

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



SKRIPSI

**OPTIMASI DAYA REAKTIF SECARA EKONOMIS
MENGUNAKAN METODE *EVOLUTIONARY PROGRAMMING*
PADA SALURAN TRANSMISI 150 kV
SUB SISTEM PAITON DAN BALI**

Disusun oleh :

**JOKO NUGROHO
00.12.034**

APRIL 2005



LEMBAR PERSETUJUAN

**OPTIMASI DAYA REAKTIF SECARA EKONOMIS MENGGUNAKAN
METODE *EVOLUTIONARY PROGRAMMING* PADA SALURAN
TRANSMISI 150 kV SUB SISTEM PAITON DAN BALI**


SKRIPSI

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


Disusun Oleh :

**JOKO NUGROHO
NIM. 00.12.034**

**Mengetahui,
Ketua Jurusan Teknik Elektro**


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

**Menyetujui,
Dosen Pembimbing**


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

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

Lembar Persembahan

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Kupersembahkan skripsiku ini special untuk



ALLAH SWT, atas semua rahmat dan hidayah-MU
yang telah Engkau berikan kepadaku
Serta junjunganku Nabi Muhammad SAW yang telah memberi
sauri teladan-NYA kepada seluruh umat manusia
Hanya kepada-MU aku berserah diri dan minta petunjuk
Dalam hidup ini dengan iman dan taqwa
Bapak dan Ibu, serta Kakak-kakakku semua : Mas Agus,
Mas Anto, Mas Arifin dan Mas Heru
Terima kasih atas doa, dorongan dan semangatnya sehingga
Joko bisa menyelesaikan kuliah, semoga Allah SWT
selalu memberikan Rahmat dan hidayat pada kita semua, Amin.
Enaku sayang, yang selalu menemaniku dalam kondisi
Susah, senang, bingung dan ragu...
Semoga Ena tetap mengertiin aku saat kondisi yang
bagaimapun, Selalu sayang dan setia...padaku [Wimas]



Thank's for.....

Dosen Pembimbing

Bapak Ir. Yusuf Ismail Nakhoda, MT yang telah membimbing dan membantu saya dalam penyusunan skripsi ini. Mohon maaf atas kesalahan yang pernah saya perbuat, semoga Bapak memaafkannya. Dan doakan saya agar menjadi orang yang sukses serta banyak rejeki Amin...

DwiAris JEMBER [00']

Suwun seng akeehh Ris, atas semua bantuan yang kau berikan padaku, Aku mungkin ga' bisa membalas semua kebaikanmu, semoga Allah SWT selalu memberikan yang terbaik untukmu, Amin...

(Good by RAMPALL... Perjuangan qta akan menjadi mEmoRy terIndah saat di 'MALANG', ojo L@LI Ketik SMS spasi JOKO MedION kirim Ke 0351-331034. Persaudaraan Kita ta'kan terpuTuS sampai Disini.. RIS)

TEman2 SePenDeriT@4N

Gu\$ (Thingho)... Thank's 4 ALL kapan2 aku maen kerumahmu lagi(Pasuruan) jo' Lali cepakno Ndok "PUYUH" O.K, Tit4_Heh inget AKU YAA>> kalo aku ke Bandung, Boleh Mampir KAN?? Juga buat GHoKTheng(AGUS cepu) BlimBinge sek Akeh Ko' tapi menek'o dewe yo..

Tak Lupa ReZA(Prendku yg baik),Lingg@,irFaN(sMaga awet ma pandu Ye),Ali(yg kocak Abis),Andre(yg baik bgt),Mansyah(COOl Men),Hari(pendiem),Dedi gusur (Editor Programku thank's ya,aku jd bisa Sminar),Yusuf,LI@(iMut),Pandu(Makasih catatannya),Dwi

TEman2 SebiMbiNg4N

UbEt,Arif,Dwi,Jantur,IkHs@N(mas AcoNg thank's, 4 SMS nya),BoWo,Nur,dan Mas Andre. Kalian Semua Sangat Baik ama Aku, Kita Tetep JaGa PersaHabat4N ini YAAA... dan JAngAn LuPa KeTik SMS ke JOKO Spasi O>K : 081334647900 , kaLO Ada LoWongan Kerja kabari AKU YOOO...

ABSTRAKSI

OPTIMASI DAYA REAKTIF SECARA EKONOMIS MENGUNAKAN METODE *EVOLUTIONARY PROGRAMMING* PADA SALURAN TRANSMISI 150 kV SUB SISTEM PAITON DAN BALI

(Joko Nugroho, Nim. 00.12.034, Teknik Elektro S-1/Teknik Energi Listrik)
(Dosen Pembimbing : Ir Yusuf Ismail Nakhoda, MT)

Kata Kunci: *Optimal Reactive Power Planning, Evolutionary programming*
Optimasi Biaya Pembangkitan.

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

Skripsi ini menganalisis optimasi daya reaktif secara ekonomis dengan menggunakan metode *Evolutionary Programming*. Hasil dari analisa tersebut nantinya dapat digunakan sebagai salah satu acuan dalam operasi pembangkitan dan penyaluran daya yang ekonomis dan optimal, terutama mengenai optimasi biaya pembangkitan. Input dari program ini adalah biaya bahan bakar (*fuel cost*), biaya pemeliharaan dan hasil perhitungan aliran daya, sedangkan hasil akhir dari program ini yaitu hasil perhitungan aliran daya, tegangan dan sudut fasa tiap-tiap bus serta biaya pembangkitan yang optimum.

Analisa dilakukan dengan bantuan program computer dengan menggunakan bahasa pemrograman Delphi versi 7.0 dan telah sukses dicoba pada sub sistem 150 kV Paiton-Bali yang terdiri dari 25 bus, dimana telah berhasil dilakukan penghematan biaya pembangkitan sebesar Rp 87.434.785,7 atau terjadi optimasi biaya sebesar 20.38 %.

KATA PENGANTAR

Dengan rahmat Allah SWT, dan mengucapkan syukur kehadiran-Nya atas karunia yang dilimpahkan kepada saya sehingga dapat menyelesaikan skripsi yang berjudul “OPTIMASI DAYA REAKTIF SECARA EKONOMIS MENGGUNAKAN METODE *EVOLUTIONARY PROGRAMMING* PADA SALURAN TRANSMISI 150 kV SUB SISTEM PAITON DAN BALI”.

