

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



**ANALISA *PEMBEBANAN EKONOMIS* DENGAN
MENGUNAKAN METODE *QUADRATIC HOPFIELD*
PADA PT. PLN PJB JAWA-BALI**

SKRIPSI

**Disusun Oleh :
JOHAN CAHYADI
NIM : 99.12.072**

SEPTEMBER 2006

LEMBAR PERSETUJUAN

**ANALISA PEMBEBANAN EKONOMIS DENGAN
MENGUNAKAN METODE *QUADRATIC HOPFIELD* PADA
PT. PLN PJB JAWA-BALI**

SKRIPSI

*Disusun dan Diajukan Untuk Melengkapi dan Memenuhi Syarat Guna Mencapai
Gelar Sarjana Teknik*

Disusun oleh

JOHAN CAHYADI

NIM : 99.12.072

**Mengetahui,
Ketua Jurusan Teknik Elektro**



Yudi Limpraptono, MT
NIP.Y. 1039500274

**Diperiksa dan Disetujui,
Dosen Pembimbing**

Ir. H. Taufik Hidayat, MT
NIP.P. 1018700151

**KOSENTRASI TEKNIK ENERGI LISTRIK
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG**

ABSTRAKSI

ANALISA *ECONOMIC DISPATCH* MENGGUNAKAN METODE *DIRECT SEARCH* DENGAN STRATEGI *MULTI-LEVEL CONVERGENCE* PADA PT. PEMBANGKITAN JAWA-BALI

(Johan Cahyadi, NIM 9912072, T. Elektro E.L S-1, 60 Hal, 2006)
(Dosen Pembimbing : Ir H. Taufik Hidayat, MT.)

Kata Kunci: *Economic Dispatch, Quadratic Hopfield*

Dalam skripsi ini membahas masalah optimasi untuk biaya bahan bakar pada suatu sistem tenaga listrik yang hasil dari analisis tersebut nantinya dapat digunakan sebagai salah satu acuan dalam mengefesiesikan biaya bahan bakar yang bisa digunakan oleh PT. Pembangkitan Jawa-Bali sehingga nilai kerugian yang sementara ini dialami oleh PT. Pembangkitan Jawa-Bali bisa berkurang, yang akhirnya nanti bisa menambah keuntungan bagi PT. PLN PJB sebagai perusahaan penyedia energi listrik di Indonesia. Untuk penyelesaian masalah *Economic Dispatch* adalah menggunakan metode *Quadratic Hopfield*, metode ini dipilih untuk menyelesaikan sejumlah kendala pertidaksamaan dan persamaan serta unit dengan beragam fungsi biaya bahan bakar. Program ini secara bersamaan, menghasilkan suatu rencana operasi yang memenuhi persyaratan pengoperasian sistem tenaga listrik dimana yang utama adalah daya yang dibangkitkan cukup memasok beban konsumen. Dengan menggunakan metode *Quadratic Hopfield* ternyata dapat melakukan optimasi yang lebih murah dibandingkan dengan optimasi PT. PLN PJB. Pada tanggal 10 Maret 2004 selisih biaya total operasional PT. PLN PJB dengan Metode *Quadratic Hopfield* 8.50 % atau Rp. 1.612.605.060. Pada tanggal 13 Maret 2004 selisih biaya total operasional PT. PLN PJB dengan Metode *Quadratic Hopfield* sebesar sebesar 7.25 % atau Rp.1.296.507.890. Pada tanggal 14 Maret 2004 selisih biaya total operasional PT. PLN PJB dengan Metode *Quadratic Hopfield* sebesar 4.86 % atau Rp. 1.953.303.020

KATA PENGANTAR

Dengan memanjatkan puji syukur kehadiran Allah SWT. Atas Karunia dan Hidayah-Nya, skripsi ini dapat terselesaikan dengan baik, guna memenuhi persyaratan untuk mendapat gelar Sarjana Teknik pada jurusan Teknik Elektro Konsentrasi Energi Listrik, Institut Teknologi Nasional Malang.

Penyusun Menyampaikan terima kasih kepada pihak yang telah membantu, membimbing dan mengarahkan mulai dari awal hingga terselesainya skripsi ini. Ucapan terimah kasih saya ucapkan kepada yang terhormat :

1. Bapak Prof. Dr. Ir. Abraham Lomi, MSEE, selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. Muchtar Asroni, MSEE, selaku Dekan Fakultas Teknologi Industri, Institut Teknologi Nasional Malang
3. Bapak Ir. F. Yudi Limpraptono, MT., Selaku Ketua Jurusan Teknik elektro, Institut Teknologi Nasional Malang.
4. Bapak Ir. H. Taufik Hidayat, MT. selaku dosen pembimbing yang memberikan masukan, dorongan semangat dan bantuannya.
5. Rekan-rekan Elektro khususnya angkatan 1999 atas segala kebersamaannya selama ini.
6. Serta semua pihak yang turut membantu dalam menyelesaikan skripsi ini.

Kami menyadari bahwa skripsi ini masih jauh sempurna, oleh sebab itu kritik dan saran akan kami terima dengan senang hati.

Akhir kata penulis berharap semoga skripsi ini dapat bermanfaat bagi kita semua dan memperkaya ilmu pengetahuan khususnya bagi mahasiswa pada Jurusan Teknik Elektro Energi Listrik.

Malang, September 2006

Penyusun

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
LEMBAR PERSETUJUAN	ii
ABSTRAKSI	iii
KATA PENGANTAR	iv
DAFTAR ISI	vi
DAFTAR GAMBAR	ix
DAFTAR TABEL	x
DAFTAR GRAFIK	xii
 BAB I PENDAHULUAN	
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan Penelitian	2
1.4. Batasan Masalah	3
1.5. Metodologi Penelitian	3
1.6. Sistematika Penulisan	4
1.7. Kontribusi	5
 BAB II TEORI DASAR	
2.1. Sistem Tenaga Listrik	6
2.2. Operasi Sistem Tenaga Listrik.....	7

2.3. Karakteristik Pembangkit	9
2.3.1. Karakteristik <i>Input-Output</i>	11
2.3.2. Karakteristik <i>Heat Reat</i>	13
2.3.3. Karakteristik <i>Incremental Hate Rate</i> dan <i>Incremental Fuel Cost</i>	14
2.4. <i>Economic Dispatch</i>	16
2.4.1. Penyelesaian <i>Economic Dispatch</i> dengan <i>Metode Pengali La Grange</i>	16
2.4.2. Penyelesaian <i>Economic Dispatch</i> dengan <i>Metode Iterasi Lamda</i>	19
2.5. Fungsi Biaya Bahan Bakar	20
2.6. <i>Economic Dispatch</i> dengan Mengabaikan Rugi-Rugi Transinisi	28

BAB III APLIKASI METODE *QUADRATIC HOPFIELD* UNTUK ECONOMIC DISPATCH

3.1. Penyelesaian dengan Quadratic Hopfield	25
3.1.1. Quadratic Hopfield	25
3.1.2. Metode Pengaturan Slope	27
3.1.3. Momentum	29
3.2. Algoritma Quadratic Hopfield	30
3.3. Flow Chart Quadratic Hopfield	31

BAB IV ANALISA DATA MENGGUNAKAN METODE *QUADRATIC*

HOPFIELD

4.1. Pendahuluan.....	33
4.2. Program Komputer <i>Economic Dispatch</i> Menggunakan Metode <i>Quadratic Hopfield</i>	33
4.3. Data Pembangkit Thermal.....	34
4.4. Aplikasi Metode <i>Quadratic Hopfield</i> di PT. PLN PJB.....	36
4.5. Beban Sistem.....	37
4.6. Hasil Perhitungan dan Analisa Data.....	39
4.6.1....Hasil Perhitungan PT. PLN PJBsebelum optimasi dan sesudah optimasi.....	39
4.6.2. Tampilan Program Komputer dan Hasil Perhitungan dengan Metode <i>Quadratic Hopfield</i>	43
4.6.2.1. Hasil Perhitungan Menggunakan Metode <i>Quadratic Hopfield</i>	48
4.6.3. Perbandingan Hasil Perhitungan PT. PLN PJB dengan Metode <i>Quadratic Hopfield</i>	51

BAB V KESIMPULAN DAN SARAN

5.1. Kesimpulan	57
5.2. Saran	58

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar	Halaman
Gambar 2.1. Unit <i>Boiler-Turbin-Generator</i>	12
Gambar 2.2. Kurva Karakteristik <i>Input-Output</i> Pembangkitan Thermal.....	13
Gambar 2.3. Kurva Karakteristik <i>hate rate</i> Unit Pembangkit.....	14
Gambar 2.4. Kurva Karakteristik <i>Incremental Fuel Cost Rate</i>	15
Gambar 2.5. N Unit Melayani Beban P_R	17
Gambar 2.6. Grafik Penyelesaian Dengan Metode Iterasi Lamda.....	20
Gambar 2.7. N Unit Pembangkit Thermal Melayani Beban P_R	22
Gambar3.1. Fungsi Perubahan sigmoidal dengan Perbedaan nilai dari Pendekatan Parameter	27
Gambar 4.1 Tampilan Program Utama.....	43
Gambar 4.2. Tampilan Input Data Pembangkitan.....	44
Gambar 4.3 Tampilan data generator.....	44
Gambar 4.4. Tampilan Data Pembebanan Pembangkit.....	45
Gambar 4.5 Tampilan Data Pembebanan PT, PLN PJB.....	45
Gambar 4.6. Tampilan Parameter yang Digunakan.....	46
Gambar 4.7. Hasil Optimasi Menggunakan <i>Quadratic Hopfield</i>	47

DAFTAR TABEL

Tabel	Halaman
4.1. Data Unit Termal PT. PLN. PJB	35
4.2. Unit Termal Siap Beroperasi	36
4.3. Data Beban Sistem Unit Termal	38
4.4. Hasil Perhitungan Biaya Operasional Perjam PT. PLN. PJB Rabu 10 Maret 2004.....	40
4.5. Hasil Perhitungan Biaya Operasional Perjam PT. PLN. PJB Sabtu 13 Maret 2004.....	41
4.6. Hasil Perhitungan Biaya Operasional Perjam PT. PLN. PJB Kamis 14 Maret 2004.....	42
4.7. Data Daya Generator PT. PLN. PJB Sebelum Dioptimasi Rabu 10 Maret 2004	48
4.8. Data Daya Generator PT. PLN. PJB Sesudah Dioptimasi Rabu 10 Maret 2004	49
4.9. Data Daya Generator PT. PLN. PJB Sebelum Dioptimasi Sabtu 13 Maret 2004	50
4.10. Data Daya Generator PT. PLN. PJB Sesudah Dioptimasi Sabtu 13 Maret 2004	51
4.11. Data Daya Generator PT. PLN. PJB Sebelum Dioptimasi Sabtu 14 Maret 2004	52

4.12. Data Daya Generator PT, PLN, PJB Sesudah Dioptimasi Sabtu 13 Maret 2004	53
4.13. Hasil Perhitungan dgn Metode Quadratic Hopfield 10 Maret 2004	55
4.14. Hasil Perhitungan dgn Metode Quadratic Hopfield 13 Maret 2004	56
4.15. Hasil Perhitungan dgn Metode Quadratic Hopfield 14 Maret 2004	57
4.16. Total Perbandingan Biaya Operasional perjam PT, PLN, PJB dg Metode Quadratic Hopfield	58
4.17. Perbandingan Total Biaya Operasional PT, PLN, PJB dg Metode Quadratic Hopfield	59

DAFTAR GRAFIK

Grafik 4.1. Perbandingan biaya operasional perjam PT, PLN, PJB dan Metode Quadratic Hopfield Rabu 10 Maret 2004	59
Grafik 4.2. Perbandingan Biaya Operasional Perjam PT, PLN, PJB dan Metode Quadratic Hopfield Sabtu 13 Maret 2004	59
Grafik 4.3. Perbandingan Biaya Operasional Perjam PT, PLN, PJB dan Metode Quadratic Hopfield Minggu 14 Maret 2004	60
Grafik 4.4. Perbandingan Total Biaya Operasional Perjam PT, PLN, PJB Dengan Metode Quadratic Hopfield	60

BAB I

PENDAHULUAN

1.1. Latar Belakang

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

Dalam pembangkitan tenaga listrik dilakukan usaha agar biaya pembangkitannya semurah mungkin. Usaha untuk mengoptimalkan biaya operasi ini, salah satunya dilakukan dengan penerapan *Economic Dispatch*. Di dalam operasi sistem tenaga listrik, *Economic Dispatch* adalah hal yang sangat perlu diperhatikan untuk mendapatkan biaya bahan bakar yang sangat ekonomis dalam suatu sistem pembangkit.

Koordinasi antara unit-unit pembangkit yang ada pada sistem tenaga listrik sangat diperlukan untuk mencapai biaya operasi yang seoptimal mungkin. Pada

skripsi ini akan dibahas metode alternatif masalah optimasi biaya pembangkitan dengan mengoptimalkan biaya operasi dengan penerapan *Economic Dispatch* menggunakan metode *Quadratic Hopfield*.

1.2. Permasalahan

Berdasarkan latar belakang diatas, maka muncul permasalahan yang ada adalah bagaimana mengoptimalkan biaya operasional pembangkitan dan menerapkan *Economic Dispatch* yang dihasilkan oleh pembangkit thermal. Sehubungan dengan itu, maka judul skripsi ini adalah :

**“ ANALISA PEMBAGIAN PEMBEBANAN EKONOMIS DENGAN
MENGUNAKAN METODE *QUADRATIC HOPFIELD* PADA PT.
PEMBANGKITAN JAWA-BALI ”.**

1.3. Tujuan

Tujuan dari skripsi ini adalah :

- ⊗ Memberikan analisa penerapan penbebanan ekonomis untuk mengoptimasikan biaya operasi pada system tenaga listrik dengan memperhatikan mutu dan keandalan.
- ⊗ Mengoptimasi perubahan pada beban system harian pada unit – unit pembangkit tenaga listrik tiap jam secara efisien
- ⊗ Melakukan pembagian pembebanan terhadap unit pembangkit yang nilai operasinya rendah dengan tetap memperhatikan kendala-kendala operasi dan tetap memenuhi beban system.

1.4. Batasan Masalah

Apabila ditinjau lebih jauh, permasalahan dalam sistem tenaga listrik merupakan permasalahan yang luas, sehingga untuk membatasi apa saja yang dibahas maka ditentukan bahwa skripsi ini :

1. Tidak membahas masalah rugi-rugi transmisi.
2. Tidak membahas biaya star-up dan biaya shut down.
3. Daya listrik yang dihasilkan setiap unit pembangkit harus disertai beberapa batasan (kapasitas minimum dan maksimum).
4. Tidak membahas masalah *spinning reserve* atau cadangan berputar.
5. Tidak membahas metode lain selain Metode *Quadratic Hopfield*.
6. Hanya membahas unit pembangkit thermal dan dibatasi pada pembangkit thermal tenaga uap dan gas yang ada pada lingkungan kerja P.T. Pembangkit Jawa-Bali.
7. Pembahasan di titik beratkan pada segi ekonomis dari pada teknis.

1.5. Metodologi Penelitian

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

1. Studi kepustakaan mengenai hal-hal yang berhubungan dengan pembahasan masalah.
2. Studi lapangan untuk mendapatkan data parameter unit thermal yang dibutuhkan dari obyek penelitian yaitu di PT. PLN PJB yang

Bab IV : Hasil Simulasi Program

Dalam bab ini diuraikan tentang seluruh hasil simulasi program untuk menyelesaikan masalah Economic Dispatch serta hasil perhitungan, analisa dan evaluasinya.

Bab V : Kesimpulan dan Saran.

1.7. Kontribusi

Dari hasil peneititan ini nantinya diharapkan dapat menjadi solusi alternative pada P.T. PLN (Persero) Pembangkitan Jawa – Bali, dalam usaha untuk meminimalkan biaya operasi khususnya biaya pembangkitan yang merupakan biaya terbesar pada sistem tenaga listrik.

BAB II

TEORI DASAR

2.1. Sistem Tenaga Listrik^[1]

Untuk keperluan penyediaan tenaga listrik bagi para pelanggan diperlukan berbagai peralatan listrik. Peralatan-peralatan tersebut dihubungkan satu sama lain yang saling berhubungan dan secara keseluruhan membentuk suatu sistem tenaga listrik. Yang dimaksud dengan sistem tenaga listrik adalah satu kesatuan yang terintegrasi antara pembangkit tenaga listrik, Gardu induk (pusat beban) yang satu sama yang lain dihubungkan.

Pengelolaan sistem tenaga listrik merupakan persoalan yang cukup rumit, sehingga diperlukan suatu manajemen operasi yang baik. Manajemen operasi sistem tenaga listrik harus memikirkan bagaimana menyediakan tenaga listrik yang seekonomis mungkin dengan tetap memperhatikan mutu dan keandalan. Mutu dan keandalan diukur dengan frekuensi, tegangan dan jumlah gangguan. Masalah mutu tenaga listrik tidak semata-mata merupakan masalah operasi tenaga listrik tetapi erat kaitanya dengan pemeliharaan instalasi tenaga listrik dan juga pengembangan sistem tenaga listrik karena mengingat konsumsi tenaga listrik oleh pelanggan selalu bertambah dari waktu ke waktu oleh karena itu hasil-hasil operasi sistem tenaga listrik perlu dianalisa dan dievaluasi untuk menjadi masukan bagi pemeliharaan instalasi serta pengembangan sistem tenaga listrik. Mutu tenaga listrik yang merupakan kendala / *counstrain* terhadap biaya pengadaan

tenaga listrik yang serendah mungkin, maka kompromi antar kedua hal yang merupakan masalah optimasi yang cukup kompleks.

