932

Total biaya selama 1 hari yang diperoleh PT. Pembangkitan Jawa-Bali sebesar Rp. 15,011,037,573. Sedangkan total biaya yang diperoleh dari hasil perhitungan metode EP-ES sebesar Rp. 14,670,844,242, sehingga didapat selisih sebesar Rp. 340,193,331.

Grafik 4.7  
 Perbandingan Biaya Operasional P.T. PLN PJB dengan Biaya  
 Operasional Menggunakan Metode EP-ES  
 Pada Hari: Rabu, 29 Maret 2006





#### 4.6.6. Perbandingan Total Biaya PT. PLN PJB dengan Metode EP-ES

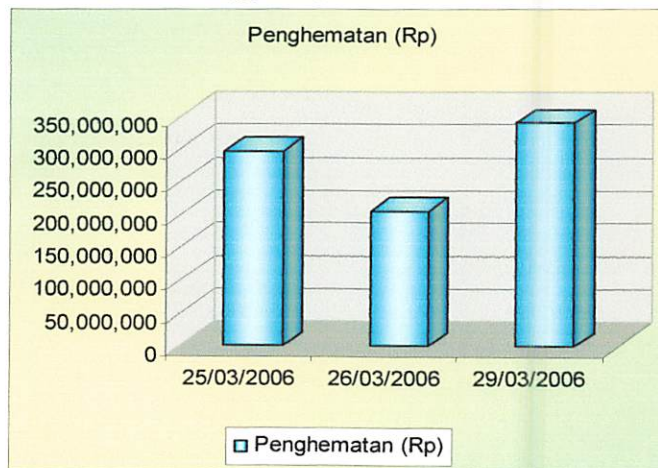
Tabel 4.15

Perbandingan Total Biaya Oprasional PT. PLN PJB dengan Metode EP-ES

TANGGAL	PT.PJB (Rp)	EP-ES (Rp)	Penghematan (Rp)	Persentase (%)
25/03/2006	14,880,506,277	14,583,379,416	297,126,861	1.996
26/03/2006	13,944,483,780	13,738,135,294	206,348,486	1.479
29/03/2006	15,011,037,573	14,670,844,242	340,193,331	2.266

Grafik 4.8

Selisih Perbandingan Total Biaya Operasional PT. PLN PJB dengan Total Biaya Operasional Metode EP-ES Pada Tanggal 25, 26 dan 29 Juli 2006



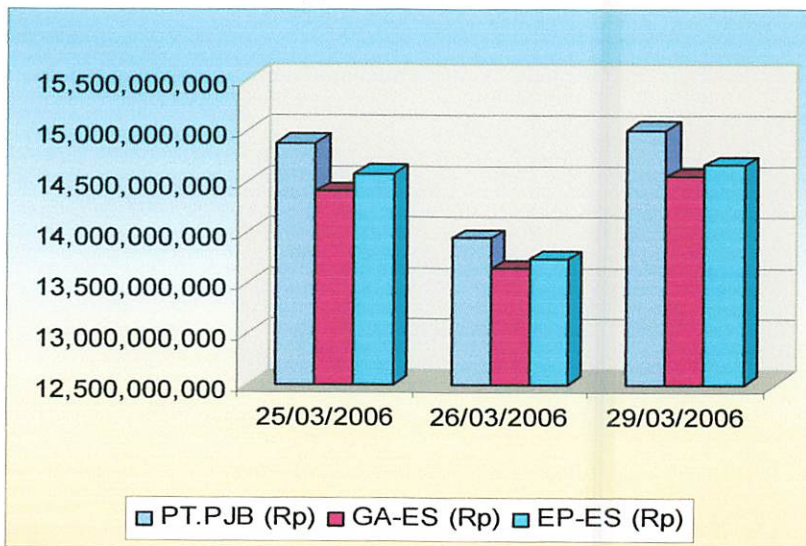
Dari tabel perbandingan biaya operasional untuk tiap jam selama 24 jam, dapat dibuat perbandingan biaya total operasional untuk periode 24 jam (satu hari) antara biaya total operasional selama 24 jam milik PT. PJB dengan biaya operasional selama 24 jam hasil optimasi metode EP-ES. Bila dilihat pada tabel diatas, tampak apabila dihitung dengan kedua cara, yaitu antara biaya yang dikeluarkan oleh PT. PJB dengan biaya hasil optimasi dengan menggunakan metode EP-ES terdapat perbedaan, ini berarti hasil yang diperoleh dari optimasi dengan menggunakan metode EP-ES menghasilkan penghematan yang relatif besar, sehingga tujuan utama dari optimasi sudah terpenuhi.

**4.6.7. Perbandingan Total Biaya Operasional Metode GA-ES Dan Metode EP-ES Terhadap PT. PLN PJB.**

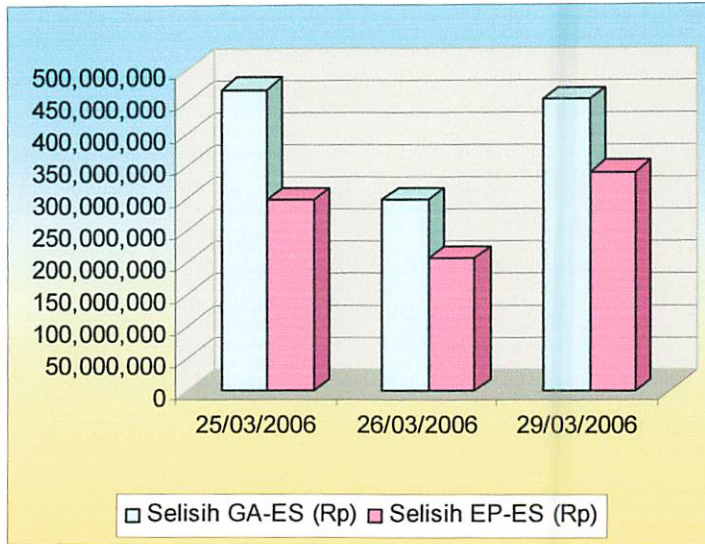
Tabel 4.16  
Perbandingan Total Biaya Operasional Metode GA-ES Dan Metode EP-ES Terhadap PT. PLN PJB

TGL	PT.PJB	GA-ES	Selisih GA-ES	Persen tase	EP-ES	Selisih EP-ES	Persen tase
	(Rp)	(Rp)	(Rp)	(%)	(Rp)	(Rp)	(%)
25/03/2006	14,880,506,277	14,410,700,488	469,805,789	3.157	14,583,379,416	297,126,861	1.996
26/03/2006	13,944,483,780	13,647,146,855	297,336,925	2.132	13,738,135,294	206,348,486	1.479
29/03/2006	15,011,037,573	14,557,881,005	453,156,568	3.018	14,670,844,242	340,193,331	2.266

Grafik 4.9  
Perbandingan Total Biaya Operasional Metode GA-ES dan Metode EP-ES Terhadap PT. PLN PJB Pada Tanggal 25, 26 dan 29 Juli 2006



Grafik 4.10  
Selisih Perbandingan Total Biaya Operasional  
Metode GA-ES dan Metode EP-  
Pada Tanggal 25, 26 dan 29 Juli 2006



Didapat selisih total biaya bahan bakar dari perbandingan metode *Genetic Algorithm-Evolution Strategies* (GA-ES) dan metode *Evolutionary Programming-Evolution Strategies* (EP-ES) sehingga tujuan utama optimasi bisa terpenuhi.