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

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

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

1. Bapak Dr. Ir. Abraham Lomi, MSEE, selaku Rektor ITN Malang.
2. Bapak Ir. F Yudi Limpraptono, MT, selaku Kajur Teknik Elektro ITN Malang.
3. Bapak Ir. Yusuf Ismail Nakhoda, MT, selaku Dosen Pembimbing.
4. Bapak Ir. Totok Prawiro, selaku pembimbing lapangan di PT. PLN (persero) P3B Jawa-Bali.

Akhirnya saya mengharapkan skripsi ini berguna dan bermanfaat bagi rekan mahasiswa khususnya jurusan Teknik Elektro.

Malang, Maret 2005

Penyusun

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PERSETUJUAN	ii
ABSTRAKSI	iii
KATA PENGANTAR	iv
DAFTAR ISI	vi
DAFTAR GAMBAR	x
DAFTAR TABEL	xii
DAFTAR GRAFIK	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan	2
1.4. Batasan Masalah	3
1.5. Metodologi Penelitian	3
1.6. Sistematika Penelitian	4
1.7. Kontribusi Penelitian	5
BAB II SISTEM TENAGA LISTRIK, OPERASI EKONOMIS DAN	
KARAKTERISTIK UNIT PEMBANGKIT	6
2.1. Sistem Tenaga Listrik	6
2.2. Saluran Transmisi	7
2.2.1. Saluran Transmisi Pendek	8

2.2.2.	Saluran Transmisi Menengah	8
2.2.3	Saluran Transmisi Panjang	10
2.3.	Daya Dalam Sistem Tenaga Listrik	11
2.3.1.	Daya Aktif (<i>Active Power</i>)	11
2.3.2.	Daya Reaktif (<i>Reactive Power</i>)	11
2.3.3.	Daya semu	12
2.4.	Sistem Per-Unit	13
2.4.1.	Mengubah Dasar Sistem Per-Unit	14
2.5.	Sistem Operasi Pada Sistem Tenaga Listrik	15
2.6.	Karakteristik Unit Pembangkit	18
2.6.1.	Karakteristik <i>Input-Output</i>	18
2.6.2.	Karakteristik <i>Heat Rate</i>	20
2.6.3.	Karakteristik <i>Incremental Heat-Rate</i> dan <i>Incremental Fuel Cost</i>	21
2.7.	<i>Economic Dispatch</i>	22
2.7.1.	Fungsi Biaya Bahan Bakar	22
2.7.2.	Fungsi Biaya Daya Reaktif Pembangkit.....	23
2.7.3.	<i>Economic Dispatch</i> Dengan Mengabaikan Rugi-rugi Transmisi.....	24
2.7.4	<i>Economic Dispatch</i> Dengan Memperhitungkan Rugi-Rugi Transmisi.....	26
BAB III OPTIMASI DAYA REAKTIF MENGGUNAKAN METODE		
	EVOLUTIONARY PROGRAMMING	28
3.1.	Analisa Aliran Daya	28
3.1.1.	Klasifikasi Bus	29
3.1.2.	Metode <i>Newton Raphson</i>	30

3.2.	<i>Evolutionary Programming (EP)</i>	32
3.2.1.	Parameter <i>Evolutionary Programming</i>	33
3.2.2.	Mekanisme <i>Evolutionary Programming</i>	35
3.3.	Formulasi Masalah Optimasi Daya Reaktif	37
3.4.	Adaptasi <i>Evolutionary Programming</i> Ke Masalah Optimasi Daya Reaktif	38
3.4.1.	Representasi solusi	39
3.4.2.	Inialisasi (<i>Initialization</i>)	39
3.4.3.	Satistik (<i>Statistics</i>)	39
3.4.4.	Mutasi (<i>Mutation</i>)	39
3.4.5.	Kompetisi (<i>Competition</i>)	40
3.4.6.	Konvergensi (<i>Convergence</i>)	41
BAB IV HASIL DAN PEMBAHASAN		42
4.1.	Program Komputer Optimasi Daya Reaktif Menggunakan Metode <i>Evolutionary Programming</i>	42
4.1.1.	Algoritma Program	42
4.1.2.	<i>Flowchart</i> Program.....	45
4.2.	Validasi Data IEEE 30 Untuk Menyelesaikan Permasalahan Optimasi Daya Reaktif Secara Ekonomis Menggunakan Metode <i>Evolutionary Programming</i>	48
4.2.1.	Hasil Validasi IEEE 30 Dengan Menggunakan Metode <i>Evolutionary Programming</i>	50
4.3.	Data Pembangkitan Thermal Pada Sub Sistem Paiton dan Bali	52
4.4.	Data Pembangkitan dan Pembebanan 150 kV Sub Sistem Paiton dan Bali	57

4.5.	Data Saluran Transmisi 150 kV Sub Sistem Paiton dan Bali	58
4.6.	Prosedur Pelaksanaan Program Perhitungan	60
4.7.	Hasil dan Analisis Hasil Perhitungan Optimasi Daya Reaktif Menggunakan Metode <i>Evolutionary Programming</i> Pada Saluran Transmisi 150 kV Sub Sistem Paiton dan Bali	66
4.7.1.	Hasil Perhitungan Sebelum Optimasi	66
4.7.2.	Hasil Perhitungan Setelah Optimasi	69
4.8.	Perbandingan Hasil Perhitungan Sebelum dan Setelah Optimasi Daya Reaktif Menggunakan Metode <i>Evolutionary Programming</i>	72
4.8.1.	Tingkat Optimum Biaya Pembangkitan	72
4.8.2.	Perbandingan Tingkat Tegangan dan Sudut Tegangan Pada Tiap Bus	75
4.8.3.	Perbandingan Tingkat Rugi-rugi Daya Pada Saluran	76
BAB V KESIMPULAN DAN SARAN		77
5.1.	Kesimpulan	77
5.2.	Saran	78