Daya yang tersedia dalam sistem tenaga listrik tergantung kepada daya yang terpasang unit-unit pembangkit dan juga pada kesiapan operasi unit-unit tersebut. Berbagai faktor seperti gangguan kerusakan dan pemeliharaan rutin, menyebabkan unit pembangkit menjadi tidak siap operasi.

2.2. Sistem Operasi Pada Sistem Tenaga Listrik

Seperti diketahui bahwa dalam masalah pengaturan beban pada suatu operasi sistem tenaga listrik harus selalu dicapai suatu keadaan operasi yang bisa diandalkan dan cukup ekonomis.

Ada beberapa kinerja yang dilakukan untuk menjamin keandalan sistem operasi antara lain, pengaturan frekuensi dan tegangan sistem untuk berada pada harga normalnya karena adanya perubahan beban sistem. Dan seperti yang diketahui dan berulang kali disebutkan bahwa tenaga listrik tidak dapat disimpan sehingga dalam operasinya harus selalu dicapai keseimbangan antara penyediaan dengan pemenuhan kebutuhan daya serta perlu juga diingat bahwa sistem selalu berubah setiap saat. Maka sudah tentu jauh-jauh sebelumnya sudah harus diketahui atau diramalkan keadaan tersebut dengan tetap yaitu keadaan beban pada hari itu dari waktu ke waktu sampai selama 24 jam. Keadaan beban ini digambarkan sebagai kebutuhan daya sebagai fungsi dari waktu yang disebut dengan lengkung beban harian. Lengkung beban harian ini adalah merupakan suatu yang amat penting disamping karakteristik-karakteristik lainnya sehingga

dalam operasi hariannya harus berdasarkan lengkung beban harian yang telah dibuat karena dengan lengkung beban ini dapat ditentukan perencanaan operasi pembangkit – pembangkit yang ada, baik itu unit pembangkit thermal ataupun hidro. Tentu saja kebutuhan beban dalam suatu harinya tidak merata akan tetapi dari jam ke jam berbeda sesuai dengan kebutuhan konsumen. Berdasarkan lengkung beban yang telah ada maka dapat ditentukan beberapa unit pembangkit yang harus bekerja dan siap bekerja pada hari itu.

Sebagai dasar pertimbangan yang sifatnya umum, untuk menentukan biaya produksi tenaga listrik yang dibutuhkan adalah dengan memperhatikan bahwa dalam keadaan beban minimum maka tenaga listrik yang dibutuhkan diberikan oleh unit pembangkit yang bekerja paling efisien pada keadaan tersebut. Pembangkit ini akan terus beroperasi atau dibebani sampai pada batas efisiensi maksimumnya. Dan apabila ternyata beban masih terus bertambah sedangkan unit pembangkit ini telah mencapai maksimumnya maka selanjutnya belum ditanggung oleh pembangkit yang lain yang belum mencapai efisiensi maksimumnya. Dengan operasi yang demikian maka dapat dicapai keadaan operasi yang cukup ekonomis.

Akan tetapi dengan semakin berkembangnya sistem itu sendiri maka diperlukan suatu perencanaan pembangkit yang optimum dengan biaya operasi yang ekonomis. Mengingat bahwa beban sistem adalah selalu berubah-ubah dari waktu ke waktu maka perlu membuat secara grafis perubahan beban terhadap waktu.

Oleh karena biaya operasi untuk memproduksi daya listrik sangat besar, suatu pembangkit khususnya pembangkit thermal, maka untuk biaya operasi pembangkit harus ditekan seekonomis mungkin untuk mendapatkan biaya operasi yang rendah, karena pada unit pembangkit thermal ini akan membutuhkan biaya operasi yang cukup tinggi sehingga usaha penghematan biaya bahan bakar akan sangat berarti. Dengan kata lain dengan mengkoordinasikan operasi pembangkit-pembangkit yang tersedia dengan tepat dan sesuai dengan beban maka didapat suatu keadaan operasi yang ekonomis.

Pembahasan mengenai operasi ekonomis adalah merupakan salah satu cara bagaimana menekan biaya produksi dari sistem tenaga listrik. Dalam hal ini maka metode yang dipakai adalah dengan memanfaatkan karakteristik dan menganalisa operasi dari sistem tersebut. Disamping karakteristik dari unit-unit pembangkit perlu juga diketahui karakteristik beban, karena karakteristik bebanlah maka dapat dianalisa pengaturan yang paling ekonomis dari setiap pembangkit. Adapun karakteristik yang perlu diketahui dari setiap unit pembangkit adalah :

1. Karakteristik input bahan bakar sebagai fungsi output daya.
2. Nilai panas sebagai fungsi output daya.
3. Kenaikan jumlah bahan bakar yang dibutuhkan jika terdapat perubahan beban.

Ketiga karakteristik tersebut merupakan pedoman menganalisa penjadwalan selanjutnya. Kemudian yang perlu diperhitungkan adalah Variabel-variabel yang terdapat pada saluran transmisi, karena variabel-variabel ini juga

sangat menentukan ekonomis tidaknya penjadwalan pembangkit yang kita tentukan.

Maka untuk mencapai suatu operasi yang ekonomis pada suatu sistem tenaga listrik adalah dengan melakukan penjadwalan pada sistem pembangkit yang ada pada suatu sistem tenaga listrik yang ditinjau tersebut dengan memanfaatkan karakteristik dari setiap masing-masing unit pembangkit yang ada pada dasarnya bertujuan untuk menekan biaya pembangkit agar diperoleh biaya yang sangat rendah sehingga dapat memuaskan pemakai energi listrik.

2.3. 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 termis pusat pembangkit tersebut. Selain tergantung dari sifat-sifat pusat pembangkit tenaga listrik itu sendiri, seperti efisien dan keandalan, lengkung masukan – keluaran itu juga tergantung dari kondisi-kondisi yang berada di luar pusat pembangkit 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 di atas.

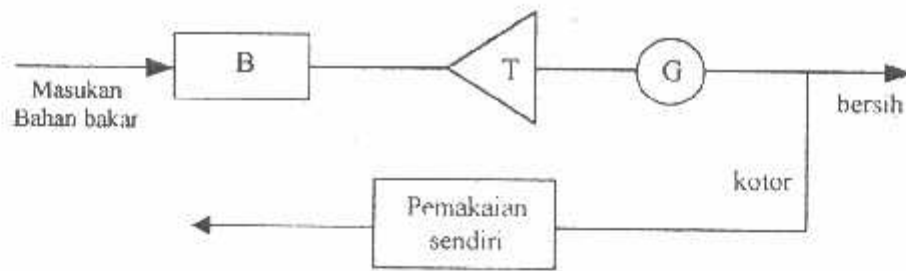
2.3.1. Karakteristik Input – Output

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 tes pembangkit. Adapun definisi dari karakteristik *Input – Output* pembangkit adalah formula yang menyatakan hubungan antara input pembangkit sebagai fungsi dari output pembangkit. Karakteristik dari pembangkit unit boiler – turbin – generator dapat digambarkan dalam gambar 2.1. dimana unit ini memuat sebuah boiler yang menghasilkan uap untuk menjalankan 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 uang 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 :

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

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



Keterangan : B = Boiler, T = Turbin Uap, G = Generator

Gambar 2-1
Unit Boliér-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).

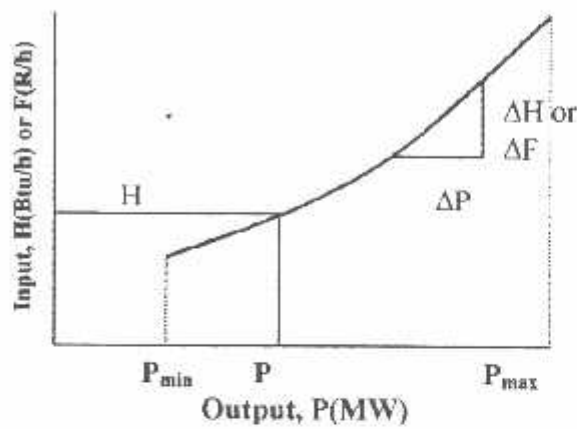
Generator akan mengeluarkan daya sesuai dengan beban yang ada. Semakin besar beban, semakin besar daya yang dikeluarkan oleh generator. Daya yang dikeluarkan generator dapat membesar sesuai dengan peningkatan beban sampai dengan daya maksimum yang dapat dibangkitkan oleh generator. Semakin besar daya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

Dari keterangan di atas, 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.

$$H = f(P).....(2.2)$$

$$\text{atau } F = f(P).....(2.3)$$

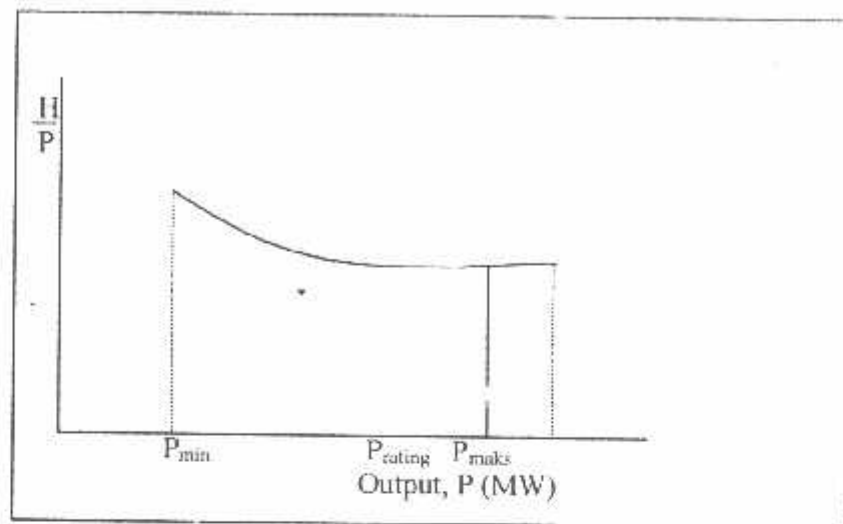


Gambar 2.2. Kurva karakteristik *Input-Output* Pembangkit Thermal^[1]

2.3.2. Karakteristik Heat Rate

Karakteristik lain yang cukup penting bagi pmbangkit 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 efisien 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 (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^[1]

2.3.3. Karakteristik Incremental Heat Rate dan Incremental Fuel Cost

Perwujudan yang lain dari karakteristik pembangkit adalah karakteristik *Incremental Heat Rate* atau perubahan tingkat laju panas dan karakteristik *Incremental Fuel Cost* atau 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 hubungan 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.

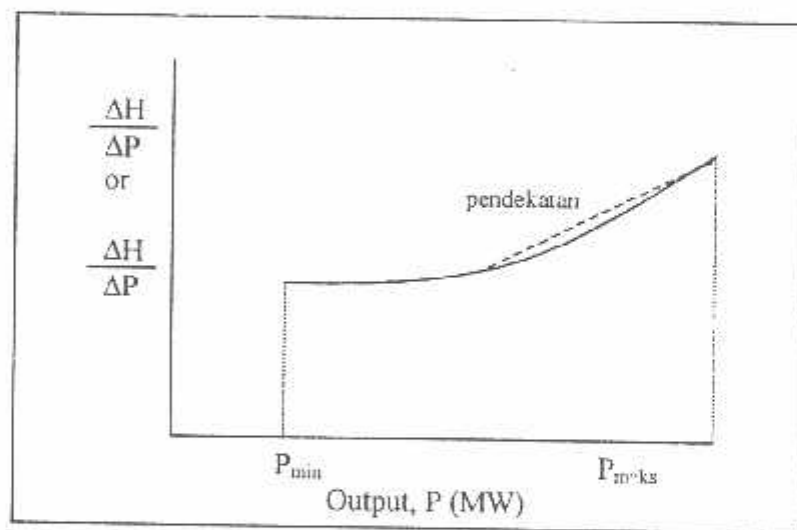
$$\text{Incremental Heat Rate} = \frac{\Delta H}{\Delta P} \left[\frac{\text{MBTU}}{\text{kwh}} \right] \dots\dots\dots (2.4)$$

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

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*^[1]

2.4. Economic Dispatch

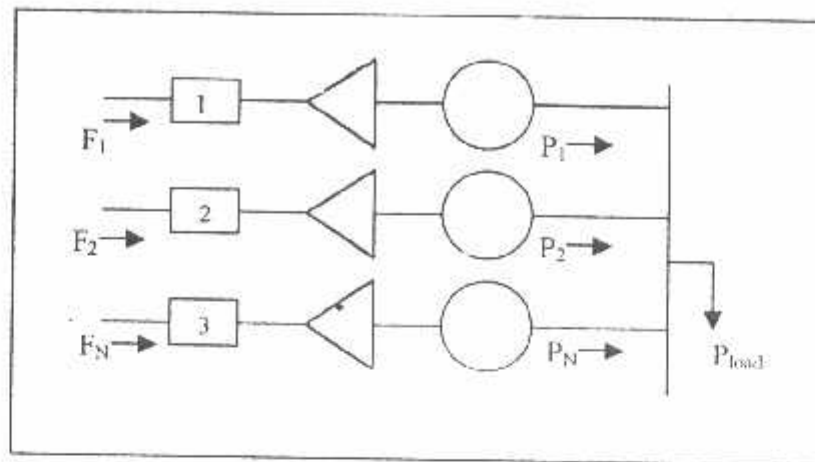
Yang dimaksud *Economic Dispatch* adalah pembagian pembebanan pada pembangkit-pembangkit yang ada dalam sistem, secara optimal ekonomi pada harga beban tertentu. Tujuan dari economic dispatch adalah untuk mendapatkan biaya bahan bakar semurah mungkin dalam suatu sistem pembangkit pada beban tertentu. Dengan dilakukan economic dispatch maka didapatkan biaya bahan bakar daya yang paling murah dalam suatu sistem pembangkit. Oleh karena 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 di bawah ini.

2.4.1. 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 bus bar yang melayani beban P_R . Input dari masing-masing unit ditunjukkan oleh F_i yang mewakili biaya dari satu unit generator dan output dari masing-masing unit P_i adalah daya yang dihasilkan oleh satu unit generator.

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



Gambar 2.5. N Unit Melayani Beban P_R

Yang menjadi permasalahan adalah meminimumkan total biaya F_T dengan memperhatikan pembatas ϕ bahwa daya yang dihasilkan generator sama dengan yang diterima beban. Secara matematika pernyataan tersebut di atas dapat dinyatakan dengan persamaan 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 menggunakan metode kalkulus tingkat lanjut yang melibatkan fungsi La Grange. Dimana fungsi La Grange didapat dengan cara menambahkan pembatas ϕ yang telah dikalikan dengan faktor pengali La Grange

ϕ pada fungsi F_T . Fungsi La Grange dapat ditunjukkan dengan persamaan di bawah ini :

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

Dimana :

F_T = fungsi tujuan

λ = faktor pengali

ϕ = fungsi pembatas (constrain) *

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

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

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

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

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

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

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 dan lebih kecil atau sama dengan daya output maksimum yang diijinkan.

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

$$\begin{aligned} \frac{\partial F_i}{\partial P_i} &= \lambda \quad \text{ada N buah persamaan} \\ P_{i \min} &\leq P_i \leq P_{i \max} \quad \text{ada 2 N buah pertidaksamaan.....(2.15)} \\ \sum_{i=1}^N P_i &= P_R \quad \text{ada 1 buah pembatas} \end{aligned}$$

Dari batasan pertidaksamaan pembatas diatas dapat diperluas menjadi :

$$\begin{aligned} \frac{\partial F_i}{\partial P_i} &= \lambda \quad \text{untuk } P_{i \min} \leq P_i \leq P_{i \max} \\ \frac{\partial F_i}{\partial P_i} &\leq \lambda \quad \text{untuk } P_i = P_{i \max} \dots\dots\dots(2.16) \end{aligned}$$

$$\frac{\partial F_i}{\partial P_i} \geq \lambda \quad \text{untuk } P_i = P_{i \min} \dots\dots\dots(2.17)$$

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

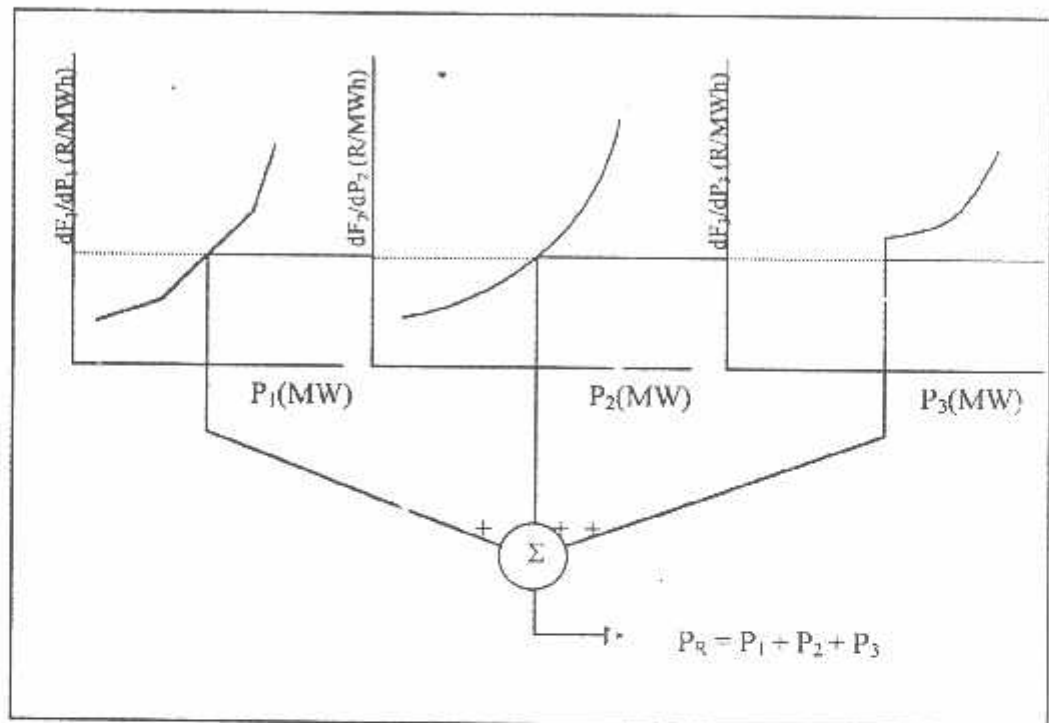
2.4.2. Penyelesaian Economic Dispatch dengan metode iterasi Lamda

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 di atas 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]

2.5. Fungsi Biaya Bahan Bakar

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

$$F_i(P_i) = a_i + b_i P_i + c_i P_i^2 \dots\dots\dots(2.18)$$

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

P_i = 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 :

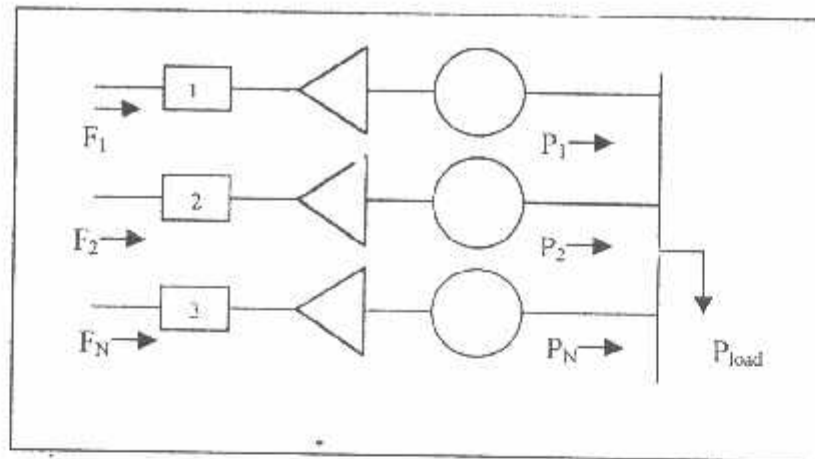
- Jenis bahan bakar
- Nilai kalori
- Harga bahan bakar

2.6. 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

2.7. Sistem ini terdiri dari N unit generator thermal yang dihubungkan pada single bus bar yang melayani beban P_d . 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.



