

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



**APLIKASI METODE *EVOLUTIONARY PROGRAMMING*
PADA PEMBAGIAN PEMBEBANAN
PEMBANGKIT THERMAL
DI PT. PLN PEMBANGKIT JAWA BALI**

SKRIPSI

Disusun Oleh:

MOHAMMAD ROYANI

NIM: 99. 12. 012

MARET 2006

MILIK
PERPUSTAKAAN
ITN MALANG

SECRET JOURNAL DOCUMENT TITLES
INTERNATIONAL POLITICAL SCIENCE
1-2 OCTOBER 1961
MAY 1962

INTERNATIONAL POLITICAL SCIENCE
MAY 1962
MAY 1962
MAY 1962

SECRET

SECRET
SECRET
SECRET

SECRET

LEMBAR PERSETUJUAN

APLIKASI METODE *EVOLUTIONARY PROGRAMMING* PADA PEMBAGIAN PEMBEBANAN PEMBANGKIT THERMAL DI PT. PLN PEMBANGKIT JAWA BALI

SKRIPSI

*Disusun Untuk Melengkapi dan Memenuhi Syarat-Syarat
Guna Mencapai Gelar Sarjana Teknik Elektro S-1*

Disusun Oleh :

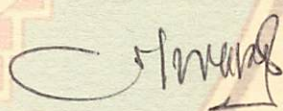
MOHAMMAD ROYANI
NIM : 99. 12. 012

Mengetahui,
Ketua Jurusan Teknik Elektro S-1



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

**Diperiksa dan Disetujui
Dosen Pembimbing**

 9/09 '06

Ir. I Made Wartana, MT
NIP. 131 991 182

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



LEMBAR PERSEMBAHAN

يَتَأْتِيهَا الَّذِينَ ءَامَنُوا إِذَا قِيلَ لَكُمْ تَفَسَّحُوا فِي الْمَجَالِسِ فَافْسَحُوا
بِفَسْحِ اللَّهِ لَكُمْ وَإِذَا قِيلَ أَنْشُرُوا فَأَنْشُرُوا يَرْفَعُ اللَّهُ الَّذِينَ ءَامَنُوا
مِنْكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ وَاللَّهُ بِمَا تَعْمَلُونَ خَبِيرٌ

"Hai orang-orang beriman apabila kamu dikatakan kepadamu: "Berlapang-lapanglah dalam majlis", maka lapangkanlah niscaya Allah akan memberi kelapangan untukmu. Dan apabila dikatakan: "Berdirilah kamu", maka berdirilah, niscaya Allah akan meninggikan orang-orang yang beriman di antaramu dan orang-orang yang diberi ilmu pengetahuan beberapa derajat. Dan Allah Maha Mengetahui apa yang kamu kerjakan" (QS. Al Mujaadilah : 58)

ALLAH SWT

Penguasa Alam Semesta dan Segala Isinya

Dengan Nama ALLAH Yang Maha Pengasih Lagi Maha Penyayang

Yang telah menjadikanku dari segumpal darah dan dari setetes air yang hina.

Yang telah meniupkan roh kedalam ragaku, yang telah mengizinkan aku menghirup udara-Mu, yang telah memberiku kesempatan berdiri di atas bumi-Mu hingga kelak aku kembali menghadap-Mu di hari akhir nanti.

Atas segala berkah, rahmat, hidayah dan karunia yang telah di anugerahkan kepadaku, sembah sujud di hadapan-Mu.

"Tuhanku, Pelihara Imanku. Agar Tetap Pada Jalan-Mu"

"Tuhanku, Aku Milik-Mu"

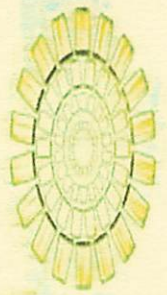
NABI MUHAMMAD SAW

Sebaik-baiknya teladan dan semulia-mulianya akhlak didunia.

Atas segala ajaran dan petunjuk yang telah disampaikan, semoga saja kelak aku dapat menjadi muslim yang sempurna dan dapat berkumpul bersama di dalam surga-Nya.

"Amin, Amin, Amin Ya Rabbal Alamin"





رَبَّنَا اغْفِرْ لِي وَلِوَالِدَيَّ وَلِلْمُؤْمِنِينَ يَوْمَ يَقُومُ الْحِسَابُ

"Ya Tuhan kami, beri ampunlah aku dan kedua ibu bapaku dan sekalian orang-orang mukmin pada hari terjadinya hisab (hari kiamat)".

(Surat Ibrahim : 41)

Ayahanda ANANG MARWANI dan Ibunda RUSMINI, SPd.

Atas nama cinta, kasih dan sayang yang selalu tercurah, cucuran keringat dan air mata yang senantiasa selalu mengalir, kalimat do'a yang terpanjatkan tiada henti, nasehat dan petuah yang terucap santun, restu dan ampunan yang tiada henti di berikan, kesabaran dan pengertian yang tiada batas. Atas segalanya itu, tak ada satupun kata dan kalimat yang pantas terucapkan untuk menyatakannya. Hanya sembah sujud dan bakti Ananda smoga Ayahanda dan Ibunda dapat berkenan menerimanya selalu. Sungguh, tak akan pernah Ananda sia-siakan sgala pengorbanan yang telah diberikan.

"Restu Kalian Damaikan Hidupku, Kasih Sayang Kalian Damaikan Jiwaku"

***"Semoga Allah SWT Menjadikan Ananda Anak Yang Shaleh,
Berbakti Kepada Ibu Bapak Dan Menjauhkan Ananda Dari Sikap Durhaka.***

Ayahanda H. HANAFI AS dan Ibunda Dra. RUHAYATI

Atas segala do'a, kesabaran dan cinta kasih yang telah diberikan, hingga pada akhirnya Ananda dapat menyelesaikan studi ini, maka tak lupa pula Ananda menghaturkan sembah sujud kepada Ayahanda dan Ibunda.

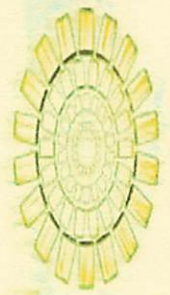
Semoga Ayah dan Ibu dapat berkenan menerimanya.

Atas segala kesempatan, pengertian dan kebijaksanaan yang telah diberikan, Sungguh tidaklah Ananda akan menyia-nyiakan semua itu.

Kehidupan terus berjalan, dan tanggung jawab wajib untuk dilaksanakan. Semoga ALLAH SWT Memberikan Kekuatan Untuk Melaksanakannya. Amin.

"Mohon Restu Untuk Kehidupan Selanjutnya"





NOOR BAIDLA ROYANI, ST

***“Masing-Masing Dari Kita Adalah Malaikat Bersayap Satu.
Dan Hanya Bisa Terbang Jika Saling Berpelukan”***

Sujud syukur kepada Allah SWT, yang telah menciptakan dirimu di muka bumi ini.

Yang telah mempertemukan kita dalam kasih dan sayang.

Yang telah menyatukan kita dalam bahtera cinta suci,

Dalam pertautan bathin yang syahdu.

Cintamu kuatkan aku menjalani hidup, sayangmu menyabarkan hatiku, dengan kasihmu kembali perlahan luruskan jalanku, senyummu adalah sumber inspirasiku, manjamu selalu redakan emosiku, kerling matamu luluhkan kerasnya egoku, tawamu selalu saja hilangkan gundahku, dan pengertianmu adalah segalanya bagiku.

Terima kasih udah mau bersabar menungguku pulang kembali dalam pelukanmu.

Usai sudah malam-malam sepi penantian panjangmu.

Usai sudah kesendirianku dalam kerinduan yang mendalam.

Takkan pernah aku pergi jauh lagi dari sisimu.

Sebab Kamu Adalah :

“Separuh Nafasku, Sepenggal Jiwaku”.

Karena Dirimu Adalah :

Anugrah Terindah Yang Pernah Kumiliki.





AKHMAD RIZANI (Alm)

Maaf atas segala salah dan khilaf yang pernah terjadi. Bukankah merupakan suatu kepastian bahwa segala yang hidup pada akhirnya pasti kembali padaNya.

Damailah Selalu Dalam Surga ALLAH SWT

Senyummu Terpatri Didalam Hati Nuraniku Yang Terdalam

Hadirmu Selalu Ada Dalam Setiap Langkahku

"I Love U My Little Brother"

FATIMAH, HAMKA dan FARID

Atas kebersamaan, pengertian, bantuan dan segala-galanya yang telah kalian berikan.

Maka tak lah lupa seuntai kalimat terima kasih yang mendalam
dipersembahkan untuk kalian.

Semoga saja kebersamaan ini Abadi adanya.

"Mari Saling Bergandeng Tangan Untuk Kehidupan Yang Lebih Baik"

SELURUH SANAK SAUDARA, KERABAT dan HANDAI TOLAN

Yang tak bisa disebutkan satu persatu. Atas dukungan moril dan motivasinya selama ini, maka izinkanlah Roy dengan segala kerendahan hati mengucapkan terima kasih yang mendalam kepada kalian semua.

Semoga ALLAH SWT memberikan balasan pahala yang setimpal. Amin.



ROY SPECIAL THANK'S TO:



Bapak Ir. I Made Wartana, MT

Atas segala bimbingan, petunjuk dan arahan yang Bapak berikan.....jujur saya ga tau mesti mengucapkan apa lagi selain beritua terima kasih. Semoga apa yang telah Bapak ajarkan dapat berguna di kehidupan saya selanjutnya. Maaf atas segala prasangka buruk saya selama ini. Semoga Tuhan YME selalu memberikan kesejahteraan kepada Bapak dan Keluarga.

Ibu Ida Barifa, SH (dosen waliku)

7 tahun lamanya kebersamaan kita, sekalipun saya ga pernah lihat Ibu marah pada kami, walaupun kami sering merepotkan Ibu. Atas segala kelembutan, bimbingan, pengertian dan senyum manis Ibu, ananda hanya bisa ber do'a smoga saja Ibu dan Keluarga selalu dalam lindungan Tuhan YME. Amin. Bagi saya Ibu bukan hanya sekedar Dosen Wali. Miss u mom.....

Bapak Ir. Almizan Abdullah, MSEE (Sang Profesor)

Bagi saya Anda bukan hanya seorang Pengajar, melainkan juga seorang Pendidik. Salut untuk kedisiplinan dan fropesionalitas Bapak. Semoga ALLAH SWT selalu memberikan rahmat dan hidayahNya kepada Bapak dan Keluarga. Amin.

Bapak Ir. H. Taufik Hidayat, MT

Atas segala pengajaran dan pendidikan yang telah diberikan dengan sepenuh cinta kasih, sabar dan welas asih, hanya do'a yang dapat saya persembahkan kepada bapak dan keluarga. Semoga Allah SWT selalu melimpahkan rahmat dan hidayahNya kepada Bapak dan Keluarga. Amin.

Mas Jayeng dan Ibu Puji

Atas segala bantuan yang pernah diberikan sejak pertama kuliah hingga terselesainya studi ini. Makasih banget lho.....

Bapak Rudy Resnawan (Walikota Banjarbaru)

Atas dedikasi Anda terhadap dunia pendidikan, atas perhatian anda terhadap mahasiswa mahasiswi asal kota Banjarbaru di Malang hingga dapat terealisasinya Asrama Mahasiswa Banjarbaru IDAMAN di Kota Malang saya ucapkan banyak terima kasih.

Seluruh Dosen T. Elektro

Atas segala pelajaran dan ilmu yang telah diberikan hingga saya dapat mengerti banyak tentang berbagai hal di dunia elektro. Kalian Pahlawan Tanpa Tanda Jasa.

Sahabat - sahabat tercinta di Asrama Mahasiswa Banjarbaru IDAMAN >> Denny BrookenFoot, Amd (jgn terlalu banyak nge-net, sasadangnya aja), Rudy Idur Udhayak, SPsi n Ika (Buzz aza), Yuliadie (alm), Arie Lesung, ST., Didi Carbone, Indra Julak, SKom n Maulida. A'at, Ardian (akhirnya kita lulus jua), Badak, calon SH., dan kalian2 The AMBARU Crew >> Smoga kebersamaan ini kan slalu ABADI slamanya.

DA 3625 CK

SiKuning yang selalu setia menemani dalam setiap langkah perjuanganku, yang telah nganter aku kemana-mana walau kadang sering ga ngerawatmu dan meski terkadang aku tak slalu bersamamu.

Eka Cemplu

Atas bantuan fasilitas** nya.thanks banget lho, juga anak** J 281 atas kebersamaannya. Kamu emang baik banget.

Nurtanto, ST

Atas segala bantuan, bimbingan, petunjuk dan dorongan motivasinya. Semoga ALLAH SWT memberikan balasan yang setimpal.Amin. dan cepat dapat kerja ya.

Rekan-rekan T.Elektro 99

Yang udah sarjana smoga cepat dpt kerja, yg udah kerja ajak temannya yg belum.Yang belum sarjana, cepetan lulus sebab waktu tak pernah menunggu.

Anak Buah Pak Made Yang Tersisa

Aris Ayu Jakarta 99, Andi Mojokerto 00, Endi Gundul Ngawi 97 Doraemon, Andi Bojonegoro 97 yang GA MESTI, Ani Sulawesi 00, Chandra 96, Widi Kediri 01 n Erwin Balikpapan 99. Kebersamaan, perjuangan dan pengorbanan kita semoga saja tidak sia-sia adanya. Thanks atas diskusi, sharing, guyon dan canda tawa kita slama bimbingan, semuanya sangat berkesan bagiku. Juga atas sgala bantuan dan kebersamaan kalian slama ini. Akhirnya kita lulus juga. Tapi perjuangan belum berakhir teman. Ini bukan akhir, namun awal bagi kehidupan kita selanjutnya.

The Police

Rekan-rekan Kesatuan Polresta Malang dan Polsek Lowokwaru (Patroli 07). #Mas Bambang, Mas Josafath, Pak Las, Pak Bejo, Mas Rambo dll nya, # Om Luki sekeluarga yang telah banyak bantu saya slama di Malang. Trima kasih atas kebersamaan slama ini, meski saya sering merepotkan kalian.

KalSel Community

Bekan-rekan Asrama MTP, Memed (thanks banget tuk pinjaman printer nya), AMKS Mandastana, AMKS PangHit (Lebak), Asrama BJM, KDG, KTB, PLH, dll yg terlalu banyak jika di sebutkan satu-satu.

Tempat Congkrukan

Wartel Wahyu n Crew # Wartel El-Prima n crew # Dory Café # Dream Café # Komunitas Warung Lamongan yg ga bisa disebutkan satu-satu karena terlalu banyak. # FotoCopy S.Ria . # Alex kunci.

Besti JKT

Thanks banget untuk pembuatan dan modifikasi programnya yang ok punya. Smoga segala sesuatunya lancar-lancar aja n sukses 4 you.

Malang Raya

Atas segala mimpi dan kenangan yang pernah tercipta. Tunggulah aku kembali dengan bertepuk dada dan dengan kepala tegak.....!!!

Dunia Maya

Atas segala kesempatan pengembaraan liarku menjelajah ruang tanpa batas waktu.

The Singer

Bryan Adam, Bob Marley, Rastafara, Radja, Kla Project, Gigi, Pance F Pondag serta para musisi Tembang Kenangan dan juga tak lupa kepada seluruh para musisi yang lagunya tersimpan didalam MP3 ku. Atas suara merdu kalian yang tak pernah lelah menyanyi menemani menyelesaikan skripsi ini. Thanks banget gitu loooooh.

My Motto

" **Hidup Adalah Perjudian** " >> pijakan dan landasanku dalam berpikir dan memotivasiku untuk slalu berusaha menjadi yang terbaik walau kadang ga slalu.

"OBSESIKU, BERI AKU WAKTU"

ABSTRAKSI

APLIKASI METODE *EVOLUTIONARY PROGRAMMING* PADA PEMBAGIAN PEMBEBANAN PEMBANGKIT THERMAL DI PT. PLN PEMBANGKIT JAWA BALI

(Mohammad Royani, NIM: 99.12. 012. Jurusan Teknik Elektro S-1. Konsentrasi Teknik Energi Listrik. Dosen Pembimbing: Ir. I Made Wartana, MT.)

Kata Kunci:

Evolutionary Programming, Pembebanan Ekonomis.

Pembangkitan tenaga listrik merupakan bagian dari permasalahan energi dan lingkungan yang dihadapi oleh Indonesia sebagai negara berkembang. Untuk suatu operasi pada beban tertentu, perhitungan ekonomis harus tetap merupakan suatu prioritas, sehingga nantinya diperlukan suatu rencana operasi yang optimum dengan tetap memenuhi beberapa persyaratan pengoperasian sistem tenaga listrik.

Dalam pembangkitan tenaga listrik dilakukan usaha agar biaya pembangkitannya semurah mungkin. Usaha untuk mengoptimalkan biaya operasi ini, salah satunya dilakukan dengan penerapan pembagian pembebanan menggunakan metode *Evolutionary Programming*.

Skripsi ini membahas masalah optimasi biaya bahan bakar pada suatu sistem tenaga listrik dengan menggunakan metode *Evolutionary Programming*. Hasil dari analisa tersebut nantinya dapat digunakan sebagai salah satu acuan dalam mengefisienkan biaya bahan bakar yang bisa dimanfaatkan oleh PT. PLN (Persero) sehingga nantinya nilai kerugian yang sementara ini ditanggung oleh PLN bisa berkurang.

Analisa dilakukan dengan bantuan program komputer menggunakan bahasa pemrograman *borland delphi versi 7.0* dan telah berhasil dicoba pada PT.PLN Pembangkit Jawa Bali yang terdiri dari 38 unit pembangkit thermal 19 yang siap operasi.

Dengan menggunakan *Evolutionary Programming* pada pembagian pembebanan di PT. PLN Pembangkit Jawa Bali maka didapatkan optimasi biaya bahan bakar yaitu pada hari Rabu, tanggal 10 Maret 2004 sebesar Rp. 1.517.574.707 (dapat dioptimasi sekitar 8,71 %). Pada hari Sabtu, tanggal 13 Maret 2004 sebesar Rp. 1.165.057.696 (dapat dioptimasi sekitar 7,41 %). Pada hari Minggu, tanggal 14 Maret 2004 sebesar Rp. 729.729.983 (dapat dioptimasi sekitar 4,96 %).

KATA PENGANTAR

Puji Syukur Kehadirat ALLAH SWT atas segala limpahan Rahmat dan Hidayah-Nya yang telah diberikan sehingga penyusunan skripsi ini dapat terselesaikan dengan baik. Karena hanya atas Kehendak-Nya semata penulis mampu dan sanggup bertahan melewati segala macam cobaan dan kendala yang ada selama penyusunan skripsi ini.

Skripsi berjudul “Aplikasi Metode *Evolutionary Programming* Pada Pembagian Pembebanan Pembangkit Thermal Di PT. PLN Pembangkit Jawa Bali” ini disusun sebagai salah satu syarat untuk mendapatkan gelar Sarjana Teknik di Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Nasional Malang.

Selama penyusunan skripsi ini, tidak terlepas dari bantuan, bimbingan dan motivasi dari berbagai pihak. Oleh karena itu atas segala yang pernah diberikan, tak lupa penulis menyampaikan rasa hormat dan terima kasih yang mendalam kepada :

1. Bapak Ir. Mochtar Asroni, MSME, Selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang.
2. Bapak Ir. F. Yudi Limpraptono, MT , Selaku Ketua Jurusan Teknik Elektro S-1 Institut Teknologi Nasional Malang.
3. Ibu Ir. Mimin Mustikawati, Selaku Sekretaris Jurusan Teknik Elektro Institut Teknologi Nasional Malang.
4. Bapak Ir. I Made Wartana, MT , Selaku Dosen Pembimbing.

5. Bapak Ir. H. Taufik Hidayat, MT , Selaku Dosen Penguji.
6. Bapak Ir. H. Almizan Abdullah, MSEE , Selaku Dosen Penguji.
7. Rekan-rekan Teknik Elektro ITN Malang yang telah banyak membantu dalam penyelesaian skripsi ini.

Penulis menyadari bahwa masih terdapat berbagai kekurangan dalam penyusunan skripsi ini. Oleh karenanya, segala saran dan kritik yang sifatnya membangun sangat penulis harapkan. Semoga skripsi ini kelak dapat berguna dan bermanfaat bagi pembaca.

Malang, Maret 2006

Penulis

DAFTAR ISI

LEMBAR PERSETUJUAN	i
ABSTRAKSI.....	ii
KATA PENGANTAR.....	iii
DAFTAR ISI.....	v
DAFTAR GAMBAR	ix
DAFTAR TABEL.....	x
DAFTAR GRAFIK.....	xiii

BAB I PENDAHULUAN

1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan	3
1.4. Batasan Masalah.....	3
1.5. Metodologi Penelitian	4
1.6. Sistematika Penulisan	4
1.7. Relevansi	6

BAB II KAJIAN PUSTAKA

2.1. Karakteristik Pembangkit.....	7
2.1.1. Karakteristik <i>Input - Output</i>	7
2.1.2. Karakteristik <i>Heat Rate</i>	10

- 2.1.3. Karakteristik *Incremental Heat Rate* dan
Incremental Fuel Cost.....11
- 2.2. *Economic Dispatch*12
- 2.2.1. Penyelesaian *Economic Dispatch* Dengan
Metode Pengali *La Grange*13
- 2.2.2. Penyelesaian *Economic Dispatch* Dengan Iterasi Lamda.....16
- 2.3. Fungsi Biaya Bahan Bakar.....17
- 2.4. *Economic Dispatch* Mengabaikan Rugi-Rugi Transmisi.....17

BAB III APLIKASI METODE *EVOLUTIONARY PROGRAMMING* UNTUK *ECONOMIC DISPATCH*

- 3.1. Teori Dasar *Evolutionary Programming*21
- 3.1.1. Konsep Dasar21
- 3.1.2. Parameter *Evolutionary Programming*23
- 3.1.3. Mekanisme *Evolutionary Programming*.....24
- 3.2. Formulasi Masalah *Economic Dispatch*28
- 3.3. Aplikasi Metode *Evolutionary Programming*
Untuk Masalah *Economic Dispatch*.....29
- 3.3.1. Inisialisasi.....29
- 3.3.2. Fungsi Tujuan (*Objective Function*).....30
- 3.3.3. Fungsi *Fitness*30
- 3.3.4. Pembangkitan *Offspring* Dengan Mutasi.....31
- 3.3.5. Seleksi Dan Kompetisi.....32
- 3.4. Algoritma Program.....33

3.4.1. Algoritma Penyelesaian <i>Economic Dispatch</i> Menggunakan Metode <i>Evolutionary Programming</i>	33
3.4.2. Algoritma <i>Subroutine</i> Fungsi <i>Fitness</i> Untuk Masalah <i>Economic Dispatch</i> Dengan Metode <i>Evolutionary Programming</i>	34
3.4.3. Flowchart <i>Economic Dispatch</i> Menggunakan Metode <i>Evolutionary Programming</i>	35
3.4.4. Flowchart <i>Subroutine</i> Fungsi <i>Fitness</i> Untuk Masalah <i>Economic Dispatch</i> Dengan Metode <i>Evolutionary Programming</i>	36

BAB IV HASIL DAN ANALISIS HASIL

4.1. Pendahuluan	37
4.2. Program Komputer <i>Economic Dispatch</i> Dengan Metode <i>Evolutionary Programming</i>	37
4.3. Data Pembangkit Thermal.....	38
4.4. Aplikasi Metode <i>Evolutionary Programming</i> Di PT. PLN Pembangkit Jawa Bali.....	40
4.5. Beban Sistem.....	41
4.6. Hasil Perhitungan Dan Analisa Hasil Perhitungan <i>Economic Dispatch</i> Menggunakan Metode <i>Evolutionary Programming</i>	45
4.6.1. Hasil Perhitungan PT. PLN Pembangkit Jawa Bali	45

4.6.2. Hasil Perhitungan Dengan Metode
Evolutionary Programming49

4.6.3. Hasil Perhitungan PT. PLN Pembangkit Jawa Bali
Dengan Metode *Evolutionary Programming*.....54

4.6.4. Perbandingan Hasil Perhitungan PT. PLN Pembangkit Jawa
Bali Dengan Metode *Evolutionary Programming*57

BAB V KESIMPULAN DAN SARAN

5.1. Kesimpulan72

5.2. Saran.....73

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar :

Halaman :

2-1.	Unit Boiler - Turbin – Generator	8
2-2.	Kurva Karakteristik <i>Input-Output</i> Pembangkit Thermal	9
2-3.	Kurva Karakteristik <i>Heat Rate</i> Unit Pembangkit.....	10
2-4.	Kurva Karakteristik <i>Incremental Fuel Cost Rate</i>	12
2-5.	N Unit Melayani Beban P_R	13
2-6.	Grafik Penyelesaian Dengan Metode Iterasi Lamda.....	16
2-7.	N Unit Pembangkit Thermal Melayani Beban P_R	18
3-1.	Ilustrasi Proses Mutasi	27
4-1.	Tampilan Program Utama	49
4-2.	Menampilkan Input Data.....	49
4-3.	Parameter <i>Evolutionary Programming</i>	50
4-4.	Tampilan Data Pembangkit.....	50
4-5.	Tampilan Data Pembebanan	51
4-6.	Tampilan Data PLN	51
4-7.	Hasil Optimasi Pada Jam 01.00	52
4-8.	Hasil Optimasi Pada Jam 18.00	52
4-9.	Hasil Optimasi Pada Jam 24.00	53

DAFTAR TABEL

Tabel :

Halaman :

4-1.	Data Unit Pembangkit Pada PT. PLN Pembangkit Jawa Bali	39
4-2.	Unit Pembangkit Thermal Yang Siap Operasi.....	40
4-3.	Data Pembebanan Unit Thermal Pada PT. PLN Pembangkit Jawa Bali, Rabu 10 Maret 2004	42
4-4.	Data Pembebanan Unit Thermal Pada PT. PLN Pembangkit Jawa Bali, Sabtu 13 Maret 2004	43
4-5.	Data Pembebanan Unit Thermal Pada PT. PLN Pembangkit Jawa Bali, Minggu 14 Maret 2004.....	44
4-6.	Hasil Perhitungan Biaya Operasional Per Jam PT. PLN Pembangkit Jawa Bali, Rabu 10 Maret 2004	46
4-7.	Hasil Perhitungan Biaya Operasional Per Jam PT. PLN Pembangkit Jawa Bali, Sabtu 13 Maret 2004	47
4-8.	Hasil Perhitungan Biaya Operasional Per Jam PT. PLN Pembangkit Jawa Bali, Minggu 14 Maret 2004.....	48
4-9.	Hasil Perhitungan Menggunakan Metode <i>Evolutionary Programming</i> , Rabu 10 Maret 2004	54
4-10.	Hasil Perhitungan Menggunakan Metode <i>Evolutionary Programming</i> , Sabtu 13 Maret 2004.....	55
4-11.	Hasil Perhitungan Menggunakan Metode <i>Evolutionary Programming</i> , Minggu 14 Maret 2004	56

4-12.	Perbandingan Biaya Operasional Per Jam PT. PLN Pembangkit Jawa Bali Dengan Metode <i>Evolutionary Programming</i> , Rabu 10 Maret 2004.....	57
4-13.	Perbandingan Biaya Operasional Per Jam PT. PLN Pembangkit Jawa Bali Dengan Metode <i>Evolutionary Programming</i> , Sabtu 13 Maret 2004	58
4-14.	Perbandingan Biaya Operasional Per Jam PT. PLN Pembangkit Jawa Bali Dengan Metode <i>Evolutionary Programming</i> , Minggu 14 Maret 2004	59
4-15.	Perbandingan Total Biaya Operasional PT. PLN Pembangkit Jawa Bali Dengan Metode <i>Evolutionary Programming</i>	62
4-16.	Pembebanan Unit Pembangkit Tiap Jam Pt. PLN Pembangkit Jawa Bali Sebelum Optimasi, Rabu 10 Maret 2004	63
4-17.	Pembebanan Unit Pembangkit Tiap Jam Pt. PLN Pembangkit Jawa Bali Sesudah Optimasi, Rabu 10 Maret 2004	63
4-18.	Pembebanan Unit Pembangkit Tiap Jam Pt. PLN Pembangkit Jawa Bali Sebelum Optimasi, Sabtu 13 Maret 2004.....	64
4-19.	Pembebanan Unit Pembangkit Tiap Jam Pt. PLN Pembangkit Jawa Bali Sesudah Optimasi, Sabtu 13 Maret 2004	64
4-20.	Pembebanan Unit Pembangkit Tiap Jam Pt. PLN Pembangkit Jawa Bali Sebelum Optimasi, Minggu 14 Maret 2004	65
4-21.	Pembebanan Unit Pembangkit Tiap Jam Pt. PLN Pembangkit Jawa Bali Sesudah Optimasi, Minggu 14 Maret 2004.....	65

4-22.	Data Penjadwalan Unit Pembangkit, Rabu 10 Maret 2004.....	69
4-23.	Data Penjadwalan Unit Pembangkit, Sabtu 13 Maret 2004.....	70
4-24.	Data Penjadwalan Unit Pembangkit, Minggu 14 Maret 2004	71

DAFTAR GRAFIK

Grafik :

Halaman :

4-1.	Perbandingan Biaya Operasional Per Jam PT. PLN Pembangkit Jawa Bali Dengan Metode <i>Evolutionary Programming</i> , Rabu 10 Maret 2004.....	60
4-2.	Perbandingan Biaya Operasional Per Jam PT. PLN Pembangkit Jawa Bali Dengan Metode <i>Evolutionary Programming</i> , Sabtu 13 Maret 2004	60
4-3.	Perbandingan Biaya Operasional Per Jam PT. PLN Pembangkit Jawa Bali Dengan Metode <i>Evolutionary Programming</i> , Minggu 14 Maret 2004	61
4-4.	Perbandingan Biaya Total Operasional PT. PLN Pembangkit Jawa Bali Dengan Metode <i>Evolutionary Programming</i>	62
4-5.	Pembebanan Unit Pembangkit PT. PLN Pembangkit Jawa Bali Sebelum Optimasi, Rabu 10 Maret 2004	66
4-6.	Pembebanan Unit Pembangkit PT. PLN Pembangkit Jawa Bali Sesudah Optimasi, Rabu 10 Maret 2004.....	66
4-7.	Pembebanan Unit Pembangkit PT. PLN Pembangkit Jawa Bali Sebelum Optimasi, Sabtu 13 Maret 2004	67
4-8.	Pembebanan Unit Pembangkit PT. PLN Pembangkit Jawa Bali Sesudah Optimasi, Sabtu 13 Maret 2004	67
4-9.	Pembebanan Unit Pembangkit PT. PLN Pembangkit Jawa Bali Sebelum Optimasi, Minggu 14 Maret 2004	68

4-10. Pembebanan Unit Pembangkit PT. PLN Pembangkit Jawa Bali

Sesudah Optimasi, Minggu 14 Maret 2004 68

BAB I

PENDAHULUAN

1.1. Latar Belakang

Pembangkitan tenaga listrik merupakan bagian dari permasalahan energi dan lingkungan yang dihadapi oleh Indonesia sebagai negara berkembang. Secara garis besar, suatu sistem tenaga listrik dibagi menjadi tiga bagian yaitu : sisi pembangkit tenaga listrik, jaringan transmisi dan 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, diantaranya daya yang dibangkitkan cukup untuk memasok beban dan rugi-rugi daya pada saluran transmisi.