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar 2.1.	Rangkaian Setara Saluran Transmisi	7
Gambar 2.2.	Rangkaian Setara Saluran Transmisi Pendek	8
Gambar 2.3.	Rangkaian Setara Saluran Transmisi Menengah	9
Gambar 2.4.	Diagram Skema Saluran Transmisi Panjang	10
Gambar 2.5.	Representasi Segitiga Daya	12
Gambar 2.6.	Unit Boiler-Turbin-Generator	19
Gambar 2.7.	Kurva Karakteristik <i>Input-Output</i> Pembangkit Thermal	20
Gambar 2.8.	Kurva Karakteristik <i>Heat-Rate</i> Unit Pembangkit	20
Gambar 2.9.	Kurva Karakteristik <i>Incremental Heat-Rate/Fuel Cost</i>	21
Gambar 2.10.	N Unit Pembangkit Thermal Melayani Beban P_R	24
Gambar 2.11.	N buah Pembangkit Thermal Melayani Beban P_R Melalui Saluran Transmisi	27
Gambar 3.1.	Mutasi Gaussian Dari Induk (<i>Parent</i>) (a) Menghasilkan Anak (<i>Offspring</i>) (b)	36
Gambar 4.1.	<i>Flowchart</i> Perhitungan Sebelum Optimasi	45
Gambar 4.2.	<i>Flowchart</i> Program Menggunakan <i>Evolutionary Programming</i>	46
Gambar 4.3.	<i>Flowchart</i> Perhitungan Aliran Daya Metode <i>Newton Rapshon</i>	47
Gambar 4.4.	Tampilan Parameter Validasi IEEE 30	50
Gambar 4.5.	Hasil <i>Load Flow</i>	51
Gambar 4.6.	Tampilan Hasil Validasi IEEE 30 Hasil Perhitungan	

	Biaya Pembangkitan	51
Gambar 4.7.	Diagram Segaris Jaringan Sistem Transmisi Listrik 150 kV Paiton-Bali	55
Gambar 4.8.	Tampilan Utama Program	60
Gambar 4.9.	Tampilan Masukan Data	60
Gambar 4.10.	Tampilan Data Bus.....	61
Gambar 4.11.	Tampilan Data Saluran	61
Gambar 4.12.	Tampilan Data Generator	62
Gambar 4.13.	Tampilan <i>Summary Load flow</i> Pada Kondisi Awal (Sebelum Optimasi)	62
Gambar 4.14.	Tampilan Hasil <i>Load flow</i> Pada Kondisi Awal (Sebelum Optimasi)	63
Gambar 4.15.	Tampilan Hasil Aliran Daya Pada Kondisi Awal (Sebelum Optimasi)..	63
Gambar 4.16.	Tampilan Parameter <i>Evolutionary Programming</i>	64
Gambar 4.17.	Tampilan Hasil <i>Load flow</i> Pada Kondisi Akhir (Setelah Optimasi).....	64
Gambar 4.18.	Tampilan Hasil Aliran Daya Pada Kondisi Akhir (Setelah Optimasi) ..	65
Gambar 4.19.	Tampilan <i>Summary</i> Hasil Akhir (Setelah Optimasi)	65
Gambar 4.20.	Tampilan Akhir Biaya Sebelum dan Setelah Optimasi.....	66

DAFTAR TABEL

Tabel 4.1.	Data Generator Dan Koefisien Biaya IEEE 30	48
Tabel 4.2.	Data Bus IEEE 30	48
Tabel 4.3.	Saluran Impedansi IEEE 30	49
Tabel 4.4.	Perbandingan Hasil Data Referensi Jurnal Dengan Data Optimasi	52
Tabel 4.5.	Parameter Unit Pembangkit Thermal.....	53
Tabel 4.6.	Persamaan Biaya Pembangkitan Unit Pembangkit Thermal Paiton dan Bali	54
Tabel 4.7.	Penomoran Bus 150 kV Sub Sistem Paiton dan Bali	56
Tabel 4.8.	Data Pembangkitan dan Pembebanan 150 kV Sub Sistem Paiton dan Bali	57
Tabel 4.9.	Data Saluran Transmisi 150 kV sub Sistem Paiton dan Bali	59
Tabel 4.10.	Hasil Perhitungan Tegangan, Sudut Tegangan, Pembangkitan dan Pembebanan Sebelum Optimasi	66
Tabel 4.11.	Hasil Perhitungan Aliran Daya Antar Saluran Sebelum Optimasi	67
Tabel 4.12.	Total Pembangkitan, Pembebanan dan Rugi-rugi Saluran Sebelum Optimasi.....	68
Tabel 4.13.	Hasil Perhitungan Daya Yang Dibangkitkan dan Biaya Operasi Sebelum Optimasi	68
Tabel 4.14.	Hasil Perhitungan Tegangan, Sudut Tegangan, Pembangkitan dan Pembebanan Setelah Optimasi	69

Tabel 4.15. Hasil Perhitungan Aliran Daya Antar Saluran Setelah Optimasi	70
Tabel 4.16. Total Peembangkitan, Pembebanan dan Rugi-rugi Saluran Setelah Optimasi	71
Tabel 4.17. Hasil Perhitungan Daya Yang Dibangkitkan dan Biaya Operasi Setelah Optimasi	71
Tabel 4.18. Perbandingan Tingkat Optimum Biaya Pembangkitan	73
Tabel 4.19. Perbandingan Total Pembangkitan, Pembebanan dan Rugi-rugi Saluran Sebelum dan Setelah Optimasi	74
Tabel 4.20. Perbandingan Tingkat Tegangan dan Sudut Tegangan Pada Tiap Bus	75

DAFTAR GRAFIK

Grafik 4.1.	Hasil Perhitungan Daya Yang Dibangkitkan Sebelum Optimasi	69
Grafik 4.2.	Hasil Perhitungan Daya Yang Dibangkitkan Setelah Optimasi	72
Grafik 4.3.	Perbandingan Biaya Pembangkitan Sebelum dan Setelah Optimasi	73
Grafik 4.4.	Perbandingan Total Pembangkitan, Pembebanan dan Rugi-rugi Saluran Sebelum dan Setelah Optimasi	74
Grafik 4.5.	Perbandingan Rugi-rugi Daya	76

BAB I

PENDAHULUAN

1.1. Latar Belakang

Sistem tenaga listrik secara garis besar dapat dibagi menjadi 3 bagian yaitu : sisi pembangkit tenaga listrik, jaringan transmisi dan beban. Dalam penyediaan tenaga listrik bagi para pelanggan, tegangan yang konstan merupakan salah satu syarat utama yang harus dipenuhi. Tetapi kondisi tersebut secara praktis sulit dicapai, sehingga kualitas suatu sistem tenaga listrik harus ditingkatkan melalui optimasi daya reaktif. Untuk memenuhi permintaan daya pada setiap saat, dengan kualitas baik dan harga yang murah, maka pada suatu operasi pada beban tertentu perhitungan ekonomis harus menjadi prioritas utama, sehingga diperlukan suatu rencana operasi yang optimum dengan tetap memenuhi beberapa persyaratan pengoperasian sistem tenaga listrik, yaitu 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.