## BAB V

### KESIMPULAN

#### 5.1 Kesimpulan.

Setelah dilakukan analisa Pembebanan Ekonomis dengan menggunakan perbandingan metode *Genetic Algorithm-Evolution Strategies* (GA-ES) dan metode *Evolutionary Programming-Evolution Strategies* (EP-ES), maka dapat diambil kesimpulan sebagai berikut:

1. Pada hari Sabtu, tanggal 25 Maret 2006 biaya yang dikeluarkan oleh PT. PJB sebesar Rp. 14.880.506.227, perbandingan optimasi dengan menggunakan metode *Genetic Algorithm-Evolution Strategies* (GA-ES) diperoleh hasil sebesar Rp. 14.410.700.488, penghematan sebesar Rp. 469.805.789 atau 3.157%, sedangkan untuk metode *Evolutionary Programming-Evolution Strategies* (EP-ES) diperoleh hasil sebesar Rp. 14.583.379.416, penghematan sebesar Rp. 297.126.861 atau 1.996%.
2. Pada hari Minggu, tanggal 26 Maret 2006 biaya yang dikeluarkan oleh PT. PJB sebesar Rp. 13.944.483.780, perbandingan optimasi dengan menggunakan metode *Genetic Algorithm-Evolution Strategies* (GA-ES) diperoleh hasil sebesar Rp. 13.647.146.855, penghematan sebesar Rp. 297.336.925 atau 2.132%, sedangkan untuk metode *Evolutionary Programming-Evolution Strategies* (EP-ES) diperoleh hasil sebesar Rp. 13.738.135.294, penghematan sebesar Rp. 206.348.486 atau 1.479%.

3. Pada hari Rabu tanggal, 29 Maret 2006 biaya yang dikeluarkan oleh PT. PJB sebesar Rp. 15.011.037.573, perbandingan optimasi dengan menggunakan metode *Genetic Algorithm-Evolution Strategies* (GA-ES) diperoleh hasil sebesar Rp. 14.557.881.005, penghematan sebesar Rp. 453.156.568 atau 3.018%, sedangkan untuk metode *Evolutionary Programming-Evolution Strategies* (EP-ES) diperoleh hasil sebesar Rp. 14.670.884.242, penghematan sebesar Rp. 340.193.331 atau 2.266%.
4. Dari data diatas dapat disimpulkan bahwa metode *Genetic Algorithm-Evolution Strategies* (GA-ES) lebih layak digunakan dibandingkan dengan metode *Evolutionary Programming-Evolution Strategies* (EP-ES), untuk mencari optimasi biaya biaya bahan bakar.

## 5.2. Saran

Agar didapatkan hasil optimasi pada permasalahan yang akan diangkat lebih akurat pada kedepannya, maka sebaiknya analisis dilakukan dengan memperhitungkan data-data pendukung yang lebih spesifik.

## DAFTAR PUSTAKA

- [1] C. E. Lin and G. L. Viviani, "*Hierarchical Economic Dispatch for Piecewise Quadratic Cost Functions*", IEEE Trans. on PAS, Vol. PAS-103, No. 6, June, 1984.
- [2] Djiteng Marsudi, Ir "*Operasi Sistem Tenaga Listrik*", Penerbit Balai Penerbit Humas ISTN, Juni 1990.
- [3] <http://zooland.alife.org/hhg2ec/>, "*What's Evolutionary Programming (EP) dan What's Evolutionary Strategy (ES)?*".
- [4] June Ho Park\*, Sung-Oh Yang\*, Hwa Seok Lee\*, Young Moon Park\*\* "*Economic Load Dispatch Using Evolutionary Algorithms*", IEEE 1996.
- [5] Kwang. Y. Lee, and Frank F. Yang, "*Optimal Reactive Power Planning Using Evolutionary Algorithms: A Comparative Study for Evolutionary Programming, Evolutionary Strategy, Genetic Algorithm, and Linier Programming*", IEEE Trans on Power System, Vol 13, No 1, February 1998.
- [6] Luke, Brian T., "*Overview of Evolutionary Programming Methods*", Learning From The Web.net.
- [7] Stevenson, William D, Jr, "*Analisa Sistem Tenaga Listrik*", Penerbit Erlangga edisi keempat, 1996.
- [8] Wood, Allan J and B. F. Wollenberg, "*Power Generation, Operation, and Control*", John Willey & Sons, Inc, 1984.

# **LAMPIRAN**



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

## BERITA ACARA UJIAN SKRIPSI FAKULTAS TEKNOLOGI INDUSTRI

Nama Mahasiswa : MAHRUS ALI  
N.I.M. : 00.12.081  
Jurusan : Teknik Elektro  
Konsentrasi : Teknik Energi Listrik S-1  
Judul Skripsi : ANALISIS PERBANDINGAN EKONOMIS  
DENGAN MENGGUNAKAN *GENETIC  
ALGORITHM-EVOLUTIONARY PROGRAMMING-  
EVOLUTION STRETEGIES*  
DI PT. PEMBANGKITAN JAWA-BALI.