Dalam pembangkitan tenaga listrik dilakukan usaha agar biaya pembangkitannya semurah mungkin. Usaha untuk mengoptimalkan biaya operasi ini , salah satunya dilakukan dengan penerapan *Economic Dispatch*. Dimana pengertian dari *Economic Dispatch* itu sendiri adalah pembagian pembebanan pada pembangkit-pembangkit yang ada dalam suatu sistem tenaga listrik secara optimum pada harga beban tertentu^[1]. Didalam operasi sistem tenaga listrik *Economic Dispatch* adalah hal yang sangat perlu diperhatikan agar bisa mendapatkan biaya bahan bakar yang 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.

Pada skripsi ini akan dibahas metode alternative masalah optimasi biaya pembangkitan dengan mengoptimalkan biaya operasional melalui penerapan *Economic Dispatch* menggunakan metode *Evolutionary Programming* yang merupakan pengembangan dari Algoritma Genetika.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, maka timbul permasalahan yaitu bagaimana mengoptimasi biaya pembangkitan pada pembangkit tenaga listrik thermal dengan menggunakan *Economic Dispatch* yang didasarkan pada metode *Evolutionary Programming* yang mempunyai keunggulan pada proses operasi, yaitu operasi *competition* (seleksi individu untuk mencari yang terbaik dan akan menghasilkan keturunan).

Berdasarkan pada gambaran permasalahan tersebut di atas, maka skripsi ini diberi judul :

“ Aplikasi Metode *Evolutionary Programming* Pada Pembagian Pembebanan Pembangkit Thermal Di PT. PLN Pembangkit Jawa Bali “

1.3. Tujuan

Memberikan analisa penerapan *Economic Dispatch* unit pembangkit thermal dengan menggunakan metode *Evolutionary Programming* untuk menentukan biaya bahan bakar yang ekonomis dan optimal dengan tetap memperhatikan besar beban yang harus ditanggung dan kapasitas maksimum saluran transmisi pada pembangkit thermal tersebut.

1.4. Batasan Masalah

Agar permasalahan mengarah sesuai dengan tujuan yang dimaksud, maka pembahasan makalah ini hanya dibatasi pada hal-hal berikut:

1. Analisa perhitungan hanya menggunakan metode *Evolutionary Programming* untuk *Economic Dispatch*.
2. Tidak membahas masalah rugi-rugi pada transmisi.
3. Tidak membahas biaya star-up dan biaya shut down.
4. Tidak membahas masalah *Spinning Reserve* (cadangan berputar).
5. Data penjadwalan unit pembangkit PT. PLN Pembangkit Jawa Bali tidak mengalami perubahan karena tidak dilakukan penjadwalan ulang.
6. Analisa dilakukan hanya pada pembangkit thermal dan dibatasi pada pembangkit thermal tenaga uap dan gas yang ada pada lingkungan kerja PT. PLN Pembangkit Jawa Bali.
7. Pembahasan di titik beratkan pada segi ekonomis daripada teknis.

1.5. Metodologi Penelitian

Metodologi yang digunakan dalam penulisan skripsi ini adalah sebagai berikut :

1. Studi kepustakaan (buku, majalah, jurnal dan internet) mengenai hal-hal yang menyangkut dan berhubungan dengan pembahasan masalah.
2. Studi lapangan untuk mendapatkan data parameter unit thermal yang diperlukan dari obyek penelitian yaitu di PT. PLN Pembangkit Jawa Bali yang diperlukan dan berpedoman pada teori yang diperoleh dari studi literature dan studi kepustakaan.
3. Perhitungan *Economic Dispatch* dengan menggunakan metode *Evolutionary Programming* yang dilakukan dengan bahasa pemrograman *Borland Dhelpi versi 7.0*.
4. Membuat evaluasi, sehingga dapat disimpulkan dari perhitungan antara sebelum optimasi dan sesudah dilakukannya optimasi *Economic Dispatch* dengan menggunakan metode *Evolutionary Programming*.

1.6. Sistimatika Penulisan

Skripsi ini terdiri dari lima bab dengan sistimatika penulisan sebagai berikut :

BAB I Pendahuluan

Berisi tentang latar belakang dari skripsi yang terdiri dari rumusan masalah, tujuan, metodologi penelitian, sistimatika penulisan dan relevansi penulisan.

BAB II Kajian Pustaka

Berisi tentang karakteristik pembangkit, *Economic Dispatch*, fungsi biaya bahan bakar, metode *Evolutionary Programming* dan aplikasi *Economic Dispatch* dengan *Evolutionary Programming*.

BAB III Aplikasi Metode *Evolutionary Programming* Untuk *Economic Dispatch*

Berisi tentang teori dasar *Evolutionary Programming*, formulasi masalah *Economic Dispatch*, aplikasi metode *Evolutionary Programming* untuk masalah *Economic Dispatch* dan algoritma program.

BAB IV Hasil Dan Analisis Hasil

Berisi tentang pendahuluan, program komputer *Economic Dispatch* dengan metode *Evolutionary Programming*, data pembangkit thermal, aplikasi metode *Evolutionary Programming* di PT. PLN Pembangkit Jawa Bali, beban sistem serta hasil perhitungan dan analisa hasil perhitungan *Economic Dispatch* menggunakan metode *Evolutionary Programming*.

BAB V Kesimpulan Dan Saran

Berisi tentang kesimpulan dan saran.

1.7. Relevansi

Dari hasil penelitian ini diharapkan dapat menjadi bahan pertimbangan ataupun sebagai solusi alternative yang nantinya dapat digunakan oleh PT. PLN Pembangkit Jawa Bali dalam usahanya untuk meminimalkan biaya operasional, khususnya biaya pembangkitan yang merupakan biaya pengeluaran terbesar pada sistem tenaga listrik.

BAB II

KAJIAN PUSTAKA

2.1. Karakteristik Pembangkit

Performa dari sebuah pusat pembangkit tenaga listrik pada prinsipnya ditentukan oleh apa yang dinamakan lengkung masukan-keluaran (*Input-Output*). Lengkung ini memberikan gambaran tentang *efisiensi thermis* pusat pembangkit tersebut. Selain tergantung pada sifat-sifat pusat tenaga listrik itu sendiri, seperti keadaan air pendingin, kualitas bahan bakar, kecakapan para operator pusat pembangkit dan bentuk lengkung beban.

Berikut ini macam-macam karakteristik pembangkit yang berhubungan dengan penjadwalan operasi pembangkit untuk memperjelas keterangan diatas.

2.1.1. Karakteristik *Input-Output*⁴¹

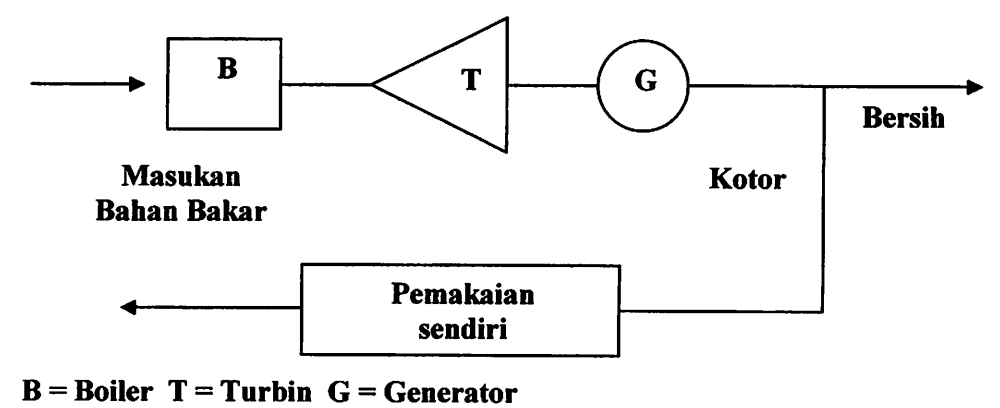
Hal yang paling mendasar dalam mengoptimalkan pembangkit secara ekonomis adalah membuat karakteristik *Input-Output* dari unit pembangkit thermal. Karakteristik ini diperoleh dari desain perencanaan atau melalui test pembangkit. Adapun definisi dari karakteristik *Input-Output* pembangkit adalah formula yang menyatakan hubungan antara *input* pembangkit sebagai fungsi dari *output* pembangkit unit boiler-turbin-generator dapat digambarkan dalam gambar 2.1. dimana unit ini membuat sebuah boiler yang menghasilkan uap untuk turbin yang dikopel dengan rotor dari generator.

Pada pembangkit thermal, *input* diberikan dalam satuan panas Btu/jam atau Kal/jam dari bahan bakar yang diberikan pada boiler untuk menghasilkan

output pembangkit. Sedangkan notasi yang digunakan adalah H(Mbtu/h) atau dalam satuan yang lain H (Mkal/h). selain itu *input* dari pembangkit dapat pula dinyatakan dalam nilai yang menyatakan besarnya biaya yang diperlukan untuk bahan bakar. Notasi yang digunakan adalah F(R/h). Hubungan antara H dan F dapat dinyatakan dalam rumus berikut ini :

$$F = H \times \frac{Rupiah}{Mbtu} \dots\dots\dots (2.1)$$

Dimana rupiah/Mbtu adalah nilai uang yang diperlukan persatuan panas dari bahan bakar.



Gambar 2.1. Unit Boiler-Turbin-Generator^[4]

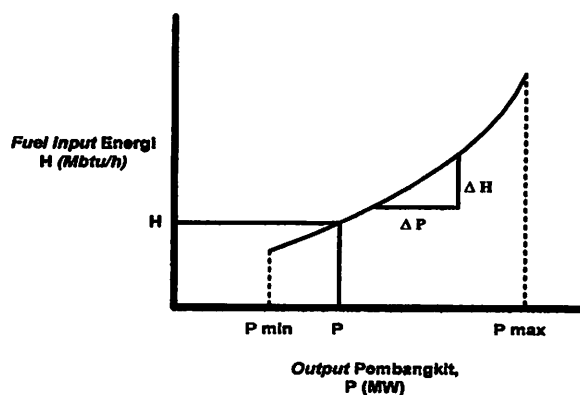
Seperti digambarkan dalam gambar 2.1. maka *output* pembangkit tidak hanya dihubungkan dengan beban tetapi juga untuk peralatan bantu dalam pembangkit. Disini *output* pembangkit didefinisikan sebagai daya yang dikeluarkan oleh generator untuk beban sistem diluar untuk keperluan pembangkit itu sendiri. Jadi untuk karakteristik *input-output*, daya *output* adalah berupa daya *netto* dari pembangkit, notasi yang digunakan adalah P (MW).

Generator akan mengeluarkan daya sesuai dengan beban yang ada. Semakin besar beban, semakin besar pula daya yang dikeluarkan oleh generator. Daya yang dikeluarkan generator dapat membesar sesuai peningkatan beban sampai dengan daya maksimum yang dapat dikeluarkan oleh generator. Semakin besar daya yang dikeluarkan oleh generator, semakin besar pula bahan bakar yang dimasukkan. Dengan kata lain jumlah bahan bakar yang dibakar merupakan fungsi dari daya keluaran generator. Namun hubungan antara bahan bakar dengan daya keluaran generator tidak *linier*, sebab bahan bakar melewati proses pembakaran yang memerlukan waktu.

Dari keterangan diatas, dapat dibentuk persamaan karakteristik *Input-Output* pembangkit yang dapat dilihat pada persamaan 2.2 dan persamaan 2.3 dibawah ini sedangkan kurva dari karakteristik *Input-Output* dapat dilihat pada gambar 2.2.

$$H = f(P), \text{ atau } \dots\dots\dots (2.2)$$

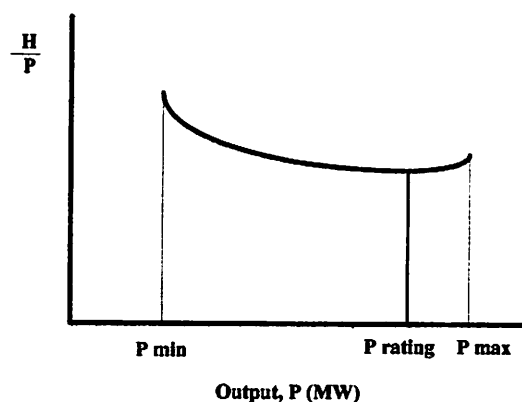
$$I = f(P) \dots\dots\dots (2.3)$$



Gambar 2.2. Kurva Karakteristik *Input-Output* Pembangkit Thermal^[4]

2.1.2. Karakteristik *Heat Rate*^[2]

Karakteristik lain yang cukup penting bagi pembangkit thermal adalah karakteristik tingkat panas atau *Heat Rate Characteristic*. Fungsi ini menyatakan hubungan antara tingkat panas terhadap tingkat beban pusat listrik. Karakteristik ini umumnya memiliki korelasi dengan efisiensi mesin kalor yang digunakan. Gambar 2.3 diplot berdasarkan nilai H/P terhadap P. Pembangkit listrik thermal konvensional memiliki efisiensi kalor antara 30% sampai 35%, sehingga tingkat panas yang dimiliki berkisar antara 11400 BTU/KWH sampai 9800 BTU/KWH (1 KWH kira-kira setara dengan 3412 BTU). Karakteristik tingkat panas pada dasarnya antara lain menyatakan keadaan uap tingkat temperatur, tekanan kondensor dan siklus fluida kerja yang terjadi selama pembangkit listrik tenaga thermal tersebut beroperasi.



Gambar 2.3. Kurva Karakteristik *Heat-Rate* Unit Pembangkit^[4]

2.1.3. Karakteristik *Incremental Heat Rate* dan *Incremental Fuel Cost*^[4]

Perwujudan yang lain dari karakteristik pembangkit adalah karakteristik *Incremental Heat Rate* atau perubahan tingkat laju panas dan karakteristik *Incremental Fuel Cost* atau tingkat perubahan tingkat laju biaya bahan bakar. Karakteristik *Incremental Heat Rate* menyatakan hubungan daya *output* sebagai fungsi *Incremental Heat Rate*. Sedangkan karakteristik *Incremental Fuel Cost* menyatakan daya *output* sebagai fungsi *Incremental Fuel Cost*. Karakteristik *Incremental Heat Rate* ini menunjukkan besarnya perubahan *Input* energi bila ada perubahan *output* pembangkit pada megawatt *output* unit pembangkit.

Kurva karakteristik *Incremental Heat Rate* atau *Fuel Cost* dapat dilihat pada gambar 2.4. sedangkan persamaan *Incremental Heat Rate* dan persamaan *Incremental Fuel Cost* dapat dilihat pada persamaan 2.4. sampai persamaan 2.7.

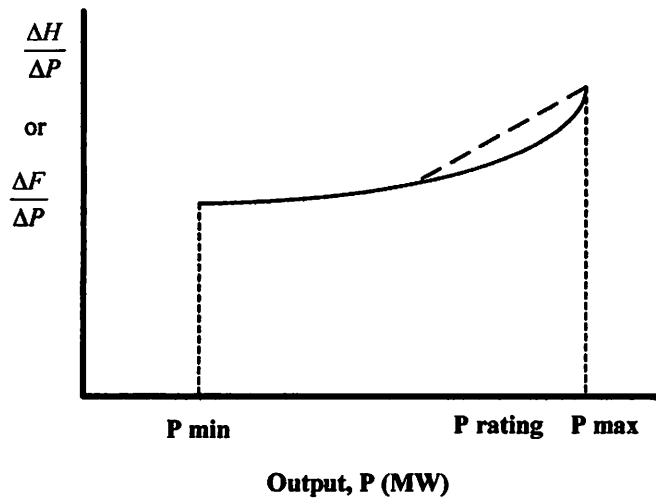
$$\text{Incremental Heat Rate} = \frac{\Delta H}{\Delta P} \left[\frac{\text{MBTU}}{\text{kwh}} \right] \dots\dots\dots (2.4)$$

$$\text{Incremental Fuel Cost} = \frac{\Delta F}{\Delta P} \left[\frac{\text{Rupiah}}{\text{kwh}} \right] \dots\dots\dots (2.5)$$

Dengan mengambil harga ΔP mendekati nol maka dapat dinyatakan dengan persamaan berikut ini

$$\text{Incremental Heat Rate} = \frac{dH}{dP} \left[\frac{\text{MBTU}}{\text{kwh}} \right] \dots\dots\dots (2.6)$$

$$\text{Incremental Fuel Cost} = \frac{dF}{dP} \left[\frac{\text{Rupiah}}{\text{kwh}} \right] \dots\dots\dots (2.7)$$



Gambar 2.4. Kurva Karakteristik *Incremental Fuel Cost Rate*^[4]

2.2. Formulasi Masalah *Economic Dispatch*^[4]

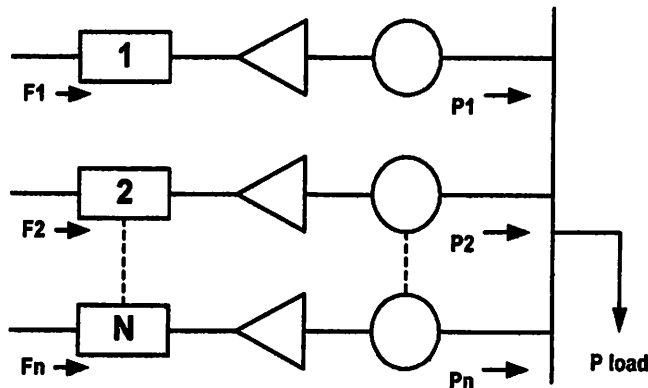
Yang dimaksud *Economic Dispatch* adalah pembagian pembebanan pada pembangkit-pembangkit yang ada dalam sistem, secara optimal ekonomis pada harga beban tertentu. Tujuan dari *Economic Dispatch* adalah untuk mendapatkan biaya harga bahan bakar semurah mungkin dalam suatu sistem pembangkit pada beban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

Economic Dispatch dapat dilakukan dengan beberapa cara yang akan dibahas pada sub bab dibawah ini.

2.2.1. Penyelesaian *Economic Dispatch* Dengan Metode Pengali La Grange

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

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



Gambar 2.5. N Unit Melayani Beban P_R ^[4]

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

$$F_T = F_1 + F_2 + F_3 + \dots + F_N$$
$$= \sum_{i=1}^N F_i(P_i) \dots\dots\dots (2.8)$$

$$\phi = 0 = P_R - \sum_{i=1}^N P_i \dots\dots\dots (2.9)$$

Persamaan ini adalah pembatas yang merupakan masalah dari optimasi dan ini dapat dipecahkan dengan metode kalkulus tingkat lanjut yang melibatkan fungsi La Grange. Dimana fungsi La Grange didapat dengan cara menambah pembatas ϕ yang telah dikalikan dengan factor pengali La Grange ϕ pada fungsi F_T . Fungsi La Grange dapat ditunjukan dengan persamaan dibawah ini :

$$L = F_T + \lambda \phi \dots\dots\dots (2.10)$$

Dimana :

- F_T = fungsi tujuan
- λ = factor pengali
- ϕ = fungsi pembatas (*constrain*)

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

$$\nabla L = 0 \dots\dots\dots (2.11)$$

$$\nabla F_T + \lambda \phi = 0 \dots\dots\dots (2.12)$$

$$\frac{\delta L}{\delta P} = \frac{\delta F_T}{\delta P_i} + \lambda \left[\frac{\delta P_R}{\delta P_i} - \frac{\delta P_i}{\delta P_i} \right] = 0 \text{ atau} \dots\dots\dots (2.13)$$

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

$$\frac{\delta F_i}{\delta P_i} = \lambda \dots\dots\dots (2.14)$$

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

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

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

$$P_{i_{\min}} \leq P_i \leq P_{i_{\max}} \text{ ada } 2N \text{ buah pertidaksamaan } \dots\dots (2.15)$$

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

Dari batasan pertidaksamaan pembatasan diatas dapat diperluas menjadi :

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

$$\frac{\delta F_i}{\delta P_i} \leq \lambda \text{ untuk } P_i = P_{i_{\max}} \dots\dots\dots (2.16)$$

$$\frac{\delta F_i}{\delta P_i} \geq \lambda \text{ untuk } P_i = P_{i_{\min}} \dots\dots\dots (2.17)$$

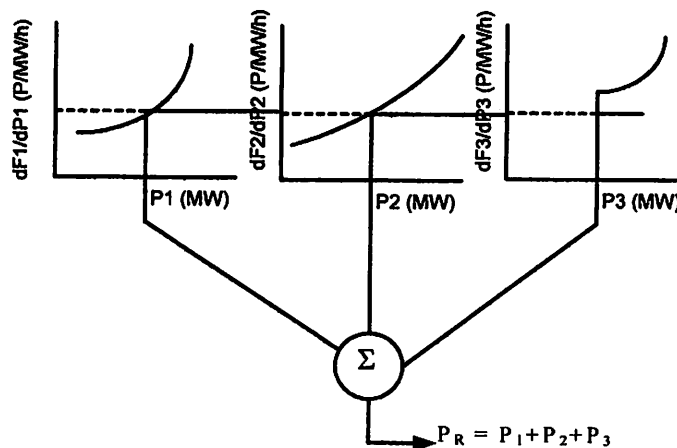
Karena F_i hanya sebagai fungsi P_i maka $\frac{\delta F_i}{\delta P_i}$ dapat diganti dengan $\frac{dF_i}{dP_i}$

2.2.2. Penyelesaian *Economic Dispatch* Dengan Metode Iterasi Lamda

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

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

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



Gambar 2.6. Grafik Penyelesaian Dengan Metode Iterasi Lamda^[4]

2.3. Fungsi Biaya Bahan Bakar

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

$$F_i(P_{it}) = a_i P_{it}^2 + b_i P_{it} + c_i \dots\dots\dots (2.18)$$

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

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

Dalam pengoperasian secara ekonomis adalah penting untuk mengetahui biaya bahan bakar yang digunakan untuk membangkitkan daya yang diperlukan :

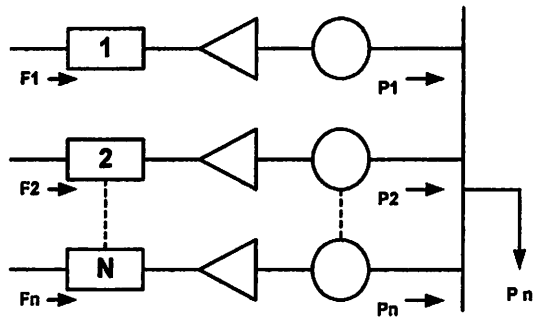
- Jenis bahan bakar
- Nilai kalori
- Harga bahan bakar

2.4. *Economic Dispatch* Dengan Mengabaikan Rugi-rugi Transmisi

Dalam sistem tenaga listrik, kerugian transmisi merupakan kehilangan daya yang harus ditanggung oleh sisi pembangkit. Jadi dengan adanya kerugian daya tersebut merupakan tambahan beban bagi sistem tenaga listrik.

Sistem dengan mengabaikan rugi-rugi transmisi dapat dilihat pada gambar 2.9. Sistem ini terdiri dari N buah pembangkit thermal yang dihubungkan pada single bus bar yang melayani beban P_r . *Input* dari masing-masing pembangkit ditunjukkan oleh F_i yang mewakili biaya dari satu unit pembangkit dan output

dari masing-masing unit P_i adalah daya yang dihasilkan oleh satu unit pembangkit.



Gambar 2.7. N Unit Pembangkit Thermal Melayani Beban P_R

Total biaya rata-rata yang harus ditanggung oleh sistem adalah jumlah biaya dari masing-masing unit pembangkit. Dan pembatas yang paling penting adalah jumlah *output* dari dari masing-masing unit pembangkit sama dengan di konsumen. Yang menjadi permasalahan adalah meminimumkan total biaya F_T dengan memperhatikan pembatas ϕ bahwa daya yang dihasilkan oleh pembangkit sama dengan beban yang diterima. Secara matematika pernyataan yang tersebut diatas dapat dinyatakan dengan persamaan sebagai berikut :

$$\begin{aligned} F_T &= F_1 + F_2 + F_3 + \dots + F_N \dots\dots\dots(2.19) \\ &= \sum_{i=1}^N F_i(P_i) \\ \phi = 0 &= P_R - \sum_{i=1}^N P_i \end{aligned}$$

Persamaan di atas adalah pembatas yang merupakan problem dari optimasi dan ini dapat dipecahkan dengan menggunakan kalkulus tingkat lanjut yang

melibatkan fungsi La Grange. Dimana fungsi ini didapat dengan cara menambahkan pembatas ϕ yang telah dikalikan dengan factor pengali La Grange λ pada fungsi tujuan F_T fungsi La Grange dapat ditunjukkan dengan persamaan dibawah ini :

$$L = F_T + \lambda . \phi$$

Persamaan La Grange di atas fungsi *output* pembangkit P_i dan faktor pengali La Grange λ keadaan dari optimal dari fungsi tujuan F_T dapat diperoleh dengan operasi gradient persamaan La Grange sama dengan 0.

$$\nabla L = 0 \dots\dots\dots (2.20)$$

$$\nabla F_T + \lambda . \phi = 0 \dots\dots\dots (2.21)$$

$$\frac{\partial L}{\partial P} = \frac{\partial F_i}{\partial P_i} + \lambda \left(\frac{\partial P_R}{\partial P_i} - \frac{\partial P_i}{\partial P_i} \right) = 0 \dots\dots\dots (2.22)$$

$$\text{atau } \frac{\partial F_i}{\partial P_i} + \lambda . (0 - 1) = 0 \dots\dots\dots (2.23)$$

$$\frac{\partial F_i}{\partial P_i} = \lambda \dots\dots\dots (2.24)$$

Persamaan terakhir ini menunjukkan bahwa bila digunakan biaya bahan bakar F_T yang paling minimum maka *incremental cost* setiap unit generator pembangkit harus sama yaitu sebesar λ kondisi optimal ini tentunya dengan tetap memperhatikan pembatas yang ada yaitu bahwa daya dari setiap unit generator pembangkit harus lebih besar atau sama dengan daya *output* minimum dan lebih kecil atau sama dengan daya *output* maksimum yang diizinkan.

Dari N buah pembangkit yang ada dalam sistem tenaga yang telah dibahas dan beban sistem sebesar P_R , maka dapat diambil kesimpulan sebagai berikut :

$\frac{\partial F_i}{\partial P_i} = \lambda$ (2.25)

$P_{i\ min} \leq P_i \leq P_{i\ maks}$ ada 2 N buah pertdaksamaan(2.26)

$\sum_{i=1}^N P_i = P_R$ (2.27)

Dari batasan pertidaksamaan pembatas di atas dapat di perluas menjadi :

$\frac{\partial F_i}{\partial P_i} = \lambda$ untuk $P_{i\ min} \leq P_i \leq P_{i\ maks}$ (2.28)

$\frac{\partial F_i}{\partial P_i} \leq \lambda$ untuk $P_i = P_{i\ maks}$(2.29)

$\frac{\partial F_i}{\partial P_i} \geq \lambda$ untuk $P_i = P_{i\ min}$ (2.30)

Karena F_i hanya sebagai fungsi P_i maka $\frac{\partial F_i}{\partial P_i}$ dapat diganti dengan $\frac{\partial F_i}{\partial P_i}$.

BAB III

APLIKASI METODE *EVOLUTIONARY PROGRAMMING* UNTUK *ECONOMIC DISPATCH*

3.1. Teori Dasar *Evolutionary Programming*^{[1][3]}

3.1.1. Konsep Dasar

Evolutionary Programming (EP) merupakan metode *stokastik* yang biasa digunakan untuk memecahkan suatu pencarian nilai dalam sebuah masalah optimasi. Metode ini didasarkan pada proses evolusi yang ada pada makhluk hidup yaitu dalam perkembangan generasi dalam sebuah populasi yang alami, secara lambat laun mengikuti prinsip mengikuti seleksi alam “siapa yang kuat , dia yang bertahan (*survive*)”. Dengan meniru proses ini, *Evolutionary Programming* dapat digunakan untuk mencari solusi permasalahan-permasalahan dalam dunia nyata.

Evolutionary Programming adalah suatu metode strategi optimasi yang merupakan cabang dari *Evolutionary Computation* yang didalamnya terdiri dari *Algoritma Genetika*, *Genetic Programming*, *Evolutionary Strategies*, dan *Evolutionary Programming*. Perbedaan yang paling mendasar antara *Evolutionary Programming* dengan *Algorithm Genetika* adalah pada proses operasi. Dalam metode *Evolutionary Programming* tidak menggunakan operasi *crossover* melainkan operasi *competition* (kompetisi).

Evolutionary Programming ditemukan oleh Lawrence.J. Fogel 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 oleh *string-string* bilangan real yang disebut *kromosom*. Seperti halnya alam, metode 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 .

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 *genetic* terkecil yaitu gen. Pemakaian bilangan real (*floating point*) sebagai *allele* (nilai gen) memungkinkan penerapan operator *Evolutionary Programming* yaitu proses produksi (*reproduction*), mutasi (*mutation*), dan kompetisi (*competition*) untuk menciptakan himpunan 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*).

Evolutionary Programming memiliki tiga 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, tidak hanya memproses satu solusi saja.

3. Hanya memperhitungkan fungsi *fitness* setiap kandidat solusi untuk mendapatkan hasil optimum global.

3.1.2 Parameter *Evolutionary Programming*

Terdapat beberapa parameter pada *Evolutionary Programming* yang digunakan untuk melihat kompleksitas dari *Evolutionary Programming* itu sendiri. Parameter yang digunakan adalah :

❖ Jumlah Generasi (MAXGEN)

Merupakan jumlah perulangan (*iterasi*) dilakukannya rekombinasi dan seleksi. Jumlah generasi ini mempengaruhi kestabilan *output* dan lama iterasi (waktu proses *Evolutionary Programming*). 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*.

❖ Ukuran Populasi (POPSIZE)

Ukuran populasi mempengaruhi kinerja dan efektifitas dari *Evolutionary Programming*. Jika ukuran populasi kecil maka populasi tidak menyediakan cukup materi untuk mencakup ruang permasalahan, sehingga pada umumnya kinerja *Evolutionary Programming* 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.

❖ Probabilitas Mutasi (P_m)

Mutasi digunakan untuk meningkatkan variasi populasi dan digunakan untuk menentukan tingkat mutasi yang terjadi, karena frekuensi terjadinya mutasi tersebut menjadi $P_m \times \text{POPSIZE} \times N$, dimana N adalah panjang struktur/gen dalam suatu 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 *Evolutionary Programming* mutasi menjalankan aturan penting yaitu :

1. Menggantikan gen-gen yang hilang selama proses seleksi.
2. Menyediakan gen-gen yang tidak muncul pada saat inisialisasi awal populasi.

❖ Panjang Kromosom ($NVAR$)

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

❖ Probabilitas Beta

Nilai minimal bobot kepastian dari setiap kromosom agar bisa bertahan untuk dijadikan solusi yang dipresentasikan dalam suatu undak biner pada aturan probabilitas sebagai pembatas nilai minimum pada range proses statistik.