Alokasi daya reaktif yang tepat dalam sistem bisa menghasilkan kondisi optimum, yaitu kondisi dengan rugi-rugi transmisi yang minimum^[1] dengan biaya investasi yang murah dan mampu memperbaiki profil tegangan sistem. Permasalahan besarnya rugi-rugi daya di saluran menjadi perhatian utama khususnya pada saluran transmisi 150 kV.

Masalah-masalah diatas dapat diatasi dan dipecahkan menggunakan metode *Evolutionary Programming*. Metode *Evolutionary Programming* ini akan mengoptimalkan penyaluran daya reaktif secara ekonomis seta mengurangi rugi-rugi daya pada jaringan transmisi 150 kV sub sistem Paiton dan Bali, sehingga dihasilkan kualitas suatu sistem tenaga listrik yang baik.

1.2. Rumusan Masalah

Variasi tegangan yang besar pada bus akibat dari variasi pembebanan daya yang besar menyebabkan perubahan aliran daya pada setiap saluran. Dari perubahan aliran daya yang besar menimbulkan rugi-rugi daya yang besar pula. Mengingat pentingnya kestabilan pelayanan sistem tenaga listrik tersebut, maka dapat timbul permasalahan sebagai berikut:

1. Bagaimana kondisi tegangan tiap bus pada saluran 150 kV sub sistem Paiton dan Bali, apakah masih memenuhi syarat minimum dan maksimum yang telah ditentukan PLN.
2. Bagaimana mengoptimalkan penyaluran daya reaktif secara ekonomis dan mengurangi rugi-rugi daya yang terjadi pada 150 kV sub sistem Paiton dan Bali dengan menggunakan metode *Evolutionary Programming*.

1.3. Tujuan

Tujuan dari penelitian skripsi ini adalah untuk mengoptimalkan penyaluran daya reaktif secara ekonomis dengan tetap memperhatikan profil tegangan serta mengurangi rugi-rugi daya yang terjadi pada saluran 150 kV sub sistem Paiton dan Bali dengan menggunakan metode *Evolutionary Programming*.

1.4. Batasan Masalah

Dalam skripsi ini akan dilakukan analisis optimasi daya reaktif secara ekonomis menggunakan metode EP pada saluran 150 kV sub sistem Paiton dan Bali. Agar pembahasan mengarah sesuai tujuan, maka pembahasan dalam skripsi ini dibatasi oleh hal sebagai berikut :

1. Analisis perhitungan optimasi daya reaktif pada saluran 150 kV sub sistem Paiton dan Bali menggunakan metode *Evolutionary Programming*.
2. Analisa dilakukan dengan asumsi bahwa sistem berada dalam operasi normal.
3. Analisa perhitungan dilakukan pada pembangkit termal.
4. Tidak membahas pengaruh kontrol tegangan.
5. Tidak membahas masalah peralatan kompensasi dan penempatannya.

1.5. Metodologi Penelitian

Metode yang digunakan dalam pembahasan skripsi ini dilakukan dengan langkah – langkah sebagai berikut :

1. Studi Literatur

Yaitu kajian pustaka yang mempelajari teori – teori yang terkait melalui literatur yang telah ada, yang berhubungan dengan permasalahan.

2. Pengumpulan data lapangan yang dipakai dalam objek penelitian yakni data impedansi saluran transmisi dan data pembebanan pada saluran 150 kV sub sistem Paiton dan Bali serta data pembangkit thermal.

3. Simulasi dan pembahasan masalah

Analisa perhitungan tegangan, sudut fasa tegangan, aliran daya pada tiap saluran, biaya pembangkitan dan biaya instalasi dengan menggunakan metode

Evolutionary Programming yang disimulasikan dengan komputer dengan bahasa pemrograman Borland Delphi versi 7.0.

1.6. Sistematika Penulisan

Adapun sistematika pembahasan pada skripsi ini adalah sebagai berikut :

- BAB I : Menguraikan latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, metode penulisan, sistematika penulisan dan kontribusi penelitian.
- BAB II : SISTEM TENAGA LISTRIK, OPERASI EKONOMIS, DAN KARAKTERISTIK UNIT PEMBANGKIT
Menguraikan pembahasan sistem tenaga listrik dan *economic dispatch* secara umum, dan *reactive power dispatch*
- BAB III : OPTIMASI DAYA REAKTIF MENGGUNAKAN METODE *EVOLUTIONARY PROGRAMMING*
Menguraikan teori dasar dari aliran daya, metode aliran daya *Newton Raphson*, teori EP, dan adaptasi EP ke permasalahan optimasi daya reaktif secara ekonomis.
- BAB IV : HASIL DAN ANALISIS HASIL
Menguraikan alur program, hasil validasi, serta hasil perhitungan optimasi daya reaktif menggunakan metode EP.
- BAB V : KESIMPULAN DAN SARAN
Memuat intisari dan hasil pembahasan, yang berisikan kesimpulan dan saran yang dapat digunakan sebagai pertimbangan untuk pengembangan selanjutnya.

1.6. Kontribusi Penelitian

Adapun kontribusi dari skripsi ini adalah diharapkan penggunaan dari metode *Evolutionary Programming* dalam mengalisa penyaluran daya reaktif sehingga dapat menekan rugi-rugi daya dengan biaya pembangkitan yang murah dalam sistem dapat diterapkan oleh pemerintah khususnya PT. PLN (Persero), sehingga akan bisa menambah keuntungan bagi PLN sebagai perusahaan penyedia energi listrik di Indonesia.

BAB II
SISTEM TENAGA LISTRIK, OPERASI EKONOMIS DAN
KARAKTERISTIK UNIT PEMBANGKIT

2.1. Sistem Tenaga Listrik^[1]

Sistem tenaga listrik ada tiga bagian utama yaitu : pusat pembangkit tenaga listrik, saluran transmisi serta sistem distribusi yang berhubungan langsung dengan konsumen. Saluran transmisi merupakan penghubung antara pusat pembangkit melalui hubungan antar sistem yang menuju sistem pada sistem yang lain.

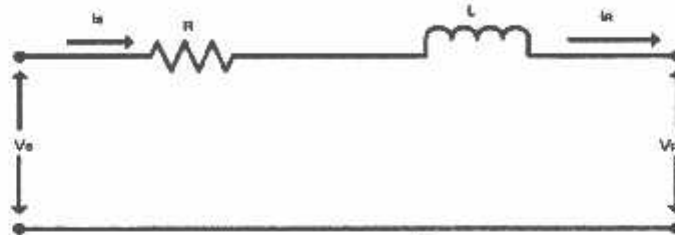
Saluran transmisi mempunyai empat parameter yang mempengaruhi kemampuannya dalam menyalurkan daya listrik. Keempat parameter tersebut yaitu : resistansi (R), induktansi (L), kapasitansi (C), serta konduktansi (G).