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

Hari : Senen  
Tanggal : 25 September 2006  
Dengan Nilai : 80.04 (A)



(Ir. Mochtar Asroni, MSME)  
Ketua

Panitia Ujian

(Ir. F. Yudi Limpraptono, MT)  
Sekretaris

Anggota Penguji

(Ir. Eko Nurcahyo)  
Penguji Pertama

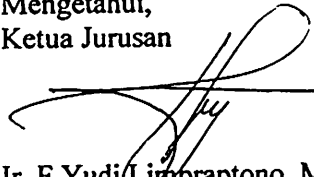
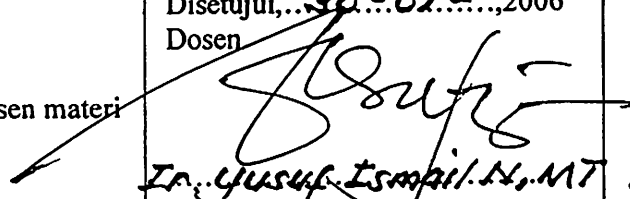
(Ir. Choirul Saleh, MT)  
Penguji Kedua





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

Konsentrasi : Teknik Energi Listrik/Teknik Elektronika \*)

1.	Nama Mahasiswa : <b>Mahrus Ali</b>	Nim : <b>00.12.081</b>		
2.	Waktu Pengajuan :	Tanggal : <b>20</b>	Bulan : <b>01</b>	Tahun : <b>2006</b>
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 .....		
4.	Konsultasikan judul sesuai materi bidang ilmu kepada Dosen *) : <i>Ir. Yusuf Ismail (M, MT)</i>	Mengetahui, Ketua Jurusan  Ir. F Yudi Limpraptono, MT Nip. Y.1039500274		
5.	Judul yang diajukan mahasiswa	<b>ANALISA PEMBEBANAN EKONOMIS DENGAN MENGGUNAKAN METODE GENETIC ALGORITHMS DAN METODE EVOLUTION STRATEGIES (GA+ES) DI PT. PEMBANGKITAN JAWA-BALI</b>		
6.	Perubahan Judul yang disetujui Dosen sesuai materi bidang ilmu	<i>Analisa Pembebanan Ekonomis Menggunakan Metode Kombinasi Genetic Algorithm - Evolutionary Programming - Evolution Strategies di PT. PJB</i>		
7.	Catatan :			
8.	Persetujuan Judul Skripsi yang dikonsultasikan kepada Kelompok Dosen materi bidang ilmu	Disetujui, <i>30.01.2006</i> , 2006 Dosen  Ir. Yusuf Ismail, M, MT		

**Perhatian :**

1. Formulir Pengajuan ini harap dikembalikan ke Jurusan paling lambat satu minggu setelah disetujui kelompok dosen keahliian 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.



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

Konsentrasi : Teknik Energi Listrik / ~~Teknik Elektronika~~ \*)

1	Nama Mahasiswa : <b>ROSTANRUS ALI</b>		Nim : <b>00.12081</b>	
2	Keterangan	Tanggal	Waktu	Tempat
	Pelaksanaan	<b>24-02-2006</b>	<b>11.15 - 1200</b>	Ruang : <b>Seminar</b>
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	Analisa pembebanan ekonomis menggunakan kombinasi Genetik ALGORITHM - Evoluinary programming - Evolution strategies di P.T. 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 ybs  Ir. Gubuk Ismail Nalenda M.T.		

Perhatian :

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

INSTITUT TEKNOLOGI NASIONAL  
Jl. Bendungan Sigura-gura No. 2  
MALANG

Lampiran : 1 (satu) berkas  
Pembimbing Skripsi

Kepada : Yth. Bapak/Ibu P. ~~Christi~~ Yusuf M.  
Dosen Institut Teknologi Nasional  
MALANG

Yang bertanda tangan di bawah ini :

Nama : MAHRUS ALI  
Nim : 00.12.081  
Jurusan : Teknik Elektro S-1  
Konsentrasi : Teknik Elektronika/Energi Listrik

Dengan ini mengajukan permohonan, kiranya Bapak/Ibu bersedia menjadi Dosen Pembimbing Utama / Pendamping <sup>\*)</sup>, untuk penyusunan Skripsi dengan judul (proposal terlampir) :

Analisa pembebanan Ekonomis menggunakan metode Kombinasi  
Genetic Algorithm - Evolutionary programming - Evolution Strategies  
di PT. PJB  
Adapun tugas tersebut sebagai salah satu syarat untuk menempuh Ujian Akhir Sarjana Teknik.

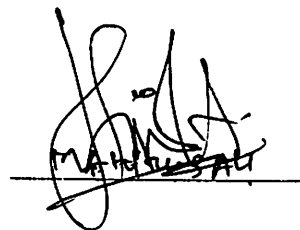
Demikian permohonan kami dan atas kesediaan Bapak/Ibu kami ucapkan terima kasih.

Malang,

Ketua  
Jurusan Teknik Elektro S-1

Hormat kami,

Ir. F. Yudi Limpraptono, MT  
Nip. 1039500274



\*) coret yang tidak perlu

**PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI**

Sesuai permohonan dari mahasiswa /i :

Nama : **Mahrus Ali**

Nim : **00.12081**

Semester : **XII (Dua Belas)**

Jurusan : **Teknik Elektro S-1**

Konsentrasi : **Energi Listrik**

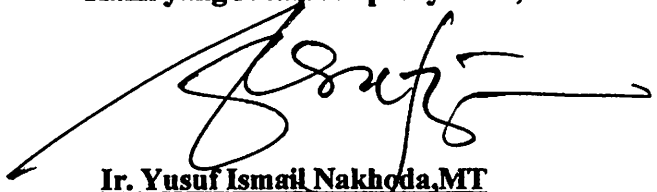
Dengan ini Menyatakan bersedia / ~~tidak bersedia~~ ) Membimbing Skripsi dari mahasiswa tersebut, dengan judul :

**“ANALISA PEMBEBANAN EKONOMIS MENGGUNAKAN KOMBINASI  
GENETIC ALGORITHM- EVOLUTIONARY PROGAMMING-EVOLUTIONARY  
STRATEGIES DI P.T. PEMBANGKITAN JAWA-BALI”**

Demikian surat Pernyataan ini kami buat agar dapat digunakan seperlunya.

Malang, 7 Febuari 2006

Kami yang Membuat pernyataan,



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

**Catatan:**

Setelah disetujui agar formulir ini  
Diserahkan mahasiswa/i yang bersangkutan  
Kepada Jurusan untuk diproses lebih lanjut.  
) Coret yang tidak perlu



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG  
INSTITUT TEKNOLOGI NASIONAL MALANG  
FAKULTAS TEKNOLOGI INDUSTRI  
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN  
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG  
BANK NIAGA 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, 9 Agt. 2006

Nomor : ITN-1643/I.TA/2/'06  
Lampiran :  
Perihal : Bimbingan Skripsi

Kepada : Yth. Sdr. Ir. YUSUF ISMAIL NAKHODA, MT

Dosen Pembimbing  
Jurusan Teknik Elektro S-1  
di  
Malang

Dengan hormat,  
Sesuai dengan permohonan dan persetujuan dalam proposal skripsi  
untuk mahasiswa:

Nama : MAHRUS ALI  
Nim : 0012081  
Fakultas : Teknologi Industri  
Jurusan : Teknik Elektro S-1  
Konsentrasi : Teknik Energi listrik