Gambar 2.7,

N Unit Pembangkit Thermal Melayani Beban P_R ^[1]

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 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 generator sama dengan yang diterima beban. Secara matematika pernyataan tersebut di atas dapat dinyatakan dengan persamaan berikut :

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

$$= \sum_{i=1}^N F_i(P_i) \dots \dots \dots (2.19)$$

$$\phi = 0 = P_R - \sum_{i=1}^N P_i \dots \dots \dots (2.20)$$

Persamaan ini adalah pembatas yang merupakan masalah dari optimasi dan ini dapat dipecahkan dengan menggunakan metode kalkulus tingkat lanjut

yang melibatkan fungsi La Grange. Dimana fungsi La Grange didapat dengan cara menambahkan pembatas ϕ yang telah dikalikan dengan faktor pengali La Grange λ pada fungsi F_T . Fungsi La Grange dapat ditunjukkan dengan persamaan di bawah ini :

$$L = F_T + \lambda \cdot \phi \dots\dots\dots(2.21)$$

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

$$\nabla L = 0 \dots\dots\dots(2.22)$$

$$\nabla F_T + \lambda \phi = 0 \dots\dots\dots(2.23)$$

$$\frac{\partial L}{\partial P} = \frac{\partial F_T}{\partial P_i} + \lambda \left[\frac{\partial P_R}{\partial P_i} - \frac{\partial P_i}{\partial P_i} \right] = 0 \dots\dots\dots(2.24)$$

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

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

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 diijinkan.

Dari N buah unit generator pen bangkit dalam system tenaga nyang telah dibahas dan beban system sebesar P_R , maka dapat diambil kesimpulan sebagai berikut :

$$\frac{\partial F_i}{\partial P_i} = \lambda \dots\dots\dots(2.26)$$

$$P_{i \min} \leq P_i \leq P_{i \max} \text{ ada 2 N buah pertidaksamaan} \dots\dots\dots(2.27)$$

$$\sum_{i=1}^N P_i = P_R \dots\dots\dots(2.28)$$

Dari batasan pertidaksamaan pembatas diatas dapat diperluas menjadi :

$$\frac{\partial F_i}{\partial P_i} = \lambda \text{ untuk } P_{i \min} \leq P_i \leq P_{i \max} \dots\dots\dots(2.29)$$

$$\frac{\partial F_i}{\partial P_i} \leq \lambda \text{ untuk } P_i = P_{i \max} \dots\dots\dots(2.30)$$

$$\frac{\partial F_i}{\partial P_i} \geq \lambda \text{ untuk } P_i = P_{i \min} \dots\dots\dots(2.31)$$

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 *QUADRATIC HOPFIELD* UNTUK *ECONOMIC DISPATCH*

3.1. Penyelesaian dengan Quadratic Hopfield

Pendekatan konvensional untuk memecahkan masalah *Economic Dispatch* dengan menggunakan metode Quadratic Hopfield membutuhkan jumlah pengulangan (iterasi) yang banyak dan seringkali berubah-ubah selama transien. Untuk mempercepat keseimbangan (convergence), maka metode pengaturan quadratic di bahas dalam skripsi ini : Metode pengaturan slope.

3.1.1. Quadratic Hopfield

Model neuron kontinyu adalah generalisasi dari jaringan Hopfield dimana energi perhitungan turun secara kontinyu seiring waktu. Untuk parameter keuangan sangat tinggi (λ) dari neuron, jaringan kontinyu melakukan dengan cara serupa seperti model diskrit. Ketika vector parameter nilai adalah simetris, maka fungsi energi dari jaringan Hopfield didefinisikan sebagai berikut ;

$$E = -1/2 \sum_i \sum_j T_{ij} V_i V_j - \sum_i I_i V_i + \sum_i \theta_i V_i \dots\dots\dots(3.1)$$

Dimana V_i adalah harga output dari neuron i , I_i adalah input, keluaran pada neuron i , dan θ_i adalah ambang batas bias.

Pergerakan (*dynamics*) dari neuron didefinisikan :

$$\frac{dU_i}{dt} = \sum_j T_{ij} V_j + I_i \dots \dots \dots (3.2)$$

dimana U_i adalah total keluaran pada neuron i dan fungsi sigmoid bias didefinisikan :

$$V_i = g_i (\lambda U_i) = g_i \left(\frac{U_i}{U_0} \right) = \frac{1}{1 + \exp \left(- \frac{U_i + \theta_i}{U_0} \right)} \dots \dots \dots (3.3)$$

Stabilitas diketahui untuk menjamin dimana fungsi energi keluar batas dan dimana kenaikannya ditemukan menjadi tidak positif.

$$\frac{dE}{dt} = - \sum_i g_i (U_i) \left(\frac{dU_i}{dt} \right)^2 \dots \dots \dots (3.4)$$

Ketika $g_i (U_i)$ adalah fungsi kenaikan non monotone, maka masing-masing bentuk dari jumlah ini adalah tidak negatif. Oleh karena itu persamaan (3.4) akan lebih kecil dari nol. Evolusi waktu dari sistem adalah sebuah gerakan pada ruang yang mencari E minimum dan sampai berhenti pada angka tersebut.

Untuk memecahkan masalah ED, fungsi energi dapat didefinisikan dengan persamaan sebagai berikut :

$$E = \frac{1}{2} A (D + L - \sum P_i)^2 + \frac{1}{2} B \sum (a_i + b_i P_i + c_i P_i^2) \dots \dots \dots (3.5)$$

dimana a_{ik} , b_{ik} , c_{ik} adalah biaya koefisien sebagai fungsi discrete P_i .

Dengan membandingkan (3.5) dengan (3.1) dimana mendekati nol, parameter bobot dan input eksternal pada neuron i dalam jaringan diberikan sebagai :

$$\begin{aligned} T_{ji} &= -A - Bc_i \\ T_{ij} &= -A \\ I_i &= A (D+L) - Bb_i / 2 \end{aligned} \dots \dots \dots (3.6)$$

dimana bobot diagonal adalah tidak nol. Hasil perubahan ini (3.2) kedalam bentuk update berikut :

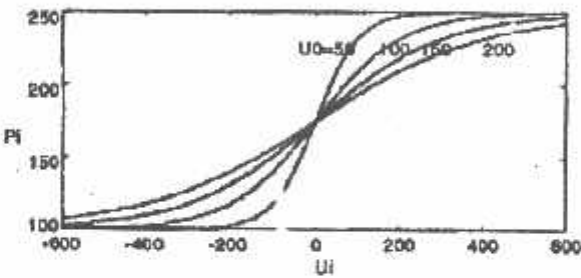
$$U_i(k)-U_i(k-1)=\sum T_{ij} V_j(k)+I_i \dots\dots\dots(3.7)$$

Fungsi sigmoid (3.3) dapat diubah menjadi bentuk batas daya sebagai berikut :

$$V_i(k+1)=(P_i^{max}-P_i^{min}) \frac{1}{1+\exp\left(-\frac{U_i(k)+\theta_i}{U_0}\right)}+P_i^{min} \dots\dots\dots(3.8)$$

3.1.2. Metode Pengaturan Slope

Dalam keadaan transien, input neuron berubah-ubah mendekati antara nol ditunjukkan dalam gambar 1.



Gambar 1. fungsi perubahan sigmoidal dengan perbedaan nilai dari pendekatan parameter

Beberapa input neuron berubah-ubah menjauhi dari nilai yang ditentukan. Jika parameter yang diperoleh terlalu tinggi, maka bentuk gelombang (osilasi) akan terjadi pada daerah saturasi. Jika parameter pada slope yang berada pada daerah tersebut terlalu tinggi maka neuron-neuron itu tidak akan dalam keadaan stabil dan akan menyebabkan ketidakstabilan.

Ketika energi diperkecil dan keseimbangannya tergantung pada parameter yang diperoleh U_o , maka metode gradient-descent dapat dipakai untuk menyesuaikan parameter tersebut sebagai berikut :

$$U_o(k+1) = U_o(k) - \eta_s \frac{\partial E}{\partial U_o} \dots\dots\dots (3.9)$$

dimana η_s adalah tingkat pembelajaran.

Dari (3.5) dan (3.8), gradien energi dengan mengacu kepada parameter yang diperoleh dapat dihitung sebagai berikut :

$$\frac{\partial E}{\partial U_o} = \sum_{i=1}^n \frac{\partial E \partial P_i}{\partial P_i \partial U_o} \dots\dots\dots (3.10)$$

Rumus terbaru (3.9) membutuhkan pilihan yang sesuai dengan tingkat pembelajaran η_s . Untuk sebuah nilai kecil η_s konvergensi dapat dipenuhi tetapi kecepatannya rendah, di lain sisi jika tingkat pembelajaran terlalu tinggi, algoritmanya menjadi tidak stabil. Untuk terpenuhinya konvergensi dan dengan cepat, sebuah metode untuk menghitung tingkat pembelajaran quadratic maka dikembangkan prosedur berikut. Ini dapat ditunjukkan bahwa konvergensi dapat terpenuhi jika tingkat pembelajaran η_s , dipilih sebagai berikut :

$$0 < \eta_s < \frac{2}{g_{s, \max}^2} \dots\dots\dots (3.11)$$

Dimana $g_{s, \max} := \max \|g_s(k)\|, g_s(k) = \partial E(k) / \partial U_o$

Lebih jauh lagi, konvergensi optimal akan sesuai dengan

$$\eta_s^* = \frac{1}{g_{s,max}^2} \quad \text{dimana } g_{s,max} = \max_{j \in \{1,2,\dots,n\}} \{g_{s,j}\} \quad (3.12)$$

Ini menunjukkan sebuah hasil yang menarik bahwa pada tingkat pembelajaran lain lebih besar dari pada η_s^* tidak memenuhi sebuah konvergensi yang cepat.

3.1.2. Momentum

Kecepatan konvergensi dapat diatur dengan menambahkan momentum dalam proses berikutnya. Momentum dapat dipakai ketika inputnya nanti dalam persamaan (3.7), parameter yang diperoleh berikutnya dalam persamaan (3.9):

$$U_i(k) - U_i(k-1) = \sum_j T_{ij} V_j(k) + \alpha_u \Delta U_i(k-1) \quad (3.13)$$

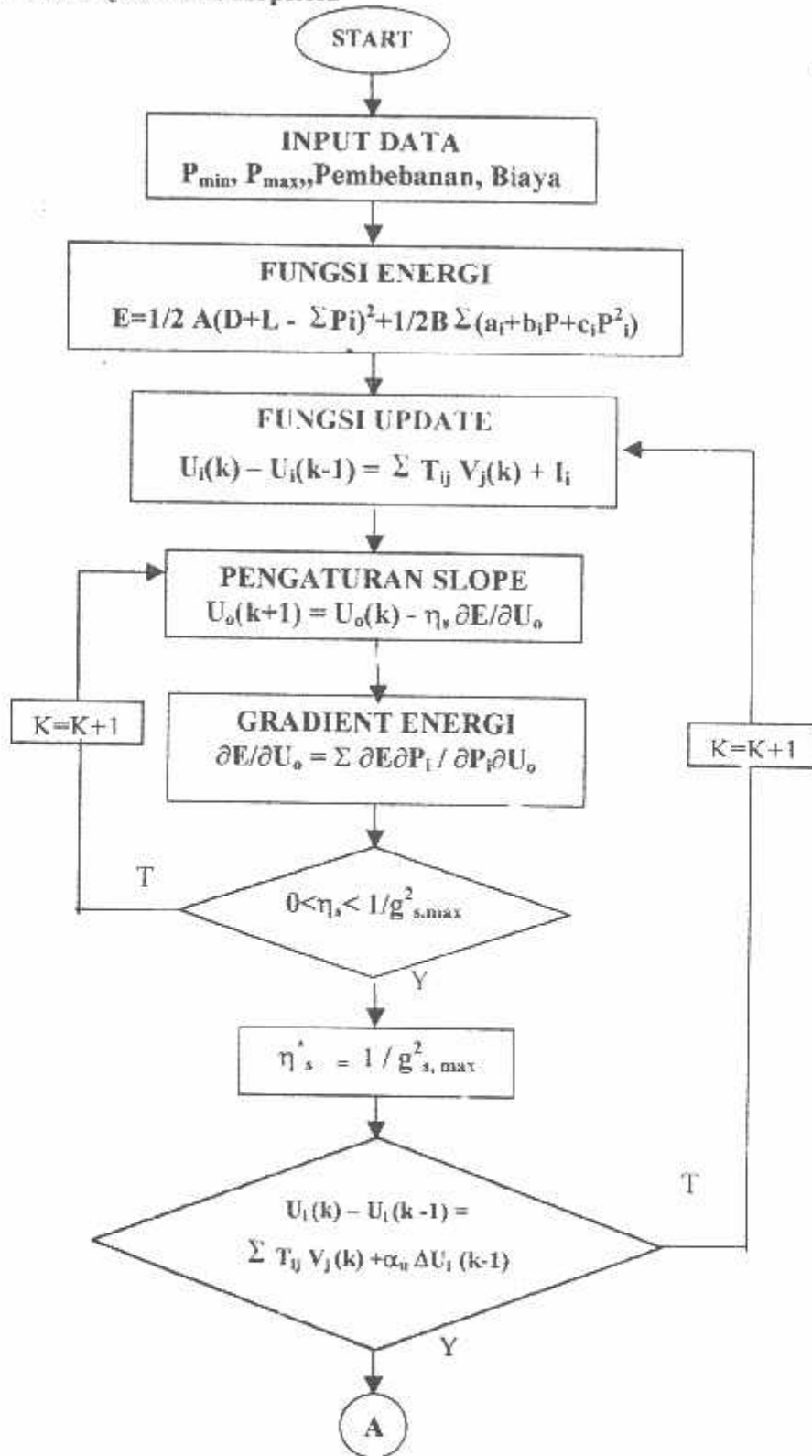
$$U_0(k) = U_0(k-1) - \eta_s \frac{\partial E}{\partial U_0} + \alpha_s \Delta U_0(k-1) \quad (3.14)$$

dimana α 's adalah faktor momentum

3.2. Algoritma Quadratic Hopfield

- Langkah 1 : Set input data P_{min} , P_{max} , Pembebanan, Koefisien Biaya
- Langkah 2 : Set Persamaan Fungsi Energi
- Langkah 3 : Set Fungsi Update $\sum T_{ij} V_j(k) + I_i$
- Langkah 4 : Set Pengaturan Slope
- Langkah 5 : Set Persamaan Gradient Energi
- Langkah 6 : Set Tingkat Pembelajaran η_s sampai mendekati nilai yang paling konvergen
- Langkah 7 : Jika tidak konvergen maka akan kembali ke langkah ke -4 di pengaturan slope
- Langkah 8 : Jika nilai Tingkat Pembelajaran η_s sudah konvergen maka diteruskan ke langkah selanjutnya
- Langkah 9 : Set persamaan momentum yaitu $U_i(k) = U_i(k-1)$.
- Langkah 10 : Jika langkah ke -9 belum sesuai maka kembali ke langkah ke -3 yaitu set update fungsi.
- Langkah 11 : Jika sudah konvergen maka kita teruskan ke persamaan slope yaitu $U_o(k)$
- Langkah 12 : Setelah nilai $U_o(k)$ diketahui maka kita dapat masukkan persamaan economic dispatch yaitu $F = a_{ik} + b_{ik}P_i + c_{ik}P_i^2$
- Langkah 13 : Kemudian langkah yang terakhir adalah mencetak hasilnya, dan proses sudah dikatakan selesai

3.3. Flow Chart Quadratic Hopfield



BAB IV

ANALISA DATA MENGGUNAKAN METODE *QUADRATIC* *HOPFIELD*

4.1. Pendahuluan

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

Dengan demikian merupakan suatu keharusan bagi seluruh jajaran P.T. Pembangkitan 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 Menggunakan Metode *Quadratic Hopfield*

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

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

4.3. Hasil uji validasi Metode

Uji validasi metode ini dilakukan test program. Yang digunakan test program adalah dari buku “ Power Generation, Operation, and Control ”, Allen J. Wood and Bruce F. wollenberg, second edition, hal 13 sampai sampai dengan hal 45. jumlah pembangkit yang digunakan sebanyak 3 unit dan periode waktu 1 jam dengan $P(\text{Daya}) = 800 \text{ MW}$, data-data selengkapnya seperti pada tabel 1. dan tabel 2.