Resistansi umumnya tergantung pada jenis penghantar sedangkan konduktansi menyatakan besarnya arus bocor antar penghantar, antar penghantar dengan tanah, tetapi harganya relatif kecil maka dapat diabaikan. Induktansi adalah parameter rangkaian yang menghubungkan tegangan yang diimbaskan oleh perubahan fluksi akibat perubahan arus, sedangkan kapasitansi suatu saluran transmisi timbul akibat adanya beda potensial antara penghantar dengan tanah, dalam hal ini kapasitansi menyebabkan penghantar bermuatan seperti yang terjadi pada pelat kapasitor.

Impedansi seri terbentuk dari resistansi dan induktansi yang terbagi merata sepanjang saluran. Sedangkan konduktansi dan kapasitansi terdapat diantara

2.2.1. Saluran Transmisi Pendek^[1]

Rangkaian ekuivalen untuk saluran transmisi pendek diperlihatkan pada gambar 2, dimana I_S dan I_R merupakan arus pada ujung pengirim dan ujung penerima, sedangkan V_S dan V_R adalah tegangan saluran terhadap netral pada ujung pengirim dan ujung penerima.



Gambar 2.2.^[1]
Rangkaian Setara Saluran Transmisi Pendek

Karena tidak ada cabang paralel (shunt), arus pada ujung – ujung pengirim dan penerima akan sama besar :

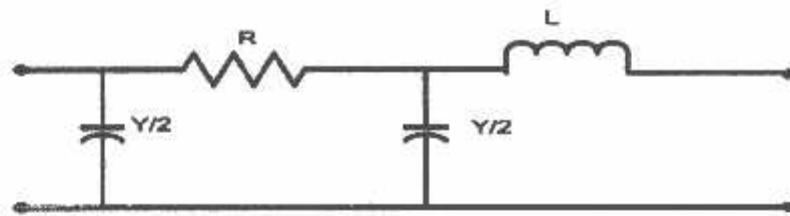
$$I_S = I_R \dots\dots\dots (2.1.)$$

Bila kondisi tegangan pada ujung penerima diketahui, maka tegangan pada ujung sisi kirim adalah :

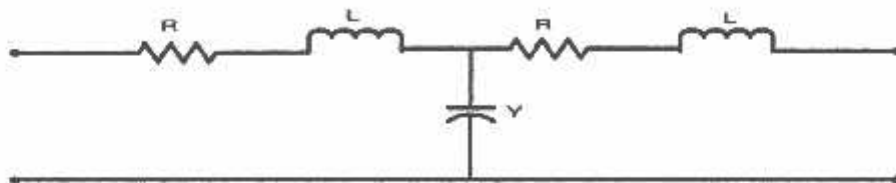
$$V_S = V_R + I_R Z \dots\dots\dots (2.2.)$$

2.2.2. Saluran Transmisi Menengah^[1]

Bertambahnya saluran menyebabkan kapasitansi shunt bertambah besar dan tidak dapat diabaikan. Saluran transmisi jarak menengah pada umumnya digambarkan dengan rangkaian π atau rangkaian T. Dari dua versi ini rangkaian π lebih umum dipakai dari pada rangkaian T.



(a) Rangkaian Ekivalen π



(b) Rangkaian Ekivalen T

Gambar 2.3.^[1]
Rangkaian Setara Saluran Transmisi Menengah

Untuk rangkaian π berlaku :

$$V_s = \left[\frac{ZY}{2} + 1 \right] V_R + ZI_R \dots\dots\dots(2.3.)$$

$$I_s = \left[\frac{ZY}{4} + 1 \right] YV_R + \left[\frac{ZY}{2} + 1 \right] I_R \dots\dots\dots(2.4.)$$

Untuk rangkaian T berlaku :

$$V_s = \left[\frac{ZY}{2} + 1 \right] V_R + \left[\frac{ZY}{4} + 1 \right] ZI_R \dots\dots\dots(2.5.)$$

$$I_s = YV_R + \left[\frac{ZY}{2} + 1 \right] I_R \dots\dots\dots(2.6.)$$

Dimana :

V_s, I_s = Tegangan, Arus sisi kirim

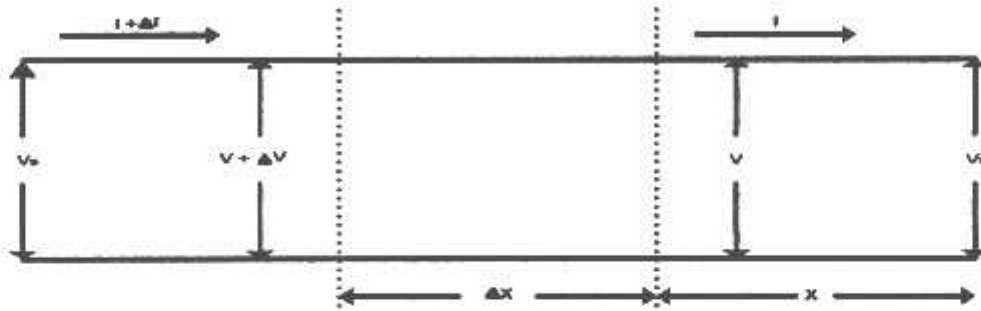
V_R, I_R = Tegangan, Arus sisi terima

Z = Impedansi seri total saluran transmisi

Y = Admitansi shunt total saluran

2.2.3. Saluran Transmisi Panjang^[1]

Pada saluran panjang parameter-parameter saluran tidak terpusat menjadi satu, melainkan tersebar merata diseluruh panjang saluran.