3.1.3. Mekanisme *Evolutionary Programming*[¹¹¹³]

A. Pengkodean atau *Representasi*

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

Secara umum EP dibentuk oleh serangkaian kromosom yang ditandai dengan x_i ($i=1,2,...N$). Setiap elemen dalam kromosom ini adalah *variabel string* yang disebut gen., berisi nilai-nilai *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 EP untuk awal melakukan pencarian.

B. Fungsi *Fitness* (Fungsi Evaluasi)

Dalam EP, 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 ($i = 1,2,...N$) dipergunakan untuk menghitung $f_k(x)$ ($k = 1,2,...POPSIZE$)

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.

C. 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 akan menghasilkan keturunan”. Pada proses seleksi ini dapat banyak menggunakan metode seperti *roulette wheel selection*, *rank selection*, *elitesm* dan lain sebagainya.

❖ *Roulette Wheel Selection*

Dimana setiap individual memiliki harga *fitness* sehingga didapatkan *probabilitas individual* ($f(i)/\sum f(t)$) tersebut dicopykan 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 *fitnessnya*.
3. Menjumlahkan *fitness* dari kromosom-kromosom dari populasi mulai 0 hingga total *fitness* lebih besar atau sama dengan nilai k lalu ambil kromosom tersebut.

❖ *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 mempunyai *fitness* 2 dan seterusnya sampai yang terakhir akan mempunyai *fitness* N. Dengan demikian semua kromosom akan mempunyai peluang untuk diseleksi.

D. *Mutation* (Mutasi)

Operator mutasi digunakan untuk memodifikasi satu atau lebih nilai gen dalam 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 Strategies* yaitu menggunakan operator *Gaussian Mutation*, dimana setiap individu akan terpilih secara acak untuk mengalami mutasi berdasarkan nomor acak Gaussian untuk untuk menciptakan individu baru (*offspring*).



Gambar 3.1. Ilustrasi Proses Mutasi

a). Mutasi Gaussian dari Induk (*parent*) b). Menghasilkan Anak (*offspring*)

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

E. *Competition* (Kompetisi)

Dalam tahap kompetisi, mekanisme seleksi dipakai untuk menghasilkan populasi baru dari populasi yang ada. Melalui penggunaan skema kompetisi setiap individu dalam populasi baik orang 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.

3.2. Formulasi Masalah *Economic Dispatch*^[1]

Economic Dispatch adalah suatu pembagian pembebanan secara ekonomis suatu daya listrik yang dihasilkan oleh generator dalam suatu periode waktu tertentu secara optimum dan ekonomis. Dengan dilakukannya *Economic Dispatch*

maka akan didapatkan biaya bahan bakar yang paling murah dalam suatu sistem pembangkit. *Objective Function* atau fungsi tujuan adalah :

$$Min F = \sum_{t=1}^T \sum_{i=1}^n F_{it}(P_{it})(3.1)$$

dimana F adalah total biaya operasi untuk penyaluran dalam periode tertentu., n adalah total unit yang bekerja pada waktu t . $F_{it}(P_{it})$ adalah biaya produksi setiap unit pembangkit dalam bentuk keluaran daya aktif P_i pada waktu t .

Biaya pembangkitan untuk setiap unit diberikan oleh persamaan:

$$Fi(Pi) = a_iP_i^2 + b_iP_i + c_i.....(3.2)$$

Dengan memperhatikan syarat batasan sebagai berikut :

- 1. Batasan daya aktif (*Real power balance constraint*)

$$\sum_{i=1}^n P_{ij} - P_{Dj} - P_{Lj} = 0(3.3)$$

dimana P_{Dj} , P_{Lj} adalah total kebutuhan beban dan rugi rugi yang terjadi pada sistem selama penyaluran.

- 2. Batas operasi daya aktif (*Real power operating limits*)

$$P_{imin} \leq P_i \leq P_{imax} \quad , \quad i = 1, ..., n.....(3.4)$$

dimana P_{imin} , P_{imax} adalah keluaran daya aktif minimum dan maksimum dari pembangkit ke i .

- 3. Batasan *ramp rate* unit pembangkit (*generating unit ramp rate limitations*)

$$P_{it} - P_{i(t-1)} \leq UR_i \quad , \quad i = 1, ..., n.....(3.5)$$

$$P_{i(t-1)} - P_{it} \leq DR_i \quad , \quad i = 1, ..., n.....(3.6)$$

dimana UR_i , DR_i merupakan batas *ramp-up rate* dan *ramp-down rate* dari unit ke i

3.3. Aplikasi Metode *Evolutionary Programming* Untuk Masalah *Economic Dispatch*

3.3.1. Inisialisasi^[1]

Setiap variabel dari setiap individu dilakukan inisialisasi secara acak, dimana N adalah ukuran populasi. Setiap x_i , $i = 1, 2, \dots, N$ diberikan suatu nilai *fitness* $\mathcal{G}(x_i) = G(f(x_i), v_i)$, dimana $f(x_i)$ merupakan *fitness* yang sejati dari gen x_i , v_i diuraikan sebagai alternatif acak dalam x_i . Jadi $G(f(x_i), v_i)$, dituliskan sebagai nilai *fitness* (*fitness score*).

3.3.2 Fungsi Tujuan (*Objective Function*)^[1]

Formulasi permasalahan dirumuskan pada suatu fungsi tujuan (*objective function*). Tujuan dari permasalahan *Economic Dispatch* adalah untuk mendapatkan biaya bahan bakar yang paling murah dalam suatu sistem pembangkit. Fungsi tujuan (*objective function*) yang dirancang untuk permasalahan tersebut adalah^[1] :

$$\text{Min } F = \sum_{t=1}^T \sum_{i=1}^n F_{it}(P_{it}) + \sum_{t=2}^T \sum_{i=2}^n \rho_{\text{ramp}} |P_{it} - P_{\text{ramp lim}}| \dots\dots\dots(3.7)$$

Dalam optimasi ini, syarat-syarat batasan dirumuskan dalam fungsi tujuan (*objective function*) menggunakan istilah *penalty* (*penalty terms*).

3.3.3 Fungsi *Fitness*

Setiap calon solusi menugaskan fungsi *fitness* untuk mengukur keoptimalannya yang berkenaan dengan sasaran yang akan dioptimalkan. Dalam permasalahan ini, fungsi *fitness* yang dirancang untuk setiap individu adalah :

$$f = \frac{A}{\sum_{t=1}^T \sum_{i=1}^n F_{it}(P_{it}) + \sum_{t=2}^T \sum_{i=1}^n \rho_{ramp} |P_{it} - P_{ramp\lim}|} \dots\dots\dots(3.8)$$

dimana ρ_{ramp} adalah parameter *penalty* dan $P_{ramp\lim}$ direpresentasikan sebagai berikut :

$$P_{ramp\lim} = \begin{cases} P_{i(t-1)} - DR_t & \text{if } P_{it} < P_{i(t-1)} - DR_t, \\ P_{i(t-1)} + UR_t & \text{if } P_{it} > P_{i(t-1)} + UR_t, \dots\dots\dots(3.9) \\ P_{it} & , \text{ sebaliknya} \end{cases}$$

Syarat batasan pada persamaan (3.3) dihandle oleh persamaan di bawah ini :

$$\text{if } \sum_{i=1}^n P_{it} > P_{Dt} \text{ then } P_{ij}^{new} = P_{ij}^{old} - \frac{\sum_{i=1}^n P_{it}^{old} - P_{Dt}}{n} \dots\dots\dots(3.10)$$

dan pendekatan yang serupa juga diimplementasikan untuk $\sum_{i=1}^n P_{it} < P_D$

Nilai *objective function* yang dinormalisasikan dalam persamaan (3.9) digunakan untuk mengindikasi *fitness* dari kandidat solusi (*offspring*) pada individu- individu.

3.3.4. Pembangkitan *Offspring* Dengan Mutasi^[1]

Populasi yang baru dari solusi masalah *Economic Dispatch* dihasilkan melalui operator mutasi. Individu yang baru p'_i dihasilkan dari setiap individu p_i , dimana dihitung dengan persamaan sebagai berikut :

$$P'_i = P_i + N(0, \sigma_i^2) \dots\dots\dots(3.10)$$

Dimana $N(0, \sigma_i^2)$ adalah nomor acak *Gaussian* dengan nilai tengah nol dan standar deviasi (simpangan baku) dari σ_i . Cara untuk merancang σ_i adalah :

$$\sigma_{it} = \beta \times \frac{f_s}{f_{\max}} (P_{it\max} - P_{it\min}) \dots \dots \dots (3.11)$$

Dimana β adalah faktor skala yang ditentukan selama proses pencarian yang paling optimum, sedangkan f_s adalah *fitness* maksimum dari populasi *parent*. Setelah penambahan nilai *Gaussian random fitness* pada *parent*, elemen *offspring* ditambahkan batasan dari pelanggaran yang diberikan pada persamaan (3.4), yang mana pelanggarannya memperhatikan persamaan dibawah :

$$P'_{it} = P_{it\min} \text{ if } P_{it} < P_{it\min} \dots \dots \dots (3.12)$$

$$P'_{it} = P_{it\max} \text{ if } P_{it} > P_{it\max} \dots \dots \dots (3.13)$$

3.3.5. Seleksi Dan Kompetisi^[1]

Berdasarkan populasi baru yang dihasilkan yakni kombinasi populasi dengan ukuran $2N$, setiap individu dalam populasi harus dikompetisi secara *stochastic* dengan anggota yang lain berdasarkan nilai *fitness*nya. Setiap x_i , $i=1,2,\dots,N$. suatu nilai w_i diberikan untuk bobot individu menurut persamaan berikut:

$$w_i = \sum_{t=1}^s w_t \dots \dots \dots (3.14)$$

dengan :

$$w_t = \begin{cases} 1, & \text{jika } \mathcal{G}(x_i) < \mathcal{G}(x_r) \\ 0, & \text{sebaliknya} \end{cases}$$

N individu pertama dituliskan bersama nilai *fitness*nya untuk menjadi awal dari generasi berikutnya. Proses akan melakukan pengulangan sampai memberikan kondisi yang paling *konvergen*.

3.4. Algoritma Program

3.4.1. Algoritma Penyelesaian *Economic Dispatch* Menggunakan Metode *Evolutionary Programming*

Urutan langkah-langkah dalam program komputer yang menggunakan metode *Evolutionary Programming* adalah sebagai berikut :

1. Memasukkan inputan data parameter unit pembangkit thermal dan data pembebanan tiap jam.
2. Memasukkan inputan parameter *Evolutionary Programming* yang meliputi jumlah populasi, jumlah generasi, probabilitas mutasi dan probabilitas betha, konstanta *penalty*, dan panjang kromosom.
3. Menentukan populasi awal ($pop = 0$), menentukan generasi awal ($gen = 0$), $Max\ Gen = 100$ dan $Max\ Pop = 50$.
4. Menghitung nilai *fitness* populasi awal atau populasi *parent*.
5. Melakukan proses statistik.
6. Melakukan proses seleksi.
7. Melakukan proses mutasi.
8. Apakah *offspring* sama dengan jumlah *parent*.
 - a. Jika “Ya” maka dilanjutkan ke langkah 9
 - b. Jika “Tidak” maka $off = off + 1$ dan kembali ke langkah 5.
9. Melakukan proses kompetisi.
10. Diperiksa, apakah kondisi sudah konvergen.
 - a. Jika “Ya” maka perhitungan selesai.
 - b. Jika “Tidak” maka dilanjutkan kelangkah 11.

11. Diperiksa, apakah generasi sudah maksimum.

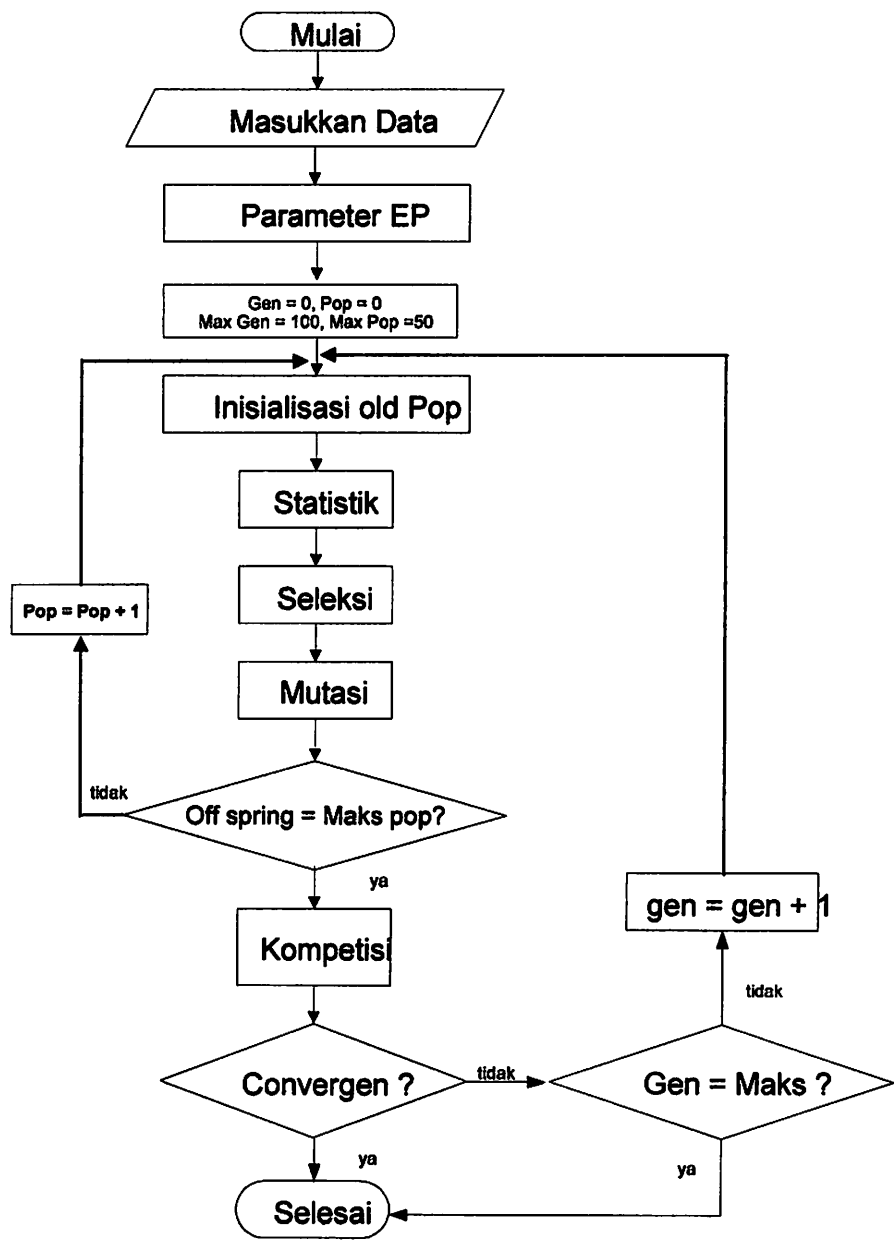
a. Jika “Ya” maka perhitungan selesai.

b. Jika “Tidak” maka $gen = gen + 1$ dan kembali ke langkah 5.

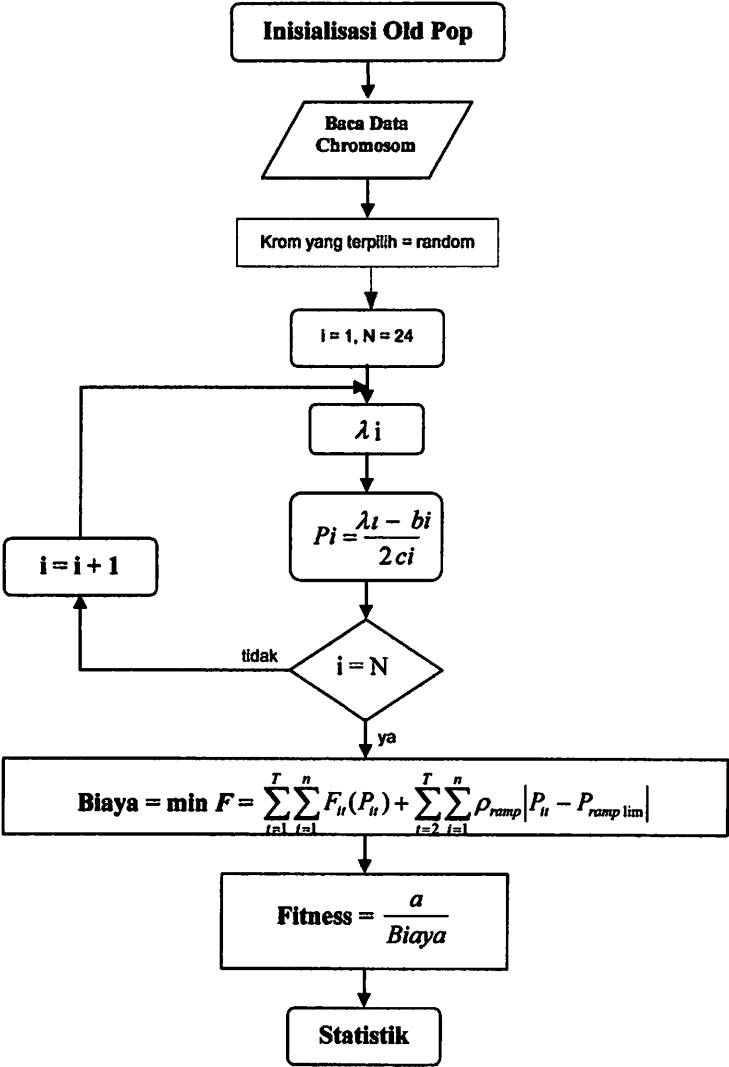
3.4.2 Algoritma Subroutine Fungsi *Fitness* Untuk Masalah *Economic Dispatch* Menggunakan Metode *Evolutionary Programming*

1. Inisialisasi Old Pop.
2. Memasukkan input kromosom yang dilakukan secara random (acak).
3. Menentukan kromosom yang terpilih secara random.
4. Menentukan $i = 1$ dan $N = 24$.
5. Menentukan nilai λ i.
6. Menentukan daya output optimal untuk tiap pembangkit pada jam ke i .
7. Diperiksa apakah $i = N$.
 Jika “Ya” maka dilanjutkan kelangkah 8.
 Jika “Tidak” maka $i = i + 1$ dan kembali ke langkah 5.
8. Menghitung biaya pembangkitan seperti pada persamaan 3.7.
9. Menghitung *Fitness* seperti pada persamaan 3.8.
10. Melakukan proses statistik.

3.4.3. Flowchart *Economic Dispatch Menggunakan Metode Evolutionary Programming.*



3.4.4. Flowchart Subroutine Fungsi *Fitness* Untuk Masalah *Economic Dispatch* Dengan Metode *Evolutionary Programming*.



BAB IV

HASIL DAN ANALISIS HASIL

4.1. Pendahuluan

PT. PLN Pembangkit Jawa Bali dalam penyelenggaraan usaha ketenagalistrikan berdasarkan prinsip industri dan perniagaan yang sehat, dituntut mampu bersaing dan mampu memanfaatkan sebesar-besarnya peluang pasar dalam bidang tenaga listrik. Dalam hal tersebut, PT. PLN Pembangkit Jawa Bali harus menjaga efisiensi dan keadilan operasional penyediaan tenaga listrik dari pembangkit-pembangkit yang dimilikinya.

Dengan demikian merupakan suatu keharusan bagi seluruh jajaran PT.PLN Pembangkit Jawa Bali agar selalu berupaya untuk meningkatkan kondisi penyediaan tenaga listrik dari pembangkit agar lebih ekonomis, bermutu dan didukung keandalan yang tinggi.

4.2. Program Komputer *Economic Dispatch* Dengan Metode *Evolutionary Programming*.

Dalam penyelesaian masalah ini digunakan bantuan program komputer. Program komputer ini sangat berguna untuk mempercepat proses perhitungan yang membutuhkan ketelitian tinggi dan sering melibatkan iterasi dengan waktu yang lama pula bila dikerjakan secara manual.

Program komputer ini menggunakan bahasa pemrograman *Borland Delphi versi 7.0*, yang merupakan bahasa pemrograman terstruktur yang relatif mudah untuk dipelajari dan mudah penggunaanya.

4.3. Data Pembangkit Thermal

Pembangkit Thermal yang berada dalam pengawasan PT.PLN Pembangkit Jawa Bali berjumlah 38 unit yang terdiri dari 5 blok Pembangkit Listrik Tenaga Gas dan Uap, 11 Pembangkit Listrik Tenaga Uap dan 5 Pembangkit Listrik Tenaga Gas. Adapun Data-data lebih lengkapnya dapat dilihat pada tabel 4.1, untuk harga bahan bakar berdasarkan statistik PLN 2004 dimana dipakai nilai tukar Rp. 9000 per satuan dolar Amerika.

Tabel 4.1.
Data Unit Thermal Pada PT.PLN Pembangkit Jawa Bali Tahun 2004

No	Nama Pembangkit	Bahan Bakar	Kapasitas (MW)		Koefisien Biaya Bahan Bakar		
			Min	Max	A	B	C
1	PLTU Paiton 1	Coal	225	370	3244978	111712,15	10,2971
2	PLTU Paiton 2	Coal	225	370	3244978	111712,15	10,2971
3	PLTGU Gresik GT 1.1	Gas	53	102	5467532,4	217963,548	34,155
4	PLTGU Gresik GT 1.2	Gas	53	102	5467532,4	217963,548	34,155
5	PLTGU Gresik GT 1.3	Gas	53	102	5467532,4	217963,548	34,155
6	PLTGU Gresik ST 1.0	Gas	115	143	10936203,3	72527,004	368,874
7	PLTGU Gresik GT 2.1	Gas	53	102	5467532,4	217963,548	34,155
8	PLTGU Gresik GT 2.2	Gas	53	102	5467532,4	217963,548	34,155
9	PLTGU Gresik GT 2.3	Gas	53	102	5467532,4	217963,548	34,155
10	PLTGU Gresik ST 2.0	Gas	115	143	10936203,3	72527,004	368,874
11	PLTGU Gresik GT 3.1	Gas	53	102	5467532,4	217963,548	34,155
12	PLTGU Gresik GT 3.2	Gas	53	102	5467532,4	217963,548	34,155
13	PLTGU Gresik GT 3.3	Gas	53	102	5467532,4	217963,548	34,155
14	PLTGU Gresik ST 3.0	Gas	115	143	10936203,3	72527,004	368,874
15	PLTU Gresik 1	Gas	43	85	1327126,68	217378,359	132,066
16	PLTU Gresik 2	Gas	43	85	1327126,68	217378,359	132,066
17	PLTU Gresik 3	Gas	90	175	5017369,5	169242,579	193,545
18	PLTU Gresik 4	Gas	90	175	5017369,5	169242,579	193,545
19	PLTG Gresik 1	Gas	5	16	352707,3	350680,77	903,969
20	PLTG Gresik 2	Gas	5	16	352707,3	350680,77	903,969
21	PLTG Gresik 3	Gas	5	16	352707,3	350680,77	903,969
22	PLTG Gilitimur 2	HSD	5	16	687181,85	683240,965	1762,3893
23	PLTG Gilitimur 2	HSD	5	16	687181,85	683240,965	1762,3893
24	PLTGU M. Karang GT 1.1	Gas	50	95	5730795	202052,97	108,045
25	PLTGU M. Karang GT 1.2	Gas	50	95	5730795	202052,97	108,045
26	PLTGU M. Karang GT 1.3	Gas	50	95	5730795	202052,97	108,045
27	PLTGU M. Karang ST 1.0	Gas	110	150	11560815	53685,135	460,845
28	PLTGU M. Tawar GT 1.1	HSD	72	138	14706521,25	433337,8	49,4605
29	PLTGU M. Tawar GT 1.2	HSD	72	138	14706521,25	433337,8	49,4605
30	PLTGU M. Tawar GT 1.3	HSD	72	138	14706521,25	433337,8	49,4605
31	PLTGU M. Tawar GT 2.1	HSD	72	138	14706521,25	433337,8	49,4605
32	PLTGU M. Tawar GT 2.2	HSD	72	138	14706521,25	433337,8	49,4605
33	PLTGU M. Tawar ST 1.0	HSD	162	202	672630	144191,717	519,1757
34	PLTU M. Karang 1	MFO	44	85	2417820,7	473895,41	120,77935
35	PLTU M. Karang 2	MFO	44	85	2417820,7	473895,41	120,77935
36	PLTU M. Karang 3	MFO	44	85	2417820,7	473895,41	120,77935
37	PLTU M. Karang 4	Gas	90	165	2949187,5	205217,145	83,79
38	PLTU M. Karang 5	Gas	90	165	2949187,5	205217,145	83,79

Sumber : Data Penawaran PT PLN PJB, JL. Ketingtang Baru NO. 11, Surabaya 60231

Catatan : Harga Batubara 235 Rp/Kg
 Harga MFO 1595 Rp/liter
 Harga HSD 1595 Rp/liter
 Harga Gas UP. Gresik 2,53 US\$/MMBTU
 Harga Gas UP. Gresik 2,45 US\$/MMBTU
 Nilai Tukar 9000 Rp/US\$

4.4. Aplikasi Metode *Evolutionary Programming* di PT.PLN Pembangkit Jawa Bali

Perhitungan dan analisa ini dilakukan pada kebutuhan daya yang ditanggung oleh PT. PLN Pembangkit Jawa Bali tanggal 10, 13 dan 14 Maret 2004. Analisa data dilakukan hanya untuk unit yang beroperasi, karena program komputer ini hanya untuk menghitung unit pembangkit yang siap beroperasi menjadi 19 unit yang dapat dilihat pada tabel 4.2.

Tabel 4.2.
Unit Pembangkit Thermal Yang Siap Operasi

No	Nama Pembangkit	Bahan Bakar	Kapasitas (MW)	
			Min	Max
1	PLTU Paiton 1	Coal	225	370
2	PLTU Paiton 2	Coal	225	370
3	PLTGU Gresik GT 1.3	Gas	53	102
4	PLTGU Gresik GT 2.3	Gas	53	102
5	PLTGU Gresik GT 3.2	Gas	53	102
6	PLTGU Gresik GT 3.3	Gas	53	102
7	PLTGU Gresik ST 1.0	Gas	250	480
8	PLTU Gresik 1	Gas	43	85
9	PLTU Gresik 2	Gas	43	85
10	PLTU Gresik 3	Gas	90	175
11	PLTU Gresik 4	Gas	90	175
12	PLTGU M. Karang ST	Gas	300	465
13	PLTGU M. Tawar ST 1.0	HSD	162	202
14	PLTGU M. Karang GT	Gas	50	95
15	PLTU M. Karang 1	MFO	44	85
16	PLTU M. Karang 2	MFO	44	85
17	PLTU M. Karang 3	MFO	44	85
18	PLTU M. Karang 4	Gas	90	165
19	PLTU M. Karang 5	Gas	90	165

4.5. Beban Sistem

Dalam wilayah Jawa Bali, pembangkit-pembangkit listrik yang ada dikoordinasi oleh PT. PLN Pembangkit Jawa Bali. Proses *Economic Dispatch* dengan metode *Evolutionary Programming* bertujuan untuk membuat rencana operasi yang optimum dalam sistem tenaga listrik sehingga 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 thermal dan beban yang ditanggung oleh PT. PLN Pembangkit Jawa Bali sebagai bahan perbandingan. Sedangkan kombinasi jadwal dan daya *output* pembangkit tenaga listrik dalam sistem PT. PLN Pembangkit Jawa Bali tanggal 10, 13 dan 14 Maret 2004 terdapat pada lampiran. Untuk beban sistem terdapat pada tabel 4.3, 4.4 dan 4.5 (beban sistem yang ditanggung oleh pembangkit thermal saja).

Tabel 4.3.
Data Pembebanan Unit Thermal
PT.PLN Pembangkit Jawa Bali
Rabu 10 Maret 2004

JAM	Beban Sistem (MW)
01:00	3108
02:00	3024
03:00	2993
04:00	2974
05:00	2976
06:00	2916
07:00	2808
08:00	3202
09:00	3265
10:00	3281
11:00	3297
12:00	3220
13:00	3225
14:00	3226
15:00	3297
16:00	3372
17:00	3499
18:00	3600
19:00	3657
20:00	3642
21:00	3403
22:00	3388
23:00	3335
24:00	3316

Tabel 4.4.
Data Pembebanan Unit Thermal
PT. PLN Pembangkit Jawa Bali
Sabtu 13 Maret 2004

JAM	Beban Sistem (MW)
01:00	2896
02:00	2864
03:00	2845
04:00	2866
05:00	2921
06:00	2806
07:00	2710
08:00	2856
09:00	3002
10:00	3020
11:00	3026
12:00	3030
13:00	3016
14:00	2901
15:00	2717
16:00	2796
17:00	2869
18:00	3374
19:00	3382
20:00	3373
21:00	3205
22:00	3015
23:00	2929
24:00	2869

Tabel 4.5.
Data Pembebanan Unit Thermal
PT. PLN Pembangkit Jawa Bali
Minggu 14 Maret 2004

JAM	Beban Sistem (MW)
01:00	2816
02:00	2678
03:00	2675
04:00	2694
05:00	2804
06:00	2611
07:00	2588
08:00	2746
09:00	2802
10:00	2816
11:00	2853
12:00	2789
13:00	2799
14:00	2657
15:00	2613
16:00	2709
17:00	2714
18:00	3255
19:00	3268
20:00	3269
21:00	2982
22:00	2876
23:00	2864
24:00	2822

4.6. Hasil Perhitungan dan Analisa Hasil Perhitungan *Economic Dispatch* Menggunakan Metode *Evolutionary Programming*.

4.6.1. Hasil Perhitungan PT. PLN Pembangkit Jawa Bali

Dari data pembebanan harian pada lampiran dapat dihitung biaya operasional tiap jamnya dengan memasukkan ke persamaan fungsi biaya bahan bakar. Sedangkan untuk hasil perhitungan beban dan biaya operasional perjamnya dapat dilihat pada tabel dibawah ini.