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya  
kepada Saudara/i selama masa waktu 6 (enam) bulan, terhitung mulai  
tanggal:

24 Feb. 2006 s/d 24 Agt. 2006

Sebagai satu syarat untuk menempuh Ujian Sarjana Teknik,  
Jurusan Teknik Elektro  
Demikian atas perhatian serta kerjasamanya yang baik kami ucapkan  
terima kasih



Ketua Jurusan  
Teknik Elektro S-1

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

Form S-4a



## FORMULIR BIMBINGAN SKRIPSI

Nama : **Mahrus Ali**  
Nim : **00.12081**  
Masa Bimbingan : **24 Febuari 2006 - 24 Agustus 2006**  
Judul Skripsi : **ANALISA PERBANDINGAN PEMBEBANAN EKONOMIS  
MENGUNAKAN KOMBINASI *GENETIC ALGORITHM* -  
*EVOLUTIONARY PROGRAMMING* - *EVOLUTION*  
*STRATEGIES* DI P.T. PEMBANGKITAN JAWA-BALI**

No	Tanggal	Uraian	Paraf Pembimbing
1	14-08-2006	BAB I : - Perbaiki Sistematika - Perbaiki Batasan masalah dan tujuan	
2	23-08-2006	BAB II : - Perbaiki Teori Dasar - Pelajari Dasar Pembebanan Ekonomis - Format Index	
3	30-08-2006	BAB III : - Perbaiki Teori Dasar Metode - Perbaiki <i>Flowchart</i>	
4	04-09-2006	BAB IV : - Tampilkan Tabel Uji Validasi - Buat Tabel Perbandingan Hasil dari Metode GA-ES dan EP-ES terhadap PT. PLN PJB - Buat Grafik Perbandimngan Hasil dari Metode GA-ES dan EP-ES terhadap PT. PLN PJB	
5	09-09-2006	BAB V : - Perbaiki Kesimpulan - Buat Tabel Perbandingan Hasil dari Metode GA-ES dan EP-ES terhadap PT. PLN PJB	
6	14-09-2006	Buat Makalah Seminar Hasil	
7	16-09-2006	Acc Seminar Hasil	
8	22-09-2006	Acc Kompre	

Malang, September 2006  
Dosen Pembimbing

**Ir. Yusuf Ismail Nakhoda, MT**  
NIP.P. 101.880.0189

Form.S-4b



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

---

## LEMBAR BIMBINGAN SKRIPSI

1. Nama : MAHRUS ALI
2. NIM : 00.12.081
3. Jurusan : Teknik Elektro
4. Konsentrasi : Teknik Energi Listrik S-1
5. Judul Skripsi : ANALISIS PERBANDINGAN EKONOMIS  
DENGAN MENGGUNAKAN *GENETIC  
ALGORITHM-EVOLUTIONARY  
PROGRAMMING-EVOLUTION  
STRETEGIES*  
DI PT. PEMBANGKITAN JAWA-BALI.
6. Tanggal Mengajukan Skripsi : 24 Februari 2006
7. Tanggal Menyelesaikan Skripsi : 24 Agustus 2006
8. Dosen Pembimbing : Ir. Yusuf Ismail Nakhoda, MT
10. Telah dievaluasi dengan nilai : 87,00 (A)

Malang, 25 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

```

unit uEvoPro;

interface

uses uUtils,uGenVar,uGenetic, SysUtils,uFitness2,uHasil,uRandom;

type
  TEvoPro=class(TGenetic)
  private
    FBetha:double;
  public
    constructor Create(const rMaxGen,rPopSize,rLength:integer;
      const rKa,rBetha:double);
    property Betha:double read FBetha write FBetha;
  end;

  TEvoPro1=class(TEvoPro)
  private
    FBatas:TBatasArr1;
    FParent,FChild:TPopDouble1;
    FBestIndi:TIndiDouble1;
    function getBatas:TBatasArr1;
    procedure setBatas(const rBatas:TBatasArr1);
    function getIndividu(const rIndi:TIndiDouble1):TIndiDouble1;
    function FindIndiMax:TIndiDouble1;
    function FindFitnessMax:double;
    procedure SwapIndi(var rIndi1,rIndi2:TIndiDouble1);
    procedure InitParent;
    procedure Statistik;
    procedure Generasi;
    procedure Kompetisi;
    procedure doHitung;
    function getBestChrom:dArr1;
  public
    constructor Create(const rMaxGen,rPopSize,rLength:integer;
      const rKa,rBetha:double;
      const rBatas:TBatasArr1);
    property Batas:TBatasArr1 read getBatas write setBatas;
    property BestChrom:dArr1 read getBestChrom;
  end;

implementation

//constructor
constructor TEvoPro.Create(const rMaxGen,rPopSize,rLength:integer;
  const rKa,rBetha:double);
begin
  inherited Create(rMaxGen,rPopSize,rLength,rKa);

```



```

    FBetha:=rBetha;
end;

//constructor
constructor TEvoPro1.Create(const rMaxGen,rPopSize,rLength:integer;
    const rKa,rBetha:double;
    const rBatas:TBatasArr1);
var i:integer;
begin
    if (high(rBatas)+1) <> rLength then
        begin
            raise Exception.Create('Length dan Batas arraynya tidak sama!');
        end;
    inherited Create(rMaxGen,rPopSize,rLength,rKa,rBetha);
    SetLength(FBatas,Length);
    for i:=0 to Length-1 do
        begin
            FBatas[i].min:=rBatas[i].min;
            FBatas[i].max:=rBatas[i].max;
        end;
    end;
end;

//data accessing
function TEvoPro1.getBatas:TBatasArr1;
var i:integer;
begin
    SetLength(result,Length);
    for i:=0 to Length-1 do
        begin
            result[i].min:=FBatas[i].min;
            result[i].max:=FBatas[i].max;
        end;
    end;
end;

procedure TEvoPro1.setBatas(const rBatas:TBatasArr1);
var i:integer;
begin
    if (high(rBatas)+1) <> Length then
        begin
            raise Exception.Create('Length dan Batas arraynya tidak sama!');
        end;
    SetLength(FBatas,Length);
    for i:=0 to Length-1 do
        begin
            FBatas[i].min:=rBatas[i].min;
            FBatas[i].max:=rBatas[i].max;
        end;
    end;
end;