Gambar 2.4.^[1]
Diagram Skema Saluran Transmisi Panjang

Persamaan tegang dan arus pada setiap titik sepanjang saluran transmisi dengan jarak x dari ujung sisi terima dapat ditulis sebagai berikut :

$$V = \frac{V_R + I_R Z_c}{2} e^{yx} + \frac{V_R - I_R Z_c}{2} e^{-yx} \dots\dots\dots(2.7.)$$

$$I = \frac{V_R}{Z_c} + I_R e^{yx} + \frac{V_R}{Z_c} - I_R e^{-yx} \dots\dots\dots(2.8.)$$

Persamaan untuk saluran transmisi panjang dapat ditulis dalam bentuk hiperbola sebagai berikut :

$$V_{(x)} = V_s \cosh \tau x + I_s Z_c \sinh \tau x \dots\dots\dots(2.9.)$$

$$I_{(x)} = I_s \cosh \tau x + \frac{V_s}{Z_c} \sinh \tau x \dots\dots\dots(2.10.)$$

Dimana :

$$\tau = \text{Konstanta rambatan pada saluran} = \sqrt{ZY}$$

$$Z_c = \text{Impedansi karakteristik saluran} = \sqrt{\frac{Z}{Y}}$$

2.3. Daya Dalam Sistem Tenaga Listrik^[1]

Dalam sistem tenaga listrik, pembangkit-pembangkit tenaga listrik harus mampu menyediakan tenaga listrik kepada pelanggan sesuai dengan permintaan beban listrik yang ada. Dan hal yang harus diperhatikan adalah sistem yang tetap konstan, dalam hal ini tegangan dan frekuensi harus tetap konstan karena berhubungan dengan daya. Daya listrik yang dibangkitkan dikenal dengan istilah:

2.3.1 Daya aktif (*Active power*)

Secara umum daya aktif dinyatakan oleh persamaan :

$$P = [V][I] \cos \varphi \dots\dots\dots (2.11.)$$

dimana : V dan I nilai efektifnya.

P adalah daya rata – rata yang disebut juga daya aktif.

2.3.2 ,Daya Reaktif (*Reactive power*)

Daya reaktif adalah daya yang timbul karena adanya pembentukan medan magnet pada beban-beban induktif (KVAR).

Persamaan daya reaktif adalah :

$$Q = [V][I] \sin \varphi \dots\dots\dots (2.12.)$$

Kita mengenal dua macam daya reaktif yaitu : daya reaktif induktif dan daya reaktif kapasitif, dimana keduanya memiliki tanda yang berlawanan. Daya

reaktif kapasitif adalah daya yang dibutuhkan oleh kapasitor yang tidak menghasilkan kerja, tetapi tersimpan dalam bentuk energi magnetis atau energi kapasitif.

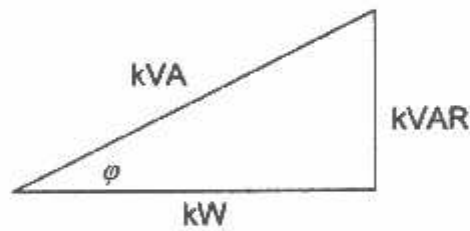
Daya reaktif induktif adalah daya listrik yang dibutuhkan untuk menghasilkan medan magnet yang dibutuhkan oleh alat-alat seperti motor induksi, transformator dan sebagainya.

2.3.3. Daya Semu (*Apparent power*)

Daya semu merupakan penjumlahan secara vektoris antara daya aktif dan daya reaktif.

Selain diatas dikenal juga istilah faktor daya / *power factor* adalah perbandingan antara daya aktif dan daya semu, sehingga dapat dirumuskan pada persamaan berikut :

$$\cos \varphi = \frac{P}{S} = \frac{P}{\sqrt{P^2 + Q^2}} = \frac{kW}{kVA} \dots\dots\dots (2.13.)$$



Gambar 2.5.^[1]
Representasi Segitiga Daya.

Dimana :

$\cos \varphi$ = Faktor daya

KW = Daya aktif

KVA = Daya semu

KVAR = Daya reaktif

2.4. Sistem Per-Unit⁽¹⁾

Untuk memudahkan proses perhitungan, dalam sistem tenaga listrik digunakan sistem Per-Unit (pu).

$$\text{Besaran Per-Unit} = \frac{\text{Besaran sebenarnya}}{\text{Besaran dasar}} \dots\dots\dots (2.14.)$$

Rumus-rumus yang digunakan untuk penentuan arus dasar dan impedansi dasar adalah :

➤ Untuk data 1 fasa

Arus dasar

$$I_d = \frac{\text{kVA dasar 1 fasa}}{\text{kV dasar L - N}} \dots\dots\dots (2.15.)$$

Impedansi dasar

$$Z_d = \frac{(\text{kV dasar L - N})^2 \times 10^{-3}}{\text{kVA dasar 1 fasa}} \dots\dots\dots (2.16.)$$

$$Z_d = \frac{(\text{kV dasar L - N})^2}{\text{MVA dasar 1 fasa}} \dots\dots\dots (2.17.)$$

➤ Untuk data 3 fasa

Arus dasar

$$I_d = \frac{\text{kVA dasar 3 fasa}}{\sqrt{3} \text{ kV dasar L - L}} \dots\dots\dots (2.18.)$$

Impedansi dasar

$$Z_d = \frac{(\text{kV dasar L - L})^2 \times 10^{-3}}{\text{kVA dasar 3 fasa}} \dots\dots\dots (2.19.)$$

$$Z_d = \frac{(\text{kV dasar L - L})^2}{\text{MVA dasar 3 fasa}} \dots\dots\dots (2.20.)$$

Dalam persamaan di atas nilai-nilai besaran diberikan untuk rangkaian satu fasa. Jadi tegangannya adalah tegangan antara fasa dengan tanah dan daya setiap fasa.

Setelah besaran-besaran dasar telah ditentukan maka besaran-besaran itu dinormalisasikan terhadap besaran dasar. Dengan demikian impedansi per satuan dari suatu elemen rangkaian didefinisikan sebagai berikut :

$$Z = \frac{\text{Impedansi sebenarnya } Z(\Omega)}{\text{Impedansi dasar } Z_d} \dots\dots\dots (2.21.)$$

2.4.1. Mengubah Dasar Sistem Per-Unit^[1]

Kadang-kadang impedansi per-unit untuk suatu komponen dari suatu sistem dinyatakan menurut dasar yang berbeda dengan dasar yang dipilih untuk bagian dari sistem dimana komponen tersebut berada. Karena semua impedansi dalam bagian manapun dari suatu sistem harus dinyatakan dengan dasar impedansi yang sama, maka dalam perhitungannya kita perlu mempunyai cara untuk dapat mengubah impedansi per-unit dari suatu dasar ke dasar yang lain. Dengan mensubstitusikan impedansi dasar yang diberikan dalam persamaan (2.17.) dan (2.20.) ke dalam persamaan (2.21.) maka diperoleh:

$$Z_u = \frac{(\text{Impedansi sebenarnya, } \Omega) \cdot (\text{MVA dasar})}{(\text{Tegangan dasar, kV})^2} \dots\dots\dots (2.22.)$$

Persamaan (2.22.) memperlihatkan bahwa impedansi per-unit berbanding lurus dengan MVA dasar serta berbanding terbalik dengan kuadrat tegangan dasar. Untuk mengubah dari impedansi per-unit menurut suatu dasar yang diberikan menjadi impedansi per-unit menurut suatu dasar yang baru, dapat dipakai persamaan berikut:

$$Z_{baru} \text{ per-unit} = Z_{diberikan} \text{ per-unit} \left(\frac{kV_{diberikan} \text{ dasar}}{kV_{baru} \text{ dasar}} \right)^2 \times \left(\frac{MVA_{baru} \text{ dasar}}{MVA_{diberikan} \text{ dasar}} \right) \dots\dots\dots (2.23.)$$

Persamaan ini tidak ada hubungannya dengan pemindahan nilai impedansi dalam ohm dari salah satu sisi transformator ke sisi yang lain.