Tabel 4.1
Kapasitas Pembangkit dan Koefisien Biaya Bahan Bakar

Unit Gen	P max (MW)	P min (MW)	A \$/h	B \$/MWh	c \$/MW ² h
Unit 1	600	100	561.0	7.92	0.001562
Unit 2	600	100	310.0	7.85	0.00194
Unit 3	200	50	78.0	7.97	0.00482

Tabel 4.2
Data Pembebanan

Unit Gen	Data Pembebanan (MW)
Unit 1	369.6871
Unit 2	315.6965
Unit 3	114.6164
Total	7738.8

Dari data validasi pada buku dapat dihitung biaya operasional tiap unit pembangkit dengan memasukkan ke persamaan fungsi biaya bahan bakar, sehingga berdasarkan data pada tabel 2. di dapatkan fungsi biaya bahan bakar :

$$F_i(P_{i,t}) = a_i + b_i P_{i,t} + c_i P_{i,t}^2$$

➤ Untuk unit 1 = 369.6871

$$F_1(P_{1,t}) = 561 + 7.92 (369.6871) + 0.001562 (369.6871)^2$$

$$= 3699.391$$

➤ Untuk unit 2 = 315.6965

$$F_2(P_{2,t}) = 310 + 7.85 (315.6965) + c_i (315.6965)^2$$

$$= 2977.58$$

➤ Untuk unit 3 = 114.6164

$$F_3(P_{3,t}) = 78 + 7.97 (114.6164) + 0.0048 (114.6164)^2$$

$$= 1054.55$$

Tabel 4.3. Hasil Perhitungan Biaya Data Jurnal

Unit	Data Jurnal	
	Daya (MW)	Biaya
Unit 1	369.6871	3699.391
Unit 2	315.6965	29977.58
Unit 3	114.6164	1054.55

Tampilan Data

Bar	Nama	Pmax	Pmin	a1	a2	a3	T
1	B 1	600	100	561	7.92	0.001562	1
2	B 2	600	100	310	7.85	0.00194	1
3	B 3	200	50	79	7.97	0.00482	1

Buttons: New, Close

Gambar 4.1. Tampilan Data Pembangkit (data jurnal)

Hasil Program

No	P GEN MW	P PEN MW	Cost IR (\$)	Cost PIR (\$)	Sisa (\$)
1	250	3.917	2.639	1.279	
2	335	400	3.154	3.760	606
3	122	200	1.123	1.865	742

Biaya GR: 6.194 Biaya PIR: 8.264 Sisa: 69

Buttons: Print, Exit

Gambar 4.2. Tampilan Hasil Perhitungan (data jurnal)

4.4. Data Pembangkit Thermal

Pembangkit thermal yang berada pada pengawasan PT. Pembangkitan 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 tahun 2004 dimana dipakai nilai tukar Rp. 9000 per satuan dollar Amerika.

Tabel 4.1.
Data Unit Termal pada PT. PLN PJB

No	Nama Pembangkit	Bahan Bakar	Kapasitas (MW)		Koefisien Biaya Bahan Bakar		
			Min	Max	A	B	C
1	PLTU Paiton 1	Coal	225	370	3241978	111712.2	10.2971
2	PLTU Paiton 2	Coal	225	370	3244978	111712.2	10.2971
3	PLTGU Gresik GT 1.1	Gas	53	102	5467532.4	217963.5	34.155
4	PLTGU Gresik GT 1.2	Gas	53	102	5467532.4	217963.5	34.155
5	PLTGU Gresik GT 1.3	Gas	53	102	5467532.4	217963.5	34.155
6	PLTGU Gresik ST 1.0	Gas	250	480	17177460.3	145165.581	4.554
7	PLTGU Gresik GT 2.1	Gas	53	102	5467532.4	217963.5	34.155
8	PLTGU Gresik GT 2.2	Gas	53	102	5467532.4	217963.5	34.155
9	PLTGU Gresik GT 2.3	Gas	53	102	5467532.4	217963.5	34.155
10	PLTGU Gresik ST 1.0	Gas	164	314	17177460.3	145165.581	4.554
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 1.0	Gas	164	314	17177460.3	145165.581	4.554
15	PLTU Gresik 1	Gas	43	85	1327126.7	217378.4	132.066
16	PLTU Gresik 2	Gas	43	85	1327126.7	217378.4	132.066
17	PLTU Gresik 3	Gas	90	175	50173695	169242.6	193.545
18	PLTU Gresik 4	Gas	90	175	50173695	169242.6	193.545
19	PLTG Gresik 1	Gas	5	16	352707.3	350680.8	903.969
20	PLTG Gresik 2	Gas	5	16	352707.3	350680.8	903.969
21	PLTG Gresik 3	Gas	5	16	352707.3	350680.8	903.969
22	PLTG Gilitimur 1	HSD	5	16	687181.85	683241	1762.3893
23	PLTG Gilitimur 2	HSD	5	16	687181.85	683241	1762.3893
24	PLTGU M.Karang GT 1.1	Gas	50	95	5730795	202053	108.045
25	PLTGU M.Karang GT 1.2	Gas	50	95	5730795	202053	108.045
26	PLTGU M.Karang GT 1.3	Gas	50	95	5730795	202053	108.045
27	PLTGU M.Karang ST 1.0	Gas	300	465	31017735	87825.15	57.33
28	PLTGU M.Tawar GT 1.1	HSD	72	138	14706521	433337.8	49.4605
29	PLTGU M.Tawar GT 1.2	HSD	72	138	14706521	433337.8	49.4605
30	PLTGU M.Tawar GT 1.3	HSD	72	138	14706521	433337.8	49.4605
31	PLTGU M.Tawar GT 2.1	HSD	72	138	14706521	433337.8	49.4605
32	PLTGU M.Tawar GT 2.2	HSD	72	138	14706521	433337.8	49.4605
33	PLTGU M.Tawar ST 1.0	HSD	162	202	672630	144191.7	49.4605
34	PLTU M.Karang 1	MFO	44	85	2417820.7	473895.4	519.1757
35	PLTU M.Karang 2	MFO	44	85	2417820.7	473895.4	120.77935
36	PLTU M.Karang 3	MFO	44	85	2417820.7	473895.4	120.77935
37	PLTU M.Karang 4	Gas	90	165	2949187.5	205217.1	83.73
38	PLTU M.Karang 5	Gas	90	165	2949187.5	205217.1	83.73

Sumber : Data Penawaran PT PJB. Jl. Ketintang Baru No.11, Surabaya 60231

Catatan : Harga Batubara 253 Rp/Kg
 Harga MFO 1595 Rp/liter
 Harga HSD 1595 Rp/liter
 Harga Gas UP. Gresik 2,53 US\$/MBTU
 Nilai Tukar 9000 Rp/\$

4.5 Aplikasi Metode *Quadratic Hopfield* di P.T. Pembangkitan Jawa-Bali

Perhitungan dan analisa ini dilakukan pada kebutuhan daya yang ditanggung P.T Pembangkitan Jawa-Bali (PJB) tanggal 10, 13 dan 14 Maret 2004. Analisa data dilakukan hanya untuk unit yang beroperasi, dimana yang siap beroperasi menjadi 18 unit pembangkit dikarenakan unit-unit pembangkit tersebut sudah mampu melayani beban permintaan. Program komputer ini hanya untuk menghitung unit pembangkit yang siap beroperasi menjadi 18 unit yang dapat dilihat pada table 4.5.

Tabel 4.5.
Unit Thermal yang Siap Beroperasi

No.	Nama Pembangkit
1	PLTU Paiton 1
2	PLTU Paiton 2
3	PLTGU Gresik GT 1.3
4	PLTGU Gresik ST 1.0
5	PLTGU Gresik GT 2.3
6	PLTGU Gresik GT 3.3
7	PLTGU Gresik ST 1.0
8	PLTU Gresik 1
9	PLTU Gresik 2
10	PLTU Gresik 3
11	PLTU Gresik 4
12	PLTGU M. Karang ST. 1.0
13	PLTGU M. Tawar ST. 1.0
14	PLTU M. Karang 1
15	PLTU M. Karang 2
16	PLTU M. Karang 3
17	PLTU M. Karang 4
18	PLTU M. Karang 5

4.6. Beban Sistem

Dalam wilayah Jawa-Bali, Pembangkit-pembangkit yang ada dikoordinasi oleh P.T. Pembangkitan Jawa-Bali (PJB). Proses *Economic Dispatch* dengan metode *Quadratic Hopfield* bertujuan untuk membuat rencana operasi yang optimum dalam sistem tenaga listrik yang dapat memenuhi kebutuhan beban dengan biaya operasi yang seekonomis mungkin.

Untuk mengetahui seberapa besar efisiensi dari metode ini, maka dilakukan evaluasi dengan mengambil data unit pembangkit termal dan beban yang ditanggung oleh P.T. PJB sebagai bahan perbandingan. Sedangkan kombinasi jadwal dan daya output pembangkit tenaga listrik dalam system P.T. PJB tanggal 10, 13 dan 14 maret 2004 terdapat pada lampiran 2. Untuk beban sistem terdapat pada tabel 4.3 (beban sistem yang ditanggung oleh pembangkit termal saja). Beban sistem yang terdapat pada tabel 4.3 dibawah ini didapatkan dari jumlah total keseluruhan data beban yang ada pada PT. PLN PJB, agar dalam perhitungan nanti baik secara manual maupun dengan memakai perangkat lunak (software) dapat dibandingkan antara sebelum dan sesudah optimasi.

Tabel 4.6
Data Beban Unit Termal pada PT. PLN PJB

JAM	BEBAN SISTEM (MW)		
	10 Maret 2004	13 Maret 2004	14 Maret 2004
1:00	3108	2896	2816
2:00	3024	2864	2678
3:00	2993	2845	2675
4:00	2974	2866	2694
5:00	2976	2921	2804
6:00	2916	2806	2611
7:00	2868	2710	2588
8:00	3202	2856	2746
9:00	3265	3002	2802
10:00	3281	3020	2816
11:00	3297	3026	2853
12:00	3220	3030	2789
13:00	3225	3016	2749
14:00	3226	2901	2657
15:00	3297	2717	2613
16:00	3372	2796	2709
17:00	3499	2869	2714
18:00	3600	3374	3255
19:00	3657	3382	3268
20:00	3642	3373	3269
21:00	3403	3205	2982
22:00	3388	3015	2876
23:00	3335	2929	2864
24:00	3316	2869	2822

4.7. Hasil Perhitungan dan Analisa Data

4.7.1. Hasil Perhitungan PT. PLN PJB sebelum optimasi dan sesudah optimasi

Dari data pembebanan harian, dapat dihitung biaya operasional tiap jamnya dengan memasukan ke persamaan fungsi biaya bahan bakar.

Pada hari Sabtu 13 Maret 2004 jam 01.00 sebelum di optimasi unit pembangkit 1 dibebankan 370 MW maka perhitungannya adalah :

$$\begin{aligned} Fi(P_i) &= a_i + b_i P_i + c_i P_i^2 \\ &= 3244978 + 111712.15 \cdot (370) + 10.2971 \cdot (370)^2 \\ &= \text{Rp } 45.988.147 \text{ (sebelum dioptimasi)} \end{aligned}$$

Setelah daya generator dioptimasi menjadi 351 MW maka :

$$\begin{aligned} Fi(P_i) &= a_i + b_i P_i + c_i P_i^2 \\ &= 3244978 + 111712.15 \cdot (351) + 10.2971 \cdot (351)^2 \\ &= \text{Rp } 43.742.956 \text{ (sesudah dioptimasi)} \end{aligned}$$

Dengan cara yang sama selama 3 hari dalam waktu 24 jam, maka hasil perhitungannya didapatkan biaya operasional perjamnya yang dapat dilihat pada tabel 4.4 ; 4.5 dan 4.6 di bawah ini.

642

Tabel 4.7
Hasil Perhitungan Biaya operasional perjam PT. PLN PJB
Rabu 10 Maret 2004

Jam	Beban Sistem (MW)	PT. 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	2868	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

Tabel 4.8.
**Hasil Perhitungan Biaya operasional perjam PT. PLN PJB
 Sabtu 13 Maret 2004**

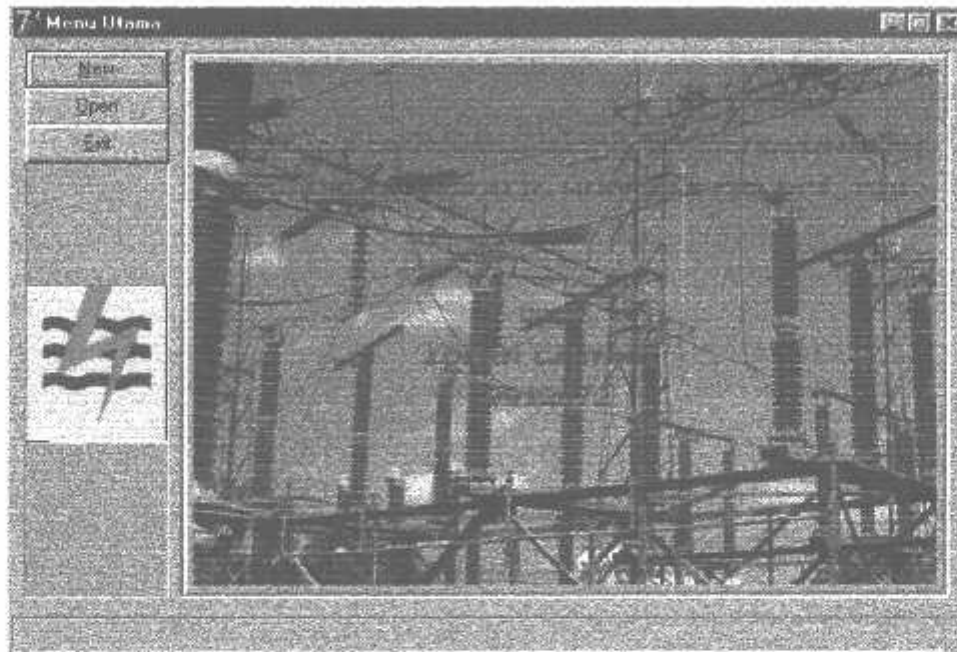
Jam	Beban Sistem (MW)	PT. PLN PJB (Rp)
01:00	2896	642.976.722
02:00	2864	627.840.757
03:00	2845	626.899.359
04:00	2866	629.878.888
05:00	2921	638.467.808
06:00	2806	622.968.844
07:00	2710	609.418.876
08:00	2856	629.093.132
09:00	3002	651.358.752
10:00	3020	655.538.620
11:00	3026	657.042.259
12:00	3030	658.662.629
13:00	3016	657.445.941
14:00	2901	637.045.941
15:00	2717	609.113.162
16:00	2796	622.230.104
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

Tabel 4.9.
Hasil Perhitungan Biaya operasional perjam PT. PLN PJB
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	2749	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

4.6.2. Tampilan Program Komputer dan Hasil Perhitungan dengan Metode *Quadratic Hopfield*.

Tampilan 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 di bawah ini :



Gambar 4.3.
Tampilan Program Utama

Kemudian setelah itu tekan tombol buka data untuk membuka file inputan data yang tersimpan, data ini ditampilkan pada hari Rabu 10 Maret 2004. Disini terlihat bahwa pada generalnya diketahui jumlah pembangkit 38 unit dan jumlah jamnya 24 jam

Tampilan Data

General | Data Generator | Data Pembebanan | Data PLN

Jumlah Pembangkit: 30

Jumlah Jam: 24

Next Close

Gambar 4.4
Tampilan Input Data pembangkit

Tampilan Data

General | Data Generator | Data Pembebanan | Data PLN

Gen	Nama	Pmax	Pmin	a0	a1	a2	f
1	PLTU Paiton 1	370	225	3244978	111712.2	10.2971	72
2	PLTU Paiton 2	370	225	3244978	111712.2	10.2971	72
3	PLTGU Gresik GT 1.1	102	53	5467532.4	217963.5	34.155	36
4	PLTGU Gresik GT 1.2	102	53	5467532.4	217963.5	34.155	36
5	PLTGU Gresik GT 1.3	102	53	5467532.4	217963.5	34.155	36
6	PLTGU Gresik ST 1.0	480	250	17177460.3	145165.581	4.954	36
7	PLTGU Gresik GT 2.1	102	53	5467532.4	217963.5	34.155	36
8	PLTGU Gresik GT 2.2	102	53	5467532.4	217963.5	34.155	36
9	PLTGU Gresik GT 2.3	102	53	5467532.4	217963.5	34.155	36
10	PLTGU Gresik ST 1.0	480	250	17177460.3	145165.581	4.954	36
11	PLTGU Gresik GT 3.1	102	53	5467532.4	217963.5	34.155	36
12	PLTGU Gresik GT 3.2	102	53	5467532.4	217963.5	34.155	36
13	PLTGU Gresik GT 3.3	102	53	5467532.4	217963.5	34.155	36

Next Close

Gambar 4.5
Tampilan data generator

Setelah menampilkan data generator, berikutnya tampilan Data Pembebanan dan nilai daya maksimum pada cadangan berputarnya sebagai berikut:

Setelah inputan data dimasukan maka parameter yang diperlukan adalah sebagai berikut:

Parameter QH	
Jumlah Herd QH	100
Memorandum	50
Jumlah Pembangkit	38
Steps Adjustment	0.3
Kt	100000000

Use Default

Rata-rata ke Bolepe
1

Parameter Objective Function
Pinale Generator: 1000000

Hitung Close

Gambar 4.8.
tampilan parameter yang digunakan

Setelah menetapkan parameter yang diperlukan maka komputasi dapat dilakukan. Untuk proses komputasi digunakan software Borland Delphi 7.0 kemudian dieksekusi dengan menggunakan komputer berspesifikasi prosesor Intel Pentium 120 GHz Ram 256 Mb. Perhitungan dimulai pada jam 01:00 – 24:00 selama 3 hari.