Tabel 4.6.
Hasil Perhitungan Biaya Operasional Per Jam
PT. PLN Pembangkit Jawa Bali
Rabu 10 Maret 2004

JAM	Beban Sistem (MW)	PT. PLN PJB (Rp)
01:00	3108	680.306.108
02:00	3024	651.394.473
03:00	2993	646.667.419
04:00	2974	644.171.861
05:00	2976	640.361.923
06:00	2916	646.587.755
07:00	2808	626.319.642
08:00	3202	713.721.328
09:00	3265	725.817.463
10:00	3281	728.989.485
11:00	3297	732.577.863
12:00	3220	720.949.264
13:00	3225	723.177.188
14:00	3226	720.370.936
15:00	3297	733.154.110
16:00	3372	747.933.301
17:00	3499	805.181.358
18:00	3600	832.153.449
19:00	3657	844.634.828
20:00	3642	842.417.383
21:00	3403	757.459.805
22:00	3388	754.703.180
23:00	3335	744.635.040
24:00	3316	740.803.704
TOTAL		17.404.488.866

Tabel 4.7.
Hasil Perhitungan Biaya Operasional Per Jam
PT. PLN Pembangkit Jawa Bali
Sabtu 13 Maret 2004

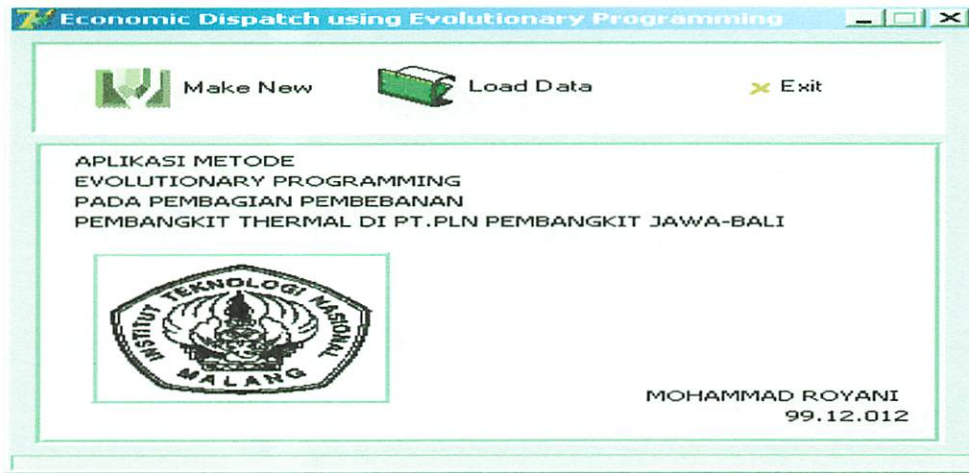
JAM	Beban Sistem (MW)	PT. PLN PJB (Rp)
01:00	2896	642.976.722
02:00	2864	627.840.757
03:00	2845	626.899.359
04:00	2866	629.878.888
05:00	2921	638.467.808
06:00	2806	622.986.844
07:00	2710	609.418.876
08:00	2856	629.093.132
09:00	3002	651.358.752
10:00	3020	655.538.620
11:00	3026	657.042.259
12:00	3030	658.662.629
13:00	3016	657.445.941
14:00	2901	637.045.376
15:00	2717	609.113.162
16:00	2796	622.239.104
17:00	2869	633.104.555
18:00	3374	755.125.576
19:00	3382	756.832.935
20:00	3373	755.769.182
21:00	3205	700.429.220
22:00	3015	657.757.376
23:00	2929	642.690.720
24:00	2869	632.703.376
TOTAL		15.710.421.169

Tabel 4.8.
Hasil Perhitungan Biaya Operasional Per Jam
PT. PLN Pembangkit Jawa Bali
Minggu 14 Maret 2004

JAM	Beban Sistem (MW)	PT. PLN PJB (Rp)
01:00	2816	618.251.323
02:00	2678	580.714.514
03:00	2675	580.742.086
04:00	2694	583.369.050
05:00	2804	601.290.761
06:00	2611	574.415.775
07:00	2588	570.967.006
08:00	2746	591.959.107
09:00	2802	601.814.600
10:00	2816	604.252.146
11:00	2853	611.074.946
12:00	2789	600.616.653
13:00	2799	595.203.725
14:00	2657	577.919.351
15:00	2613	570.955.908
16:00	2709	587.564.000
17:00	2714	588.687.073
18:00	3255	728.245.361
19:00	3268	730.147.988
20:00	3269	729.197.563
21:00	2982	639.723.642
22:00	2876	617.848.361
23:00	2864	615.562.714
24:00	2822	606.757.754
TOTAL		14.707.281.407

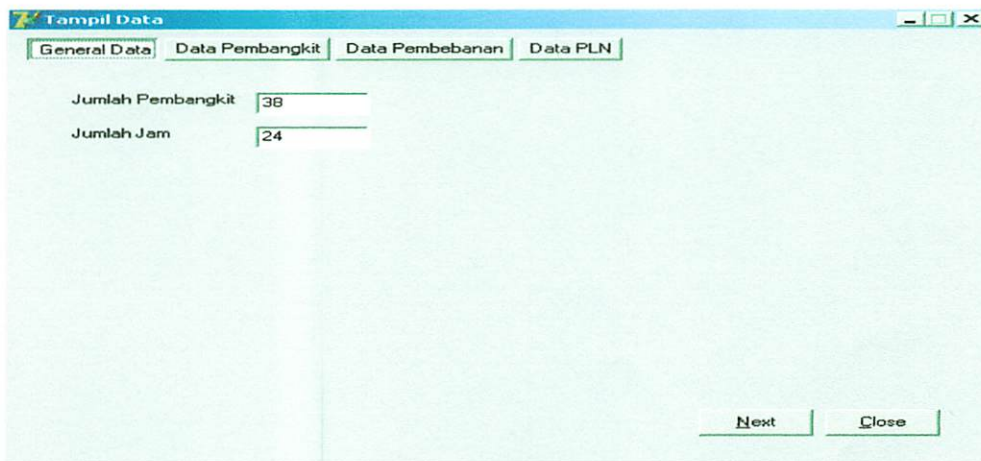
4.6.2. Hasil Perhitungan dengan Metode *Evolutionary Programming*

Hasil perhitungan dari program komputer yang telah dibuat ini adalah hasil perhitungan beban dan biaya operasional yang optimal, sedangkan tampilan utama dari program dapat dilihat pada gambar 4.1. dibawah ini.



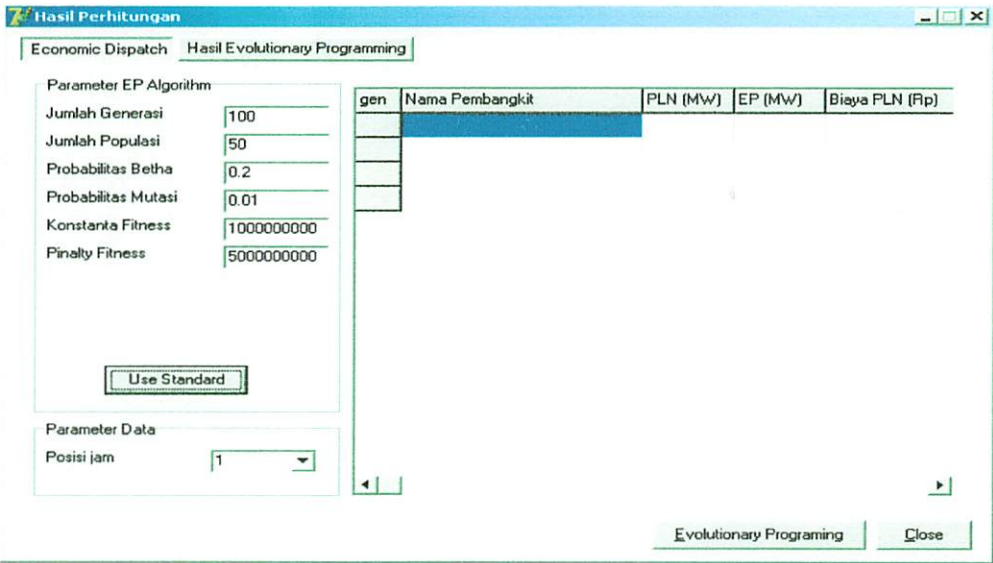
Gambar 4.1. Tampilan Program Utama

Kemudian setelah itu tekan tombol buka data untuk membuka file yang tersimpan

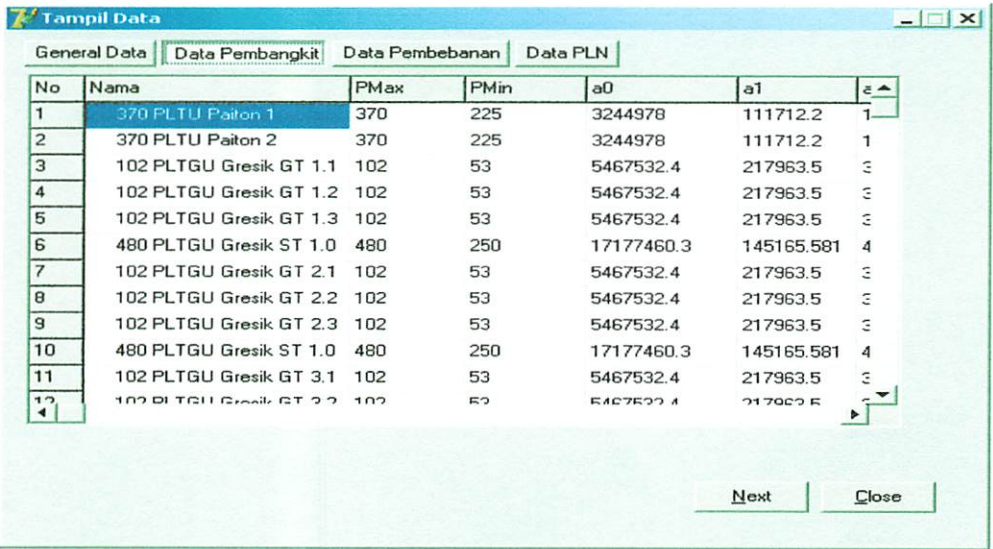


Gambar 4.2. Menampilkan Input Data

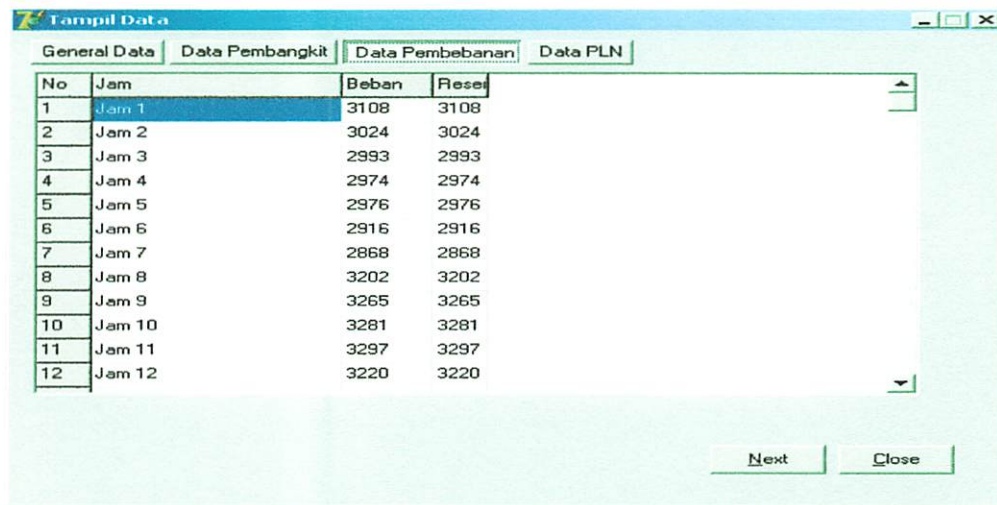
Setelah data yang ada diinputkan kedalam komputer maka parameter-parameter yang digunakan adalah sebagai berikut :



Gambar 4.3. Parameter *Evolutionary Programming*



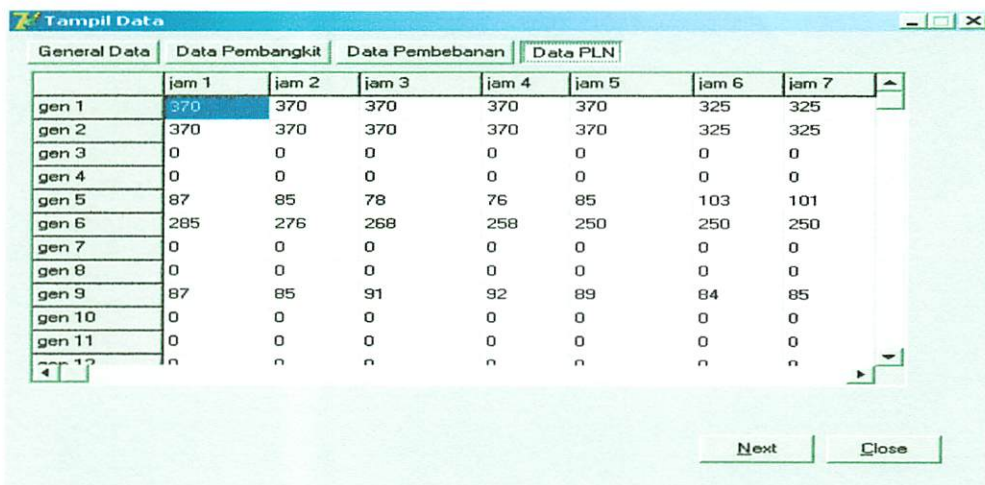
Gambar 4.4. Tampilan Data Pembangkit



No	Jam	Beban	Reser
1	Jam 1	3108	3108
2	Jam 2	3024	3024
3	Jam 3	2993	2993
4	Jam 4	2974	2974
5	Jam 5	2976	2976
6	Jam 6	2916	2916
7	Jam 7	2868	2868
8	Jam 8	3202	3202
9	Jam 9	3265	3265
10	Jam 10	3281	3281
11	Jam 11	3297	3297
12	Jam 12	3220	3220

Next Close

Gambar 4.5. Tampilan Data Pembebanan



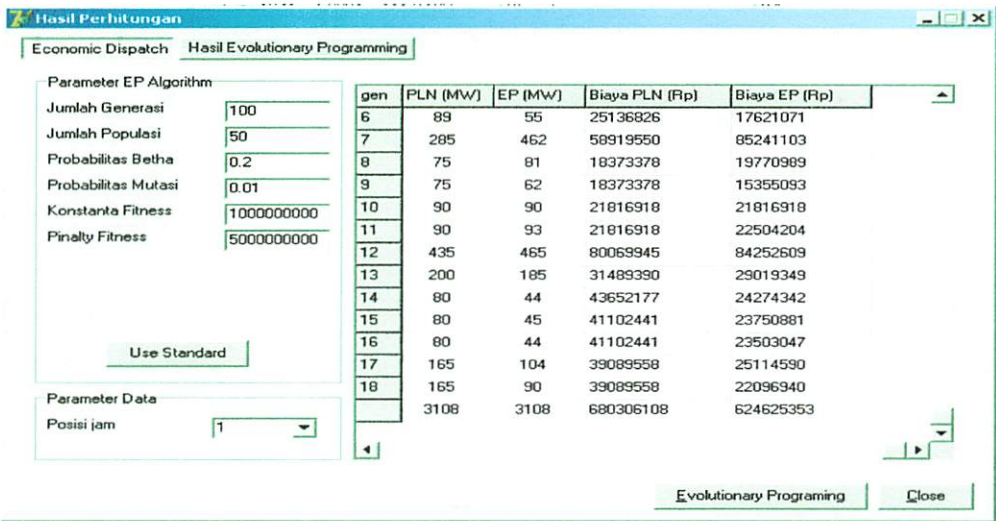
	jam 1	jam 2	jam 3	jam 4	jam 5	jam 6	jam 7
gen 1	370	370	370	370	370	325	325
gen 2	370	370	370	370	370	325	325
gen 3	0	0	0	0	0	0	0
gen 4	0	0	0	0	0	0	0
gen 5	87	85	78	76	85	103	101
gen 6	285	276	268	258	250	250	250
gen 7	0	0	0	0	0	0	0
gen 8	0	0	0	0	0	0	0
gen 9	87	85	91	92	89	84	85
gen 10	0	0	0	0	0	0	0
gen 11	0	0	0	0	0	0	0
gen 12	0	0	0	0	0	0	0

Next Close

Gambar 4.6. Tampilan Data PLN

Setelah menetapkan parameter tsrsebut maka komputasi dapat dilakukan, untuk menentukan kebutuhan beban yang disuplai dengan biaya yang minimum. Untuk proses komputasi digunakan software Borland Delphi versi 7.0 kemudian dieksekusi dengan menggunakan Komputer berspesifikasi Intel Pentium IV 2.8 GHz Ram 512 Mb. Perhitungan dimulai pada jam 01.00 dengan beban 3108 MW.

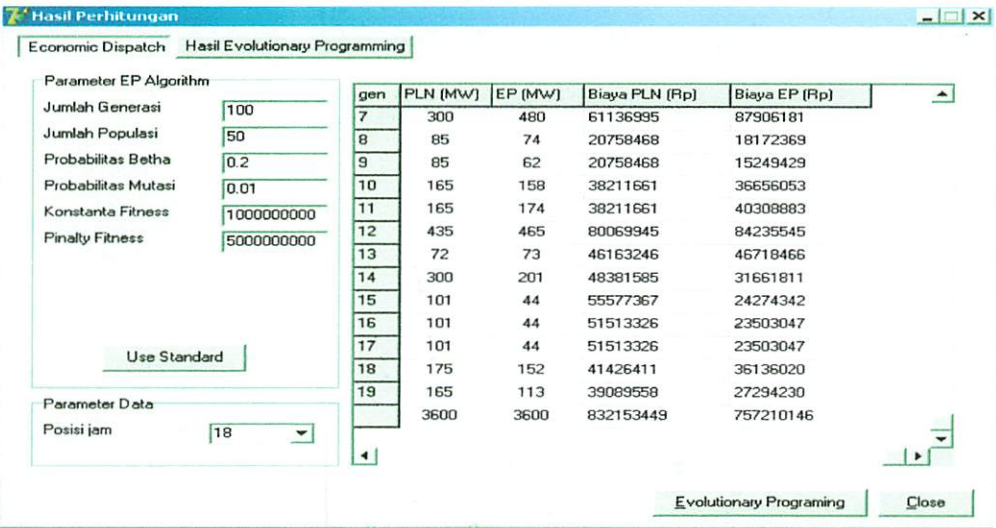
Komputasi pada jam 01.00 dengan beban 3108 MW.



Gambar 4.7. Hasil Optimasi pada jam 01.00.

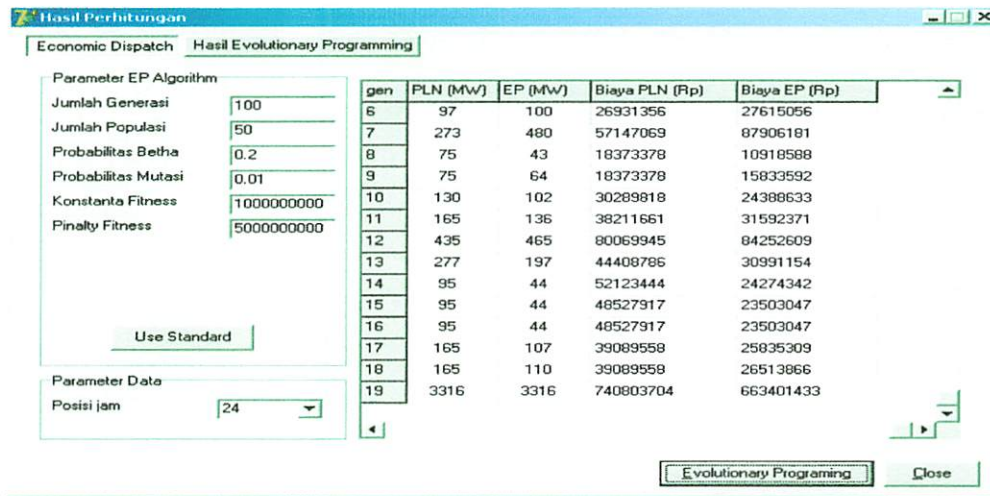
Dengan parameter dan cara yang sama dapat dicari pembangkit yang dapat melayani permintaan beban dengan biaya yang minimum.

Komputasi pada jam 18.00 dengan beban 3600 MW



Gambar 4.8. Hasil Optimasi pada jam 18.00.

Komputasi pada jam 24.00 dengan beban 3316 MW.



Gambar 4.9. Hasil Optimasi pada jam 24.00

4.6.3. Hasil Perhitungan PT. PLN Pembangkit Jawa Bali dengan Metode *Evolutionary Programming*.

Berikut ini adalah yang berisi hasil biaya pada PT. PLN Pembangkit Jawa Bali dengan metode *Evolutionary Programming*.

Tabel 4.9.
Hasil Perhitungan Menggunakan Metode *Evolutionary Programming*
Rabu 10 Maret 2004

JAM	Beban Sistem (MW)	Metode EP (Rp)
01:00	3108	624.625.353
02:00	3024	610.509.679
03:00	2993	606.523.972
04:00	2974	602.921.299
05:00	2976	603.727.480
06:00	2916	598.928.914
07:00	2808	588.425.487
08:00	3202	646.090.548
09:00	3265	657.342.310
10:00	3281	659.053.626
11:00	3297	663.614.674
12:00	3220	649.571.239
13:00	3225	649.587.909
14:00	3226	648.141.465
15:00	3297	663.614.674
16:00	3372	675.623.989
17:00	3499	735.566.426
18:00	3600	758.070.983
19:00	3657	771.596.010
20:00	3642	768.281.540
21:00	3403	682.182.201
22:00	3388	680.212.390
23:00	3335	672.703.022
24:00	3316	669.998.969
TOTAL		15.886.914.159

Tabel 4.10.
Hasil Perhitungan Menggunakan Metode *Evolutionary Programming*
Sabtu 13 Maret 2004

JAM	Beban Sistem (MW)	Metode EP (Rp)
01:00	2896	592.234.666
02:00	2864	591.382.127
03:00	2845	588.732.617
04:00	2866	584.789.490
05:00	2921	600.847.912
06:00	2806	578.089.497
07:00	2710	565.918.746
08:00	2856	588.991.922
09:00	3002	606.390.750
10:00	3020	610.377.069
11:00	3026	611.090.862
12:00	3030	611.618.478
13:00	3016	608.246.845
14:00	2901	596.460.220
15:00	2717	567.322.706
16:00	2796	575.190.881
17:00	2869	590.569.764
18:00	3374	677.708.776
19:00	3382	678.597.248
20:00	3373	676.509.634
21:00	3205	642.949.666
22:00	3015	609.844.208
23:00	2929	600.929.625
24:00	2869	590.569.764
TOTAL		14.545.363.473

Tabel 4.11.
Hasil Perhitungan Menggunakan Metode *Evolutionary Programming*
Minggu 14 Maret 2004

JAM	Beban Sistem (MW)	Metode EP (Rp)
01:00	2816	581.308.841
02:00	2678	561.154.609
03:00	2675	561.419.104
04:00	2694	563.401.918
05:00	2804	577.159.748
06:00	2611	550.857.862
07:00	2588	546.747.502
08:00	2746	573.485.245
09:00	2802	577.361.722
10:00	2816	581.308.841
11:00	2853	588.186.740
12:00	2789	575.988.995
13:00	2799	566.756.010
14:00	2657	556.116.224
15:00	2613	551.233.894
16:00	2709	565.689.752
17:00	2714	567.266.484
18:00	3255	654.084.776
19:00	3268	658.300.198
20:00	3269	655.498.056
21:00	2982	604.410.167
22:00	2876	586.960.579
23:00	2864	591.382.127
24:00	2822	581.472.030
TOTAL		13.977.551.424

4.6.4. Perbandingan Hasil Perhitungan PT. PLN PJB Dengan Metode *Evolutionary Programming*.

Berikut ini adalah perbandingan biaya operasional pada PT. PLN PJB dan metode *Evolutionary Programming*

Tabel 4.12.
Perbandingan Biaya Operasional Per Jam
PT. PLN Pembangkit Jawa Bali dengan Metode *Evolutionary Programming*
Rabu 10 Maret 2004

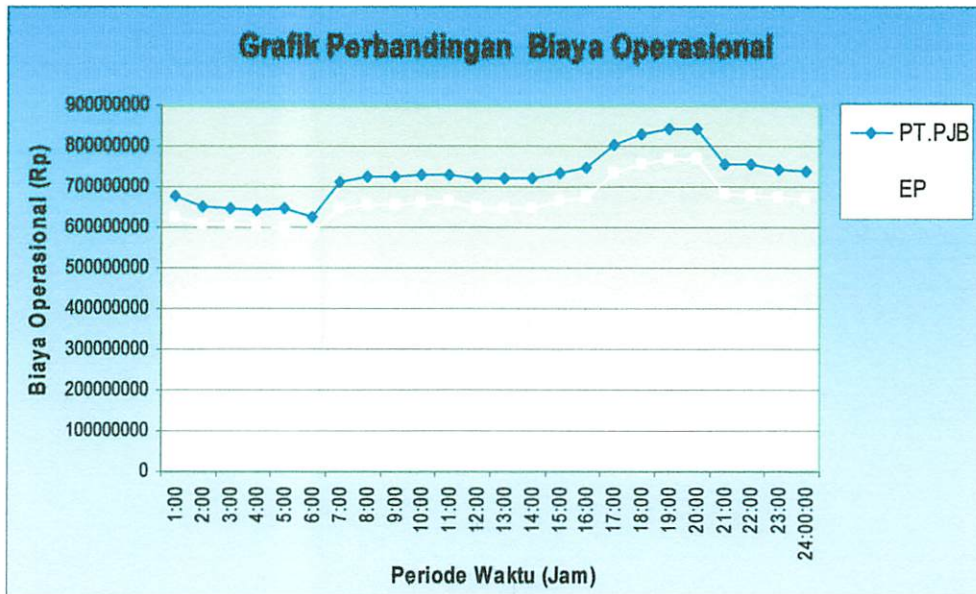
JAM	Beban Sistem (MW)	PT. PLN PJB (Rp)	Metode EP (Rp)	Penghematan (Rp)
01:00	3108	680.306.108	624.625.353	55.680.755
02:00	3024	651.394.473	610.509.679	40.884.794
03:00	2993	646.667.419	606.523.972	40.143.447
04:00	2974	644.171.861	602.921.299	41.250.562
05:00	2976	640.361.923	603.727.480	36.634.443
06:00	2916	646.587.755	598.928.914	47.658.841
07:00	2808	626.319.642	588.425.487	37.894.155
08:00	3202	713.721.328	646.090.548	67.630.780
09:00	3265	725.817.463	657.342.310	68.475.153
10:00	3281	728.989.485	659.053.626	69.935.859
11:00	3297	732.577.863	663.614.674	68.963.189
12:00	3220	720.949.264	649.571.239	71.378.025
13:00	3225	723.177.188	649.587.909	73.589.279
14:00	3226	720.370.936	648.141.465	72.229.471
15:00	3297	733.154.110	663.614.674	69.539.436
16:00	3372	747.933.301	675.623.989	72.309.312
17:00	3499	805.181.358	735.566.426	69.614.932
18:00	3600	832.153.449	758.070.983	74.082.466
19:00	3657	844.634.828	771.596.010	73.038.818
20:00	3642	842.417.383	768.281.540	74.135.843
21:00	3403	757.459.805	682.182.201	75.277.604
22:00	3388	754.703.180	680.212.390	74.490.790
23:00	3335	744.635.040	672.703.022	71.932.018
24:00	3316	740.803.704	669.998.969	70.804.735
TOTAL		17.404.488.866	15.886.914.159	1.517.574.707

Tabel 4.13.
Perbandingan Biaya Operasional Per Jam
PT. PLN Pembangkit Jawa Bali dengan Metode *Evolutionary Programming*
Sabtu 13 Maret 2004

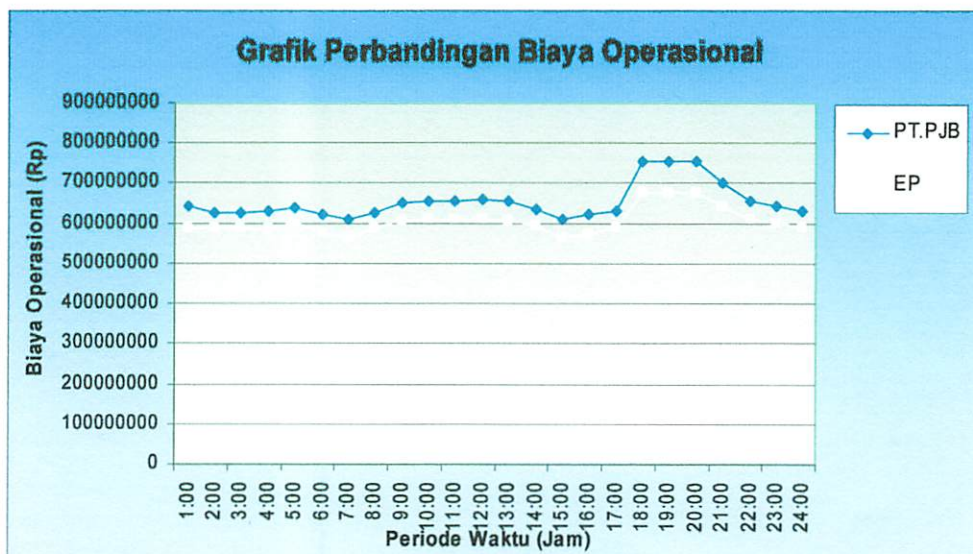
JAM	Beban Sistem (MW)	PT. PLN PJB (Rp)	Metode EP (Rp)	Penghematan (Rp)
01:00	2896	642.976.722	592.234.666	50.742.056
02:00	2864	627.840.757	591.382.127	36.458.630
03:00	2845	626.899.359	588.732.617	38.166.742
04:00	2866	629.878.888	584.789.490	45.089.398
05:00	2921	638.467.808	600.847.912	37.619.896
06:00	2806	622.986.844	578.089.497	44.897.347
07:00	2710	609.418.876	565.918.746	43.500.130
08:00	2856	629.093.132	588.991.922	40.101.210
09:00	3002	651.358.752	606.390.750	44.968.002
10:00	3020	655.538.620	610.377.069	45.161.551
11:00	3026	657.042.259	611.090.862	45.951.397
12:00	3030	658.662.629	611.618.478	47.044.151
13:00	3016	657.445.941	608.246.845	49.199.096
14:00	2901	637.045.376	596.460.220	40.585.156
15:00	2717	609.113.162	567.322.706	41.790.456
16:00	2796	622.239.104	575.190.881	47.048.223
17:00	2869	633.104.555	590.569.764	42.534.791
18:00	3374	755.125.576	677.708.776	77.416.800
19:00	3382	756.832.935	678.597.248	78.235.687
20:00	3373	755.769.182	676.509.634	79.259.548
21:00	3205	700.429.220	642.949.666	57.479.554
22:00	3015	657.757.376	609.844.208	47.913.168
23:00	2929	642.690.720	600.929.625	41.761.095
24:00	2869	632.703.376	590.569.764	42.133.612
TOTAL		15.710.421.169	14.545.363.473	1.165.057.696

Tabel 4.14.
Perbandingan Biaya Operasional Per Jam
PT. PLN Pembangkit Jawa Bali dengan Metode *Evolutionary Programming*
Minggu 14 Maret 2004

JAM	Beban Sistem (MW)	PT. PLN PJB (Rp)	Metode EP (Rp)	Penghematan (Rp)
01:00	2816	618.251.323	581.308.841	36.942.482
02:00	2678	580.714.514	561.154.609	19.559.905
03:00	2675	580.742.086	561.419.104	19.322.982
04:00	2694	583.369.050	563.401.918	19.967.132
05:00	2804	601.290.761	577.159.748	24.131.013
06:00	2611	574.415.775	550.857.862	23.557.913
07:00	2588	570.967.006	546.747.502	24.219.504
08:00	2746	591.959.107	573.485.245	18.473.862
09:00	2802	601.814.600	577.361.722	24.452.878
10:00	2816	604.252.146	581.308.841	22.943.305
11:00	2853	611.074.946	588.186.740	22.888.206
12:00	2789	600.616.653	575.988.995	24.627.658
13:00	2799	595.203.725	566.756.010	28.447.715
14:00	2657	577.919.351	556.116.224	21.803.127
15:00	2613	570.955.908	551.233.894	19.722.014
16:00	2709	587.564.000	565.689.752	21.874.248
17:00	2714	588.687.073	567.266.484	21.420.589
18:00	3255	728.245.361	654.084.776	74.160.585
19:00	3268	730.147.988	658.300.198	71.847.790
20:00	3269	729.197.563	655.498.056	73.699.507
21:00	2982	639.723.642	604.410.167	35.313.475
22:00	2876	617.848.361	586.960.579	30.887.782
23:00	2864	615.562.714	591.382.127	24.180.587
24:00	2822	606.757.754	581.472.030	25.285.724
TOTAL		14.707.281.407	13.977.551.424	729.729.983



Grafik 4.1.
Perbandingan Biaya Operasional Per jam
PT. PLN Pembangkit Jawa Bali dengan Metode *Evolutionary Programming*
Rabu 10 Maret 2004



Grafik 4.2.
Perbandingan Biaya Operasional Per jam
PT. PLN Pembangkit Jawa Bali dengan Metode *Evolutionary Programming*
Sabtu 13 Maret 2004

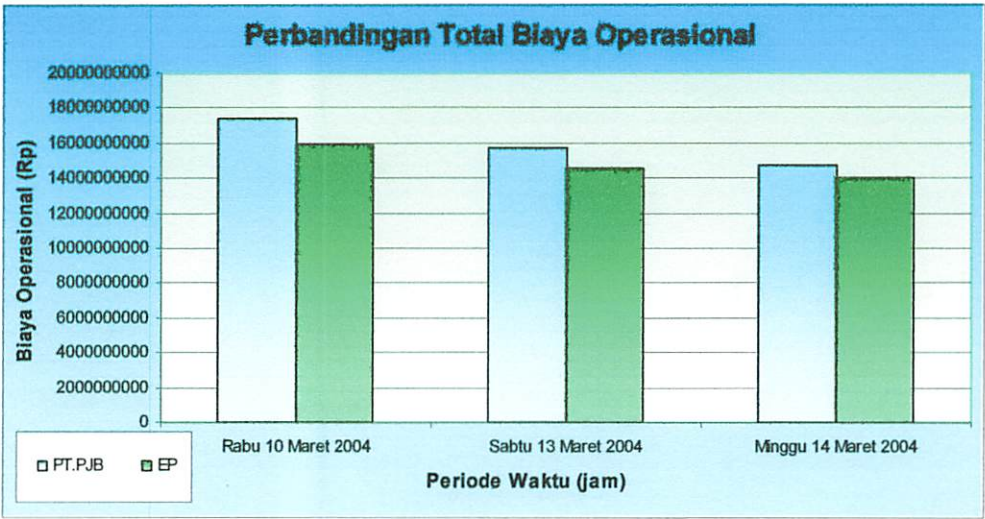


Grafik 4.3.
Perbandingan Biaya Operasional Per jam
PT. PLN Pembangkit Jawa Bali dengan Metode *Evolutionary Programming*
Minggu 14 Maret 2004

Pada tabel 4.15 di bawah ini menunjukkan bahwa dengan metode *Evolutionary Programming* terjadi pengurangan total biaya operasional operasi pembangkit dalam tiap periode 1 hari dalam 24 jam. Hal ini berarti bahwa pengurangan biaya operasi pembangkitan yang cukup besar bila dibandingkan dengan biaya operasi PT. PLN Pembangkit Jawa Bali. Perbandingan hasil optimalisasi total biaya operasi pembangkitan sebagai fungsi daya beban dari proses.