2.5. Sistem Operasi Pada Sistem Tenaga Listrik^[2]

Seperti telah 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 kerja yang harus dilaksanakan untuk menjamin keandalan sistem operasi antara lain, pengaturan frekuensi dan tegangan sistem untuk berada pada harga normalnya karena adanya perubahan beban pada 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 tepat 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 biasa disebut dengan lengkung beban harian. Lengkung beban harian ini adalah merupakan sesuatu yang sangat penting disamping karakteristik-karakteristik lainnya sehingga dalam operasi hariannya harus berdasarkan lengkung beban harian yang telah dibuat karena dengan lengkung beban harian ini dapat ditentukan perencanaan operasi pembangkit-pembangkit yang ada, baik itu unit

pembangkit thermal maupun 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 berapa 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 beban ditanggung oleh unit pembangkit yang lain yang belum mencapai efisiensi maksimumnya. Dengan dasar operasi yang demikian maka dapat dicapai keadaan operasi yang cukup ekonomis.

Akan tetapi dengan semakin berkembangnya sistem itu sendiri maka diperlukan suatu perencanaan pembangkitan yang optimum dengan biaya operasi yang ekonomis dan harus memperhitungkan rugi-rugi yang terjadi pada saluran transmisi. Mengingat bahwa beban sistem adalah selalu berubah-ubah dari waktu ke waktu maka perlu untuk membuat secara grafis perubahan beban terhadap waktu.

Oleh karena biaya operasi untuk memproduksi daya listrik, suatu pembangkit hidro (PLTA) sangat kecil jika dibandingkan dengan pembangkit thermal (PLTU, PLTG, PLTGU, PLTD) maka pembahasan selanjutnya untuk mendapatkan biaya operasi yang ekonomis sebagian besar ditekankan pada unit

pembangkit thermal saja karena disini 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 akan 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 dari 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 unit pembangkit. Adapun karakteristik yang perlu diketahui dari setiap unit pembangkit adalah :

1. Karakteristik *input* bahan bakar sebagai fungsi dari *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 juga 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 produksi listrik agar harga dari listrik yang dihasilkan dapat ditekan serendah mungkin sehingga dapat memuaskan pemakai listrik.

2.6. Karakteristik Unit Pembangkit^[2]

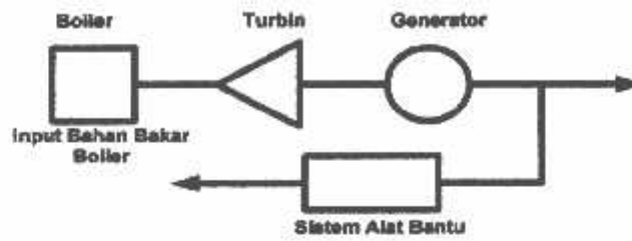
2.6.1. Karakteristik *Input-Output*^[2]

Hal yang paling mendasar dalam operasi pembangkitan yang ekonomis adalah dengan membuat karakteristik *input-output* dari unit pembangkit thermal. Karena ini diperoleh dari desain perencanaan atau melalui test pembangkit. Adapun definisi dari karakteristik *input-output* dari pembangkit itu sendiri adalah formula yang menyatakan hubungan antara *input* pembangkit sebagai fungsi dari *output* pembangkit. Sedangkan ciri dari unit boiler-turbin-generator dapat digambarkan dalam gambar 2.6., 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 Kalori/jam dari bahan bakar yang diberikan boiler untuk menghasilkan *output* pembangkit. Sedangkan notasi yang digunakan adalah H (MBtu/h) atau dalam satuan yang lain H (MKal/h). Adapaun dalam skripsi ini, perhitungan dilakukan adalah dalam satuan MKal/jam. 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 (Rp/h). Hubungan antara H dan F dapat dinyatakan dalam rumus sebagai berikut ini :

$$F = H \times \frac{\text{Rupiah}}{\text{MBtu}} \dots\dots\dots (2.24.)$$

Dimana $\frac{\text{Rupiah}}{\text{MBtu}}$ adalah nilai uang yang diperlukan per satuan panas dari bahan bakar.



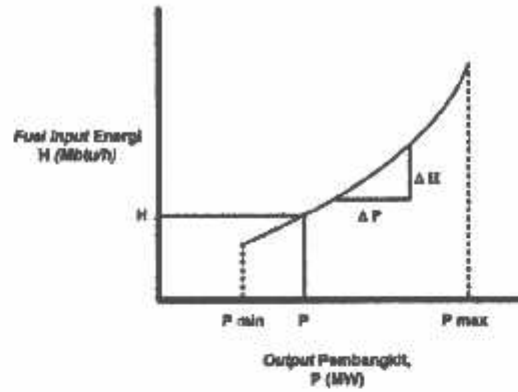
Gambar 2.6.^[2]
Unit Boiler-Turbin-Generator

Seperti digambarkan dalam gambar 2.6., maka *output* dari pembangkit tidak hanya dihubungkan dengan sistem saja akan tetapi juga untuk sistem peralatan bantu pembangkit didefinisikan sebagai daya yang dikeluarkan oleh generator karakteristik *input-output*, daya *output* adalah berupa daya netral dari pembangkit, notasi yang digunakan adalah P (MW).