Hasil Program

Parameter (24) Hasil QH					
No	P QH (MW)	P PLN (MW)	Cost QH (Rp)	Cost PLN (Rp)	Selanj (Rp)
1	370	370	45,980,165	45,980,165	0
2	370	370	45,980,165	45,980,165	0
3	56	87	17,676,326	24,600,876	7,009,950
4	472	285	86,714,566	58,919,550	27,795,017
5	53	87	17,115,539	24,688,876	7,573,337
6	53	89	17,115,539	25,136,826	8,021,286
7	480	285	87,906,181	58,919,550	28,986,631
8	43	75	10,816,088	18,373,378	7,454,790
9	44	75	11,234,885	18,373,378	7,138,482
10	90	90	21,816,318	21,816,318	0
11	99	90	23,713,538	21,816,318	1,896,620
12	461	435	83,685,940	80,069,340	3,595,395
13	198	200	29,510,901	31,488,390	1,978,489
14	44	80	24,274,342	43,652,177	19,377,835
Rata Rata		516,624,423	Rata Rata		580,300,100
			Rata Rata		61,681,685

Hasil Optimalisasi

Gambar 4.8
Hasil Optimasi Menggunakan Metode *Quadratic Hopfield*

Dengan begitu proses proses komputasi dengan menggunakan Software Borland Delphi 7.0 dan dieksekusi dengan menggunakan komputer, maka dapat dilihat pada Gambar 4.6 tampilan program sesudah dioptimasi. Proses perhitungan dimulai pada jam 01:00 – 24:00 selama 3 hari. Untuk mengetahui lebih lengkapnya perbandingan antara daya generator sebelum dioptimasi dan sesudah dioptimasi dilihat pada tabel 4.8 dibawah ini :

4.2.2.1. Hasil Perhitungan Menggunakan Metode *Quadratic Hopfield*

Berikut ini adalah hasil perhitungan biaya dengan menggunakan metode *Quadratic Hopfield*

Tabel 4.10.
Hasil Perhitungan
Rabu 10 Maret 2004

Jam	Beban Sistem (MW)	<i>Quadratic Hopfield</i> (Rp)
01:00	3108	618.624.423
02:00	3024	604.850.067
03:00	2993	601.406.428
04:00	2974	598.812.721
05:00	2976	598.738.150
06:00	2916	598.022.182
07:00	2868	583.229.530
08:00	3202	638.099.200
09:00	3265	651.777.571
10:00	3281	655.449.516
11:00	3297	658.861.451
12:00	3220	642.276.200
13:00	3225	642.733.830
14:00	3226	642.986.332
15:00	3297	658.662.841
16:00	3372	675.541.792
17:00	3499	734.630.893
18:00	3600	757.641.781
19:00	3657	770.555.631
20:00	3642	767.042.874
21:00	3403	682.050.072
22:00	3388	679.192.791
23:00	3335	667.246.772
24:00	3316	663.450.756

Tabel 4.11.
Hasil Perhitungan
Sabtu 13 Maret 2004

Jam	Beban Sistem (MW)	<i>Quadratic Hopfield</i> (Rp)
01:00	2896	587.661.288
02:00	2864	581.630.841
03:00	2845	578.340.570
04:00	2866	583.022.144
05:00	2921	591.150.619
06:00	2806	576.533.644
07:00	2710	560.392.555
08:00	2856	580.418.483
09:00	3002	602.375.967
10:00	3020	605.232.608
11:00	3026	606.413.877
12:00	3030	606.388.179
13:00	3016	604.839.148
14:00	2901	587.275.250
15:00	2717	560.134.202
16:00	2796	572.487.149
17:00	2869	582.270.020
18:00	3374	675.849.634
19:00	3382	677.515.636
20:00	3373	675.757.401
21:00	3205	638.745.608
22:00	3015	605.190.961
23:00	2929	591.516.635
24:00	2869	582.770.856

Tabel 4.12.
Hasil Perhitungan
Minggu 14 Maret 2004

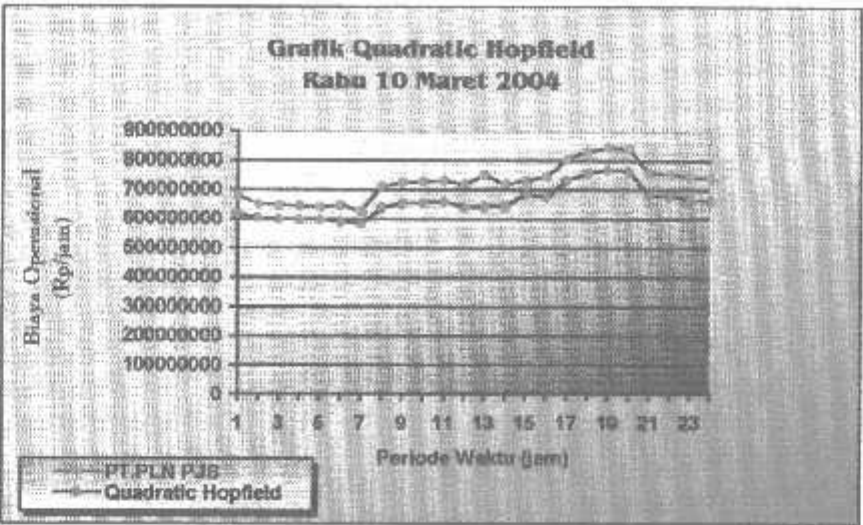
Jam	Beban Sistem (MW)	<i>Quadratic Hopfield</i> (Rp)
01:00	2816	575.414.453
02:00	2678	553.776.192
03:00	2675	553.709.570
04:00	2694	556.685.135
05:00	2804	572.907.136
06:00	2611	547.553.686
07:00	2588	542.091.070
08:00	2746	564.326.109
09:00	2802	572.460.081
10:00	2816	574.230.556
11:00	2853	580.284.220
12:00	2789	570.661.438
13:00	2749	564.828.044
14:00	2657	553.204.842
15:00	2613	545.129.343
16:00	2709	559.067.701
17:00	2714	559.502.569
18:00	3255	649.539.447
19:00	3268	652.146.768
20:00	3269	652.710.059
21:00	2982	600.007.857
22:00	2876	583.936.832
23:00	2864	584.402.359
24:00	2822	575.976.209

4.6.3. Perbandingan Hasil Perhitungan PT. PLN PJB Dengan Metode Quadratic Hopfield

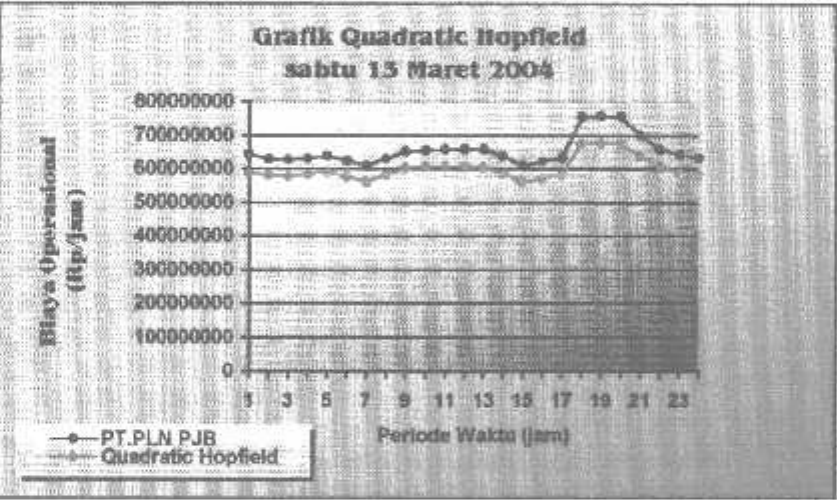
Berikut ini adalah yang berisi perbandingan biaya pada PT. PLN PJB dan Metode Quadratic Hopfield.

Tabel 4.13.
Perbandingan Biaya Operasional Perjam
PT. PLN PJB dengan Metode Quadratic Hopfield

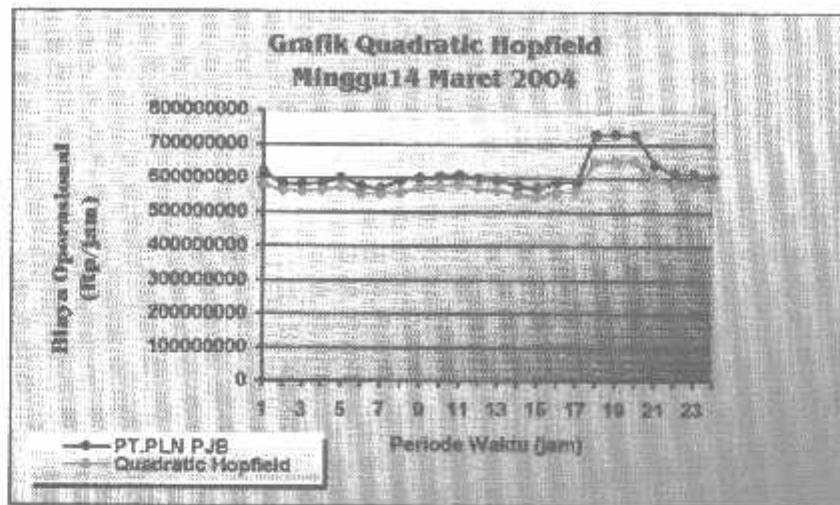
JAM	PERBANDINGAN BIAYA OPERASIONAL PERJAM					
	10 Maret 2004		13 Maret 2004		14 Maret 2004	
	Quadratic Hopfield	PT. PLN PJB	Quadratic Hopfield	PT. PLN PJB	Quadratic Hopfield	PT. PLN PJB
1:00	618.624.423	680.306.108	587.661.288	642.976.722	575.414.453	618.251.323
2:00	604.850.067	651.394.473	581.630.841	627.840.757	553.776.192	580.714.514
3:00	601.406.428	646.667.419	578.340.570	626.899.359	553.709.570	580.742.086
4:00	598.812.721	644.171.861	583.022.144	629.878.888	556.685.135	583.369.050
5:00	598.738.150	640.361.923	591.150.619	638.467.808	572.907.136	601.290.761
6:00	598.022.182	646.587.755	576.533.644	622.968.844	547.553.686	574.415.775
7:00	583.229.530	626.319.642	560.392.555	609.418.876	542.091.070	570.967.006
8:00	638.099.200	713.721.328	580.418.483	629.093.132	564.326.109	591.959.107
9:00	651.777.571	725.817.463	602.375.967	651.358.752	572.460.081	601.814.600
10:00	655.449.516	728.989.485	605.232.608	655.538.620	574.230.556	604.252.146
11:00	658.861.451	732.577.863	606.413.877	657.042.259	580.284.220	611.074.946
12:00	642.276.200	720.949.264	606.388.179	658.662.629	570.661.438	600.616.653
13:00	642.733.830	723.177.188	604.839.148	657.445.941	564.828.044	595.203.725
14:00	642.986.332	720.370.936	587.275.250	637.045.941	553.204.842	577.919.351
15:00	658.662.841	733.154.110	560.134.202	609.113.162	545.129.343	570.955.908
16:00	675.541.792	747.933.301	572.487.149	622.239.104	559.067.701	587.564.000
17:00	734.630.893	805.181.358	582.270.020	633.104.555	559.502.569	588.687.073
18:00	757.641.781	832.153.449	675.849.634	755.125.576	649.539.447	728.245.361
19:00	770.555.631	844.634.828	677.515.636	756.832.935	652.146.768	730.147.988
20:00	767.042.874	842.417.383	675.757.401	755.769.182	652.710.059	729.197.563
21:00	682.050.072	757.459.805	638.745.608	700.429.220	600.007.857	639.723.642
22:00	679.192.791	754.703.180	605.190.961	657.757.376	583.936.832	617.848.361
23:00	667.246.772	744.635.040	591.516.635	642.690.720	584.402.359	615.562.714
0:00	663.450.756	740.803.704	582.770.856	632.703.376	575.976.209	606.757.754



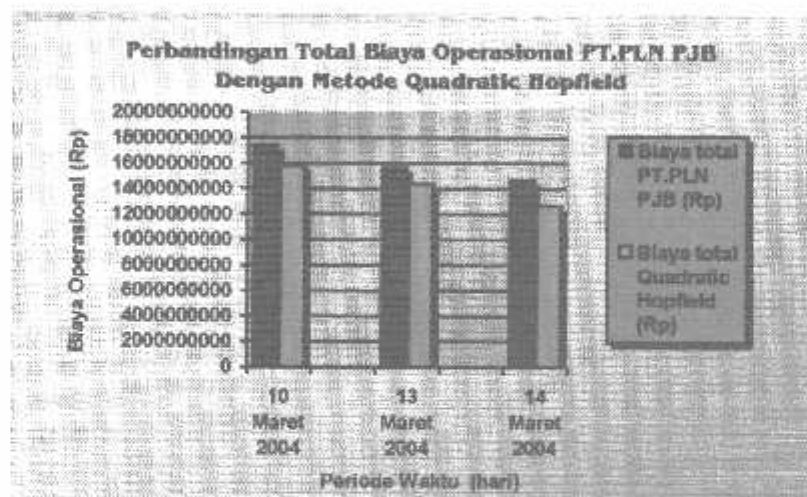
Grafik 4.1.
Perbandingan Biaya Operasional perjam PT. PLN PJB dan Metode
Quadratic Hopfield Rabu 10 Maret 2004



Grafik 4.2.
Perbandingan Biaya Operasional perjam PT. PLN PJB dan Metode
Quadratic Hopfield
Sabtu 13 Maret 2004



Grafik 4.3.
Perbandingan Biaya Operasional perjam PT. PLN PJB dan Metode *Quadratic Hopfield*
Minggu 14 Maret 2004



Grafik 4.4.
Perbandingan Total Biaya Operasional PT. PJB dengan Metode *Quadratic Hopfield*

Tabel 4.13.
Perbandingan Total Biaya Operasional
PT. PLN PJB Dengan Metode *Quadratic Hopfield*

Periode Waktu (24 jam)	PT. PLN PJB (Rp)	<i>Quadratic Hopfield</i> (Rp)	Penghematan (Rp)
10 Maret 2004	17.404.488.866	15.791.883.804	1.612.605.060
13 Maret 2004	15.710.421.169	14.413.913.275	1.296.507.890
14 Maret 2004	14.697.281.407	12.743.978.385	1.953.303.020

Pada Tabel 4.13. dapat diketahui bahwa perbandingan total biaya operasional pembangkitan antara PT. PJB dengan Metode *Quadratic Hopfield* cukup besar. Pada hari Rabu tanggal 10 Maret 2004 selisih sebesar Rp. 1.612.605.060, hari Sabtu tanggal 13 Maret 2004 selisih sebesar Rp. 1.296.507.890 dan hari Minggu tanggal 14 Maret 2004 selisih sebesar Rp. 1.953.303.020.

Berdasarkan perbandingan tersebut di atas maka dapat disimpulkan bahwa optimasi *economic dispatch* menggunakan Metode *Quadratic Hopfield* mendapatkan hasil yang lebih ekonomis dan akurat.

BAB V

PENUTUP

5.1 Kesimpulan

Setelah melakukan analisa dan perhitungan pada pembangkit tenaga terhadap beban yang ditanggung PT. PLN Pembangkit Jawa Bali pada tanggal 10, 13, 14 maret 2004 dengan menggunakan metode Quadratic Hopfield pada skripsi ini, maka dapat ditarik kesimpulan sebagai berikut :

- 1 Bahwa pendekatan metode Quadratic Hopfield pada tanggal 10 maret 2004 memiliki selisih nilai biaya total sebesar 8.50 % atau Rp. 1.612.605.060, 13 Maret 2004 sebesar 7.25 % atau Rp.1.296.507.890 dan 14 maret 2004 sebesar 4.86 % atau Rp. 1.953.303.020.
- 2 Setelah melakukan analisa terhadap biaya operasional terhadap unit pembangkit, pada hari Rabu, Sabtu dan Minggu dapat dilihat bahwa daya yang dibangkitkan oleh unit pembangkit yang beroperasi yaitu unit 1 – 18 selama 24 jam terlihat adanya perubahan terhadap daya yang dibangkitkan oleh unit pembangkit yang beroperasi. Daya yang dibangkitkan oleh pembangkit sangat berpengaruh terhadap biaya operasionalnya terutama pada biaya bahan bakarnya, dengan membandingkan antara daya sebelum dioptimasi dengan daya sesudah dioptimasi maka dapat diperoleh suatu biaya yang sehemat mungkin.