Tabel 4.15.
Perbandingan Total Biaya Operasional
PT. PLN Pembangkit Jawa Bali dengan Metode *Evolutionary Programming*

Hari Tanggal	PT. PLN PJB (Rupiah)	Metode EP (Rupiah)	Penghematan (Rupiah)	Penghematan (%)
Rabu 10/03/2004	17.404.488.866	15.886.914.159	1.517.574.707	8, 71
Sabtu 13/03/2004	15.710.421.169	14.545.363.473	1.165.057.696	7, 41
Minggu 14/03/2004	14.707.281.407	13.977.551.424	729.729.983	4, 96



Grafik 4.4.
Perbandingan Biaya Total Operasional
PT. PLN Pembangkit Jawa Bali dengan Metode *Evolutionary Programming*

Tabel 4.16
Pembebanan Unit Pembangkit Tiap Jam PT. PJB (Sebelum Optimalisasi)
Rabu 10 Maret 2004

No	Pembangkit	Pembebanan Perjam (MW)																								
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
1	PLTU Paiton 1	370	370	370	370	370	325	325	370	370	372	372	370	370	370	370	370	370	370	370	370	370	350	350	350	350
2	PLTU Paiton 2	370	370	370	370	370	325	325	370	370	374	374	350	350	372	370	370	370	370	370	370	370	350	350	350	350
3	PLTGU Gresik GT 1.3	87	85	78	76	85	103	101	84	88	102	103	102	102	90	85	99	103	105	108	108	105	103	100	98	
4	PLTGU Gresik ST 1.0	285	278	288	259	250	250	250	300	300	300	300	300	300	300	275	265	265	275	285	300	300	285	265	275	
5	PLTGU Gresik GT 2.3	87	85	91	92	89	84	85	86	88	85	88	97	98	87	98	85	84	105	115	125	125	105	105	101	
6	PLTGU Gresik GT 3.3	88	87	88	91	89	84	87	92	93	93	97	98	98	87	92	94	99	105	108	108	101	98	98	97	
7	PLTGU Gresik ST 3.0	285	288	285	252	250	250	250	275	300	300	300	300	300	275	275	290	300	300	300	285	277	273	273	273	
8	PLTU Gresik 1	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	85	85	75	75	75	75	
9	PLTU Gresik 2	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	85	85	85	75	75	75	75	
10	PLTU Gresik 3	90	90	90	90	90	90	90	135	165	165	165	165	165	165	165	165	165	165	165	165	155	134	130	130	
11	PLTU Gresik 4	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	110	140	155	165	165	165	165	165	165	
12	PLTGU M. Tawar GT 1.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	72	72	72	72	0	0	0	0	
13	PLTGU M. Tawar ST 1.0	435	410	410	410	425	410	410	435	435	435	435	435	435	435	435	435	435	435	435	435	435	435	435	435	
14	PLTGU M. Karang ST 1.0	200	200	200	200	200	200	200	200	200	200	200	185	200	200	265	275	285	300	320	320	300	285	277	277	
15	PLTU M. Karang 1	80	65	65	65	60	60	60	95	95	95	95	95	95	95	95	95	95	101	103	103	95	95	95	95	
16	PLTU M. Karang 2	80	65	65	65	60	60	60	95	95	95	95	95	95	95	95	95	95	101	103	103	95	95	95	95	
17	PLTU M. Karang 3	80	65	65	65	62	60	60	95	95	95	95	95	95	95	95	95	95	101	103	103	95	95	95	95	
18	PLTU M. Karang 4	165	165	165	165	165	165	165	165	165	165	165	165	165	165	165	165	165	175	175	175	165	165	165	165	
19	PLTU M. Karang 5	165	165	165	165	165	165	165	165	165	165	165	150	165	165	165	165	165	165	165	165	165	165	165	165	
TOTAL		3108	3024	2993	2974	2976	2918	2888	3202	3265	3281	3297	3220	3226	3228	3297	3372	3459	3600	3657	3642	3403	3358	3335	3316	

Tabel 4.17
Pembebanan Unit Pembangkit Tiap Jam PT. PJB (Setelah Optimalisasi)

No	Pembangkit	Pembebanan Perjam (MW)																								
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
1	PLTU Paiton 1	370	370	344	370	368	341	370	370	370	327	370	370	370	370	370	370	370	367	370	370	370	370	368	370	370
2	PLTU Paiton 2	331	366	354	367	370	351	370	370	370	367	370	367	370	362	369	370	370	370	365	370	370	370	370	370	370
3	PLTGU Gresik GT 1.3	59	53	55	55	53	55	63	56	71	80	74	69	56	53	63	63	92	66	102	102	102	70	65	84	81
4	PLTGU Gresik ST 1.0	472	420	443	351	322	390	310	454	480	395	480	392	488	480	480	480	480	478	476	480	480	468	408	428	405
5	PLTGU Gresik GT 2.3	56	66	68	66	62	55	54	88	88	81	71	57	69	94	53	92	83	70	96	102	102	91	83	65	65
6	PLTGU Gresik GT 3.3	55	55	64	66	64	70	53	68	88	69	80	55	92	68	65	96	84	73	102	101	93	81	83	74	74
7	PLTGU Gresik ST 3.0	462	437	436	432	460	468	363	480	480	480	480	480	429	456	480	480	480	480	480	480	480	480	480	480	480
8	PLTU Gresik 1	81	65	43	44	43	50	45	48	53	45	43	85	59	43	83	43	52	64	78	44	59	59	65	55	55
9	PLTU Gresik 2	62	49	44	44	43	43	51	53	60	53	55	84	46	74	49	65	61	71	76	80	72	85	81	81	81
10	PLTU Gresik 3	90	117	100	126	93	90	96	116	125	144	132	122	112	94	145	156	153	174	166	171	111	168	136	126	126
11	PLTU Gresik 4	83	111	90	90	90	90	94	90	94	111	148	129	114	126	138	134	171	175	175	175	168	144	147	151	175
12	PLTGU M. Tawar GT 1.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	72	72	74	76	0	0	0	0	
13	PLTGU M. Tawar ST 1.0	465	370	445	465	459	411	465	464	465	465	462	465	447	463	465	465	484	465	465	465	456	465	448	454	
14	PLTGU M. Karang ST 1.0	185	188	180	181	200	194	197	169	183	202	202	202	199	202	202	202	182	202	186	202	202	202	202	202	202
15	PLTU M. Karang 1	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44
16	PLTU M. Karang 2	45	44	44	44	44	44	44	44	44	45	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44
17	PLTU M. Karang 3	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	46	44	44	44	44	44	44	44
18	PLTU M. Karang 4	104	91	105	95	97	96	93	101	95	165	111	111	112	103	92	105	134	161	165	165	165	165	128	149	165
19	PLTU M. Karang 5	90	124	90	90	100	90	92	131	102	161	90	97	140	99	90	90	128	148	128	145	117	148	91	91	91
TOTAL		3108	3024	2993	2974	2976	2918	2888	3202	3268	3281	3297	3220	3226	3228	3297	3372	3459	3600	3657	3642	3403	3368	3335	3316	3316

Tabel 4.18
Pembebanan Unit Pembangkit Tiap Jam PT. PJB (Sebelum Optimalisasi)
Sabtu 13 Maret 2004

No	Pembangkit	Pembebanan Perjam (MW)																							
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	PLTU Paton 1	370	350	350	350	370	316	316	360	370	370	370	370	360	360	350	350	350	370	370	370	370	370	370	370
2	PLTU Paton 2	370	350	350	350	370	316	316	360	370	370	370	370	360	360	350	350	350	370	370	370	370	370	370	370
3	PLTGU Gresik GT 1.3	85	78	78	77	87	102	100	85	90	103	105	110	112	90	102	99	101	115	115	118	105	102	99	88
4	PLTGU Gresik ST 1.0	155	165	131	148	165	180	180	161	177	175	181	182	182	100	153	175	175	185	188	188	175	174	182	150
5	PLTGU Gresik GT 2.3	87	91	90	92	82	83	85	85	87	83	95	102	103	83	83	87	89	86	105	108	105	97	82	87
6	PLTGU Gresik GT 3.3	88	85	95	92	82	84	88	84	85	86	100	100	98	87	89	86	86	105	108	105	97	82	85	85
7	PLTGU Gresik ST 3.0	134	147	142	148	159	158	139	132	155	154	147	157	155	156	156	157	167	162	163	162	156	156	148	135
8	PLTU Gresik 1	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	88	89	89	89	72	73	72
9	PLTU Gresik 2	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	88	88	88	87	72	73	73
10	PLTU Gresik 3	120	122	121	121	121	121	121	121	121	120	121	121	121	121	121	120	119	172	175	174	114	114	115	114
11	PLTU Gresik 4	121	120	120	121	120	120	120	121	121	120	120	120	121	121	122	121	125	179	180	181	180	122	120	120
12	PLTGU M. Tawar ST 1.0	420	450	422	423	410	415	350	375	438	438	438	422	421	432	325	325	350	445	445	435	415	410	400	385
13	PLTGU M. Karang ST 1.0	335	335	365	365	350	320	315	388	385	387	385	377	387	385	316	325	345	390	390	390	382	382	320	320
14	PLTU M. Karang 1	75	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	90	90	90	90	65	65	65
15	PLTU M. Karang 2	75	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	90	90	90	90	65	65	65
16	PLTU M. Karang 3	75	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	90	90	90	90	65	65	65
17	PLTU M. Karang 4	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	150	150	150	150	150	150	150	150
18	PLTU M. Karang 5	150	150	150	150	150	150	150	155	167	165	165	167	167	165	150	150	150	170	170	170	162	162	165	165
TOTAL		2898	2884	2945	2935	2921	2908	2719	2838	3002	3020	3028	3030	3016	2901	2717	2795	2893	3374	3382	3373	3205	3016	2928	2859

Sumber : Data Pembebanan PT PLN PJB, JL. Kedhangan Baru NO. 11, Surabaya 60231

Tabel 4.19
Pembebanan Unit Pembangkit Tiap Jam PT. PJB (Setelah Optimalisasi)
Sabtu, 13 Maret 2004

No	Pembangkit	Pembebanan Perjam (MW)																							
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	PLTU Paton 1	370	370	368	370	368	370	370	370	370	370	370	370	370	370	370	370	370	370	370	370	370	370	370	370
2	PLTU Paton 2	368	342	370	368	370	370	288	370	368	343	368	367	370	297	370	370	369	370	369	370	370	370	370	370
3	PLTGU Gresik GT 1.3	58	55	54	53	55	59	61	54	63	63	68	63	53	57	53	54	56	62	54	54	63	55	56	56
4	PLTGU Gresik ST 1.0	302	328	293	275	313	310	383	318	387	385	415	375	414	285	350	287	322	468	478	480	450	385	319	322
5	PLTGU Gresik GT 2.3	53	53	59	53	67	68	62	63	59	61	53	53	53	59	55	53	53	62	79	100	70	54	65	63
6	PLTGU Gresik GT 3.3	68	78	54	53	89	53	77	68	63	68	58	66	63	62	63	66	65	84	63	78	65	67	75	65
7	PLTGU Gresik ST 3.0	480	510	322	484	321	373	308	286	480	468	355	480	453	334	317	372	302	480	480	480	480	331	302	302
8	PLTU Gresik 1	47	73	72	49	44	46	43	59	48	45	53	45	43	50	55	43	77	60	74	43	51	57	65	77
9	PLTU Gresik 2	48	46	58	44	71	51	46	52	43	43	43	44	49	56	67	59	59	76	80	72	48	71	59	59
10	PLTU Gresik 3	80	80	120	102	103	80	90	80	80	82	101	90	80	90	80	90	80	120	135	128	104	98	90	88
11	PLTU Gresik 4	90	83	90	82	82	82	98	80	91	80	104	103	103	112	106	85	80	122	119	132	111	83	83	80
12	PLTGU M. Tawar ST 1.0	368	484	485	485	402	485	393	384	485	388	429	485	482	485	374	405	484	483	483	455	388	483	483	484
13	PLTGU M. Karang ST 1.0	193	177	180	202	188	178	198	166	188	202	178	182	198	185	198	198	188	198	202	202	179	191	194	180
14	PLTU M. Karang 1	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44
15	PLTU M. Karang 2	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44
16	PLTU M. Karang 3	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44
17	PLTU M. Karang 4	100	103	103	118	128	97	110	110	99	88	102	98	95	91	99	90	80	151	132	145	132	90	118	90
18	PLTU M. Karang 5	133	150	110	90	112	128	95	102	127	133	121	133	122	139	90	113	150	137	151	132	124	121	115	150
TOTAL		2898	2894	2945	2888	2921	2896	2710	2838	3002	3020	3028	3030	3016	2901	2717	2795	2893	3374	3382	3373	3205	3016	2928	2859

Tabel 4.20
 Pembebanan Unit Pembangkit Tiap Jam PT. PJB (Sebelum Optimalisasi)
 Minggu 14 Maret 2004

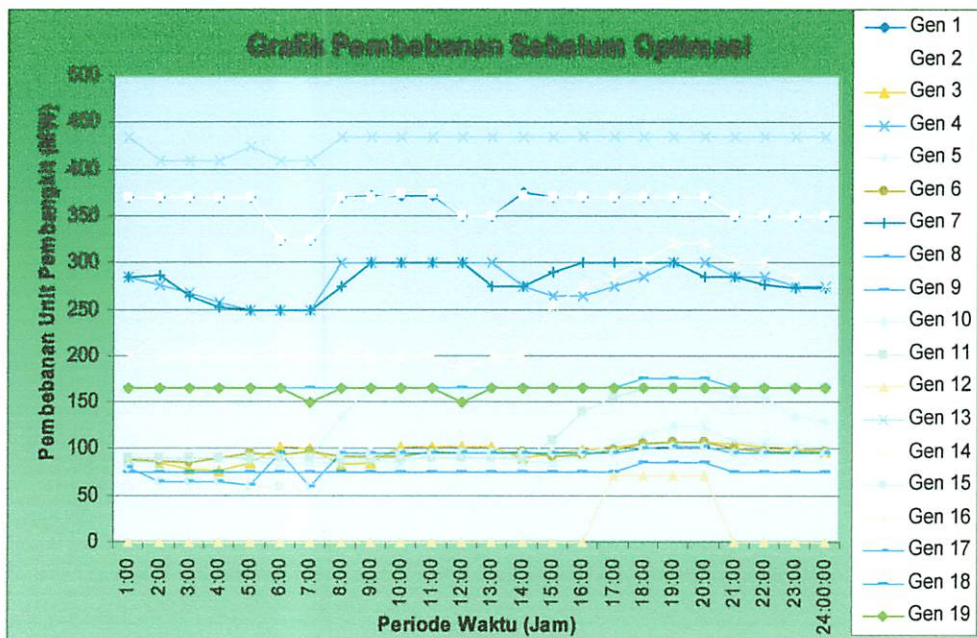
No	Pembangkit	Pembangunan Perjam (MW)																							
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	PLTU Paiton 1	370	350	350	350	370	316	316	380	360	360	360	360	360	360	360	360	370	370	370	370	370	370	370	370
2	PLTU Paiton 2	370	350	350	350	370	316	316	380	360	360	360	360	360	360	360	360	370	370	370	370	370	370	370	370
3	PLTGU Gresik GT 1.3	80	73	73	72	82	87	85	80	81	87	100	105	103	85	87	94	96	100	102	102	100	97	94	83
4	PLTGU Gresik GT 1.0	140	130	128	143	150	175	175	158	156	166	178	148	190	141	143	170	172	178	178	175	175	178	162	152
5	PLTGU Gresik GT 2.3	88	88	90	92	85	82	88	81	83	86	93	87	87	82	81	97	98	117	118	117	115	89	95	82
6	PLTGU Gresik GT 3.3	89	88	94	91	89	82	89	82	91	93	86	93	87	89	88	98	103	102	103	95	82	87	83	83
7	PLTGU Gresik ST 3.0	129	142	137	141	153	153	131	127	134	143	142	146	148	127	121	152	150	134	147	162	142	127	131	132
8	PLTU Gresik 1	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
9	PLTU Gresik 2	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
10	PLTU Gresik 3	165	150	150	150	150	150	150	165	165	165	165	150	150	150	150	150	150	165	165	165	165	165	165	160
11	PLTU Gresik 4	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90
12	PLTGU M. Tawar ST 1.0	410	400	400	400	400	350	350	400	425	425	425	400	360	360	350	350	350	425	425	425	400	400	400	400
13	PLTGU M. Karang ST 1.0	350	325	325	325	350	300	300	350	350	350	350	350	350	325	282	300	300	380	380	380	350	350	350	350
14	PLTU M. Karang 1	65	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
15	PLTU M. Karang 2	65	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
16	PLTU M. Karang 3	65	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
17	PLTU M. Karang 4	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	150	150	150	150	150	150	150
18	PLTU M. Karang 5	150	150	150	150	150	150	150	155	165	165	165	150	150	150	150	150	150	165	165	165	160	150	150	150
TOTAL		2816	2678	2676	2694	2804	2611	2663	2746	2692	2616	2683	2769	2748	2667	2613	2709	2714	3266	3268	3269	2982	2676	2664	2622

Sumber : Data Pembebanan PT PLN PJB, J.L. Kedirang Bana NO. 11, Surabaya 60231

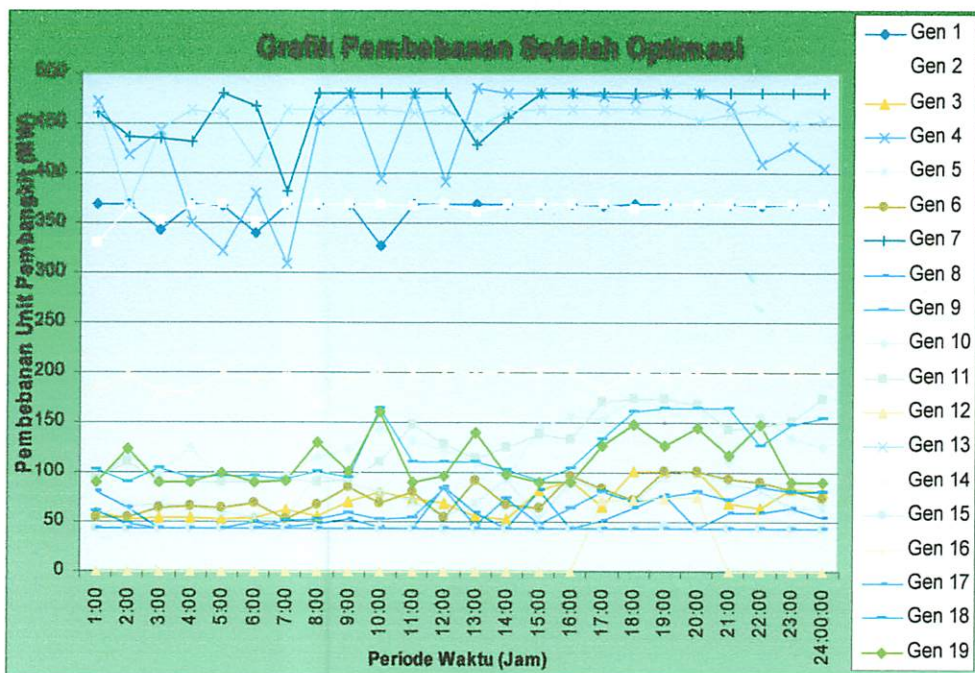
Tabel 4.21
 Pembebanan Unit Pembangkit Tiap Jam PT. PJB (Setelah Optimalisasi)
 Minggu, 14 Maret 2004

No	Pembangkit	Pembangunan Perjam (MW)																								
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
1	PLTU Paiton 1	369	370	316	370	367	314	293	370	370	369	370	370	369	367	351	370	370	370	370	370	369	370	370	370	370
2	PLTU Paiton 2	350	329	370	277	349	359	351	273	370	350	370	359	370	332	321	274	320	370	370	370	370	370	342	369	369
3	PLTGU Gresik GT 1.3	53	53	60	53	54	82	78	53	60	53	64	53	53	53	91	53	68	54	54	54	53	64	55	59	
4	PLTGU Gresik GT 1.0	285	342	286	364	318	280	320	369	307	285	283	272	322	342	307	364	384	443	405	405	389	274	328	283	
5	PLTGU Gresik GT 2.3	55	54	53	55	53	53	53	78	53	53	59	67	69	53	53	53	54	91	91	53	53	68	53	69	
6	PLTGU Gresik GT 3.3	58	55	74	71	81	88	53	73	53	59	67	69	53	71	53	58	71	58	91	91	66	53	78	54	
7	PLTGU Gresik ST 3.0	311	288	344	309	371	278	261	268	307	311	333	329	339	314	288	366	303	471	480	480	478	459	310	316	
8	PLTU Gresik 1	73	55	55	55	48	50	44	47	47	73	64	43	54	50	60	60	81	43	43	43	43	73	43	43	
9	PLTU Gresik 2	54	50	65	46	52	44	44	43	43	52	43	47	53	56	43	50	80	80	80	80	50	46	51		
10	PLTU Gresik 3	105	90	83	94	91	90	90	115	112	105	90	90	91	90	90	90	90	115	115	90	90	90	90	90	
11	PLTU Gresik 4	82	109	94	97	94	100	97	116	106	83	83	97	90	80	80	97	98	104	118	119	91	85	83		
12	PLTGU M. Tawar ST 1.0	485	349	366	399	403	382	390	405	402	485	484	447	398	340	398	359	340	482	485	485	378	409	484	485	
13	PLTGU M. Karang ST 1.0	201	171	170	171	200	182	182	202	192	202	193	189	189	176	186	181	201	202	202	188	203	177	170	170	
14	PLTU M. Karang 1	44	44	44	45	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	
15	PLTU M. Karang 2	44	44	45	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	
16	PLTU M. Karang 3	44	44	47	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	
17	PLTU M. Karang 4	102	125	91	110	90	129	112	93	91	102	102	96	96	95	112	98	108	129	128	126	94	90	103	102	
18	PLTU M. Karang 5	111	90	80	90	121	90	90	80	131	111	150	132	108	90	90	90	90	155	155	155	133	122	150	147	
TOTAL		2816	2678	2676	2694	2804	2611	2663	2746	2692	2616	2683	2769	2748	2667	2613	2709	2714	3266	3268	3269	2982	2676	2664	2622	

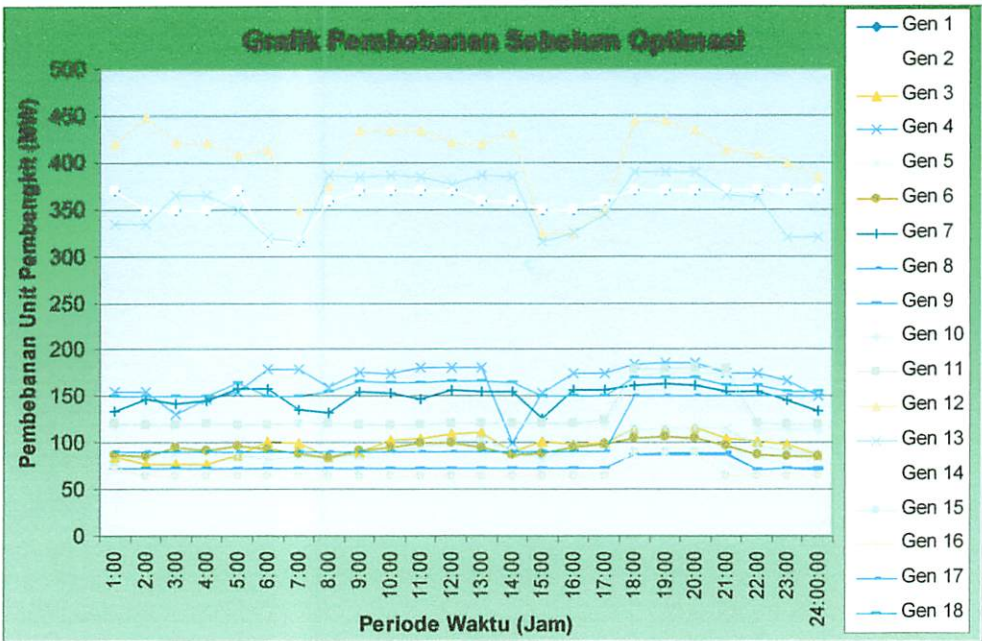
Grafik 4.5
Grafik Pembebanan Unit Pembangkit PT. PLN Pembangkit Jawa Bali
Sebelum Optimasi
Rabu 10 Maret 2004



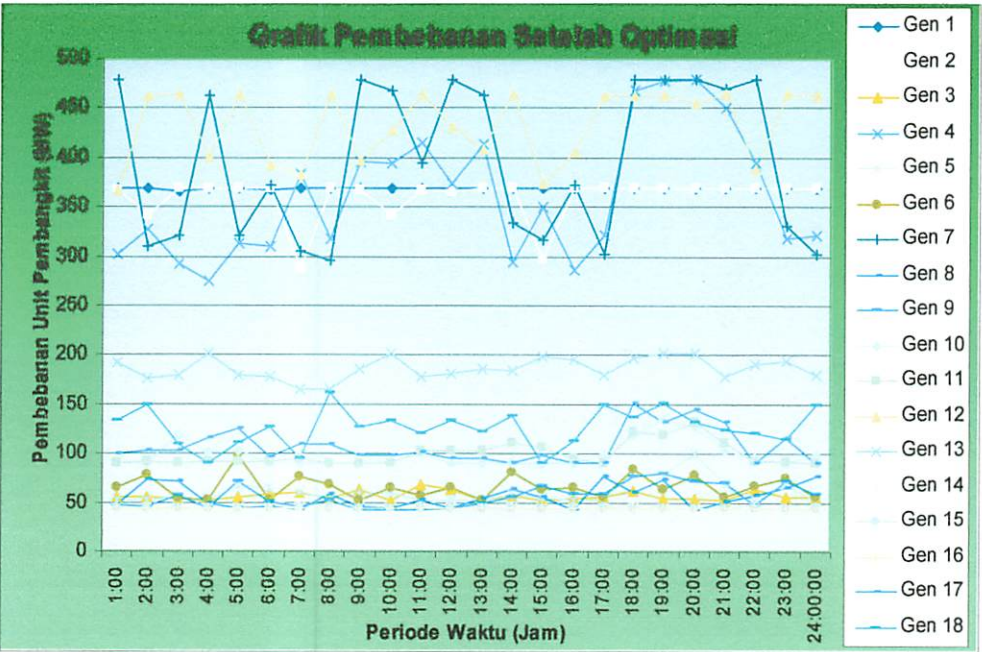
Grafik 4.6
Grafik Pembebanan Unit Pembangkit PT. PLN Pembangkit Jawa Bali
Setelah Optimasi
Rabu 10 Maret 2004