```

```

end;

//data processing
function TEvoPro1.getIndividu(const rIndi:TIndiDouble1):TIndiDouble1;
var i:integer;
begin
  SetLength(result.chrom,Length);
  for i:=0 to Length-1 do
    begin
      result.chrom[i]:=rIndi.chrom[i];
    end;
  result.fitness:=rIndi.fitness;
end;

function TEvoPro1.FindIndiMax:TIndiDouble1;
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;

function TEvoPro1.FindFitnessMax:double;
var i:integer;
begin
  result:=FParent[0].fitness;
  for i:=1 to PopSize-1 do
    begin
      if result<FParent[i].fitness then
        begin
          result:=FParent[i].fitness;
        end;
    end;
end;

procedure TEvoPro1.SwapIndi(var rIndi1,rIndi2:TIndiDouble1);
var tmpIndi:TIndiDouble1;
begin
  tmpIndi:=getIndividu(rIndi1);
  rIndi1:=getIndividu(rIndi2);
  rIndi2:=getIndividu(tmpIndi);
end;

```

```

procedure TEvoPro1.InitParent;
var i,j:integer;
begin
    SetLength(FParent,PopSize);
    SetLength(FChild,PopSize);
    SetLength(FMin,MaxGen);
    SetLength(FAvg,MaxGen);
    SetLength(FMax,MaxGen);
    for i:=0 to PopSize-1 do
        begin
            SetLength(FParent[i].chrom,Length);
            SetLength(FChild[i].chrom,Length);
        end;
    SetLength(FBestIndi.chrom,Length);
    for i:=0 to PopSize-1 do
        begin
            for j:=0 to Length-1 do
                begin
                    FParent[i].chrom[j]:=GetBatasToReal(FRandom.NextDouble,FBatas[j].min,
                        FBatas[j].max);
                end;
            FParent[i].fitness:=Ka/gFitness2.doCalcCost(FParent[i].chrom);
        end;
    end;

```

```

procedure TEvoPro1.Statistik;
var i:integer;
begin
    Min1:=FParent[0].fitness;
    Max1:=FParent[0].fitness;
    SumFitness:=FParent[0].fitness;
    for i:=1 to PopSize-1 do
        begin
            if Min1>FParent[i].fitness then
                begin
                    Min1:=FParent[i].fitness;
                end;
            if Max1<FParent[i].fitness then
                begin
                    Max1:=FParent[i].fitness;
                end;
            SumFitness:=SumFitness+FParent[i].fitness;
        end;
    Avg1:=SumFitness/PopSize;
end;

```

```

procedure TEvoPro1.Generasi;
var i,j:integer;

```

```

Fmax,fitEP,fitES:double;
tho:double;
chromEP,chromES:dArr1;
begin
Fmax:=FindFitnessMax;
SetLength(chromEP,Length);
SetLength(chromES,Length);
for i:=0 to PopSize-1 do
begin
for j:=0 to Length-1 do
begin
//tho:=Betha*(FBatas[j].max-FBatas[j].min)*FMax/FParent[i].fitness;
tho:=(FBatas[j].max-FBatas[j].min)*((Fmax-FParent[i].fitness)/
Fmax+Betha);
chromEP[j]:=FParent[i].chrom[j]+FRandom.NextGaussian(0,sqr(tho));
if chromEP[j]>FBatas[j].max then
begin
chromEP[j]:=FBatas[j].max;
end;
if chromEP[j]<FBatas[j].min then
begin
chromEP[j]:=FBatas[j].min;
end;
end;
fitEP:=Ka/gFitness2.doCalcCost(chromEP);
for j:=0 to Length-1 do
begin
chromES[j]:=FParent[i].chrom[j]+FRandom.NextGaussian(0,Betha);
if chromES[j]>FBatas[j].max then
begin
chromES[j]:=FBatas[j].max;
end;
if chromES[j]<FBatas[j].min then
begin
chromES[j]:=FBatas[j].min;
end;
end;
fitES:=Ka/gFitness2.doCalcCost(chromES);
if fitEP>fitES then
begin
for j:=0 to Length-1 do
begin
FChild[i].chrom[j]:=chromEP[j];
end;
FChild[i].fitness:=fitEP;
end
else
begin

```

```

    for j:=0 to Length-1 do
    begin
        FChild[i].chrom[j]:=chromES[j];
    end;
    FChild[i].fitness:=fitES;
end;
end;
end;

procedure TEvoPro1.Kompetisi;
var i,j,sa,Ntmp:integer;
    tmpPop:TPopDouble1;
    sort:iArr1;
begin
    Ntmp:=2*PopSize;
    SetLength(tmpPop,Ntmp);
    SetLength(sort,Ntmp);
    for i:=0 to PopSize-1 do
    begin
        tmpPop[i]:=getIndividu(FParent[i]);
        tmpPop[PopSize+i]:=getIndividu(FChild[i]);
        sort[i]:=0;
        sort[PopSize+i]:=0;
    end;
    for i:=0 to Ntmp-1 do
    begin
        for j:=0 to Ntmp-2 do
        begin
            repeat
                sa:=FRandom.NextInt(0,(Ntmp-1));
            until sa<>i;
            if tmpPop[i].fitness>tmpPop[sa].fitness then
            begin
                inc(sort[i]);
            end;
        end;
    end;
    for i:=0 to Ntmp-2 do
    begin
        for j:=i to Ntmp-1 do
        begin
            if sort[i]<sort[j] then
            begin
                Swap(sort[i],sort[j]);
                SwapIndi(tmpPop[i],tmpPop[j]);
            end;
        end;
    end;
end;
end;
end;

```

```

for i:=0 to PopSize-1 do
begin
  FParent[i]:=getIndividu(tmpPop[i]);
end;
end;

procedure TEvoPro1.doHitung;
var gen:integer;
    TempIndi:TIndiDouble1;
begin
  InitParent;
  Statistik;
  FBestIndi:=FindIndiMax;
  gen:=1;
  repeat
    Generasi;
    Kompetisi;
    Statistik;
    TempIndi:=FindIndiMax;
    if FBestIndi.fitness<TempIndi.fitness then
    begin
      FBestIndi:=GetIndividu(TempIndi);
    end;
    FMin[gen-1]:=Min1;
    FAvg[gen-1]:=Avg1;
    FMax[gen-1]:=Max1;
    frmHasil.pblterasi.StepBy(1);
    inc(gen);
  until (gen>MaxGen);
end;

function TEvoPro1.getBestChrom:dArr1;
var i:integer;
begin
  doHitung;
  SetLength(result,Length);
  for i:=0 to Length-1 do
  begin
    result[i]:=FBestIndi.chrom[i];
  end;
end;
end.