Dari hasil perhitungan analisa data yaitu :

- tanggal 10 Maret 2004 unit 1-18 selama 24 jam sebelum dioptimasi sebesar Rp. 17.404.488.866, setelah dioptimasi mendapatkan penghematan sebesar Rp. 15.791.883.804 ;
 - 13 Maret 2004 dari unit 1-18 selama 24 jam sebelum dioptimasi Rp.15.710.421.169, setelah dioptimasi akan mengalami penurunan biaya Rp. 14.413.913.275;
 - 14 Maret 2004 unit 1-18 selama 24 jam sebelum dioptimasi Rp.14.697.281.407 sesudah dioptimasi sebesar Rp. 12.743.978.385,
- 3 Untuk mengoptimasikan beban system yang ekonomis yaitu dengan cara mengurangi biaya pembangkitan pada system yang mempunyai biaya operasi tinggi untuk dibagikan nilai daya pembangkitan pada unit yang mempunyai nilai operasinya rendah dengan memperhatikan kendala operasi dan tetap memenuhi beban system.

5.2. Saran

Berdasarkan kesimpulan di atas, saran yang mungkin bisa diajukan antara lain :

1. Melakukan Studi lebih lanjut untuk melakukan optimasi pada jumlah unit yang lebih besar atau jangkah waktu yang dipelajari lebih lama untuk mengetahui sejauh mana kemampuan analisa metode ini apakah masih relevan untuk digunakan dalam system yang lebih besar.
2. Memungkinkan untuk dapat diaplikasikan pada PT. Pembangkitan Tenaga Listrik Jawa-Bali (PJB) karena menghasilkan biaya operasional yang lebih ekonomis.
3. Perlu dilakukan koordinasi antara mahasiswa dengan pihak PT.PJB agar didapat data yang memuaskan.



DAFTAR PUSTAKA

- [1] Djiteng Marsudi. Ir. “ **Operasi Sistem Tenaga Listrik** “, ISTN 1990.
- [2] A. J. Wood dan B. F. Wollenberg, “ **Power Generation Operation and Control** “ Second Edition, John Wiley & Sons. 1996.
- [3] William D. Stevenson, Jr. “ **Analisa Sistem Tenaga Listrik** “, Edisi Keempat, Erlangga, Jakarta 1993.
- [4] Kusumadewi, Sri, “**Artificial Intelegence** “ , Graha Ilmu, Jakarta 2003
- [5] Kwang Y. Lee, Arthit Sode-Yome and June Ho Park, “ Economic Dispatch for Quadratic Hopfield Neural Network “, IEEE Transactions on Power System, Vol. 13, no.2, May 1998.



LAMPIRAN

1. *Berita Acara Ujian Skripsi*
2. *Lembar Bimbingan Skripsi*
3. *Formulir Bimbingan Skripsi*
4. *Surat Permohonan Penelitian*
5. *Data Pembangkit Thermal*
PT. PLN PJB
6. *Data Daya Sebelum &*
Sesudah Optimasi
7. *Listing Program*



**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

1. Nama : JOHAN CAHYADI
2. N.I.M. : 99.12.072
3. Jurusan : Teknik Elektro S-1
4. Kosentrasi : Teknik Energi Listrik
5. Judul Skripsi :

**ANALISA PEMBEBANAN EKONOMIS DENGAN
MENGUNAKAN METODE QUADRATIC HOPFIELD
PADA PT. PLN. PJB.**

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

6. Hari : Senin
7. Tanggal : 25 September 2006
8. Dengan nilai : 73.4 (B+) 74



(Ir. Mochtar Asroni, MSME.)
Ketua

Panitia Ujian Skripsi

(Ir. F. Yudi Limpraptono, MT.)
Sekretaris

Anggota Penguji

(Ir. Widodo Pudji Muljanto, MT.)
Penguji Pertama

(Irrine Budi. S, ST, MT.)
Penguji Kedua



LEMBAR BIMBINGAN SKRIPSI

Nama : JOHAN CAHYADI
NIM : 99.12.072
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Energi Listrik
Judul Skripsi : ANALISA PEMBEBANAN EKONOMIS DENGAN
MENGUNAKAN METODE QUADRATIC
HOPFIELD PADA PT. PLN. PJB.
Tanggal Mengajukan Skripsi : 16 Juni 2006
Tanggal Menyelesaikan Skripsi : 25 September 2006
Dosen Pembimbing : Ir. H. Taufik Hidayat, MT.
Telah dievaluasi dengan nilai : 80 (Delapan Puluh) *H*

Mengetahui,
Ketua Jurusan Teknik Elektro S-1

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

Diperiksa dan disetujui,
Dosen Pembimbing

(Ir. H. Taufik Hidayat, MT)
NIP. P. 10187000151



FORMULIR BIMBINGAN SKRIPSI

Nama : JOHAN CAHYADI
Nim : 99.12.072
Masa Bimbingan : 16 Juni 2006 s/d 16 Des 2006
Judul Skripsi : ANALISA PEMBEBANAN EKONOMIS DENGAN
MENGUNAKAN METODE QUADRATIC HOPFIELD
PADA P.T. PLN. PJB.

No	Tanggal	Uraian	Parap Pembimbing
1.	2-08-2006	Revisi Bab I. Tujuan	Ah
2.	2-08-2006	ACC Bab I	Ah
3.	4-07-2006	Revisi Bab II. Teori Dasar Sistem Tenaga Listrik	Ah
4.	4-08-2006	ACC Bab II	Ah
5.	29-08-2006	ACC Bab III	Ah
6.	6-09-2006	Revisi Bab IV. Tabel sebelum-sesudah optimasi	Ah
7.	7-09-2006	ACC Bab IV	Ah
8.	8-09-2006	Revisi Bab V Kesimpulan	Ah
9.	8-09-2006	ACC Bab V + ACC Makalah Seminar Hasil	Ah
10.	22-08-2006	ACC Ujian Skripsi	Ah

Malang, 27 September 2006
Dosen Pembimbing


Ir. H. Taufik Hidayat, MT
Nip : 1018800190

Form S-4b

Nomor : E2042565
Sifat : Biasa
Lampiran : -

Surabaya, 25 Juni 2006

KEPADA
Yth. DEKAN FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
Jl. Bendungan Sigura-gura No. 2
MALANG – 65145

Perihal : Ijin Penelitian

Menunjuk surat Saudara, No.ITN.1775/III.TA/2/2006 tgl. 19 Juni 2006, perihal : Survey / Permintaan Data untuk Tugas Akhir, maka perlu kami sampaikan hal-hal sebagai berikut :

1. Pada prinsipnya kami memberikan ijin kepada mahasiswa saudara yang bernama :

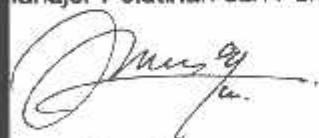
JOHAN CAHYADI – Nim 99 12 072

Jurusan Teknik Elektro Fakultas Teknologi Industri Institut Teknologi Nasional Malang, untuk melakukan Survey / Permintaan Data di PT Pembangkitan Jawa Bali - Kantor Pusat.

2. Kegiatan dimaksud akan dilaksanakan di PT Pembangkitan Jawa Bali - Kantor Pusat - Sub Direktorat Manajemen Energi (EMG), Jl. Ketintang Baru No. 11 Surabaya 60231, selama kurun waktu 1 (satu) minggu, terhitung mulai tanggal 17 Juli 2006 sampai dengan 22 Juli 2006.
3. Kepada yang bersangkutan diberikan ketentuan bahwa :
 - o Tidak diperkenankan memanfaatkan data dan informasi untuk kepentingan di luar PT PJB sebelum mendapat persetujuan dari Penjabat yang berwenang.
 - o Seusai kegiatan harus membuat laporan kegiatan yang telah disahkan oleh penjabat PT PJB yang berwenang.
 - o Selama melaksanakan kegiatan wajib mengikuti ketentuan dan aturan yang berlaku bagi Karyawan / Tamu di lingkungan PT PJB.

Atas perhatian Saudara, kami sampaikan terima kasih.

Format Kami,
Manajer Pelatihan dan Pengembangan



Gede Adnya Karsana

Pembusan
Manajer Pusat Manajemen Energi (EMC)
Slamet Mudani
Johan Cahyadi

PT PEMBANGKITAN JAWA - BALI

Jl. Ketintang Baru 11 Surabaya 60231 - Indonesia, Tlp : 62-31-8283180 (hunting) Fax : 62-31-8291280 E-mail : pjb@pjb2.com



DATA PENAWARAN
PT PLN PEMBANGKITAN JAWA BALI
AGUSTUS 2004

No	NAMA PEMBANGKIT	KAPASITAS			WAKTU (JAM)			BIAYA STARTUP (RUPIAH)			KOEFSIEN BIAYA BAHAN BAKAR		
		Daya (MW)	MIN (MW)	MAX (MW)	UP TIME (MIN)	DOWN TIME (MIN)	COLD STARTUP (HOUR)	COLD STARTUP (HOUR)	COLD STARTUP (HOUR)	COLD STARTUP (HOUR)	150	150	150
P	UP. PAITON PLTU #1/2 (COAL)	2 x 400	225	370	72	48	17	4	682.98	149.58	3244978	111712.15	102971
2	UP. GRESIK												
	GT 1-9 OC (GAS)	9 x 112	53	102	36	10	1	0	7.82	0	5437532.4	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	11785770.8	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	4.584
	PLTU #1/2 (GAS)	100	43	85	48	10	9	1	143.74	40.59	1327126.68	217978.359	132.066
	PLTU #3/4 (GAS)	200	90	175	48	10	9	2	229.5	92.52	5017369.5	169242.579	193.545
3	PLTG GRESIK 1-3 (GAS)	3 x 20	5	16	3	1	1	0	6.13	0	352707.3	350680.77	903.969
	PLTG GRESIK 1-2 (HSD)	2 x 20	5	16	3	1	1	0	6.33	0	687181.85	683240.955	1762.3893
	UP. MUARA KARANG												
	GT 1/2/3 - OC	3 x 107	50	95	36	10	1	0	7.35	0	5730795	202052.97	108.045
	CC-1.1.1 (GAS)	153	110	150	36	10	3	1	54.22	29.67	11560815	53685.135	460.845
	CC-2.2.1 (GAS)	317	200	300	36	10	3	2	61.57	36.92	18010064	127203.655	35.28
	CC-3.3.1 (GAS)	508	300	455	36	10	3	2	68.92	44.27	31017735	87825.15	57.33
4	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	288509.995	7.6504
	PLTU #1/2/3 (MFO)	3 x 100	44	85	48	10	6	1	122.58	31.08	2417820.7	473895.41	130.77935
	PLTU #4/5 (Gas)	2 x 200	90	165	48	10	11	2	215.34	89.29	2949187.5	205217.145	03.79

SUB SYSTEM REGION 1

RENCANA : HARI/TANGGAL: SABTU, 13 MARET 2004

PT. PLN PENGANGKUTAN TENAGA LISTRIK JAWA-BALI

[illegible]

PT. PLN PENDINGKATAN TENAGA LISTRIK JAWA (PLN)		Page-2																							
Jum		13.00	13.30	14.00	14.30	15.00	15.30	16.00	16.30	17.00	17.30	18.00	18.30	19.00	19.30	20.00	20.30	21.00	21.30	22.00	22.30	23.00	23.30	24.00	24.30
PLTA PLTA PLTA	Area 4 SUTAMA BUNTAS	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21
		90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90
		93	93	93	93	93	93	93	93	93	93	93	93	93	93	93	93	93	93	93	93	93	93	93	93
		360	360	360	360	360	360	360	360	360	360	360	360	360	360	360	360	360	360	360	360	360	360	360	360
PLTA PLTA	PITON #1 PITON #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	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	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	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PLTA PLTA	GRSM10C GRSM10C1 GRSM10C2 GRSM10C3	112	90	102	102	99	99	101	101	115	115	115	115	115	115	115	115	115	115	115	115	115	115	115	115
		182	100	100	153	153	173	173	173	185	185	185	185	185	185	185	185	185	185	185	185	185	185	185	185
		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	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PLTA PLTA	GRSM10C GRSM10C1 GRSM10C2 GRSM10C3	103	93	93	93	93	93	93	93	93	93	93	93	93	93	93	93	93	93	93	93	93	93	93	93
		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	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	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PLTA PLTA	GRSM10C GRSM10C1 GRSM10C2 GRSM10C3	96	87	87	87	87	87	87	87	87	87	87	87	87	87	87	87	87	87	87	87	87	87	87	87
		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	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		154	156	156	156	156	156	156	156	156	156	156	156	156	156	156	156	156	156	156	156	156	156	156	156
PLTA PLTA	GRSM10C GRSM10C1 GRSM10C2 GRSM10C3	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121
		121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121
		73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73
		73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73
PLTA PLTA	GLTMR GRSM GRSM GRSM	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	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	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	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PLTA PLTA	GRSM10C GRSM10C1 GRSM10C2 GRSM10C3	3517	3510	3477	3476	3476	3476	3476	3476	3476	3476	3476	3476	3476	3476	3476	3476	3476	3476	3476	3476	3476	3476	3476	3476
		2050	2204	2210	2230	2230	2230	2230	2230	2230	2230	2230	2230	2230	2230	2230	2230	2230	2230	2230	2230	2230	2230	2230	2230
		1467	1306	1287	1196	1196	1196	1196	1196	1196	1196	1196	1196	1196	1196	1196	1196	1196	1196	1196	1196	1196	1196	1196	1196
		65	63	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62
PLTA PLTA	GRSM10C GRSM10C1 GRSM10C2 GRSM10C3	65	63	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62
		65	63	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62
		65	63	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62
		65	63	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62

Tabel 4.7

Data Dava Generator Sebelum Optimalisasi 10 maret 2004

[illegible]

2004, 2005a,b)

[illegible]

Tabel 4.8

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------

BEGAN SISTEM
UNIT TERMAL

JAM	REBAN SISTEM (MW)
1:00	3108
2:00	3024
3:00	2993
4:00	2974
5:00	2978
6:00	2915
7:00	2882
8:00	3202
9:00	3268
10:00	3261
11:00	3277
12:00	3220
13:00	3226
14:00	3226
15:00	3297
16:00	3372
17:00	3499
18:00	3600
19:00	3657
20:00	3642
21:00	3493
22:00	3385
23:00	3336
0:00	3316

Tabel 4.9
Data Daya Generator sebelum optimasi 13 Maret 2004

Unit / Jam	Jan 1	Jan 2	Jan 3	Jan 4	Jan 5	Jan 6	Jan 7	Jan 8	Jan 9	Jan 10	Jan 11	Jan 12	Jan 13	Jan 14	Jan 15	Jan 16	Jan 17	Jan 18	Jan 19	Jan 20	Jan 21	Jan 22	Jan 23	Jan 24
Unit 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
Unit 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
Unit 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
Unit 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 5	85	78	78	77	87	102	100	85	90	103	105	110	112	90	102	99	101	115	115	118	105	102	98	88
Unit 6	155	155	131	148	155	180	180	181	177	175	181	182	182	100	153	175	175	184	188	188	175	174	186	180
Unit 7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 9	87	81	80	82	87	83	85	85	87	93	95	102	103	93	83	97	98	115	115	118	115	97	92	87
Unit 10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 12	88	85	85	82	97	94	89	84	83	96	100	100	96	87	89	98	98	105	106	105	97	87	85	86
Unit 13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 14	134	147	142	148	158	158	136	132	155	154	147	157	155	158	126	157	157	162	163	162	156	156	148	135
Unit 15	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	88	88	88	88	72	73	72
Unit 16	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	88	88	88	87	72	73	73
Unit 17	120	122	121	121	121	121	121	121	121	120	121	121	121	121	121	120	119	172	175	175	174	114	115	114
Unit 18	121	120	120	121	120	120	121	121	120	120	121	121	121	121	122	121	126	179	180	181	180	122	120	120
Unit 19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 27	420	450	422	423	410	415	390	375	436	436	436	422	421	432	325	325	350	445	445	435	415	410	400	365
Unit 28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 33	335	335	365	365	350	320	315	366	365	367	365	377	387	365	315	325	345	390	390	390	365	362	370	370
Unit 34	75	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	90	90	90	85	85	85	85
Unit 35	75	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	90	90	90	85	85	85	85
Unit 36	75	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	90	90	90	85	85	85	85
Unit 37	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	90	90	90	85	85	85	85
Unit 38	150	150	150	150	185	150	150	135	187	185	185	187	187	185	150	150	150	170	170	170	182	182	155	155
T. Beban	1896	1864	2240	2488	2921	2886	2779	3656	4022	4020	3828	3938	3884	3988	3225	3296	3488	3794	3882	3778	3605	3017	2959	2608

Jam	BEBAN SISTEM (MW)
1.00	2896
2.00	2864
3.00	2845
4.00	2865
5.00	2921
6.00	2806
7.00	2710
8.00	2856
9.00	3002
10.00	3020
11.00	3026
12.00	3030
13.00	3016
14.00	2881
15.00	2717
16.00	2796
17.00	2869
18.00	3074
19.00	3382
20.00	3373
21.00	3205
22.00	3015
23.00	2929
0.00	2869