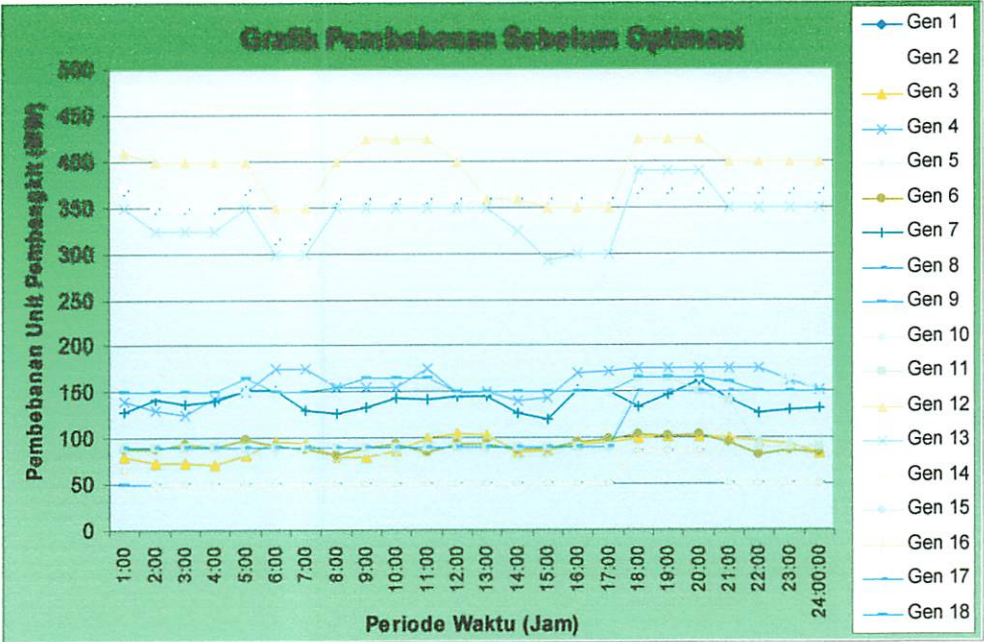
Grafik 4.7
Grafik Pembebanan Unit Pembangkit PT. PLN Pembangkit Jawa Bali
Sebelum Optimasi
Sabtu 13 Maret 2004



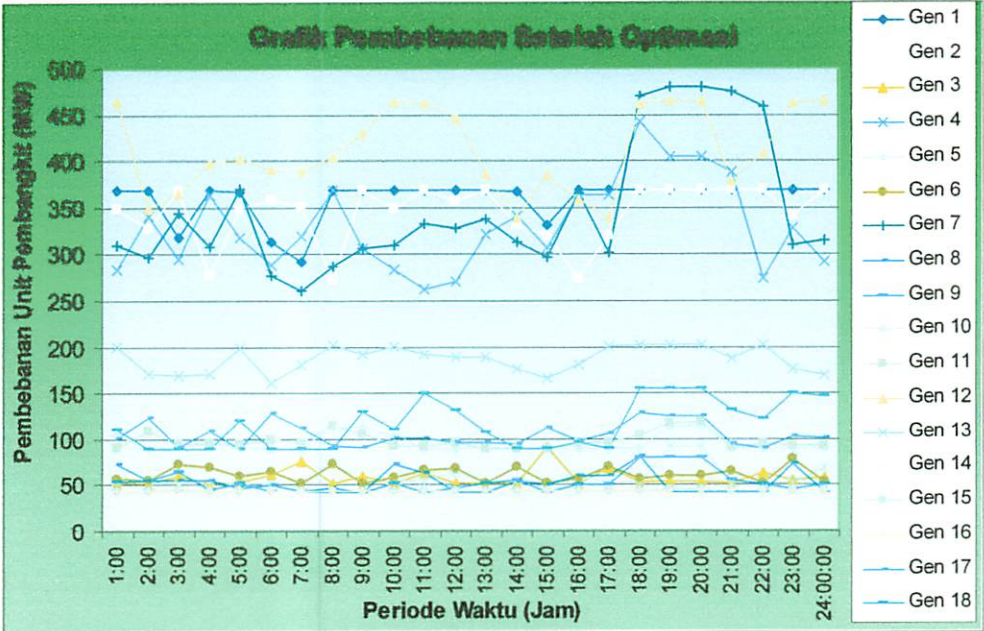
Grafik 4.8
Grafik Pembebanan Unit Pembangkit PT. PLN Pembangkit Jawa Bali
Setelah Optimasi
Sabtu 13 Maret 2004



Grafik 4.9
Grafik Pembebanan Unit Pembangkit PT. PLN Pembangkit Jawa Bali
Sebelum Optimasi
Minggu 14 Maret 2004



Grafik 4.10
Grafik Pembebanan Unit Pembangkit PT. PLN Pembangkit Jawa Bali
Setelah Optimasi
Minggu 14 Maret 2004



Tabel 4.22
Data Penjadwalan Unit Pembangkit
Rabu, 10 Maret 2004

No	Pembangkit	Jam																							
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	PLTU Paiton 1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	PLTU Paiton 2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	PLTGU Gresik GT 1.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	PLTGU Gresik GT 1.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	PLTGU Gresik GT 1.3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	PLTGU Gresik ST 1.0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	PLTGU Gresik GT 2.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	PLTGU Gresik GT 2.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	PLTGU Gresik GT 2.3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	PLTGU Gresik ST 2.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	PLTGU Gresik GT 3.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	PLTGU Gresik GT 3.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	PLTGU Gresik GT 3.3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14	PLTGU Gresik ST 3.0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
15	PLTU Gresik 1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
16	PLTU Gresik 2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
17	PLTU Gresik 3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
18	PLTU Gresik 4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
19	PLTG gresik 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	PLTG gresik 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	PLTG gresik 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	PLTG Gilitimur 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	PLTG Gilitimur 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	PLTGU M. Karang GT 1.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	PLTGU M. Karang GT 1.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	PLTGU M. Karang GT 1.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	PLTGU M. Karang ST 1.0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
28	PLTGU M. Tawar GT 1.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	PLTGU M. Tawar GT 1.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	PLTGU M. Tawar GT 1.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	PLTGU M. Tawar GT 2.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	PLTGU M. Tawar GT 2.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	PLTGU M. Tawar ST 1.0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
34	PLTU M. Karang 1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
35	PLTU M. Karang 2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
36	PLTU M. Karang 3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
37	PLTU M. Karang 4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
38	PLTU M. Karang 5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Tabel 4.23
Data Penjadwalan Unit Pembangkit
Sabtu, 13 Maret 2004

No	Pembangkit	Jam																							
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	PLTU Paiton 1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	PLTU Paiton 2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	PLTGU Gresik GT 1.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	PLTGU Gresik GT 1.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	PLTGU Gresik GT 1.3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	PLTGU Gresik ST 1.0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	PLTGU Gresik GT 2.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	PLTGU Gresik GT 2.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	PLTGU Gresik GT 2.3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	PLTGU Gresik ST 2.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	PLTGU Gresik GT 3.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	PLTGU Gresik GT 3.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	PLTGU Gresik GT 3.3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14	PLTGU Gresik ST 3.0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
15	PLTU Gresik 1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
16	PLTU Gresik 2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
17	PLTU Gresik 3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
18	PLTU Gresik 4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
19	PLTG gresik 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	PLTG gresik 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	PLTG gresik 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	PLTG Gilitmur 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	PLTG Gilitmur 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	PLTGU M. Karang GT 1.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	PLTGU M. Karang GT 1.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	PLTGU M. Karang GT 1.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	PLTGU M. Karang ST 1.0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
28	PLTGU M. Tawar GT 1.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	PLTGU M. Tawar GT 1.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	PLTGU M. Tawar GT 1.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	PLTGU M. Tawar GT 2.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	PLTGU M. Tawar GT 2.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	PLTGU M. Tawar ST 1.0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
34	PLTU M. Karang 1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
35	PLTU M. Karang 2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
36	PLTU M. Karang 3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
37	PLTU M. Karang 4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
38	PLTU M. Karang 5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Tabel 4.24
Data Penjadwalan Unit Pembangkit
Minggu 14 Maret 2004

No	Pembangkit	Jam																							
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	PLTU Païton 1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	PLTU Païton 2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	PLTGU Gresik GT 1.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	PLTGU Gresik GT 1.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	PLTGU Gresik GT 1.3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	PLTGU Gresik ST 1.0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	PLTGU Gresik GT 2.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	PLTGU Gresik GT 2.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	PLTGU Gresik GT 2.3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	PLTGU Gresik ST 2.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	PLTGU Gresik GT 3.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	PLTGU Gresik GT 3.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	PLTGU Gresik GT 3.3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14	PLTGU Gresik ST 3.0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
15	PLTU Gresik 1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
16	PLTU Gresik 2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
17	PLTU Gresik 3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
18	PLTU Gresik 4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
19	PLTG Gresik 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	PLTG Gresik 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	PLTG Gresik 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	PLTG Gilitimur 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	PLTG Gilitimur 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	PLTGU M. Karang GT 1.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	PLTGU M. Karang GT 1.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	PLTGU M. Karang GT 1.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	PLTGU M. Karang ST 1.0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
28	PLTGU M. Tawar GT 1.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	PLTGU M. Tawar GT 1.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	PLTGU M. Tawar GT 1.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	PLTGU M. Tawar GT 2.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	PLTGU M. Tawar GT 2.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	PLTGU M. Tawar ST 1.0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
34	PLTU M. Karang 1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
35	PLTU M. Karang 2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
36	PLTU M. Karang 3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
37	PLTU M. Karang 4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
38	PLTU M. Karang 5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

BAB V

KESIMPULAN DAN SARAN

5. 1. Kesimpulan

Setelah dilakukan analisa perhitungan *Economic Dispatch* dengan metode *Evolutionary Programming* pada PT. PLN Pembangkit Jawa Bali selama tiga hari yaitu hari Rabu tanggal 10 Maret 2004, hari Sabtu tanggal 13 Maret 2004 dan hari Minggu tanggal 14 Maret 2004 maka dapat diambil suatu kesimpulan sebagai berikut :

1. Proses Metode *Evolutionary Programming* memberikan analisa penyelesaian yang cukup efektif dalam mengoptimalkan pembagian pembebanan (*Economic Dispatch*) yang dapat dilihat pada tabel 4. 16 sampai tabel 4. 21 dan sekaligus penghematan biaya total operasional PT. PLN Pembangkit Jawa Bali.
2. Dengan mengoptimalkan batas daya maksimum dan batas daya minimum pembangkit sehingga setelah dilakukan *Economic Dispatch* dengan metode *Evolutionary Programming*, maka pembangkit tetap dapat memenuhi kebutuhan beban dengan biaya yang lebih minimal dibandingkan dengan sebelum dilakukannya optimasi.
3. Dengan menerapkan *Economic Dispatch* dengan metode *Evolutionary Programming* maka didapatkan biaya operasional pembangkit yang lebih optimum pada hari Rabu, tanggal 10 Maret 2004 sebelum optimasi sebesar Rp.17.404.488.866 dan sesudah optimasi menjadi sebesar Rp. 15.886.914.159. Jadi selisih biaya operasional sebelum dan sesudah dilakukannya optimasi adalah sebesar Rp. 1.517.574.707 (dapat dioptimasi sekitar 8,71 %). Pada hari

Sabtu, tanggal 13 Maret 2004 sebelum optimasi sebesar Rp. 15.710.421.169 dan sesudah optimasi menjadi sebesar Rp. 14.545.363.473. Jadi selisih biaya operasional sebelum dan sesudah dilakukannya optimasi adalah sebesar Rp. 1.165.057.696 (dapat dioptimasi sekitar 7,41 %). Pada hari Minggu, tanggal 14 Maret 2004 sebelum optimasi sebesar Rp. 14.707.281.407 dan sesudah optimasi menjadi sebesar Rp. 13.977.551.424. Jadi selisih biaya operasional sebelum dan sesudah dilakukannya optimasi adalah Rp. 729.729.983 (dapat dioptimasi sekitar 4,96 %).

5. 2. Saran

Berdasarkan kesimpulan di atas, maka dapat diajukan saran yang berhubungan dengan skripsi ini :

1. Dalam *Economic Dispatch* terdapat beberapa metode yang dapat diterapkan atau dipakai pada proses penganalisaan nya, maka dari itu diharapkan pada studi selanjutnya untuk mencoba melakukan penggabungan dari beberapa metode sekaligus karena tidak tertutup kemungkinan adanya suatu penggabungan dua atau lebih metode yang digunakan dalam proses analisa nya, agar dapat mencapai hasil yang lebih sempurna dan maksimal.
2. Untuk mencapai hasil yang lebih maksimal dimasa-masa mendatang, maka diharapkan pula pada studi analisa selanjutnya agar dapat meng update data-data terbaru dari pihak PT. PLN Pembangkit Jawa Bali sebagai obyek penelitian.

3. *Economic Dispatch* dengan metode *Evolutionary Programming* ini seyogyanya dapat diterapkan pada sistem tenaga listrik di PT. PLN Pembangkit Jawa Bali yang sudah terinterkoneksi seperti pada sistem tenaga listrik se-Jawa Bali karena hasil perhitungan dengan metode ini memberikan biaya pembangkitan total yang lebih murah di bandingkan dengan biaya total PT. PLN Pembangkit Jawa Bali.

DAFTAR PUSTAKA

- [1]. Djiteng Marsudi . Ir., “ *Operasi Sistem Tenaga Listrik* “, ISTN 1990.
- [2]. P.A Haviryanupap, H. Kita, E. Tanaka dan J. Hasegawa, “ *Evolutionary Programming based Dynamic Economic Dispatch* “.
- [3]. William D. Stevenson, Jr. “ *Analisa Sistem Tenaga Listrik* “ , Edisi Keempat, Erlangga, Jakarta, 1993.
- [4]. A.J Wood dan B. F. Wollenberg, “ *Power Generation Operation, and Control* “ Second Edition, John Wiley & Sons. 1996.
- [5]. <http://zooland.alife.org/hhg2ec/>, *What's Evolutionary Programming (EP) ?*

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 : MOHAMMAD ROYANI
N I M : 99. 12. 012
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Energi Listrik
Judul Skripsi : APLIKASI METODE *EVOLUTIONARY PROGRAMMING* PADA PEMBAGIAN PEMBEBANAN PEMBANGKIT THERMAL DI PT. PLN PEMBANGKIT JAWA BALI

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

Hari : Sabtu
Tanggal : 18 Maret 2006
Dengan Nilai : 80,5 (A) *Bm*

Panitia Ujian Skripsi



Ketua

(Ir. Mochtar Asroni, MSME.)
NIP. Y. 101 810 0036

Sekretaris

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

Anggota Majelis Penguji

Penguji Pertama

(Ir. H. Taufik Hidayat, MT)
NIP. 108 700 151

Penguji Kedua

(Ir.H. Almizan Abdullah MSEE.)
NIP. Y. 103 900 0208



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

PERSETUJUAN PERBAIKAN SKRIPSI

Dari Hasil Ujian Skripsi Jurusan Teknik Elektro Jenjang Strata Satu (S-1)
Yang Diselenggarakan Pada :

Hari : Sabtu
Tanggal : 18 Maret 2006

Telah Dilakukan Perbaikan Skripsi Oleh :

Nama : MOHAMMAD ROYANI
NIM : 99. 12. 012
Jurusan : Teknik Elektro
Konsentrasi : Teknik Energi Listrik
Judul Skripsi : APLIKASI METODE *EVOLUTIONARY PROGRAMMING*
PADA PEMBAGIAN PEMBEBANAN PEMBANGKIT
THERMAL DI PT. PLN PEMBANGKIT JAWA BALI

Perbaikan Meliputi :

No	Materi Perbaikan	Ket
1	Harus ada iterasi pada flowchart 3.4.4 (<i>Fitness</i>)	si
2	Perlu tampilan penjadwalan unit	si
3	Sinkronkan antara kesimpulan dan tujuan	si
4	Tambahkan uji validasi dan errornya	si
5	Lampirkan proses training dari jumlah generasi, populasi dan probabilitas mutasi	si
6	Tambahkan penjelasan probabilitas betha (hal 24)	si

Diperiksa dan Disetujui Oleh :
Anggota Majelis Penguji

Penguji Pertama

Ir. H. Taufik Hidayat, MT.
NIP. 108 700 151

Penguji Kedua

Ir. H. Almizan Abdullah, MSEE
NIP. Y. 103 9000 208

Mengetahui,
Dosen Pembimbing

Ir. I Made Wartana, MT
NIP. 131 991 182



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

LEMBAR BIMBINGAN SKRIPSI

Nama : MOHAMMAD ROYANI
N I M : 99. 12. 012
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Energi Listrik
Judul Skripsi : APLIKASI METODE *EVOLUTIONARY PROGRAMMING* PADA PEMBAGIAN PEMBEBANAN PEMBANGKIT THERMAL DI PT. PLN PEMBANGKIT JAWA BALI

Tanggal Mengajukan Skripsi : 11 Desember 2004
Tanggal Menyelesaikan Skripsi : 18 Maret 2006
Dosen Pembimbing : Ir. I Made Wartana, MT
Telah Dievaluasi Dengan Nilai : 85 (Delapan Puluh Lima) *sm*

Mengetahui,
Ketua Jurusan Teknik Elektro S-1

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

Diperiksa dan Disetujui,
Dosen Pembimbing

Ir. I Made Wartana, MT
NIP. 131 991 182



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

FORMULIR BIMBINGAN SKRIPSI

Nama : MOHAMMAD ROYANI
 N I M : 99. 12. 012
 Masa Bimbingan : 07 Februari 2005 s/d 07 Agustus 2005
 Judul Skripsi : APLIKASI METODE EVOLUTIONARY PROGRAMMING PADA PEMBAGIAN PEMBEBANAN PEMBANGKIT THERMAL DI PT. PLN PEMBANGKITAN JAWA BALI

No.	Tanggal	Uraian	Paraf Pembimbing
1.	2 Juni '05	- Data up to date - Pelajari EP & EP	2/
2.	7 Juli '05	- Pelajari Flocut & EP kontinuitas, EP, EP	2/
3.	26 Juli 2005	- Pelajari load & beban	2/
4.	15 Juli 2005	- Uraikan masalah & solusi EP. bagaimana pengujian & validasi	2/2/
5.	22 Jan 2006	Isenpasi & simulasi tugas, buat model & log	2/2/
6.	24 Jan 2006	kekeliruan, rumus, model, tugas dan kesimpulan diperlukan, metode & data	2/2/
7.	26 Jan 2006	Revisi buat soal & pembahasan 2 kasus, kesimpulan, dan solusi program, metode & FI	2/2/
8.	13. Feb 2006	Penggunaan etas. optimasi	2/
9.	15 Feb '06	Ada extra survey	2/
10.			

Malang, 2005
 Dosen Pembimbing

Ir. I Made Wartana, MT
 NIP. 131. 991. 182



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

FORMULIR BIMBINGAN SKRIPSI

Nama : MOHAMMAD ROYANI
N I M : 99. 12. 012
Masa Bimbingan : 07 Maret 2006 s/d 07 Agustus 2006
Judul Skripsi : APLIKASI METODE EVOLUTIONARY
PROGRAMMING PADA PEMBAGIAN
PEMBEBANAN PEMBANGKIT THERMAL
DI PT. PLN PEMBANGKITAN JAWA BALI

No	Tanggal	Uraian	Paraf Pembimbing
1.	2 - 02 - 06	Revisi bab 1 & 2 Ade Satrio	ad
2.			
3.			
4.			
5.			
6.			
7.			
8.			
9.			
10.			

Malang, 2006
Dosen Pembimbing

Ir. I Made Wartana, MT
NIP. 131. 991. 182

Form. S-4b

CAMINO A V 2

Pengujian Program Dengan Validasi Pada Jurnal

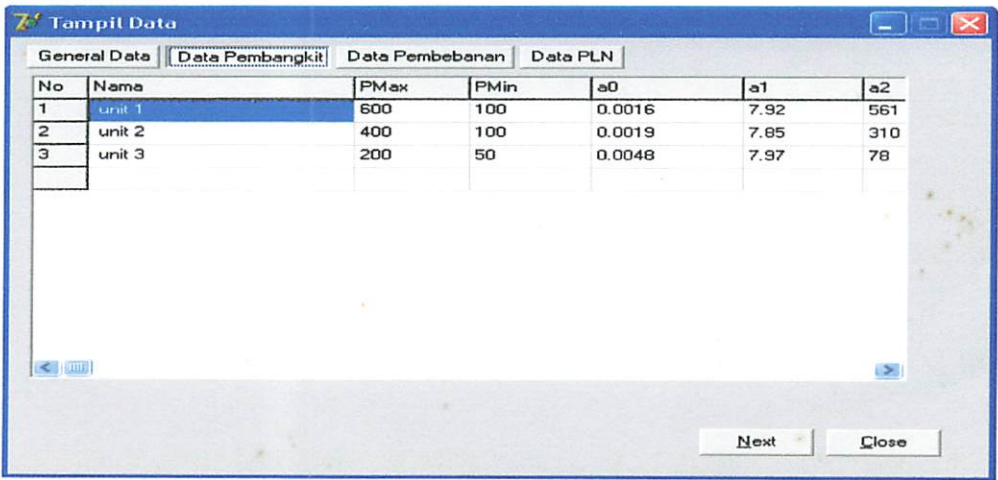
Sebelum dilakukan perhitungan menggunakan metode EP terlebih dahulu dilakukan uji validasi untuk melihat kelayakan dari program tersebut. Selanjutnya untuk pengujian validasi berpedoman pada data jurnal P.A Haviryanupap, H. Kita, E. Tanaka dan J. Hasegawa, “ *Evolutionary Programming based Economic Dispatch* “. Adapun data-data yang dipakai adalah sebagai berikut:

Tabel 1
Data system 3 Pembangkit termal untuk uji validasi

Unit	a	b	c	Pmax	Pmin
1	0.001562	7.92	561	600	100
2	0.00194	7.85	310	400	100
3	0.00482	7.97	78	200	50

Dengan data permintaan beban 850 MW

Sumber: P.A Haviryanupap, H. Kita, E. Tanaka dan J. Hasegawa, “ *Evolutionary Programming based Economic Dispatch* “



Gambar 1. Data pembangkit untuk uji validasi

No	Jam	Beban	Reser
1	Jam 1	850	850

Gambar 2. Data pembebanan untuk uji validasi

gen	PLN (MW)	EP (MW)	Biaya PLN (Rp)	Biaya EP (Rp)
1	393	250	86698395	35064480
2	335	400	34716815	49603140
3	122	200	1166727	3121594
	850	850	122581938	87789214

Gambar 3. Hasil validasi

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 Rp.87.789.214 yang mendekati perhitungan dengan jurnal P.A Haviryanupap, H. Kita, E. Tanaka dan J. Hasegawa, “*Evolutionary Programming based Economic Dispatch* “. yaitu Rp.87.618.150 dengan error 0,19% sehingga program ini valid untuk digunakan.

LAMPIRAN

HASIL TRAINING PROGRAM DARI BEBERAPA INPUTAN PARAMETER YANG BERBEDA

Jumlah Generasi
Jumlah Populasi
Probabilitas Beta
Probabilitas Mutasi

JUMLAH GENERASI
Gambar 1.
Parameter Jumlah Generasi 150
Hasil Tidak Mengalami Perubahan

gen	PLN (MW)	EP (MW)	Biaya PLN (Rp)	Biaya EP (Rp)
6	89	55	25136826	17621071
7	285	473	58919550	86916634
8	75	75	18373378	18311965
9	75	54	18373378	13379047
10	90	90	21816918	21816918
11	90	90	21816918	21824450
12	435	465	80069945	84252609
13	200	180	31489390	26201077
14	80	44	43652177	24274342
15	80	44	41102441	23503047
16	80	44	41102441	23503047
17	165	99	39089558	23985881
18	165	90	39089558	22096940
	3108	3108	680306108	624625353

Gambar 2.
Parameter Jumlah Generasi 100
Hasil Yang Digunakan

gen	PLN (MW)	EP (MW)	Biaya PLN (Rp)	Biaya EP (Rp)
6	89	55	25136826	17621071
7	285	462	58919550	85241103
8	75	81	18373378	19770989
9	75	62	18373378	15355093
10	90	90	21816918	21816918
11	90	93	21816918	22504204
12	435	465	80069945	84252609
13	200	185	31489390	29019349
14	80	44	43652177	24274342
15	80	45	41102441	23750681
16	80	44	41102441	23503047
17	165	104	39089558	25114590
18	165	90	39089558	22096940
	3108	3108	680306108	624625353

Gambar 3.
Parameter Jumlah Generasi 80
Hasil Lebih Mahal

gen	PLN (MW)	EP (MW)	Biaya PLN (Rp)	Biaya EP (Rp)
6	89	79	25136826	22987854
7	285	445	58919550	82650464
8	75	47	18373378	11860980
9	75	68	18373378	16814529
10	90	92	21816918	22314137
11	90	101	21816918	24063138
12	435	446	80069945	81580922
13	200	186	31489390	29207963
14	80	44	43652177	24274342
15	80	45	41102441	23750681
16	80	44	41102441	23503047
17	165	94	39089558	22933929
18	165	140	39089558	33218206
	3108	3108	680306108	628258155

JUMLAH POPULASI
Gambar 4.
Parameter Jumlah Populasi 80
Hasil Lebih Mahal

gen	PLN (MW)	EP (MW)	Biaya PLN (Rp)	Biaya EP (Rp)
6	89	53	25136826	17166959
7	285	423	58919550	79405820
8	75	43	18373378	10918588
9	75	70	18373378	17138245
10	90	109	21816918	25839441
11	90	117	21816918	27571220
12	435	440	80069945	80794184
13	200	176	31489390	27584921
14	80	44	43652177	24274342
15	80	45	41102441	23746486
16	80	44	41102441	23503047
17	165	103	39089558	24928771
18	165	109	39089558	26356688
	3108	3108	680306108	625431367

Gambar 5.
Parameter Jumlah Populasi 50
Hasil Yang Digunakan

gen	PLN (MW)	EP (MW)	Biaya PLN (Rp)	Biaya EP (Rp)
6	89	55	25136826	17621071
7	285	462	58919550	85241103
8	75	81	18373378	19770989
9	75	62	18373378	15355093
10	90	90	21816918	21816918
11	90	93	21816918	22504204
12	435	465	80069945	84252609
13	200	185	31489390	29019349
14	80	44	43652177	24274342
15	80	45	41102441	23750881
16	80	44	41102441	23503047
17	165	104	39089558	25114590
18	165	90	39089558	22096940
	3108	3108	680306108	624625353

Gambar 6.
Parameter Jumlah Populasi 30
Hasil Lebih Mahal

gen	PLN (MW)	EP (MW)	Biaya PLN (Rp)	Biaya EP (Rp)
6	89	71	25136826	21060914
7	285	433	58919550	80953082
8	75	63	18373378	15581010
9	75	43	18373378	10918588
10	90	101	21816918	24125060
11	90	124	21816918	28946728
12	435	465	80069945	84252609
13	200	176	31489390	27506836
14	80	44	43652177	24274342
15	80	44	41102441	23503047
16	80	44	41102441	23503047
17	165	109	39089558	26391543
18	165	109	39089558	26304454
	3108	3108	680306108	626305092

PROBABILITAS BETHA
Gambar 7.
Parameter Probabilitas Betha 0.3
Hasil Lebih Mahal

Parameter EP Algorithm

Jumlah Generasi	100
Jumlah Populasi	50
Probabilitas Betha	0.3
Probabilitas Mutasi	0.01
Konstanta Fitness	1000000000
Penalty Fitness	5000000000

Parameter Data
Posisi jam: 1

Evolutionary Programming [Close]

gen	PLN (MW)	EP (MW)	Biaya PLN (Rp)	Biaya EP (Rp)
6	89	60	25136826	10777524
7	285	469	58919550	86328087
8	75	59	18373378	14722203
9	75	47	18373378	11930928
10	90	90	21816918	21816918
11	90	104	21816918	24650397
12	435	427	80069945	78914106
13	200	172	31489390	26859251
14	80	44	43652177	24274342
15	80	44	41102441	23503047
16	80	44	41102441	23503047
17	165	152	39089558	36172455
18	165	90	39089558	22096940
	3108	3108	680306108	624785239

Gambar 8.
Parameter Probabilitas Betha 0.2
Hasil Yang Digunakan

Parameter EP Algorithm

Jumlah Generasi	100
Jumlah Populasi	50
Probabilitas Betha	0.2
Probabilitas Mutasi	0.01
Konstanta Fitness	1000000000
Penalty Fitness	5000000000

Parameter Data
Posisi jam: 1

Evolutionary Programming [Close]

gen	PLN (MW)	EP (MW)	Biaya PLN (Rp)	Biaya EP (Rp)
6	89	55	25136826	17621071
7	285	462	58919550	85241103
8	75	81	18373378	19770989
9	75	62	18373378	15355093
10	90	90	21816918	21816918
11	90	93	21816918	22504204
12	435	465	80069945	84252609
13	200	105	31489390	29019349
14	80	44	43652177	24274342
15	80	45	41102441	23750881
16	80	44	41102441	23503047
17	165	104	39089558	25114590
18	165	90	39089558	22096940
	3108	3108	680306108	624625353

Gambar 9.
Parameter Probabilitas Betha 0.1
Hasil Lebih Mahal

Parameter EP Algorithm

Jumlah Generasi	100
Jumlah Populasi	50
Probabilitas Betha	0.1
Probabilitas Mutasi	0.01
Konstanta Fitness	1000000000
Penalty Fitness	5000000000

Parameter Data
Posisi jam: 1

Evolutionary Programming [Close]

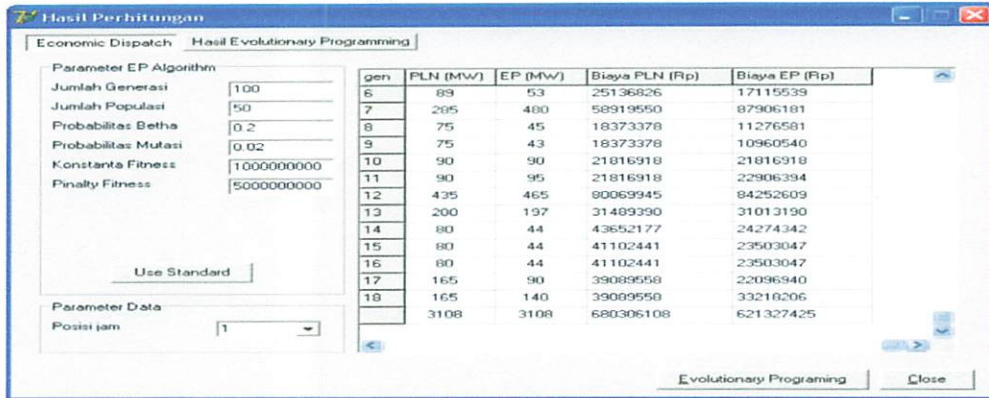
gen	PLN (MW)	EP (MW)	Biaya PLN (Rp)	Biaya EP (Rp)
6	89	71	25136826	21224087
7	285	446	58919550	82789065
8	75	52	18373378	13060970
9	75	43	18373378	11017419
10	90	90	21816918	21816918
11	90	114	21816918	26816326
12	435	450	80069945	82141097
13	200	176	31489390	27544067
14	80	44	43652177	24274342
15	80	44	41102441	23503047
16	80	44	41102441	23503047
17	165	131	39089558	31335475
18	165	118	39089558	28367279
	3108	3108	680306108	628433014

PROBABILITAS MUTASI

Gambar 10.