```

```

unit uFitness2;

interface

uses uUtils,uGenerator,uRandom;

type
  TFitness2=class
  private
    FNgen:integer;
    FLoad,FPinGen:double;
    FGen:TGenArr;
    function getGen:TGenArr;
    procedure setGen(const rGen:TGenArr);
    function getBatasGen:TBatasArr1;
  public
    constructor Create(const rPinGen:double;
      var rGen:TGenArr;
      const rLoad:double);
    destructor Destroy;override;
    function getRandomGen:dArr1;
    function doCalcCost(var rChrom:dArr1):double;
    function doRecombination(var rChrom1,rChrom2:dArr1):dArr1;
    function doMutasi(const rNo:integer;
      const rPgen,rCk,rCmin,rBetha:double;
      var rRand:TRandomu):double;
    property Ngen:integer read FNgen write FNgen;
    property Load:double read FLoad write FLoad;
    property Gen:TGenArr read getGen write SetGen;
    property PinGen:double read FPinGen write FPinGen;
    property BatasGen:TBatasArr1 read getBatasGen;
  end;

var gFitness2:TFitness2;

implementation

//constructor
constructor TFitness2.Create(const rPinGen:double;
  var rGen:TGenArr;
  const rLoad:double);
var i:integer;
begin
  inherited Create;
  FNgen:=high(rGen)+1;
  SetLength(FGen,FNgen);
  for i:=0 to FNgen-1 do
  begin

```

```

    FGen[i]:=TPembangkit.Create(rGen[i]);
end;
FLoad:=rLoad;
FPinGen:=rPinGen;
end;

//destructor
destructor TFitness2.Destroy;
var i:integer;
begin
    try
        for i:=0 to FNgen-1 do
            begin
                FGen[i].Free;
            end;
        finally
            inherited Destroy;
        end;
    end;
end;

//data accessing
function TFitness2.getGen:TGenArr;
var i:integer;
begin
    SetLength(result,FNgen);
    for i:=0 to FNgen-1 do
        begin
            result[i]:=TPembangkit.Create(FGen[i]);
        end;
    end;
end;

procedure TFitness2.setGen(const rGen:TGenArr);
var i:integer;
begin
    FNgen:=high(rGen);
    SetLength(FGen,FNgen);
    for i:=0 to FNgen-1 do
        begin
            FGen[i]:=TPembangkit.Create(rGen[i]);
        end;
    end;
end;

//data processing
function TFitness2.getBatasGen:TBatasArr1;
var i:integer;
begin
    SetLength(result,FNgen);
    for i:=0 to FNgen-1 do

```



```

begin
  result[i].min:=FGen[i].Pmin;
  result[i].max:=FGen[i].Pmax;
end;
end;

```

```

function TFitness2.getRandomGen:dArr1;
var i,pos:integer;
    sumGen:double;
begin
  SetLength(result,FNgen);
  for i:=0 to FNgen-1 do
  begin
    result[i]:=FGen[i].Pmin+random*(FGen[i].Pmax-FGen[i].Pmin);
  end;
  pos:=round(1+random*(FNgen-1));
  pos:=pos-1;
  sumGen:=0;
  for i:=0 to FNgen-1 do
  begin
    if i<>pos then
    begin
      sumGen:=sumGen+result[i];
    end;
  end;
  result[pos]:=FLoad-sumGen;
  if result[pos]>FGen[pos].Pmax then result[pos]:=FGen[pos].Pmax;
  if result[pos]<FGen[pos].Pmin then result[pos]:=FGen[pos].Pmin;
end;

```

```

function TFitness2.doCalcCost(var rChrom:dArr1):double;
var i,pos:integer;
    sumGen,pinalty:double;
begin
  pos:=round(1+random*(FNgen-1));
  pos:=pos-1;
  sumGen:=0;
  for i:=0 to FNgen-1 do
  begin
    if i<>pos then
    begin
      sumGen:=sumGen+rChrom[i];
    end;
  end;
  rChrom[pos]:=FLoad-sumGen;
  pinalty:=0;
  if rChrom[pos]>FGen[pos].Pmax then
  begin

```

```

    pinalty:=rChrom[pos]-FGen[pos].Pmax;
    rChrom[pos]:=FGen[pos].Pmax;
end;
if rChrom[pos]<FGen[pos].Pmin then
begin
    pinalty:=FGen[pos].Pmin-rChrom[pos];
    rChrom[pos]:=FGen[pos].Pmin;
end;
result:=0;
for i:=0 to FNgen-1 do
begin
    result:=result+FGen[i].GetBiaya(rChrom[i]);
end;
result:=result+FPinGen*pinalty;
end;

```

```

function TFitness2.doRecombination(var rChrom1,rChrom2:dArr1):dArr1;
var i,pos:integer;
    sumGen:double;
begin
    SetLength(result,FNgen);
    for i:=0 to FNgen-1 do
begin
    result[i]:=rChrom1[i]+random*(rChrom2[i]-rChrom1[i]);
    if result[i]>FGen[i].Pmax then result[i]:=FGen[i].Pmax;
    if result[i]<FGen[i].Pmin then result[i]:=FGen[i].Pmin;
end;
pos:=round(1+random*(FNgen-1));
pos:=pos-1;
sumGen:=0;
for i:=0 to FNgen-1 do
begin
    if i<>pos then
begin
    sumGen:=sumGen+result[i];
end;
end;
result[pos]:=FLoad-sumGen;
if result[pos]>FGen[pos].Pmax then result[pos]:=FGen[pos].Pmax;
if result[pos]<FGen[pos].Pmin then result[pos]:=FGen[pos].Pmin;
end;

```

```

function TFitness2.doMutasi(const rNo:integer;
    const rPgen,rCk,rCmin,rBeta:double;
    var rRand:TRandomu):double;
var tho,dPgen:double;
begin
    tho:=rCk/rCmin*(FGen[rNo].Pmax-FGen[rNo].Pmin)*rBeta;

```

```
dPgen:=rRand.NextGaussian(0,tho);  
result:=rPgen+dPgen;  
if result>FGen[rNo].Pmax then result:=FGen[rNo].Pmax;  
if result<FGen[rNo].Pmin then result:=FGen[rNo].Pmin;  
end;  
  
end.
```