Tabel 4.10

Data Daya Generator Sesudah Optimasi 13 Maret 2004

Unit/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
Gen 1	341	351	370	343	366	370	370	370	354	355	361	357	370	370	370	360	370	369	376	370	370	361	360	362
Gen 2	341	356	354	354	359	370	370	370	370	370	366	357	370	370	370	370	345	370	370	370	370	343	361	370
Gen 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
Gen 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Gen 5	33	53	53	56	61	89	53	61	53	14	53	60	54	58	53	53	53	67	102	91	70	50	55	56
Gen 6	453	230	277	420	350	319	271	271	375	453	438	433	490	345	353	371	470	480	480	400	480	449	412	347
Gen 7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Gen 8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Gen 9	56	53	53	51	53	55	53	53	58	53	64	58	53	56	54	56	54	53	78	78	69	54	53	62
Gen 10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Gen 11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Gen 12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Gen 13	53	54	53	56	54	54	53	53	61	53	53	53	53	55	59	54	58	102	60	66	69	53	54	57
Gen 14	362	468	462	371	440	576	321	321	471	431	462	456	394	415	300	292	306	480	480	400	409	455	418	478
Gen 15	43	42	43	41	51	43	46	46	48	43	46	48	43	43	43	46	43	44	47	44	45	46	43	47
Gen 16	49	41	47	43	44	48	43	43	43	43	56	46	46	49	40	43	43	71	54	45	43	43	44	43
Gen 17	30	30	30	30	30	30	30	30	106	90	90	96	96	90	90	97	92	137	126	145	102	94	54	100
Gen 18	102	90	93	90	92	95	90	93	90	95	92	91	104	90	90	93	95	126	166	166	91	93	97	91
Gen 19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Gen 20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Gen 21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Gen 22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Gen 23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Gen 24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Gen 25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Gen 26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Gen 27	456	414	463	422	456	447	461	451	435	465	451	455	455	465	398	451	447	405	465	463	465	465	427	456
Gen 28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	72	72	72	72	0	0	0	0
Gen 29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Gen 30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Gen 31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Gen 32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Gen 33	162	201	162	100	170	164	169	169	202	166	164	163	184	162	200	179	178	202	202	202	202	162	173	189
Gen 34	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44
Gen 35	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44
Gen 36	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44
Gen 37	92	90	90	92	93	90	90	90	90	91	90	90	90	90	90	91	91	118	126	123	103	90	97	90
Gen 38	93	90	90	90	90	90	90	90	90	90	90	90	90	91	90	98	92	159	102	101	101	102	90	90
T. beban	2591	2864	2845	2864	2921	2806	2710	2556	3002	3020	3026	3030	3016	2901	2717	2796	2169	3374	3362	3313	3105	3015	2729	2869

Unit	13 Maret 2004
Gen 1	4155
Gen 2	4155
Gen 3	4155
Gen 4	4155
Gen 5	4155
Gen 6	4155
Gen 7	4155
Gen 8	4155
Gen 9	4155
Gen 10	4155
Gen 11	4155
Gen 12	4155
Gen 13	4155
Gen 14	4155
Gen 15	4155
Gen 16	4155
Gen 17	4155
Gen 18	4155
Gen 19	4155
Gen 20	4155
Gen 21	4155
Gen 22	4155
Gen 23	4155
Gen 24	4155
Gen 25	4155
Gen 26	4155
Gen 27	4155
Gen 28	4155
Gen 29	4155
Gen 30	4155
Gen 31	4155
Gen 32	4155
Gen 33	4155
Gen 34	4155
Gen 35	4155
Gen 36	4155
Gen 37	4155
Gen 38	4155
Gen 39	4155
Gen 40	4155
Gen 41	4155
Gen 42	4155
Gen 43	4155
Gen 44	4155
Gen 45	4155
Gen 46	4155
Gen 47	4155
Gen 48	4155
Gen 49	4155
Gen 50	4155
Gen 51	4155
Gen 52	4155
Gen 53	4155
Gen 54	4155
Gen 55	4155
Gen 56	4155
Gen 57	4155
Gen 58	4155
Gen 59	4155
Gen 60	4155
Gen 61	4155
Gen 62	4155
Gen 63	4155
Gen 64	4155
Gen 65	4155
Gen 66	4155
Gen 67	4155
Gen 68	4155
Gen 69	4155
Gen 70	4155
Gen 71	4155
Gen 72	4155
Gen 73	4155
Gen 74	4155
Gen 75	4155
Gen 76	4155
Gen 77	4155
Gen 78	4155
Gen 79	4155
Gen 80	4155
Gen 81	4155
Gen 82	4155
Gen 83	4155
Gen 84	4155
Gen 85	4155
Gen 86	4155
Gen 87	4155
Gen 88	4155
Gen 89	4155
Gen 90	4155
Gen 91	4155
Gen 92	4155
Gen 93	4155
Gen 94	4155
Gen 95	4155
Gen 96	4155
Gen 97	4155
Gen 98	4155
Gen 99	4155
Gen 100	4155

Tabel 4.11
Data Daya Generator sebelum optimasi 14 Maret 2004

Unit	Jan 1	Jan 2	Jan 3	Jan 4	Jan 5	Jan 6	Jan 7	Jan 8	Jan 9	Jan 10	Jan 11	Jan 12	Jan 13	Jan 14	Jan 15	Jan 16	Jan 17	Jan 18	Jan 19	Jan 20	Jan 21	Jan 22	Jan 23	Jan 24
Unit 1	370	350	350	350	370	316	316	360	360	360	360	360	360	360	360	360	360	370	370	370	370	370	370	370
Unit 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
Unit 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
Unit 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 5	80	73	73	72	82	87	86	80	81	87	100	105	103	85	87	84	98	100	102	102	100	97	94	83
Unit 6	340	320	326	343	350	375	366	356	356	376	348	350	343	370	372	376	376	376	376	375	376	376	382	452
Unit 7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 9	48	50	50	50	50	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55
Unit 10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 13	129	142	137	141	153	183	131	127	134	143	142	143	146	137	121	152	150	134	147	162	142	137	131	132
Unit 14	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
Unit 15	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
Unit 16	145	150	150	150	150	150	150	150	150	150	150	150	150	150	150	150	150	150	150	150	150	150	150	150
Unit 17	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90
Unit 18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 26	350	325	325	325	350	300	350	350	350	350	350	350	350	350	350	350	350	350	350	350	350	350	350	350
Unit 27	65	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
Unit 28	65	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
Unit 29	65	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
Unit 30	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90
Unit 31	150	150	150	150	150	150	150	150	150	150	150	150	150	150	150	150	150	150	150	150	150	150	150	150
Unit 32	2815	2678	2675	2694	2804	3011	2788	2746	2802	2816	2853	2768	2749	2657	2613	2709	2714	3255	3365	3369	3882	3876	2864	2832

Unit	BEAL (SYSTEM, MW)
Unit 1	370
Unit 2	0
Unit 3	0
Unit 4	0
Unit 5	80
Unit 6	340
Unit 7	0
Unit 8	0
Unit 9	48
Unit 10	0
Unit 11	0
Unit 12	0
Unit 13	129
Unit 14	50
Unit 15	50
Unit 16	145
Unit 17	90
Unit 18	0
Unit 19	0
Unit 20	0
Unit 21	0
Unit 22	0
Unit 23	0
Unit 24	0
Unit 25	0
Unit 26	350
Unit 27	65
Unit 28	65
Unit 29	65
Unit 30	90
Unit 31	150
Unit 32	2815

Tabel 4.12
Data Daya Generator sesudah optimasi 14 Maret 2004

	Jan 1	Jan 2	Jan 3	Jan 4	Jan 5	Jan 6	Jan 7	Jan 8	Jan 9	Jan 10	Jan 11	Jan 12	Jan 13	Jan 14	Jan 15	Jan 16	Jan 17	Jan 18	Jan 19	Jan 20	Jan 21	Jan 22	Jan 23	Jan 24
Unit 1	370	344	367	370	370	335	366	349	358	360	362	365	365	346	370	370	370	368	370	370	370	370	370	365
Unit 2	370	370	367	370	370	285	370	370	368	368	364	370	368	337	356	365	370	370	370	370	355	368	370	363
Unit 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
Unit 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 5	53	53	53	53	63	53	58	54	53	56	58	53	53	53	53	53	53	55	72	90	55	53	53	58
Unit 6	421	314	315	301	280	284	275	369	307	306	304	403	256	269	289	289	490	490	460	460	436	347	481	500
Unit 7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 9	53	53	53	53	53	53	53	53	53	56	53	53	53	53	53	53	53	76	90	96	57	53	56	55
Unit 10	0	0	0	0	0	0	0	0	94	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 11	0	0	0	0	0	0	0	0	0	97	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 12	53	53	53	53	53	63	56	53	56	53	53	53	53	61	53	55	59	66	56	53	54	60	54	53
Unit 13	0	0	0	0	0	0	0	0	82	0	82	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 14	381	301	292	299	367	351	260	278	391	414	425	298	271	337	260	290	335	476	460	480	422	430	467	470
Unit 15	45	43	43	44	43	45	46	45	46	43	43	48	44	54	49	53	43	43	58	43	43	50	44	53
Unit 16	46	46	46	46	43	43	43	46	43	43	46	43	45	45	43	44	45	48	57	43	51	43	43	43
Unit 17	97	90	97	95	92	91	90	90	90	91	96	94	100	92	94	92	90	133	111	159	90	92	90	90
Unit 18	80	80	80	80	101	80	96	84	80	82	84	84	83	86	82	80	96	119	140	87	84	86	80	99
Unit 19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Unit 33	162	167	162	166	181	168	186	192	184	182	168	198	162	198	182	185	162	198	201	197	192	170	169	172
Unit 34	44	44	44	44	44	44	44	46	44	44	44	44	44	44	44	46	44	44	44	44	44	44	44	44
Unit 35	44	44	44	44	44	44	44	46	44	44	44	44	44	44	45	44	44	44	44	44	44	44	44	44
Unit 36	44	44	44	44	44	44	44	46	44	44	44	44	44	44	45	44	44	44	44	44	44	44	44	44
Unit 37	87	90	93	96	81	84	80	80	80	80	80	80	80	80	80	85	80	124	107	123	80	90	90	90
Unit 38	80	80	80	80	90	90	90	90	91	90	90	90	90	90	90	90	93	96	109	130	94	90	90	90
T. Beban	2816	2664	2845	2666	2923	2806	2716	2846	3002	3020	3036	3036	3036	2901	2717	2796	2869	3374	3382	3373	3385	3015	2929	2869
Jam																								
BEBAN SISTEM (MW)																								
14 Maret 2004	3598	3598	3598	3598	3598	3598	3598	3598	3598	3598	3598	3598	3598	3598	3598	3598	3598	3598	3598	3598	3598	3598	3598	3598

unit uAbout;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs;

type

TfrmAbout = class(TForm)

private

{ Private declarations }

public

{ Public declarations }

end;

var

frmAbout: TfrmAbout;

implementation

{SR *.dfm}

end.

unit uComplex;

interface

uses uUtils;

type

TComplex=class

private

FReal,FImag:double;

public

constructor Create;overload;

constructor Create(const aReal:double);overload;

constructor Create(const aReal,aImag:double);overload;

constructor Create(const aComplex:TComplex);overload;

function GetAbs:double;

function GetAngleRad:double;

function GetAngleDeg:double;

function Add(const aReal:double):TComplex;overload;

function Add(const aComplex:TComplex):TComplex;overload;

function Subtract(const aReal:double):TComplex;overload;

function Subtract(const aComplex:TComplex):TComplex;overload;

function Multiply(const aReal:double):TComplex;overload;

function Multiply(const aComplex:TComplex):TComplex;overload;

```

function Divide(const aReal:double):TComplex;overload;
function Divide(const aComplex:TComplex):TComplex;overload;
function Conj:TComplex;
function Negative:TComplex;
function toStringI(const rLen:integer):string;
function toStringJ(const rLen:integer):string;
property Real:double read FReal write FReal;
property Imag:double read FImag write FImag;
end;

```

```

CArr1=array of TComplex;
CArr2=array of array of TComplex;

```

implementation

```

{ TComplex }
//constructor
constructor TComplex.Create;
begin
    inherited Create;
    FReal:=0.0;
    FImag:=0.0;
end;

constructor TComplex.Create(const aReal:double);
begin
    inherited Create;
    FReal:=aReal;
    FImag:=0.0;
end;

constructor TComplex.Create(const aReal,almag:double);
begin
    inherited Create;
    FReal:=aReal;
    FImag:=almag;
end;

constructor TComplex.Create(const aComplex:TComplex);
begin
    inherited Create;
    FReal:=aComplex.FReal;
    FImag:=aComplex.FImag;
end;

//data operation

function TComplex.Add(const aReal:double):TComplex;
begin

```

```

    result:=TComplex.Create((FReal+aReal),FImag);
end;

function TComplex.Add(const aComplex:TComplex):TComplex;
begin
    result:=TComplex.Create((FReal+aComplex.FReal),(FImag+aComplex.FImag));
end;

function TComplex.Subtract(const aReal:double):TComplex;
begin
    result:=TComplex.Create((FReal-aReal),FImag);
end;

function TComplex.Subtract(const aComplex:TComplex):TComplex;
begin
    result:=TComplex.Create((FReal-aComplex.FReal),(FImag-aComplex.FImag));
end;

function TComplex.Multiply(const aReal:double):TComplex;
begin
    result:=TComplex.Create((aReal*FReal),(aReal*FImag));
end;

function TComplex.Multiply(const aComplex:TComplex):TComplex;
begin
    result:=TComplex.Create((FReal*aComplex.FReal-FImag*aComplex.FImag),
        (FReal*aComplex.FImag+FImag*aComplex.FReal));
end;

function TComplex.Divide(const aReal:double):TComplex;
begin
    result:=TComplex.Create((FReal/aReal),(FImag/aReal));
end;

function TComplex.Divide(const aComplex:TComplex):TComplex;
var denote:double;
begin
    denote:=sqrt(aComplex.FReal)+sqrt(aComplex.FImag);

    result:=TComplex.Create(((FReal*aComplex.FReal+FImag*aComplex.FImag)/den
ote),
        ((FImag*aComplex.FReal-FReal*aComplex.FImag)/denote));
end;

function TComplex.GetAbs:double;
begin
    result:=sqrt(sqrt(FReal)+sqrt(FImag));
end;

```

```

function TComplex.GetAngleRad:double;
begin
  result:=arctan(FImag/FReal);
end;

function TComplex.GetAngleDeg:double;
var phi:double;
begin
  phi:=4*arctan(1);
  result:=GetAngleRad*180/phi;
end;

function TComplex.Conj:TComplex;
begin
  result:=TComplex.Create(FReal,-FImag);
end;

function TComplex.Negative:TComplex;
begin
  result:=TComplex.Create((FReal*-1),(FImag*-1));
end;

function TComplex.toStringI(const rLen:integer):string;
begin
  if FImag<0 then
    begin
      result:=RealToStr(FReal,rLen)+'-'+RealToStr(FImag,rLen)+'i';
    end
  else
    begin
      result:=RealToStr(FReal,rLen)+'+'+RealToStr(FImag,rLen)+'i';
    end;
end;

function TComplex.toStringJ(const rLen:integer):string;
begin
  if FImag<0 then
    begin
      result:=RealToStr(FReal,rLen)+'- j'+RealToStr(FImag,rLen);
    end
  else
    begin
      result:=RealToStr(FReal,rLen)+'+ j'+RealToStr(FImag,rLen);
    end;
end;

end.

```

unit uMatrix;

interface

uses uUtils, SysUtils;

```
function MatrixAdd(const mat1:dArr2;const aValue:double):dArr2;overload;  
function MatrixAdd(const mat1,mat2:dArr2):dArr2;overload;  
function MatrixSub(const mat1:dArr2;const aValue:double):dArr2;overload;  
function MatrixSub(const mat1,mat2:dArr2):dArr2;overload;  
function MatrixMul(const mat1:dArr2;const aValue:double):dArr2;overload;  
function MatrixMul(const mat1:dArr2;const mat2:dArr1):dArr1;overload;  
function MatrixMul(const mat1,mat2:dArr2):dArr2;overload;  
function MatrixInvers(const mat1:dArr2):dArr2;  
function MatrixTranspose(const mat1:dArr2):dArr2;  
function MatrixNegative(const mat1:dArr2):dArr2;  
function EllGauss(const mat1:dArr2;const mat2:dArr1):dArr1;
```

implementation

```
function MatrixAdd(const mat1:dArr2;const aValue:double):dArr2;  
var i,j:integer;  
begin  
  SetLength(result,high(mat1)+1,high(mat1[0])+1);  
  for i:=0 to high(mat1) do  
    begin  
      for j:=0 to high(mat1[0]) do  
        begin  
          result[i,j]:=mat1[i,j]+aValue;  
        end;  
      end;  
    end;  
end;  
  
function MatrixAdd(const mat1,mat2:dArr2):dArr2;  
var i,j:integer;  
begin  
  if (high(mat1) <> high(mat2)) or (high(mat1[0]) <> high(mat2[0])) then  
    begin  
      raise Exception.Create('row dan col kedua matrik tidak sama!');  
    end;  
  SetLength(result,high(mat1)+1,high(mat1[0])+1);  
  for i:=0 to high(mat1) do  
    begin  
      for j:=0 to high(mat1[0]) do  
        begin  
          result[i,j]:=mat1[i,j]+mat2[i,j];  
        end;  
      end;  
    end;  
end;
```

```

function MatrixSub(const mat1:dArr2;const aValue:double):dArr2;
var i,j:integer;
begin
  SetLength(result,high(mat1)+1,high(mat1[0])+1);
  for i:=0 to high(mat1) do
    begin
      for j:=0 to high(mat1[0]) do
        begin
          result[i,j]:=mat1[i,j]-aValue;
        end;
      end;
    end;
end;

```

```

function MatrixSub(const mat1,mat2:dArr2):dArr2;
var i,j:integer;
begin
  if (high(mat1) <> high(mat2)) or (high(mat1[0]) <> high(mat2[0])) then
    begin
      raise Exception.Create('Row dan Col kedua matrik tidak sama!');
    end;
  SetLength(result,high(mat1)+1,high(mat1[0])+1);
  for i:=0 to high(mat1) do
    begin
      for j:=0 to high(mat1[0]) do
        begin
          result[i,j]:=mat1[i,j]-mat2[i,j];
        end;
      end;
    end;
end;