Parameter Probabilitas Mutasi 0.02

Terlalu Rendah, Sehingga Gen-Gen Yang Berpotensi Tidak Dicoba



Parameter EP Algorithm

Jumlah Generasi: 100
Jumlah Populasi: 50
Probabilitas Beta: 0.2
Probabilitas Mutasi: 0.02
Konstanta Fitness: 1000000000
Penalty Fitness: 5000000000

Use Standard

Parameter Data
Posisi jam: 1

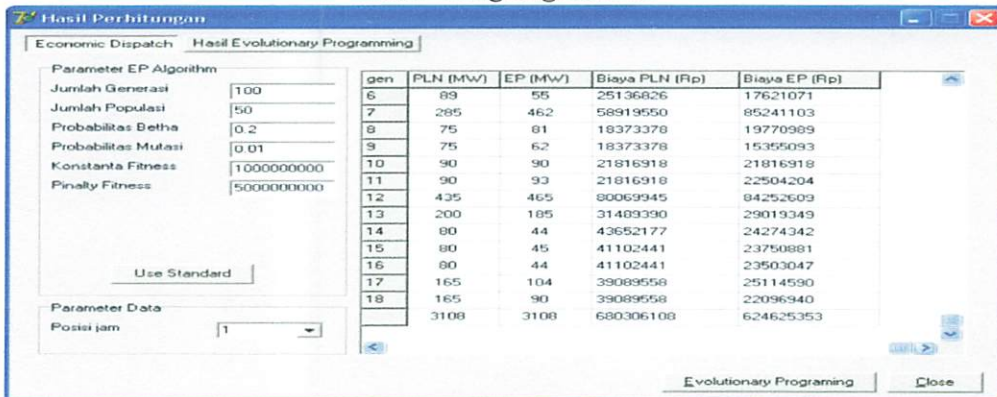
gen	PLN (MW)	EP (MW)	Biaya PLN (Rp)	Biaya EP (Rp)
6	89	53	25136826	17115539
7	285	480	58919550	87906181
8	75	45	18373378	11276581
9	75	43	18373378	10960540
10	90	90	21816918	21816918
11	90	95	21816918	22906394
12	435	465	90069945	84252609
13	200	197	31489390	31013190
14	80	44	43652177	24274342
15	80	44	41102441	23503047
16	80	44	41102441	23503047
17	165	90	39089558	22096940
18	165	140	39089558	33218206
	3108	3108	680306108	621327425

Evolutionary Programming Close

Gambar 11.

Parameter Probabilitas Mutasi 0.01

Hasil Yang Digunakan



Parameter EP Algorithm

Jumlah Generasi: 100
Jumlah Populasi: 50
Probabilitas Beta: 0.2
Probabilitas Mutasi: 0.01
Konstanta Fitness: 1000000000
Penalty Fitness: 5000000000

Use Standard

Parameter Data
Posisi jam: 1

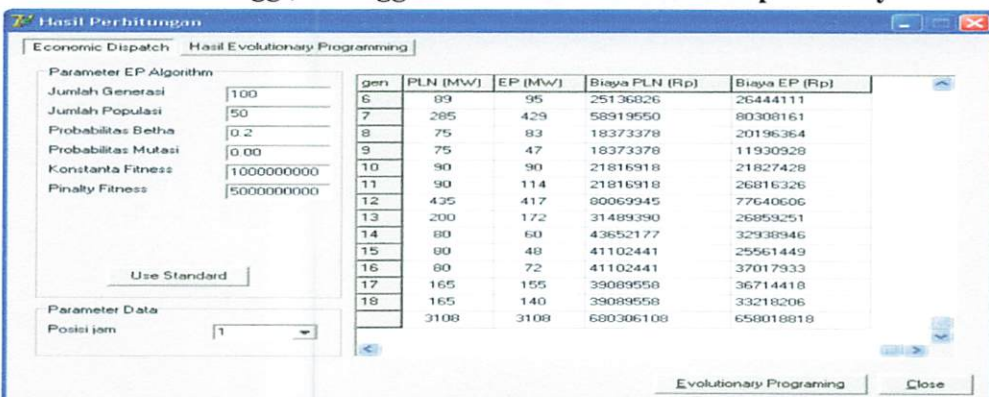
gen	PLN (MW)	EP (MW)	Biaya PLN (Rp)	Biaya EP (Rp)
6	89	58	25136826	17621071
7	285	462	58919550	85241103
8	75	81	18373378	19770989
9	75	62	18373378	15355093
10	90	90	21816918	21816918
11	90	93	21816918	22504204
12	435	465	90069945	84252609
13	200	185	31489390	29019349
14	80	44	43652177	24274342
15	80	45	41102441	23750881
16	80	44	41102441	23503047
17	165	104	39089558	25114590
18	165	90	39089558	22096940
	3108	3108	680306108	624625353

Evolutionary Programming Close

Gambar 12.

Parameter Probabilitas Mutasi 0.00

Terlalu Tinggi, Sehingga Keturunan Semakin Mirip Induknya



Parameter EP Algorithm

Jumlah Generasi: 100
Jumlah Populasi: 50
Probabilitas Beta: 0.2
Probabilitas Mutasi: 0.00
Konstanta Fitness: 1000000000
Penalty Fitness: 5000000000

Use Standard

Parameter Data
Posisi jam: 1

gen	PLN (MW)	EP (MW)	Biaya PLN (Rp)	Biaya EP (Rp)
6	89	95	25136826	26444111
7	285	429	58919550	80308161
8	75	83	18373378	20196364
9	75	47	18373378	11930928
10	90	90	21816918	21827428
11	90	114	21816918	26816326
12	435	417	90069945	77640606
13	200	172	31489390	26859251
14	80	60	43652177	32938946
15	80	48	41102441	25561449
16	80	72	41102441	37017933
17	165	155	39089558	36714418
18	165	140	39089558	33218206
	3108	3108	680306108	658018818

Evolutionary Programming Close



DATA PENAWARAN
PT PLN PEMBANGKITAN JAWA BALI
AGUSTUS 2002

No.	NAMA PEMBANGKIT	KAPASITAS			LAMA WAKTU (JAM)				BIAYA STARTUP (JUTA Rp)		KOEFSIEN BIAYA BAHAN BAKAR		
		Daya Terpasang	MIN (MW)	MAX (MW)	MIN UP TIME	MIN DOWN TIME	COLD START UP	HOT START UP	COLD START UP	HOT START UP	a0	a1	a2
P	UP. PAITON PLTU #1/2 (COAL)	2 x 400	225	370	72	48	17	4	682.98	149.68	3244978	111712.15	10.2971
2	UP. GRESIK												
ST-0	GT 1-9 OC (GAS)	9 x 112	53	102	36	10	1	0	7.82	0	5467532.4	217963.548	34.155
	CC - 1.1.1 (GAS)		115	143	36	10	3	1	57.68	31.46	10936203.3	72527.004	368.874
	CC - 2.2.1 (GAS)		164	314	36	10	3	2	65.5	39.28	11795770.8	152515.737	6.831
	CC - 3.3.1 (GAS)	3 x 526	250	480	36	10	3	2	73.32	47.1	17177460.3	145165.581	4.554
	PLTU # 1/2 (GAS)	100	43	85	48	10	9	1	143.74	40.59	1327126.68	217378.359	132.066
	PLTU # 3/4 (GAS)	200	90	175	48	10	9	2	229.5	92.52	5017369.5	169242.579	193.545
	PLTG GRESIK 1-3 (GAS)	3 x 20	5	16	3	1	1	0	6.13	0	352707.3	350680.77	903.969
	PLTG GILITIMUR 1-2 (HSD)	2 x 20	5	16	3	1	1	0	6.33	0	687181.85	683240.965	1762.3893
3	UP. MUARA KARANG												
ST-0	GT 1/2/3 - OC	3 x 107	50	95	36	10	1	0	7.35	0	5730795	202052.97	108.045
	CC - 1.1.1 (GAS)	153	110	150	36	10	3	1	54.22	29.67	11560815	53685.135	460.845
	CC - 2.2.1 (GAS)	317	200	300	36	10	3	2	61.57	36.92	16010064	127208.655	35.28
	CC - 3.3.1 (GAS)	508	300	465	36	10	3	2	68.92	44.27	31017735	87825.15	57.33
	MTW GT 1/2 - OC (HSD)	2 x 140	72	138	36	10	0	0	0	0	14706521.25	433337.8	49.4605
	MTW CC - 1.1.1 (HSD)	200	162	202	36	10	3	1	118.08	64.4	672630	144191.717	519.1757
	MTW CC - 2.2.1 (HSD)	420	210	403	36	10	3	2	134.1	80.42	30123040	303208.82	11.64715
	MTW CC - 3.3.1 (HSD)	640	315	605	36	10	3	2	160.1	96.42	43043399	288609.995	7.6584
ST-0	PLTU # 1/2/3 (MFO)	3 x 100	44	85	48	10	6	1	122.58	31.08	2417820.7	473895.41	120.77935
	PLTU # 4/5 (Gas)	2 x 200	90	165	48	10	11	2	215.34	89.29	2949187.5	205217.145	83.79

PTA	Area 4	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19
PTA	SUTAMI	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90
PTA	BRANTAS	68	68	68	68	68	68	68	68	68	68	68	68	68	68	68	73	73	73	73	73	73	73	73	73	73
PTU	PITON #1	370	370	370	370	370	370	370	370	370	370	370	370	370	370	370	370	370	370	370	370	370	370	370	370	370
PTU	PITON #2	370	370	370	370	370	370	370	370	370	370	370	370	370	370	370	370	370	370	370	370	370	370	370	370	370
	GRSIK110C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK10C1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK20C1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK120C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK130C	87	87	85	85	78	78	78	78	85	85	103	103	101	101	84	84	86	86	102	102	103	103	102	102	102
	GRSIK11CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK12CC	285	285	278	278	268	268	258	258	250	250	250	250	250	250	300	300	300	300	300	300	300	300	300	300	300
	GRSIK13CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK210C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK10C2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK20C2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK220C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK230C	87	87	85	85	91	91	92	92	89	89	84	84	85	86	86	89	89	85	85	96	96	97	97	96	96
	GRSIK21CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK22CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK23CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK11CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK10CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK20CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK330C	89	89	87	87	86	86	91	91	95	95	94	94	97	97	92	92	93	93	93	93	97	97	96	96	95
	GRSIK31CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK32CC	285	285	288	288	265	265	252	252	250	250	250	250	250	250	275	275	300	300	300	300	300	300	300	300	275
	GRSIK33CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK #3	90	90	90	90	90	90	90	90	90	90	90	90	90	90	135	135	165	165	165	165	165	165	165	165	165
	GRSIK #1	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75
	GRSIK #2	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75	75
	GLTMR #1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK #1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK #2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK #3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(*)	Perhitungan Area -1	3245	3245	3245	3245	3245	3245	3245	3428	3497	3497	3202	2974	2844	2844	3087	3272	3422	3522	3522	3522	3522	3522	3522	3522	3522
(**)	Perhitungan Area -1	2074	2065	2054	2052	2032	2032	2030	2031	2032	2030	2030	1747	1717	1665	1618	1692	1982	2082	2089	2030	2092	1998	2038	2073	2027
(***)	Perhitungan Area -1	1171	1180	1191	1193	1213	1213	1215	1395	1465	1467	1252	1217	1127	1179	1251	1380	1460	1440	1453	1492	1430	1528	1484	1409	1260
(****)	Perhitungan Area -1	65	65	100	108	108	106	108	108	72	72	67	93	92	92	98	59	61	61	61	61	61	61	61	61	61
(*****)	Perhitungan Area -1	604	604	644	695	695	695	655	695	624	624	624	624	624	624	952	1054	854	704	604	604	604	604	604	604	604

RENCANA: HARI/TANGGAL: SABTU, 13 MARET 2004

SUB SISTEM REGION 1

PT. PLN PEMBANGKITAN TENAGA LISTRIK JAWA-BALI

[illegible]

RENCANA : HARI/TANGGAL: SABTU, 13 MARET 2004

SUB SYSTEM REGION_1

PT. PLN PENBAKSI TANENAGA USTRIK JAWA. GALL

[illegible]

REKAMAYA : HARITANGGAL MINGGU, 14 MARET 2004
PT. PLN PEMBANGKITAN TENAGA LISTRIK JAWA-BALI

SUB SYSTEM REGION_1

PT. PLN PEMBANGKITAN TENAGA LISTRIK JAWA-BALI

		Juni																																															
		00.30	01.00	01.30	02.00	02.30	03.00	03.30	04.00	04.30	05.00	05.30	06.00	06.30	07.00	07.30	08.00	08.30	09.00	09.30	10.00	10.30	11.00	11.30	12.00	12.30																							
PLTU	MHRNG10C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																								
	MHRNG10C1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																								
	MHRNG20C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																								
	MHRNG20C1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																								
	MHRNG30C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																								
	MHRNG30C1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																								
	MHRNG30C2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																								
	MHRNG30C3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																								
	MHRNG30C4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																								
	MHRNG30C5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																								
PLTU	MHRNG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																								
	#4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	#5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	#1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	#2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
PLTU	MHRNG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																								
	#1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	#2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	#3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	MHRNG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	#1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	#2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	#3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	MHRNG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																								
	#1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																								
PLTU	MHRNG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																								
	#1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	#2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	#3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	MHRNG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	#1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	#2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	#3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	MHRNG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																								
	#1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																								
PLTU	MHRNG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																								
	#1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	#2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	#3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	MHRNG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	#1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	#2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	#3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	MHRNG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																								
	#1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																								
PLTU	MHRNG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																								
	#1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	#2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	#3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	MHRNG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	#1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	#2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	#3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	MHRNG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																								
	#1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																								
PLTU	MHRNG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																								
	#1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	#2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	#3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	MHRNG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	#1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	#2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	#3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	MHRNG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																								
	#1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																								
PLTU	MHRNG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																								
	#1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	#2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	#3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	MHRNG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	#1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	#2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	#3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	MHRNG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																								
	#1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																								
PLTU	MHRNG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																								
	#1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	#2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	#3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																									
	MHRNG	0	0	0	0	0	0	0	0	0																																							


```

unit EvolutionaryFunction;

interface

uses Komplex;

type
  TEP=class
  private
    maxgen,popsize,lchrom,Nparam:integer;
    pcross,pmutat,pflip,ka:double;
    function GetMaxgen:integer;
    function GetPopsize:integer;
    function GetLchrom:integer;
    function GetNparam:integer;
    function GetPcross:double;
    function GetPmutat:double;
    function GetPflip:double;
    function GetKa:double;
    procedure SetMaxgen(dMaxgen:integer);
    procedure SetPopsize(dPopsize:integer);
    procedure SetLchrom(dLchrom:integer);
    procedure SetNparam(dNparam:integer);
    procedure SetPcross(dPcross:double);
    procedure SetPmutat(dPmutat:double);
    procedure SetPflip(dPflip:double);
    procedure SetKa(dKa:double);
  public
    constructor Create;
    function GetFlip(const param:double):boolean;
    function GetRandom(const min,max:integer):integer;
    destructor Destroy;override;
    property gaMaxgen:integer read GetMaxgen write SetMaxgen;
    property gaPopsize:integer read GetPopsize write SetPopsize;
    property gaLchrom:integer read GetLchrom write SetLchrom;
    property gaNparam:integer read GetNparam write SetNparam;
    property gaPcross:double read GetPcross write SetPcross;
    property gaPmutat:double read getPmutat write SetPmutat;
    property gaPflip:double read GetPflip write SetPflip;
    property gaKa:double read GetKa write SetKa;
  end;

implementation

//constructor
constructor TEP.Create;
begin
  inherited Create;
  maxgen:=1;
  popsize:=1;
  Nparam:=1;
  ka:=1;
end;

//data accessing
function TEP.GetMaxgen:integer;
begin

```

```

    result:=maxgen;
end;

function TEP.GetPopsize:integer;
begin
    result:=popsize;
end;

function TEP.GetLchrom:integer;
begin
    result:=lchrom;
end;

function TEP.GetNparam:integer;
begin
    result:=Nparam;
end;

function TEP.GetPcross:double;
begin
    result:=pcross;
end;

function TEP.GetPmutat:double;
begin
    result:=pmutat;
end;

function TEP.GetPflip:double;
begin
    result:=pflip;
end;

function TEP.GetKa:double;
begin
    result:=ka;
end;

procedure TEP.SetMaxgen(dMaxgen:integer);
begin
    maxgen:=dMaxgen;
end;

procedure TEP.SetPopsize(dPopsize:integer);
begin
    popsize:=dpopsize;
end;

procedure TEP.SetLchrom(dLchrom:integer);
begin
    lchrom:=dLchrom;
end;

procedure TEP.SetNparam(dNparam:integer);
begin
    Nparam:=dNparam;
end;

```

```

procedure TEP.SetPcross(dPcross:double);
begin
    pcross:=dPcross;
end;

procedure TEP.SetPmutat(dPmutat:double);
begin
    pmutat:=dPmutat;
end;

procedure TEP.SetPflip(dPflip:double);
begin
    pflip:=dPflip;
end;

procedure TEP.SetKa(dKa:double);
begin
    ka:=dka;
end;

//data processing
function TEP.GetFlip(const param:double):boolean;
var rand:double;
begin
    rand:=random;
    if rand<=param then
    begin
        result:=true;
    end
    else
    begin
        result:=false;
    end;
end;

function TEP.GetRandom(const min,max:integer):integer;
var rand:double;
begin
    rand:=random;
    result:=round(min+rand*(max-min));
end;

destructor TEP.Destroy;
begin
    inherited Destroy;
end;

end.

```

```

unit DecodeChromosome;

interface

uses Komplex, TypDatEP;

type
  TDecBinToFloat1=class
  private
    lchrom:integer;
    input:TChromBin1;
    minParam,maxParam,hasil:double;
    procedure setLchrom(const dLchrom:integer);
    procedure setInput(const dInput:TChromBin1);
    procedure setMinParam(const dMinParam:double);
    procedure setMaxParam(const dMaxParam:double);
    procedure doHitung;
    function getHasil:double;
  public
    constructor Create;overload;
    constructor Create(const dlchrom:integer);overload;
    constructor Create(const dInput:TChromBin1;
                        const dminParam,dmaxParam:double);overload;
    destructor Destroy;override;
    property decLchrom:integer write setLchrom;
    property decInput:TChromBin1 write setInput;
    property decMinParam:double write setMinParam;
    property decMaxParam:double write setMaxParam;
    property decHasil:double read getHasil;
  end;

  TDecBinToFloat2=class
  private
    lchrom,Nparam:integer;
    input:TChromBin2;
    minParam,maxParam,hasil:Arr1;
    procedure setLchrom(const dLchrom:integer);
    procedure setNparam(const dNparam:integer);
    procedure setInput(const dInput:TChromBin2);
    procedure setMinParam(const dMinParam:Arr1);
    procedure setMaxParam(const dMaxParam:Arr1);
    procedure doHitung;
    function getHasil:Arr1;
  public
    constructor Create;overload;
    constructor Create(const dlchrom,dNparam:integer);overload;
    constructor Create(const dInput:TChromBin2;
                        const dminParam,dmaxParam:Arr1);overload;
    destructor Destroy;override;
    property decLchrom:integer write setLchrom;
    property decNparam:integer write setNparam;
    property decInput:TChromBin2 write setInput;
    property decMinParam:Arr1 write setMinParam;
    property decMaxParam:Arr1 write setMaxParam;
    property decHasil:Arr1 read getHasil;
  end;

```

implementation

//constructors

constructor TDecBinToFloat1.Create;

begin

 inherited Create;

 lchrom:=0;

 minParam:=0;

 maxParam:=0;

end;

constructor TDecBinToFloat1.Create(const dLchrom:integer);

begin

 inherited Create;

 lchrom:=dLchrom;

 setLength(input,lchrom+1);

 minParam:=0;

 maxParam:=0;

end;

constructor TDecBinToFloat1.Create(const dInput:TChromBin1;

 const dminParam,dmaxParam:double);

var i:integer;

begin

 inherited Create;

 lchrom:=high(dInput);

 setLength(input,lchrom+1);

 for i:=1 to lchrom do

 begin

 input[i]:=dInput[i];

 end;

 minParam:=dminParam;

 maxParam:=dmaxParam;

end;

//data accessing

procedure TDecBinToFloat1.setLchrom(const dLchrom:integer);

begin

 lchrom:=dLchrom;

end;

procedure TDecBinToFloat1.setInput(const dInput:TChromBin1);

var i:integer;

begin

 setLength(input,lchrom+1);

 for i:=1 to lchrom do

 begin

 input[i]:=dInput[i];

 end;

end;

procedure TDecBinToFloat1.setMinParam(const dMinParam:double);

begin

 minParam:=dMinParam;

end;

procedure TDecBinToFloat1.setMaxParam(const dMaxParam:double);

```

begin
    maxParam:=dMaxParam;
end;

//data processing
procedure TDecBinToFloat1.doHitung;
var i:integer;
    accum,powerof2,Sa:double;
begin
    Sa:=Pangkat(2,lchrom);
    accum:=0;
    powerof2:=1;
    for i:=1 to lchrom do
        begin
            if input[i]=true then accum:=accum+powerof2;
            powerof2:=powerof2*2;
        end;
    hasil:=minParam+accum*(maxParam-minParam)/(Sa-1);
end;

function TDecBinToFloat1.getHasil:double;
begin
    doHitung;
    result:=hasil;
end;

//destructor
destructor TDecBinToFloat1.Destroy;
begin
    inherited Destroy;
end;

//constructors
constructor TDecBinToFloat2.Create;
begin
    inherited Create;
    lchrom:=0;
    Nparam:=0;
end;

constructor TDecBinToFloat2.Create(const dlchrom,dNparam:integer);
begin
    inherited Create;
    lchrom:=dlchrom;
    Nparam:=dNparam;
    setLength(input,Nparam+1,lchrom+1);
    setLength(minParam,Nparam+1);
    setLength(maxParam,Nparam+1);
end;

constructor TDecBinToFloat2.Create(const dInput:TChromBin2;
    const dminParam,dmaxParam:Arr1);
var i,j:integer;
begin
    inherited Create;
    Nparam:=high(dInput);
    lchrom:=high(dInput[0]);

```



```

lchrom:=high(dInput[0]);
setLength(input,Nparam+1,lchrom+1);
setLength(minParam,Nparam+1);
setLength(maxParam,Nparam+1);
for i:=1 to Nparam do
begin
  for j:=1 to lchrom do
  begin
    input[i,j]:=dInput[i,j];
  end;
  minParam[i]:=dMinParam[i];
  maxParam[i]:=dMaxParam[i];
end;
end;

//data accessing
procedure TDecBinToFloat2.setLchrom(const dLchrom:integer);
begin
  lchrom:=dLchrom;
end;

procedure TDecBinToFloat2.setNparam(const dNparam:integer);
begin
  Nparam:=dNparam;
end;

procedure TDecBinToFloat2.setInput(const dInput:TChromBin2);
var i,j:integer;
begin
  setLength(input,Nparam+1,lchrom+1);
  for i:=1 to Nparam do
  begin
    for j:=1 to lchrom do
    begin
      input[i,j]:=dInput[i,j];
    end;
  end;
end;

procedure TDecBinToFloat2.setMinParam(const dMinParam:Arr1);
var i:integer;
begin
  setLength(minParam,Nparam+1);
  for i:=1 to Nparam do
  begin
    minParam[i]:=dMinParam[i];
  end;
end;

procedure TDecBinToFloat2.setMaxParam(const dMaxParam:Arr1);
var i:integer;
begin
  setLength(maxParam,Nparam+1);
  for i:=1 to Nparam do
  begin
    maxParam[i]:=dMaxParam[i];
  end;
end;

```

```

end;

//data processing
procedure TDecBinToFloat2.doHitung;
var i,j:integer;
    accum,powerof2,Sa:double;
begin
    setLength(hasil,Nparam+1);
    Sa:=Pangkat(2,lchrom);
    for i:=1 to Nparam do
    begin
        accum:=0;
        powerof2:=1;
        for j:=1 to lchrom do
        begin
            if input[i,j]=true then accum:=accum+powerof2;
            powerof2:=powerof2*2;
        end;
        hasil[i]:=minParam[i]+accum*(maxParam[i]-minParam[i])/(Sa-1);
    end;
end;

function TDecBinToFloat2.getHasil:Arr1;
var i:integer;
begin
    setLength(result,Nparam+1);
    doHitung;
    for i:=1 to Nparam do
    begin
        result[i]:=hasil[i];
    end;
end;

//destructor
destructor TDecBinToFloat2.Destroy;
begin
    inherited Destroy;
end;

end.

```

```

unit EcoManager;

interface

uses Komplex, UnitGenerator, SysUtils, Dialogs;

type
  TEcoManager=class
  private
    Ngen,Njam:integer;
    gen:TGenArr;
    PLN:Arr2;
    Beban,Res:Arr1;
    function GetNgen:integer;
    function GetNjam:integer;
    function GetGen:TGenArr;
    function GetPLN:Arr2;
    function GetBeban:Arr1;
    function GetRes:Arr1;
    procedure SetNgen(const dNgen:integer);
    procedure SetNjam(const dNjam:integer);
    procedure SetGen(const dGen:TGenArr);
    procedure SetPLN(const dPLN:Arr2);
    procedure SetBeban(const dBeban:Arr1);
    procedure SetRes(const dRes:Arr1);
  public
    constructor Create;overload;
    constructor Create(const dNgen,dNjam:integer);overload;
    constructor Create(const dGen:TGenArr;const dPLN:Arr2;
      const dBeban,dRes:Arr1);overload;
    destructor Destroy;override;
    procedure GetData(const djam:integer;
      var dNgen:integer;
      var dGen:TGenArr;
      var dPLN:Arr1;
      var dBeban,dRes:double);
    property emNgen:integer read GetNgen write SetNgen;
    property emNjam:integer read GetNjam write SetNjam;
    property emGen:TGenArr read GetGen write SetGen;
    property emPLN:Arr2 read getPLN write SetPLN;
    property emBeban:Arr1 read GetBeban write SetBeban;
    property emRes:Arr1 read GetRes write SetRes;
  end;

var ecom:TEcoManager;

implementation

//constructor
constructor TEcoManager.Create;
begin
  inherited Create;
  Ngen:=0;
  Njam:=0;
end;

constructor TEcoManager.Create(const dNgen,dNjam:integer);

```

```

begin
    inherited Create;
    Ngen:=dNgen;
    Njam:=dNjam;
end;

constructor TEcoManager.Create(const dGen:TGenArr;const dPLN:Arr2;
                               const dBeban,dRes:Arr1);
var i,j,Ncek:integer;
begin
    Ngen:=high(dGen);
    Ncek:=high(dPLN);
    if Ngen<>Ncek then
        begin
            raise Exception.Create('Ukuran Data tidak sama!');
        end;
    Njam:=high(dPLN[0]);
    Ncek:=high(Beban);
    if Njam<>Ncek then
        begin
            raise Exception.Create('Ukuran Data tidak sama!');
        end;
    Ncek:=high(Res);
    if Njam<>Ncek then
        begin
            raise Exception.Create('Ukuran Data tidak sama!');
        end;
    SetLength(gen,Ngen+1);
    SetLength(PLN,Ngen+1,Njam+1);
    SetLength(Beban,Njam+1);
    SetLength(Res,Njam+1);
    for i:=1 to Ngen do
        begin
            gen[i]:=TPembangkit.Create;
            gen[i].Assign(dGen[i]);
            for j:=1 to Njam do
                begin
                    PLN[i,j]:=dPLN[i,j];
                end;
            end;
        for i:=1 to Njam do
            begin
                Beban[i]:=dBeban[i];
                Res[i]:=dRes[i];
            end;
        end;
end;

//data accessing
function TEcoManager.GetNgen:integer;
begin
    result:=Ngen;
end;

function TEcoManager.GetNjam:integer;
begin
    result:=Njam;
end;

```

```

function TEcoManager.GetGen:TGenArr;
var i:integer;
begin
  SetLength(result,Ngen+1);
  for i:=1 to Ngen do
    begin
      result[i]:=TPembangkit.Create;
      result[i].Assign(gen[i]);
    end;
  end;

function TEcoManager.GetPLN:Arr2;
var i,j:integer;
begin
  SetLength(result,Ngen+1,Njam+1);
  for i:=1 to Ngen do
    begin
      for j:=1 to Njam do
        begin
          result[i,j]:=PLN[i,j];
        end;
      end;
    end;
  end;

function TEcoManager.GetBeban:Arr1;
var i:integer;
begin
  SetLength(result,Njam+1);
  for i:=1 to Njam do
    begin
      result[i]:=Beban[i];
    end;
  end;

function TEcoManager.GetRes:Arr1;
var i:integer;
begin
  SetLength(result,Njam+1);
  for i:=1 to Njam do
    begin
      result[i]:=Beban[i];
    end;
  end;

procedure TEcoManager.SetNgen(const dNgen:integer);
begin
  Ngen:=dNgen;
end;

procedure TEcoManager.SetNjam(const dNjam:integer);
begin
  Njam:=dNjam;
end;

procedure TEcoManager.SetGen(const dGen:TGenArr);
var i,NgenCek:integer;
begin

```

```

NgenCek:=high(dGen);
if Ngen<>0 then
begin
  if Ngen<>NgenCek then
  begin
    raise Exception.Create('Dimansi Matrik tidak sama!');
  end;
end
else
begin
  Ngen:=NgenCek;
end;
SetLength(gen,Ngen+1);
for i:=1 to Ngen do
begin
  gen[i]:=TPembangkit.Create;
  gen[i].Assign(dGen[i]);
end;
end;

procedure TEcoManager.SetPLN(const dPLN:Arr2);
var i,j,NgenCek,NjamCek:integer;
begin
  NgenCek:=high(dPLN);
  NjamCek:=high(dPLN[0]);
  if Ngen<>0 then
  begin
    if Ngen<>NgenCek then
    begin
      raise Exception.Create('Dimensi Matrik tidak sama!');
    end;
  end
  else
  begin
    Ngen:=NgenCek;
  end;
  if Njam<>0 then
  begin
    if Njam<>NjamCek then
    begin
      raise Exception.Create('Dimensi Matrik tidak sama!');
    end;
  end
  else
  begin
    Njam:=NjamCek;
  end;
  SetLength(PLN,Ngen+1,Njam+1);
  for i:=1 to Ngen do
  begin
    for j:=1 to Njam do
    begin
      PLN[i,j]:=dPLN[i,j];
    end;
  end;
end;
end;

```

```

procedure TEcoManager.SetBeban(const dBeban:Arr1);
var i,NjamCek:integer;
begin
  NjamCek:=high(dBeban);
  if Njam<>0 then
  begin
    if Njam<>NjamCek then
    begin
      raise Exception.Create('Dimensi Matrik tidak sama!');
    end;
  end
  else
  begin
    Njam:=NjamCek;
  end;
  SetLength(beban,Njam+1);
  for i:=1 to Njam do
  begin
    beban[i]:=dBeban[i];
  end;
end;

```

```

procedure TEcoManager.SetRes(const dRes:Arr1);
var i,NjamCek:integer;
begin
  NjamCek:=high(dRes);
  if Njam<>0 then
  begin
    if Njam<>NjamCek then
    begin
      raise Exception.Create('Dimensi Matrik tidak sama!');
    end;
  end
  else
  begin
    Njam:=NjamCek;
  end;
  SetLength(res,Njam+1);
  for i:=1 to Njam do
  begin
    res[i]:=dRes[i];
  end;
end;

```

```

//data output
procedure TEcoManager.GetData(const djam:integer;
  var dNgen:integer;
  var dGen:TGenArr;
  var dPLN:Arr1;
  var dBeban,dRes:double);
var i:integer;
begin
  try
    dNgen:=0;
    for i:=1 to Ngen do
    begin
      if PLN[i,djam]<>0 then

```



```

begin
    dNgen:=dNgen+1;
end;
end;
SetLength(dGen,dNgen+1);
SetLength(dPLN,dNgen+1);
dNgen:=0;
for i:=1 to Ngen do
begin
    if PLN[i,djam]<>0 then
    begin
        dNgen:=dNgen+1;
        dGen[dNgen]:=TPembangkit.Create;
        dGen[dNgen].Assign(gen[i]);
        dPLN[dNgen]:=PLN[i,djam];
    end;
end;
dBeban:=beban[djam];
dRes:=res[djam];
except
on E:Exception do
begin
    MessageDlg(E.Message,mtWarning,[mbOK],0);
end;
end;
end;
end;

```

```

//destructor
destructor TEcoManager.Destroy;
var i:integer;
begin
    try
        for i:=1 to Ngen do
        begin
            gen[i].Free;
        end;
    finally
        inherited Destroy;
    end;
end;
end.