**unit uHasil;**

**interface**

**uses**

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

**type**

TfrmHasil = class(TForm)  
  TabSheet5: TTabSheet;  
  TabSheet6: TTabSheet;  
  TabSheet7: TTabSheet;  
  Panel1: TPanel;  
  btnClose: TButton;  
  btnHitungEP: TButton;  
  TabSheet4: TPageControl;  
  pbIterasi: TProgressBar;  
  GroupBox1: TGroupBox;  
  Label2: TLabel;  
  Label3: TLabel;  
  Label4: TLabel;  
  Label5: TLabel;  
  Label6: TLabel;  
  Label11: TLabel;  
  edtMaxGen: TEdit;  
  edtPopSize: TEdit;  
  edtLength: TEdit;  
  edtBetha: TEdit;  
  edtKa: TEdit;  
  btnUseDefault: TButton;  
  GroupBox2: TGroupBox;  
  cmbJam: TComboBox;  
  fgDaya: TStringGrid;  
  Chart2: TChart;  
  Series3: TLineSeries;  
  Series4: TLineSeries;  
  Series5: TLineSeries;  
  GroupBox3: TGroupBox;  
  Label1: TLabel;  
  edtPinGen: TEdit;  
  Label8: TLabel;  
  Label9: TLabel;  
  edtCostEvo: TEdit;  
  edtCostPLN: TEdit;  
  Label17: TLabel;  
  edtSelisih: TEdit;

```

    procedure btnCloseClick(Sender: TObject);
    procedure btnUseDefaultClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure btnHitungEPClick(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    frmHasil: TfrmHasil;

implementation

uses uObjFunc, uUtils, uEvoPro, uGenerator, uFitness2;

{$R *.dfm}

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

procedure TfrmHasil.btnUseDefaultClick(Sender: TObject);
begin
    edtMaxGen.Text:='500';
    edtPopSize.Text:='100';
    edtBetha.Text:='1.5';
    edtKa.Text:='1000000000';
    edtPinGen.Text:='1000000';
end;

procedure TfrmHasil.FormCreate(Sender: TObject);
begin
    fgDaya.Cells[0,0]:='No';
    fgDaya.Cells[1,0]:='P EPdanES (MW)';
    fgDaya.Cells[2,0]:='P PLN (MW)';
    fgDaya.Cells[3,0]:='Cost EPdanES (Rp)';
    fgDaya.Cells[4,0]:='Cost PLN (Rp)';
    fgDaya.Cells[5,0]:='Selisih (Rp)';
end;

procedure TfrmHasil.btnHitungEPClick(Sender: TObject);
var evo:TEvoPro1;
    i,sa,jam,MaxGen,PopSize,Length:integer;
    Load,Betha,Ka:double;
    sumEvo,sumPLN,pinGen:double;

```

```

    PLN:dArr2;
    LoadAsli,BestChrom,Min,Avg,Max:dArr1;
    Gen,GenAsli:TGenArr;
    BatasGen:TBatasArr1;
begin
    PLN:=gObjFunc.PLN;
    LoadAsli:=gObjFunc.Beban;
    GenAsli:=gObjFunc.Gen;
    jam:=StrToInt(cmbJam.Text);
    sa:=0;
    for i:=1 to high(PLN) do
    begin
        if PLN[i,jam]>0 then
        begin
            inc(sa);
        end;
    end;
    SetLength(Gen,sa);
    sa:=0;
    for i:=1 to gObjFunc.Ngen do
    begin
        if PLN[i,jam]>0 then
        begin
            Gen[sa]:=TPembangkit.Create(GenAsli[i]);
            inc(sa);
        end;
    end;
    Length:=sa;
    edtLength.Text:=IntToStr(Length);
    Load:=LoadAsli[jam];
    PinGen:=StrToFloat(edtPinGen.Text);
    gFitness2:=TFitness2.Create(PinGen,Gen,Load);
    BatasGen:=gFitness2.BatasGen;
    for i:=1 to high(GenAsli) do
    begin
        GenAsli[i].Free;
    end;
    MaxGen:=StrToInt(edtMaxGen.Text);
    pbIterasi.Max:=MaxGen;
    PopSize:=StrToInt(edtPopSize.Text);
    Betha:=StrToFloat(edtBetha.Text);
    Ka:=StrToFloat(edtKa.Text);
    evo:=TEvoPro1.Create(MaxGen,PopSize,Length,Ka,Betha,BatasGen);
    BestChrom:=evo.BestChrom;
    Min:=evo.Min;
    Avg:=evo.Avg;
    Max:=evo.Max;
    Series3.Clear;

```

```

Series4.Clear;
Series5.Clear;
for i:=0 to high(Min) do
begin
  Series3.Add(Min[i],IntToStr(i+1));
  Series4.Add(Avg[i],IntToStr(i+1));
  Series5.Add(Max[i],IntToStr(i+1));
end;
evo.Free;
fgDaya.RowCount:=high(BestChrom)+2;
sa:=0;
sumEvo:=0;
sumPLN:=0;
for i:=1 to gObjFunc.Ngen do
begin
  if PLN[i,jam]<>0 then
  begin
    fgDaya.Cells[0,sa+1]:=IntToStr(sa+1);
    fgDaya.Cells[1,sa+1]:=FormatFloat('#,##0',BestChrom[sa]);
    fgDaya.Cells[2,sa+1]:=FormatFloat('#,##0',PLN[i,jam]);

fgDaya.Cells[3,sa+1]:=FormatFloat('#,##0',Gen[sa].GetBiaya(BestChrom[sa]));
    fgDaya.Cells[4,sa+1]:=FormatFloat('#,##0',Gen[sa].GetBiaya(PLN[i,jam]));
    fgDaya.Cells[5,sa+1]:=FormatFloat('#,##0',Gen[sa].GetBiaya(PLN[i,jam])-
      Gen[sa].GetBiaya(BestChrom[sa]));
    sumEvo:=sumEvo+Gen[sa].GetBiaya(BestChrom[sa]);
    sumPLN:=sumPLN+Gen[sa].GetBiaya(PLN[i,jam]);
    inc(sa);
  end;
end;
edtCostEvo.Text:=FormatFloat('#,##0',sumEvo);
edtCostPLN.Text:=FormatFloat('#,##0',sumPLN);
edtSelisih.Text:=FormatFloat('#,##0',sumPLN-sumEvo);
for i:=0 to high(Gen) do
begin
  Gen[i].Free;
end;
gFitness2.Free;
end;

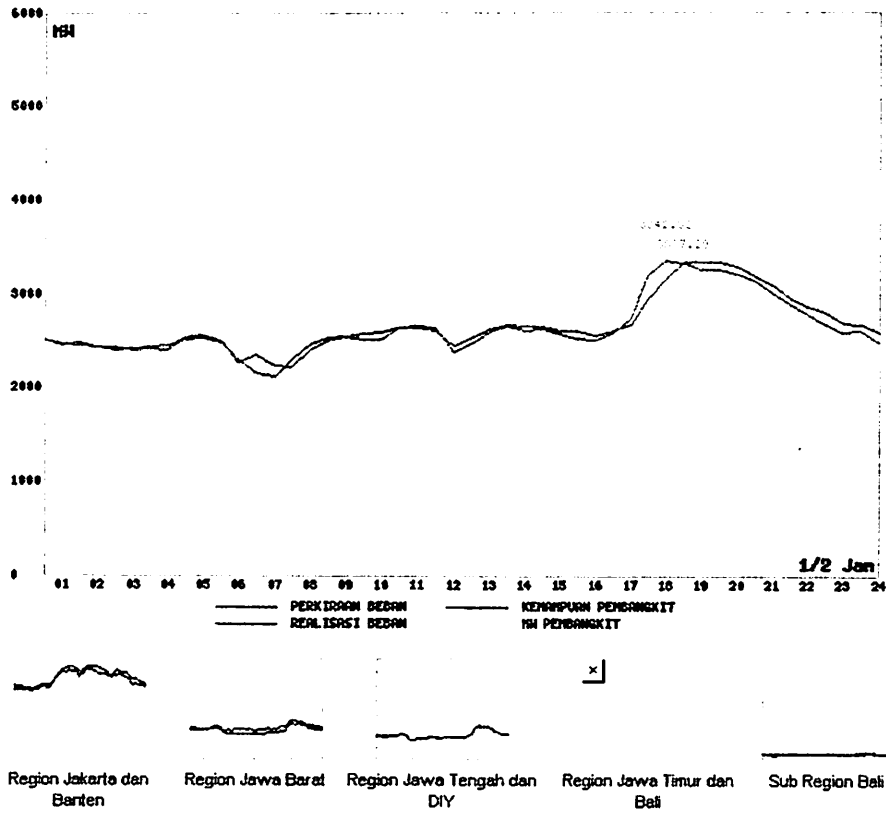
end.