```

```

function MatrixMul(const mat1:dArr2;const aValue:double):dArr2;
var i,j:integer;
begin
  SetLength(result,high(mat1)+1,high(mat1[0])+1);
  for i:=0 to high(mat1) do
    begin
      for j:=0 to high(mat1[0]) do
        begin
          result[i,j]:=mat1[i,j]*aValue;
        end;
      end;
    end;
end;

```

```

function MatrixMul(const mat1:dArr2;const mat2:dArr1):dArr1;
var i,j,k:integer;
    sum:double;
begin
  if high(mat1[0]) <> high(mat2) then

```

```

begin
    raise Exception.Create('Jumlah kolom matrik1 dan jumlah baris matrik2 tidak
sama');
end;
SetLength(result,high(mat2)+1);
for i:=0 to high(mat1) do
begin
    for j:=0 to high(mat2) do
begin
        sum:=0.0;
        for k:=0 to high(mat1[0]) do
begin
            sum:=sum+mat1[i,k]*mat2[k];
end;
        result[i]:=sum;
end;
end;
end;

function MatrixMul(const mat1,mat2:dArr2):dArr2;
var i,j,k:integer;
    sum:double;
begin
    if high(mat1[0])<>high(mat2) then
begin
        raise Exception.Create('Jumlah kolom matrik1 dan jumlah baris matrik2 tidak
sama');
end;
SetLength(result,high(mat1)+1,high(mat2[0])+1);
for i:=0 to high(mat1) do
begin
    for j:=0 to high(mat2[0]) do
begin
        sum:=0;
        for k:=0 to high(mat1[0]) do
begin
            sum:=sum+mat1[i,k]*mat2[k,j];
end;
        result[i,j]:=sum;
end;
end;
end;

function MatrixInvers(const mat1:dArr2):dArr2;
var i,j,k:integer;
    A,D:double;
begin
    if high(mat1)<>high(mat1[0]) then
begin

```

```

    raise Exception.Create('Bukan matrik bujur sangkat!');
end;
SetLength(result,high(mat1)+1,high(mat1[0])+1);
for i:=0 to high(mat1) do
begin
    for j:=0 to high(mat1[0]) do
    begin
        result[i,j]:=mat1[i,j];
    end;
end;
try
    for i:=0 to high(result) do
    begin
        D:=result[i,i];
        result[i,i]:=1;
        for j:=0 to high(result[0]) do
        begin
            if D=0 then
            begin
                D:=0.00001;
            end;
            result[i,j]:=result[i,j]/D;
        end;
        for k:=0 to high(result) do
        begin
            if k<>i then
            begin
                A:=result[k,i];
                result[k,i]:=0;
                for j:=0 to high(result[0]) do
                begin
                    result[k,j]:=result[k,j]-A*result[i,j];
                end;
            end;
        end;
    end;
except
    raise exception.Create('matrik tidak bisa diinvers!');
end;
end;

```

```

function MatrixTranspose(const mat1:dArr2):dArr2;
var i,j:integer;
begin
    SetLength(result,high(mat1)+1,high(mat1[0])+1);
    for i:=0 to high(mat1) do
    begin
        for j:=0 to high(mat1[0]) do
        begin

```

```

    result[j,i]:=mat1[i,j];
  end;
end;
end;

```

```

function MatrixNegative(const mat1:dArr2):dArr2;
var i,j:integer;
begin
  SetLength(result,high(mat1)+1,high(mat1[0])+1);
  for i:=0 to high(mat1) do
    begin
      for j:=0 to high(mat1[0]) do
        begin
          result[i,j]:=mat1[i,j]*-1;
        end;
      end;
    end;
  end;
end;

```

```

function EllGauss(const mat1:dArr2;const mat2:dArr1):dArr1;
var i,j,k,Nmat:integer;
    konst,value,DE,AF,sum:double;
    tmp:dArr2;
begin
  if high(mat1)<>high(mat1[0]) then
    begin
      raise Exception.Create('Matrik 1 bukan matrik bujur sangkar!');
    end;
  if high(mat1[0])<>high(mat2) then
    begin
      raise Exception.Create('Jumlah kolom matrik 1 tidak sama dengan jumlah baris matrik 2!');
    end;
  Nmat:=high(mat1)+1;
  SetLength(tmp,Nmat,Nmat+1);
  for i:=0 to Nmat-1 do
    begin
      for j:=0 to Nmat-1 do
        begin
          tmp[i,j]:=mat1[i,j];
        end;
      end;
    end;
  for i:=0 to Nmat-1 do
    begin
      tmp[i,Nmat]:=mat2[i];
    end;
  for i:=0 to Nmat-1 do
    begin
      if tmp[i,i]≠0 then
        begin

```

```

for k:=i+1 to Nmat-1 do
begin
  if tmp[k,i]<>0 then
  begin
    for j:=0 to Nmat do
    begin
      Konst:=tmp[i,j];
      value:=tmp[k,j];
      tmp[i,j]:=value;
      tmp[k,j]:=Konst;
    end;
  end
end;
end;
end;
for i:=0 to Nmat-1 do
begin
  DE:=tmp[i,i];
  for j:=0 to Nmat do
  begin
    tmp[i,j]:=tmp[i,j]/DE;
  end;
  for k:=i to Nmat-1 do
  begin
    if k<>i then
    begin
      AE:=tmp[k,i];
      for j:=0 to Nmat do
      begin
        tmp[k,j]:=tmp[k,j]-AE*tmp[i,j];
      end;
    end;
  end;
end;
SetLength(result,Nmat);
for i:=Nmat-1 downto 0 do
begin
  if i<>Nmat-1 then
  begin
    Sum:=0;
    for j:=i+1 to Nmat-1 do
    begin
      Sum:=Sum+tmp[i,j]*result[j];
    end;
    result[i]:=tmp[i,Nmat]-Sum;
  end
  else
  begin
    result[i]:=tmp[i,Nmat];
  end
end;

```

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

```
end.
```

```
unit uMenu;
```

```
interface
```

```
uses
```

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

```
type
```

```
    TfrmMenuUtama = class(TForm)
```

```
        Panel1: TPanel;
```

```
        btnNew: TButton;
```

```
        btnOpen: TButton;
```

```
        btnExit: TButton;
```

```
        StatusBar1: TStatusBar;
```

```
        Panel2: TPanel;
```

```
        OpenDialog1: TOpenDialog;
```

```
        Label1: TLabel;
```

```
        Label2: TLabel;
```

```
        Label3: TLabel;
```

```
        Label4: TLabel;
```

```
        Button1: TButton;
```

```
        Button2: TButton;
```

```
        Button3: TButton;
```

```
        Label5: TLabel;
```

```
        Label6: TLabel;
```

```
        Image2: TImage;
```

```
        procedure btnNewClick(Sender: TObject);
```

```
        procedure btnExitClick(Sender: TObject);
```

```
        procedure btnOpenClick(Sender: TObject);
```

```
    private
```

```
        { Private declarations }
```

```
    public
```

```
        { Public declarations }
```

```
end;
```

```
var
```

```
    frmMenuUtama: TfrmMenuUtama;
```

```
implementation
```

```
uses uInput, uLamda, uGenerator, uUtils;
```

```
{SR *.dfm}
```

```
procedure TfrmMenuUtama.btnNewClick(Sender: TObject);  
begin  
  frmInput.Caption:='Input Data';  
  frmInput.btnNext.Caption:='&Save';  
  frmInput.Show;  
end;
```

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

```
procedure TfrmMenuUtama.btnOpenClick(Sender: TObject);  
var Output: TextFile;  
    Pmax, Pmin, a0, a1, a2, Sh, Sc, Ramp, Load1, Res1: double;  
    tup, tdown, tcold, iniSt, Ngen1, Njam1: integer;  
    Nama, NamaFile: string;  
    gen: TGenArr;  
    Beban, Res: dArr1;  
    PLN: dArr2;  
    i, j: byte;  
begin  
  fit := TLamda.Create;  
  if OpenFileDialog1.Execute then  
  begin  
    NamaFile := OpenFileDialog1.FileName;  
    AssignFile(output, Namafile);  
    Reset(output);  
    Readln(output, Ngen1);  
    Readln(output, Njam1);  
    fit.Ngen := Ngen1;  
    fit.Njam := Njam1;  
    SetLength(gen, Ngen1 + 1);  
    SetLength(Beban, Njam1 + 1);  
    SetLength(Res, Njam1 + 1);  
    SetLength(PLN, Ngen1 + 1, Njam1 + 1);  
    frmInput.edtNgen.Text := IntToStr(Ngen1);  
    frmInput.edtNjam.Text := IntToStr(Njam1);  
    for i := 1 to Ngen1 do  
    begin  
      Readln(output, Pmax, Pmin, a0, a1, a2, Tup, Tdown, Sh, Sc, tcold, iniSt,  
        Ramp, Nama);  
      gen[i] := TGenerator.Create;
```

```

gen[i].Nama:=Nama;
gen[i].Pmin:=Pmin;
gen[i].Pmax:=Pmax;
gen[i].a2:=a2;
gen[i].a1:=a1;
gen[i].a0:=a0;
gen[i].Sh:=Sh;
gen[i].Sc:=Sc;
gen[i].RampRate:=Ramp;
gen[i].Tup:=tup;
gen[i].Tdown:=Tdown;
gen[i].Tcold:=Tcold;
gen[i].InitSt:=iniSt;
frmInput.fgGen.Cells[0,i]:=IntToStr(i);
frmInput.fgGen.Cells[1,i]:=Nama;
frmInput.fgGen.Cells[2,i]:=RealToStr(Pmin,0);
frmInput.fgGen.Cells[3,i]:=RealToStr(Pmax,0);
frmInput.fgGen.Cells[4,i]:=RealToStr(a2,5);
frmInput.fgGen.Cells[5,i]:=RealToStr(a1,4);
frmInput.fgGen.Cells[6,i]:=RealToStr(a0,4);
frmInput.fgGen.Cells[7,i]:=RealToStr(Sh,0);
frmInput.fgGen.Cells[8,i]:=RealToStr(Sc,0);
frmInput.fgGen.Cells[9,i]:=RealToStr(Ramp,0);
frmInput.fgGen.Cells[10,i]:=IntToStr(tup);
frmInput.fgGen.Cells[11,i]:=IntToStr(tdown);
frmInput.fgGen.Cells[12,i]:=IntToStr(tcold);
frmInput.fgGen.Cells[13,i]:=IntToStr(iniSt);
end;
fit.Gen:=gen;
for i:=1 to Njam1 do
begin
  Readln(Output,Load1,Res1);
  Beban[i]:=Load1;
  Res[i]:=Res1;
  frmInput.fgBeban.Cells[0,i]:=IntToStr(i);
  frmInput.fgBeban.Cells[1,i]:=RealToStr(Load1,0);
  frmInput.fgBeban.Cells[2,i]:=RealToStr(Res1,0);
end;
fit.Beban:=Beban;
for i:=1 to Ngen1 do
begin
  gen[i].Free;
  for j:=1 to Njam1 do
  begin
    Read(Output,PLN[i,j]);
    frmInput.fgPLN.Cells[j,i]:=RealToStr(PLN[i,j],0);
  end;
  Readln(output);
end;

```

```

    CloseFile(output);
    fit.PLN:=PLN;
    frmInput.Caption:='Tampilan Data';
    frmInput.btnNext.Caption:='&Next';
    frmInput.Show;
end;
end;

end.

unit uUtils;

interface

uses SysUtils;

type
    TSort=(asc,dec);

    TBatas=record
        min,max:double;
    end;

    TBatasArr1=array of TBatas;
    TBatasArr2=array of array of TBatas;

    dArr1=array of double;
    dArr2=array of array of double;
    iArr1=array of integer;
    iArr2=array of array of integer;
    bArr1=array of boolean;
    bArr2=array of array of boolean;
    sArr1=array of String;

    TAlleleTCSC=record
        Lokasi,TypeAlat,Setting:double;
    end;

    TChromTCSC1=array of TAlleleTCSC;

    TAlleleUpfc=record
        Status:boolean;
        TypeAlat,Iap,Sudut:double;
    end;

    TChromUpfc1=array of TAlleleUpfc;

    function RealToStr(Num:double;Pecahan:byte):String;
    function StrToReal(Huruf:string):double;

```

```

function Pangkat(Val,pangkat:double):double;

procedure Swap(var X,Y:byte);overload;
procedure Swap(var X,Y:integer);overload;
procedure Swap(var X,Y:word);overload;
procedure Swap(var X,Y:double);overload;
procedure Swap(var X,Y:extended);overload;
procedure Swap(var X,Y:string);overload;
procedure Swap(var X,Y:boolean);overload;

procedure BubleSort(var aData:dArr1;const aType:TSort);overload;
procedure BubleSort(var aData:iArr1;const aType:TSort);overload;
procedure BubleSort(var aData:sArr1;const aType:TSort);overload;

function DecodeBinToFloat1(const aData:bArr1):double;
function DecodeBinToFloat2(const aData:bArr2):dArr1;
function DecodeFloat1ToBin(const aLength:integer;
    const aData:double):bArr1;
function DecodeFloat2ToBin(const aLength:integer;
    const aData:dArr1):bArr2;

function GetBatas(const aValue,aMin,aMax:double):double;
function GetFlip(const aFlip:double):boolean;

function GetBatasToReal(const aValue,aMin,aMax:double):double;
function GetRealToBatas(const aValue,aMin,aMax: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;

```

end;

```
procedure Swap(var X,Y:byte);  
var tmp:byte;  
begin  
  tmp:=X;  
  X:=Y;  
  Y:=tmp;  
end;
```

```
procedure Swap(var X,Y:integer);  
var tmp:integer;  
begin  
  tmp:=X;  
  X:=Y;  
  Y:=tmp;  
end;
```

```
procedure Swap(var X,Y:word);  
var tmp:word;  
begin  
  tmp:=X;  
  X:=Y;  
  Y:=tmp;  
end;
```

```
procedure Swap(var X,Y:double);  
var tmp:double;  
begin  
  tmp:=X;  
  X:=Y;  
  Y:=tmp;  
end;
```

```
procedure Swap(var X,Y:extended);  
var tmp:extended;  
begin  
  tmp:=X;  
  X:=Y;  
  Y:=tmp;  
end;
```

```
procedure Swap(var X,Y:string);  
var tmp:string;  
begin  
  tmp:=X;  
  X:=Y;  
  Y:=tmp;  
end;
```

```

procedure Swap(var X,Y:boolean);
var tmp:boolean;
begin
  tmp:=X;
  X:=Y;
  Y:=tmp;
end;

```

```

procedure BubleSort(var aData:dArr1;const aType:TSort);
var i,j:integer;
begin
  for i:=1 to (high(aData)-1) do
    begin
      for j:=i to high(aData) do
        begin
          if aType=asc then
            begin
              if aData[i]>aData[j] then
                begin
                  Swap(aData[i],aData[j]);
                end;
            end
          else if aType=dec then
            begin
              if aData[i]<aData[j] then
                begin
                  Swap(aData[i],aData[j]);
                end;
            end;
          end;
        end;
      end;
    end;
  end;
end;

```

```

procedure BubleSort(var aData:iArr1;const aType:TSort);
var i,j:integer;
begin
  for i:=1 to (high(aData)-1) do
    begin
      for j:=i to high(aData) do
        begin
          if aType=asc then
            begin
              if aData[i]>aData[j] then
                begin
                  Swap(aData[i],aData[j]);
                end;
            end
          else if aType=dec then

```

```

begin
  if aData[i]<aData[j] then
    begin
      Swap(aData[i],aData[j]);
    end;
  end;
end;
end;
end;

procedure BubbleSort(var aData:sArr1;const aType:TSort);
var i,j:integer;
begin
  for i:=1 to (high(aData)-1) do
    begin
      for j:=i to high(aData) do
        begin
          if aType=asc then
            begin
              if aData[i]>aData[j] then
                begin
                  Swap(aData[i],aData[j]);
                end;
            end
          else if aType=dec then
            begin
              if aData[i]<aData[j] then
                begin
                  Swap(aData[i],aData[j]);
                end;
            end;
          end;
        end;
      end;
    end;
  end;

function DecodeBinToFloat1(const aData:bArr1):double;
var i:integer;
    powerof2,sa:double;
begin
  result:=0;
  powerof2:=1;
  sa:=pangkat(2,high(aData))-1;
  for i:=high(aData) downto 1 do
    begin
      if aData[i]=true then
        begin
          result:=result+powerof2;
        end;
        powerof2:=powerof2*2;
      end;
    end;
  end;
end;

```

```

end;
result:=result/sa;
end;

```

```

function DecodeBinToFloat2(const aData:bArr2):dArr1;
var i,j:integer;
    Data:bArr1;
begin
    SetLength(Data,high(aData[0])+1);
    SetLength(result,high(aData)+1);
    for i:=1 to high(aData) do
        begin
            for j:=1 to high(aData[0]) do
                begin
                    Data[j]:=aData[i,j];
                end;
            result[i]:=DecodeBinToFloat1(Data);
        end;
    end;
end;

```

```

function DecodeFloat1ToBin(const aLength:integer,
    const aData:double):bArr1;
var i,Sa,cek,value:integer;
begin
    Sa:=round(pangkat(2,aLength))-1;
    value:=round(aData*Sa);
    SetLength(result,aLength+1);
    for i:=1 to aLength do
        begin
            result[i]:=false;
        end;
    i:=aLength;
    Repeat
        Cek:=value mod 2;
        if Cek=1 then
            begin
                result[i]:=true;
            end
        else
            begin
                result[i]:=false;
            end;
        value:=value div 2;
        i:=i-1;
    until value=0;
end;

```

```

function DecodeFloat2ToBin(const aLength:integer,
    const aData:dArr1):bArr2;

```