```

```

unit EvolutionaryProgramming;

interface

uses Komplex;

type
  TEP=class
  private
    maxgen,popsize,lchrom,Nparam:integer;
    pcross,pmutat,pflip,ka:double;
    function GetMaxgen:integer;
    function GetPopsize:integer;
    function GetLchrom:integer;
    function GetNparam:integer;
    function GetPcross:double;
    function GetPmutat:double;
    function GetPflip:double;
    function GetKa:double;
    procedure SetMaxgen(dMaxgen:integer);
    procedure SetPopsize(dPopsize:integer);
    procedure SetLchrom(dLchrom:integer);
    procedure SetNparam(dNparam:integer);
    procedure SetPcross(dPcross:double);
    procedure SetPmutat(dPmutat:double);
    procedure SetPflip(dPflip:double);
    procedure SetKa(dKa:double);
  public
    constructor Create;
    function GetFlip(const param:double):boolean;
    function GetRandom(const min,max:integer):integer;
    destructor Destroy;override;
    property gaMaxgen:integer read GetMaxgen write SetMaxgen;
    property gaPopsize:integer read GetPopsize write SetPopsize;
    property gaLchrom:integer read GetLchrom write SetLchrom;
    property gaNparam:integer read GetNparam write SetNparam;
    property gaPcross:double read GetPcross write SetPcross;
    property gaPmutat:double read getPmutat write SetPmutat;
    property gaPflip:double read GetPflip write SetPflip;
    property gaKa:double read GetKa write SetKa;
  end;

implementation

//constructor
constructor TEP.Create;
begin
  inherited Create;
  maxgen:=1;
  popsize:=1;
  Nparam:=1;
  ka:=1;
end;

//data accessing
function TEP.GetMaxgen:integer;
begin

```

```

    result:=maxgen;
end;

function TEP.GetPopsiz:integer;
begin
    result:=popsiz;
end;

function TEP.GetLchrom:integer;
begin
    result:=lchrom;
end;

function TEP.GetNparam:integer;
begin
    result:=Nparam;
end;

function TEP.GetPcross:double;
begin
    result:=pcross;
end;

function TEP.GetPmutat:double;
begin
    result:=pmutat;
end;

function TEP.GetPflip:double;
begin
    result:=pflip;
end;

function TEP.GetKa:double;
begin
    result:=ka;
end;

procedure TEP.SetMaxgen(dMaxgen:integer);
begin
    maxgen:=dMaxgen;
end;

procedure TEP.SetPopsiz(dPopsiz:integer);
begin
    popsiz:=dpopsiz;
end;

procedure TEP.SetLchrom(dLchrom:integer);
begin
    lchrom:=dLchrom;
end;

procedure TEP.SetNparam(dNparam:integer);
begin
    Nparam:=dNparam;
end;

```

```

procedure TEP.SetPcross(dPcross:double);
begin
    pcross:=dPcross;
end;

procedure TEP.SetPmutat(dPmutat:double);
begin
    pmutat:=dPmutat;
end;

procedure TEP.SetPflip(dPflip:double);
begin
    pflip:=dPflip;
end;

procedure TEP.SetKa(dKa:double);
begin
    ka:=dka;
end;

//data processing
function TEP.GetFlip(const param:double):boolean;
var rand:double;
begin
    rand:=random;
    if rand<=param then
    begin
        result:=true;
    end
    else
    begin
        result:=false;
    end;
end;

function TEP.GetRandom(const min,max:integer):integer;
var rand:double;
begin
    rand:=random;
    result:=round(min+rand*(max-min));
end;

destructor TEP.Destroy;
begin
    inherited Destroy;
end;

end.

```

```

unit fitness;

interface

uses Komplex, UnitGenerator, SysUtils, Dialogs, DecodeChromosome;

type
  Tfitness=class
  private
    Ngen, lchrom:integer;
    gen:TGenArr;
    chrom:Arr1;
    beban, alpha1:double;
    function GetNgen:integer;
    procedure SetNgen(const dNgen:integer);
    function GetGen:TGenArr;
    function GetChrom:Arr1;
    procedure SetChrom(const dChrom:Arr1);
    procedure SetGen(const dGen:TGenArr);
    function GetBeban:double;
    procedure SetBeban(const dBeban:double);
    function GetAlpha1:double;
    procedure SetAlpha1(const dAlpha1:double);
    function GetRandom(const dmin,dmax:integer):integer;
    function GetValue:double;
    function GetDayaGen:Arr1;
  public
    constructor Create;overload;
    constructor Create(const dGen:TGenArr);overload;
    destructor Destroy;override;
    property fitNgen:integer read GetNgen write SetNgen;
    property fitGen:TGenArr read GetGen write SetGen;
    property fitChrom:Arr1 read GetChrom write SetChrom;
    property fitBeban:double read GetBeban write SetBeban;
    property fitAlpha1:double read GetAlpha1 write SetAlpha1;
    property fitValue:double read GetValue;
    property fitDayaGen:Arr1 read GetDayaGen;
  end;

var fita:Tfitness;

implementation

//constructor
constructor Tfitness.Create;
begin
  inherited Create;
  Ngen:=0;
end;

constructor Tfitness.Create(const dGen:TGenArr);
var i:integer;
begin
  inherited Create;
  Ngen:=high(dGen);
  SetLength(gen,Ngen+1);
  for i:=1 to Ngen do

```

```

begin
    gen[i]:=TPembangkit.Create;
    gen[i].Assign(dGen[i]);
end;
end;

//data accessing
function Tfitness.GetNgen:integer;
begin
    result:=Ngen;
end;

procedure Tfitness.SetNgen(const dNgen:integer);
begin
    Ngen:=dNgen;
end;

function Tfitness.GetGen:TGenArr;
var i:integer;
begin
    SetLength(result,Ngen+1);
    for i:=1 to Ngen do
        begin
            result[i]:=TPembangkit.Create;
            result[i].Assign(gen[i]);
        end;
    end;
end;

procedure Tfitness.SetGen(const dGen:TGenArr);
var i:integer;
begin
    Ngen:=high(dGen);
    SetLength(gen,Ngen+1);
    for i:=1 to Ngen do
        begin
            gen[i]:=TPembangkit.Create;
            gen[i].Assign(dGen[i]);
        end;
    end;
end;

function Tfitness.GetChrom:Arr1;
var i:integer;
begin
    SetLength(result,Ngen+1);
    for i:=1 to Ngen do
        begin
            result[i]:=chrom[i];
        end;
    end;
end;

procedure Tfitness.SetChrom(const dChrom:Arr1);
var i,NgenCek:integer;
begin
    NgenCek:=high(dChrom);
    if Ngen<>0 then
        begin
            if Ngen<>NgenCek then

```

```

begin
    raise Exception.Create('Dimensi Matrik tidak sama!');
end;
end
else
begin
    Ngen:=NgenCek;
end;
SetLength(chrom,Ngen+1);
for i:=1 to Ngen do
begin
    chrom[i]:=dChrom[i];
end;
end;
end;

function Tfitness.GetBeban:double;
begin
    result:=beban;
end;

procedure Tfitness.SetBeban(const dBeban:double);
begin
    beban:=dBeban;
end;

function Tfitness.GetAlpha:double;
begin
    result:=alpha;
end;

procedure Tfitness.SetAlpha(const dAlpha:double);
begin
    alpha:=dAlpha;
end;

//data processing
function Tfitness.GetRandom(const dmin,dmax:integer):integer;
var rand:double;
begin
    rand:=dmin+random(dmax-dmin);
    result:=round(rand);
end;

function Tfitness.GetValue:double;
var i,id:integer;
    sBeban,pinDaya,sFcost:double;
begin
    try
        id:=GetRandom(1,Ngen);
        sBeban:=0;
        for i:=1 to Ngen do
            begin
                if i<>id then
                    begin
                        sBeban:=sBeban+chrom[i];
                    end;
                end;
            end;
        end;
    end;
end;

```



```

chrom[id]:=Beban-sBeban;
pinDaya:=0;
if chrom[id]<gen[id].GenPmin then
begin
    pinDaya:=gen[id].GenPmin-chrom[id];
    chrom[id]:=gen[id].GenPmin;
end
else if chrom[id]>gen[id].GenPmax then
begin
    pinDaya:=chrom[id]-gen[id].GenPmax;
    chrom[id]:=gen[id].GenPmax;
end;
sFcost:=0;
for i:=1 to Ngen do
begin
    sFcost:=sFcost+gen[i].GetBiaya(chrom[i]);
end;
result:=sFcost+alpha*pinDaya;
except
on E:Exception do
begin
    MessageDlg(E.Message,mtWarning,[mbOK],0);
end
end;
end;
end;

```

```

function Tfitness.GetDayaGen:Arr1;
var i:integer;
    sBeban:double;
begin
    try
        SetLength(result,Ngen+1);
        sBeban:=0;
        for i:=1 to lchrom do
            begin
                sBeban:=sBeban+chrom[i];
            end;
            result[1]:=beban-sBeban;
            for i:=1 to lchrom do
                begin
                    result[i+1]:=chrom[i];
                end;
            end;
        except
            on E:Exception do
                begin
                    MessageDlg(E.Message,mtwarning,[mbOK],0);
                end;
            end;
        end;
    end;
end;

```

```

//destructor
destructor Tfitness.Destroy;
var i:integer;
begin
    try
        for i:=1 to Ngen do
            begin

```

```
    gen[i].Free;  
    end;  
    finally  
        inherited Destroy;  
    end;  
    end;  
end.
```

```

unit TypDatEP;

interface

uses Komplex;

type
  TchromBin1=array of boolean;
  TchromBin2=array of array of boolean;
  TchromFloat1=Arr1;
  TchromFloat2=Arr2;
  TchromInt1=iArr1;
  TchromInt2=iArr2;

  TindividuBin1=record
    chrom:TchromBin1;
    fitness:double;
  end;

  TindividuBin2=record
    chrom:TchromBin2;
    fitness:double;
  end;

  TindividuFloat1=record
    chrom:TchromFloat1;
    fitness:double;
  end;

  TindividuFloat2=record
    chrom:TchromFloat2;
    fitness:double;
  end;

  TindividuInt1=record
    chrom:TchromInt1;
    fitness:double;
  end;

  TindividuInt2=record
    chrom:TchromInt2;
    fitness:double;
  end;

implementation

end.

```

```

unit UnitGenerator;

interface

uses Komplex;

type
  TPembangkit=class
  private
    Nama:string;
    Pmax,Pmin,a2,a1,a0,Sh,Sc,AFLC,Daya:double;
    tup,tdown,tcold,IniSt:integer;
    function GetNama:string;
    function GetPmax:double;
    function GetPmin:double;
    function Geta2:double;
    function Geta1:double;
    function Geta0:double;
    function GetSh:double;
    function GetSc:double;
    function GetTup:integer;
    function GetTdown:integer;
    function GetTcold:integer;
    function GetIniSt:integer;
    procedure SetNama(const dNama:string);
    procedure SetPmax(const dPmax:double);
    procedure SetPmin(const dPmin:double);
    procedure Seta2(const da2:double);
    procedure Seta1(const da1:double);
    procedure Seta0(const da0:double);
    procedure SetSh(const dSh:double);
    procedure SetSc(const dSc:double);
    procedure SetTup(const dtup:integer);
    procedure SetTdown(const dtdown:integer);
    procedure SetTcold(const dtcold:integer);
    procedure SetIniSt(const dIniSt:integer);
    procedure SetDaya(const dDaya:double);
    procedure HitungAFLC;
    function GetAFLC:double;
    function GetDaya:double;
  public
    function GetBiaya(const dDaya:double):double;
    procedure Assign(const dGen:TPembangkit);
    property GenNama:string read GetNama write SetNama;
    property GenPmax:double read GetPmax write SetPmax;
    property GenPmin:double read GetPmin write SetPmin;
    property Gena2:double read Geta2 write Seta2;
    property Genal:double read Geta1 write Setal;
    property Gena0:double read Geta0 write Seta0;
    property GenSh:double read GetSh write SetSh;
    property GenSc:double read GetSc write SetSc;
    property GenTup:integer read GetTup write SetTup;
    property GenTdown:integer read GetTdown write SetTdown;
    property GenTcold:integer read GetTcold write SetTcold;
    property GenIniSt:integer read GetIniSt write SetIniSt;
    property GenDaya:double read GetDaya write SetDaya;
    property GenAFLC:double read GetAFLC;

```

```

end;

TGenArr=array of TPembangkit;

implementation

procedure TPembangkit.Assign(const dGen:TPembangkit);
begin
    Nama:=dGen.GenNama;
    Pmax:=dGen.GenPmax;
    Pmin:=dGen.GenPmin;
    a2:=dGen.Gena2;
    a1:=dGen.Gena1;
    a0:=dGen.Geta0;
    Sh:=dGen.GenSh;
    Sc:=dGen.GenSc;
    tup:=dGen.GenTup;
    tdown:=dGen.GenTdown;
    tcold:=dGen.GenTcold;
    iniSt:=dGen.GenIniSt;
end;

function TPembangkit.GetNama:string;
begin
    Result:=Nama;
end;

function TPembangkit.GetPmax:double;
begin
    Result:=Pmax;
end;

function TPembangkit.GetPmin:double;
begin
    Result:=Pmin;
end;

function TPembangkit.Geta2:double;
begin
    Result:=a2;
end;

function TPembangkit.Geta1:double;
begin
    Result:=a1;
end;

function TPembangkit.Geta0:double;
begin
    Result:=a0;
end;

function TPembangkit.GetSh:double;
begin
    Result:=Sh;
end;

```

```

function TPembangkit.GetSc:double;
begin
    Result:=Sc;
end;

function TPembangkit.GetTup:integer;
begin
    Result:=tup;
end;

function TPembangkit.GetTdown:integer;
begin
    Result:=tdown;
end;

function TPembangkit.GetTcold:integer;
begin
    Result:=tcold;
end;

function TPembangkit.GetIniSt:integer;
begin
    Result:=IniSt;
end;

function TPembangkit.GetBiaya(const dDaya:double):double;
begin
    Result:=0;
    if dDaya<>0 then
        begin
            Result:=a2*sqr(dDaya)+a1*dDaya+a0;
        end;
    end;
end;

function TPembangkit.GetAFLC:double;
begin
    HitungAFLC;
    Result:=AFLC;
end;

function TPembangkit.GetDaya:double;
begin
    Result:=Daya;
end;

procedure TPembangkit.SetNama(const dNama:string);
begin
    Nama:=dNama;
end;

procedure TPembangkit.SetPmax(const dPmax:double);
begin
    Pmax:=dPmax;
end;

procedure TPembangkit.SetPmin(const dPmin:double);
begin

```

```

    Pmin:=dPmin;
end;

procedure TPembangkit.Seta2(const da2:double);
begin
    a2:=da2;
end;

procedure TPembangkit.Seta1(const da1:double);
begin
    a1:=da1;
end;

procedure TPembangkit.Seta0(const da0:double);
begin
    a0:=da0;
end;

procedure TPembangkit.SetSh(const dSh:double);
begin
    Sh:=dSh
end;

procedure TPembangkit.SetSc(const dSc:double);
begin
    Sc:=dSc;
end;

procedure TPembangkit.SetTup(const dtup:integer);
begin
    tup:=dtup;
end;

procedure TPembangkit.SetTdown(const dtdown:integer);
begin
    tdown:=dtdown;
end;

procedure TPembangkit.SetTcold(const dtcold:integer);
begin
    tcold:=dtcold;
end;

procedure TPembangkit.SetIniSt(const diniSt:integer);
begin
    iniSt:=diniSt;
end;

procedure TPembangkit.SetDaya(const dDaya:double);
begin
    Daya:=dDaya;
end;

procedure TPembangkit.HitungAFLC;
begin
    AFLC:=a0/Pmax+a1+a2*Pmax;
end;

```

```

unit Kompleks;

interface

uses Dialogs;

type
  Arr1=array of double;
  Arr2=array of array of double;
  iArr1=array of integer;
  iArr2=array of array of integer;

  TKomplex=class
  private
    re,im:double;
    function GetRe:double;
    function GetIm:double;
    procedure SetRe(const dRe:double);
    procedure SetIm(const dIm:double);
  public
    constructor Create;overload;
    constructor Create(const aRe:double);overload;
    constructor Create(const aRe,aIm:double);overload;
    constructor Create(const r,theta:extended);overload;
    destructor Destroy;override;
    function GetStringJ(len:byte):string;
    function GetStringI(len:byte):string;
    function GetAbs:double;
    function GetAngleRad:double;
    function GetAngleDeg:double;
    procedure Assign(const aValue:TKomplex);overload;
    procedure Assign(const aRe,aIm:double);overload;
    procedure DoTambah(const aValue:TKomplex);overload;
    procedure DoTambah(const aRe,aIm:double);overload;
    procedure DoTambah(const aValue1,aValue2:TKomplex);overload;
    procedure DoTambah(const aRe1,aIm1,aRe2,aIm2:double);overload;
    procedure DoKurang(const aValue:TKomplex);overload;
    procedure DoKurang(const aRe,aIm:double);overload;
    procedure DoKurang(const aValue1,aValue2:TKomplex);overload;
    procedure DoKurang(const aRe1,aIm1,aRe2,aIm2:double);overload;
    procedure DoKali(const aValue:TKomplex);overload;
    procedure DoKali(const aRe,aIm:double);overload;
    procedure DoKali(const aValue1,aValue2:TKomplex);overload;
    procedure DoKali(const aRe1,aIm1,aRe2,aIm2:double);overload;
    procedure DoBagi(const aValue:TKomplex);overload;
    procedure DoBagi(const aRe,aIm:double);overload;
    procedure DoBagi(const aValue1,aValue2:TKomplex);overload;
    procedure DoBagi(const aRe1,aIm1,aRe2,aIm2:double);overload;
    procedure DoConj;overload;
    procedure DoConj(const aValue:TKomplex);overload;
    procedure DoConj(const aRe,aIm:double);overload;
    procedure DoNegative;overload;
    procedure DoNegative(const aValue:TKomplex);overload;
    procedure DoNegative(const aRe,aIm:double);overload;
    procedure DoPangkat(pangkat:double);
    property xRe:double read GetRe write SetRe;
    property xIm:double read GetIm write SetIm;

```



```

end;

CArr1=array of TKomplex;
CArr2=array of array of TKomplex;

function RealToStr(Num:double;Pecahan:byte):String;
function StrToReal(Huruf:string):double;
function Pangkat(Val,pangkat:double):double;

implementation

function RealToStr(Num:double;Pecahan:byte):String;
var Hasil:String;
    le:byte;
begin
    le:=sizeof(Num);
    Str(Num:le:Pecahan,Hasil);
    Result:=Hasil;
end;

function Pangkat(Val,pangkat:double):double;
begin
    Result:=exp(Pangkat*ln(Val));
end;

function StrToReal(Huruf:string):double;
var Temp:double;
    Code:integer;
begin
    val(Huruf,Temp,Code);
    Result:=Temp;
end;

constructor TKomplex.Create;
begin
    inherited Create;
    re:=0;
    im:=0;
end;

constructor TKomplex.Create(const aRe:double);
begin
    inherited Create;
    re:=aRe;
    im:=0;
end;

constructor TKomplex.Create(const aRe,aIm:double);
begin
    inherited Create;
    re:=aRe;
    im:=aIm;
end;

constructor TKomplex.Create(const r,theta:extended);
begin
    inherited Create;

```

```

    re:=r*cos(theta);
    im:=r*sin(theta);
end;

destructor TKomplex.Destroy;
begin
    inherited Destroy;
end;

function TKomplex.GetRe:double;
begin
    result:=re;
end;

function TKomplex.GetIm:double;
begin
    result:=im;
end;

function TKomplex.GetStringJ(len:byte):string;
begin
    result:=RealToStr(re,len);
    if im<0 then
    begin
        result:=result+' - j'+RealToStr(abs(im),len);
    end
    else if im>0 then
    begin
        result:=result+' + j'+RealToStr(abs(im),len);
    end;
end;

function TKomplex.GetStringI(len:byte):string;
begin
    result:=RealToStr(re,len);
    if im<0 then
    begin
        result:=result+' - '+RealToStr(abs(im),len)+'i';
    end
    else if im>0 then
    begin
        result:=result+' + '+RealToStr(abs(im),len)+'i';
    end;
end;

function TKomplex.GetAbs:double;
begin
    result:=sqrt(sqr(re)+sqr(im));
end;

function TKomplex.GetAngleRad:double;
begin
    try
        result:=arctan(im/re);
    except
        result:=0;
    end;
end;

```

```

        MessageDlg('Bilangan tidak bisa dicari
sudutnya!',mtWarning,[mbOK],0);
    end;
end;

function TKomplex.GetAngleDeg:double;
var pi:double;
begin
    try
        pi:=4*arctan(1);
        result:=arctan(im/re)*180/pi;
    except
        result:=0;
        MessageDlg('Bilangan tidak bisa dicari
sudutnya!',mtWarning,[mbOK],0);
    end;
end;

procedure TKomplex.Assign(const aValue:TKomplex);
begin
    re:=aValue.re;
    im:=aValue.im;
end;

procedure TKomplex.Assign(const aRe,aIm:double);
begin
    re:=aRe;
    im:=aIm;
end;

procedure TKomplex.SetRe(const dRe:double);
begin
    re:=dRe;
end;

procedure TKomplex.SetIm(const dIm:double);
begin
    im:=dIm;
end;

procedure TKomplex.DoTambah(const aValue:TKomplex);
begin
    re:=re+aValue.re;
    im:=im+aValue.im;
end;

procedure TKomplex.DoTambah(const aRe,aIm:double);
begin
    re:=re+aRe;
    im:=im+aIm;
end;

procedure TKomplex.DoTambah(const aValue1,aValue2:TKomplex);
begin
    re:=aValue1.re+aValue2.re;
    im:=aValue1.im+aValue2.im;
end;

```

```

procedure TKomplex.DoTambah(const aRe1,aIm1,aRe2,aIm2:double);
begin
    re:=aRe1+aRe2;
    im:=aIm1+aIm2;
end;

procedure TKomplex.DoKurang(const aValue:TKomplex);
begin
    re:=re-aValue.re;
    im:=im-aValue.im;
end;

procedure TKomplex.DoKurang(const aRe,aIm:double);
begin
    re:=re-aRe;
    im:=im-aIm;
end;

procedure TKomplex.DoKurang(const aValue1,aValue2:TKomplex);
begin
    re:=aValue1.re-aValue2.re;
    im:=aValue1.im-aValue2.im;
end;

procedure TKomplex.DoKurang(const aRe1,aIm1,aRe2,aIm2:double);
begin
    re:=aRe1-aRe2;
    im:=aIm1-aIm2;
end;

procedure TKomplex.DoKali(const aValue:TKomplex);
var tmpRe,tmpIm:double;
begin
    tmpRe:=re;
    tmpIm:=im;
    re:=tmpRe*aValue.re-tmpIm*aValue.im;
    im:=tmpRe*aValue.im+tmpIm*aValue.re;
end;

procedure TKomplex.DoKali(const aRe,aIm:double);
var tmpRe,tmpIm:double;
begin
    tmpRe:=re;
    tmpIm:=im;
    re:=tmpRe*aRe-tmpIm*aIm;
    im:=tmpRe*aIm+tmpIm*aRe;
end;

procedure TKomplex.DoKali(const aValue1,aValue2:TKomplex);
var tmpRe1,tmpIm1,tmpRe2,tmpIm2:double;
begin
    tmpRe1:=aValue1.xRe;tmpIm1:=aValue1.xIm;
    tmpRe2:=aValue2.xRe;tmpIm2:=aValue2.xIm;
    re:=tmpRe1*tmpRe2-tmpIm1*tmpIm2;
    im:=tmpRe1*tmpIm2+tmpIm1*tmpRe2;
end;

```

```

procedure TKomplex.DoKali(const aRe1,aIm1,aRe2,aIm2:double);
begin
    re:=aRe1*aRe2-aIm1*aIm2;
    im:=aRe1*aIm2+aIm1*aRe2;
end;

procedure TKomplex.DoBagi(const aValue:TKomplex);
var tmpRe,tmpIm:double;
begin
    try
        tmpRe:=re;
        tmpIm:=im;

re:=(tmpRe*aValue.re+tmpIm*aValue.im)/(sqr(aValue.re)+sqr(aValue.im));
im:=(tmpIm*aValue.re-
tmpRe*aValue.im)/(sqr(aValue.re)+sqr(aValue.im));
    except
        MessageDlg('Pembagian salah atau bilangan pembagi nol!',
            mtwarning,[mbOK],0);
    end;
end;

procedure TKomplex.DoBagi(const aRe,aIm:double);
var tmpRe,tmpIm:double;
begin
    try
        tmpRe:=re;
        tmpIm:=im;
        re:=(tmpRe*aRe+tmpIm*aIm)/(sqr(aRe)+sqr(aIm));
        im:=(tmpIm*aRe-tmpRe*aIm)/(sqr(aRe)+sqr(aIm));
    except
        MessageDlg('Pembagian salah atau bilangan pembagi nol!',
            mtwarning,[mbOK],0);
    end;
end;

procedure TKomplex.DoBagi(const aValue1,aValue2:TKomplex);
var tmpRe1,tmpIm1,tmpRe2,tmpIm2:double;
begin
    try
        tmpRe1:=aValue1.xRe;tmpIm1:=aValue1.xIm;
        tmpRe2:=aValue2.xRe;tmpIm2:=aValue2.xIm;
        re:=(tmpRe1*tmpRe2+tmpIm1*tmpIm2)/(sqr(tmpRe2)+sqr(tmpIm2));
        im:=(tmpIm1*tmpRe2-tmpRe1*tmpIm2)/(sqr(tmpRe2)+sqr(tmpIm2));
    except
        MessageDlg('Pembagian salah atau bilangan pembagi nol!',
            mtwarning,[mbOK],0);
    end;
end;

procedure TKomplex.DoBagi(const aRe1,aIm1,aRe2,aIm2:double);
begin
    try
        re:=(aRe1*aRe2+aIm1*aIm2)/(sqr(aRe2)+sqr(aIm2));
        im:=(aIm1*aRe2-aRe1*aIm2)/(sqr(aRe2)+sqr(aIm2));
    except
        MessageDlg('Pembagian salah atau bilangan pembagi nol!',

```

```

        mtwarning, [mbOK], 0);
    end;
end;

procedure TKomplex.DoConj;
begin
    re:=re;
    im:=-im;
end;

procedure TKomplex.DoConj(const aValue:TKomplex);
begin
    re:=aValue.xRe;
    im:=-aValue.xIm;
end;

procedure TKomplex.DoConj(const aRe,aIm:double);
begin
    re:=aRe;
    im:=-aIm;
end;

procedure TKomplex.DoNegative;
begin
    re:=-re;
    im:=-im;
end;

procedure TKomplex.DoNegative(const aValue:TKomplex);
begin
    re:=-aValue.xRe;
    im:=-aValue.xIm;
end;

procedure TKomplex.DoNegative(const aRe,aIm:double);
begin
    re:=-aRe;
    im:=-aIm;
end;

procedure TKomplex.DoPangkat(pangkat:double);
var theta,sum:double;
begin
    try
        theta:=arctan(im/re);
        sum:=exp((pangkat/2)*ln(sqr(re)+sqr(im)));
        re:=sum*cos(pangkat*theta);
        im:=sum*sin(pangkat*theta);
    except
        MessageDlg('Bilangan tidak bisa dipangkatkan!',mtWarning,[mbOK],0);
    end;
end;

end.

```

```

unit Param;

interface

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

type
  TfrmParam = class(TForm)
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
    lblBeban: TLabel;
    GroupBox1: TGroupBox;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    edtmaxgen: TEdit;
    edtpopsize: TEdit;
    edtpmutasi: TEdit;
    edtKa: TEdit;
    btnUseDefault: TButton;
    edtBeban: TEdit;
    rbtJurnal: TRadioButton;
    rbtPLN: TRadioButton;
    GroupBox2: TGroupBox;
    CheckBox1: TCheckBox;
    CheckBox2: TCheckBox;
    CheckBox3: TCheckBox;
    CheckBox4: TCheckBox;
    CheckBox5: TCheckBox;
    CheckBox6: TCheckBox;
    CheckBox7: TCheckBox;
    CheckBox8: TCheckBox;
    CheckBox9: TCheckBox;
    CheckBox10: TCheckBox;
    CheckBox11: TCheckBox;
    CheckBox12: TCheckBox;
    CheckBox13: TCheckBox;
    CheckBox14: TCheckBox;
    CheckBox15: TCheckBox;
    CheckBox16: TCheckBox;
    CheckBox17: TCheckBox;
    CheckBox18: TCheckBox;
    CheckBox19: TCheckBox;
    CheckBox20: TCheckBox;
    CheckBox21: TCheckBox;
    CheckBox22: TCheckBox;
    CheckBox23: TCheckBox;
    CheckBox24: TCheckBox;
    CheckBox25: TCheckBox;
    CheckBox26: TCheckBox;
    CheckBox27: TCheckBox;
  end;

```

```

    CheckBox28: TCheckBox;
    CheckBox29: TCheckBox;
    CheckBox30: TCheckBox;
    CheckBox31: TCheckBox;
    TabSheet2: TTabSheet;
    Label1: TLabel;
    fgHasil: TStringGrid;
    edtTotalBiaya: TEdit;
    TabSheet3: TTabSheet;
    fgHasilEP: TStringGrid;
    TabSheet4: TTabSheet;
    Chart1: TChart;
    Series1: TLineSeries;
    Series2: TLineSeries;
    Series3: TLineSeries;
    btnHitungEP: TButton;
    btnClose: TButton;
    Label6: TLabel;
    procedure btnCloseClick(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    frmParam: TfrmParam;

implementation

{$R *.dfm}

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

end.

```