```

• **Data Pembebanan hari Rabu, 29 Maret 2006**

Kondisi Sistem: Rabu, 29 Maret 2006  
 Region Jawa Timur dan Bali

Grafik Hari Lainnya < >

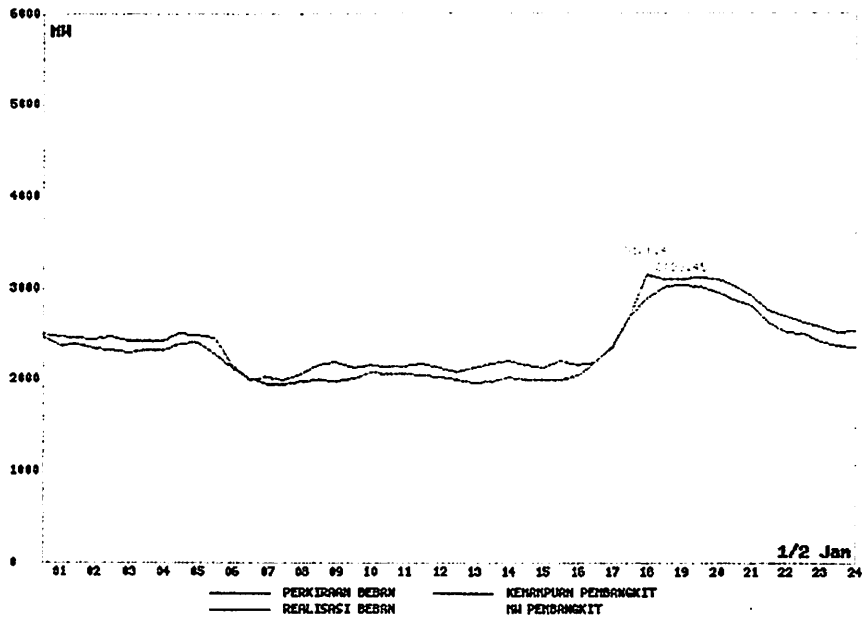




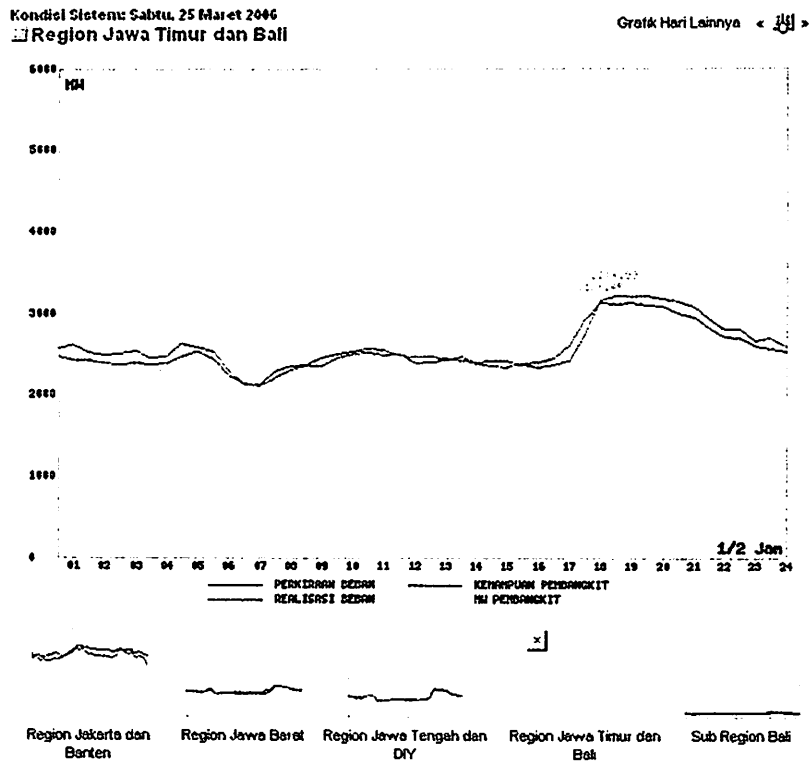
• **Data Pembebanan Minggu, 26 Maret 2006**

Kondisi Sistem: Minggu, 26 Maret 2006  
 Region Jawa Timur dan Bali

Grafik Hari Lainnya < >



- **Data Pembebanan Hari Sabtu, 25 Maret 2006**



Data Pembebanan unit Pembangkit pada tanggal 25, 26, dan 29, Maret 2006

Sumber : "WWW.Ubos.PLN-JAWA-BALI.COM

Waktu	Data Pembebanan		
	25-03-2006	26-03-2006	29-03-2006
1	2605	2459	2439
2	2465	2427	2429
3	2519	2401	2397
4	2464	2409	2397
5	2570	2465	2547
6	2293	2168	2254
7	2119	2024	2237
8	2336	2045	2396
9	2433	2181	2529
10	2528	2149	2513
11	2544	2132	2653
12	2449	2118	2380
13	2436	2114	2588
14	2395	2201	2608
15	2409	2117	2577
16	2331	2149	2507
17	2412	2325	2735
18	3157	3130	3342
19	3202	3091	3254
20	3175	3086	3205
21	3073	2912	3012
22	2803	2678	2777
23	2650	2573	2587
24	2583	2527	2478