Persamaan karakteristik *input-output* pembangkit dapat dilihat pada persamaan (2.25.) dan (2.26.) dibawah ini, sedangkan kurva dari karakteristik *input-output* pembangkit dapat dilihat pada gambar 2.7.

$$H = f(P), \text{ atau } \dots\dots\dots(2.25.)$$

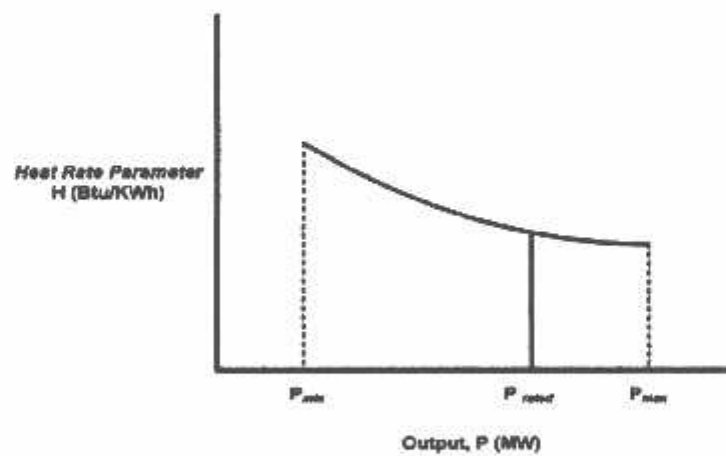
$$F = f(P) \dots\dots\dots(2.26.)$$



Gambar 2.7.^[2]
Kurva Karakteristik *Input-Output* Pembangkit Thermal

2.6.2. Karakteristik *Heat-Rate*^[2]

Karakteristik *heat-rate* merupakan karakteristik yang menunjukkan efisiensi dari sebuah mesin. Karakteristik *heat-rate* sebuah unit pembangkit menunjukkan *input* kalor yang diberikan untuk menghasilkan energi sebesar 1 kiloWatt jam pada MegaWatt *output* dari suatu unit. Kurva dari karakteristik *heat-rate* ini dapat dilihat pada gambar 2.8. di bawah ini.



Gambar 2.8.^[2]
Kurva Karakteristik *Heat-Rate* Unit Pembangkit

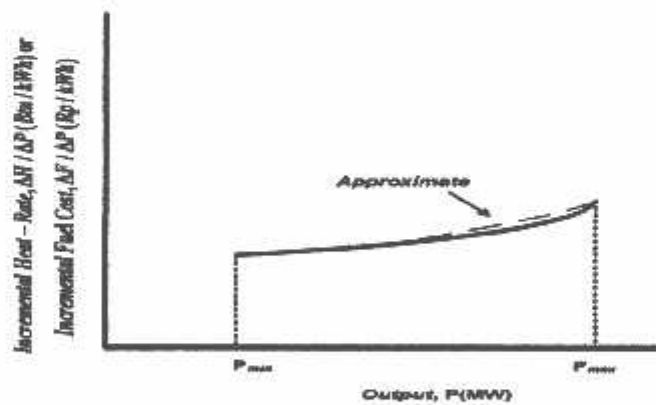
2.6.3 Karakteristik *Incremental Heat-Rate* dan *Incremental Fuel Cost*^[2]

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 bahan bakar. Karakteristik ini menyatakan hubungan daya keluaran pembangkit sebagai fungsi *Incremental Heat-Rate* atau *Incremental Fuel Cost*. Karakteristik *Incremental Heat-Rate* ini menunjukkan besarnya perubahan *input* energi bila ada perubahan *output* pada unit pembangkit.

Kurva dari karakteristik *Incremental Heat-Rate* atau *Incremental Fuel Cost* dapat dilihat pada gambar 2.9. Sedangkan persamaan *Incremental Heat-Rate* dan persamaan *Incremental Fuel Cost* dapat dilihat pada persamaan (2.27.) hingga persamaan (2.28.).

$$\text{Incremental Heat-Rate} = \frac{\Delta H}{\Delta P} \left(\frac{\text{MBtu}}{\text{kWh}} \right) \dots\dots\dots (2.27.)$$

$$\text{Incremental Fuel Cost} = \frac{\Delta F}{\Delta P} \left(\frac{\text{Rupiah}}{\text{kWh}} \right) \dots\dots\dots (2.28.)$$



Gambar 2.9.^[2]
Kurva Karakteristik *Incremental Heat-Rate/Fuel Cost*

Bila harga Δ sangat kecil maka dapat dinyatakan dengan persamaan berikut ini :

$$\text{Incremental Heat-Rate} = \frac{dH}{dP} \left(\frac{\text{MBtu}}{\text{kWh}} \right) \dots\dots\dots(2.29.)$$

$$\text{Incremental Fuel Cost} = \frac{dF}{dP} \left(\frac{\text{Rupiah}}{\text{kWh}} \right) \dots\dots\dots(2.30.)$$

2.7 Economic Dispatch^[2]

Dalam pembahasan tentang OPF dan operasi pada sistem tenaga listrik yang ekonomis, maka kita selalu membicarakan *economic dispatch*. *Economic dispatch* adalah pembagian pembebanan pada pembangkit-pembangkit yang ada dalam suatu sistem tenaga listrik, secara optimum dan ekonomis pada beban tertentu. Dengan dilakukan *economic dispatch* maka akan didapatkan biaya bahan bakar 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 pula.

2.7.1 Fungsi Biaya Bahan Bakar^[2]

Persamaan dari biaya bahan bakar dari setiap unit generator merupakan fungsi dari daya aktif saja, sehingga dengan demikian biaya operasi tiap pembangkit hanya tergantung dari daya aktif saja. Hal ini bisa dilihat dari persamaan di bawah ini:

$$C_F = \sum_{i \in N_g} C_i(P_i) \dots\dots\dots(2.31.)$$

$$C_i(P_i) = a_i + b_i P_i + c_i P_i^2 \dots\dots\dots(2.32.)$$

Dimana:

- $C_i(P_i)$ = Biaya pembangkitan pada unit generator i
- P_i = Besar daya aktif yang terbangkit pada unit generator i
- a, b, c = Koefisien persamaan kuadrat.

Akan tetapi sebenarnya dalam suatu sistem tenaga listrik, keberadaan daya reaktif yang terbangkit juga harus diperhitungkan, sehingga daya reaktif yang terbangkit ikut serta secara tak langsung mempengaruhi biaya operasi. Dengan demikian daya reaktif tidak hanya digunakan untuk perhitungan total rugi pada transmisi, akan tetapi juga harus digunakan untuk masalah optimasi

2.7.2. Fungsi Biaya Daya Reaktif Pembangkit^[3]

Biaya ini merupakan hubungan antara biaya pengoperasian dan besarnya daya reaktif yang terbangkit pada unit generator. Hubungan tersebut akan dijelaskan pada persamaan di bawah ini:

$$C_i(Q_i) = \sum_{i \in Ng} \{c_{qi} |\bar{Q}_{qi}|\} \dots\dots\dots (2.33.)$$

Dimana:

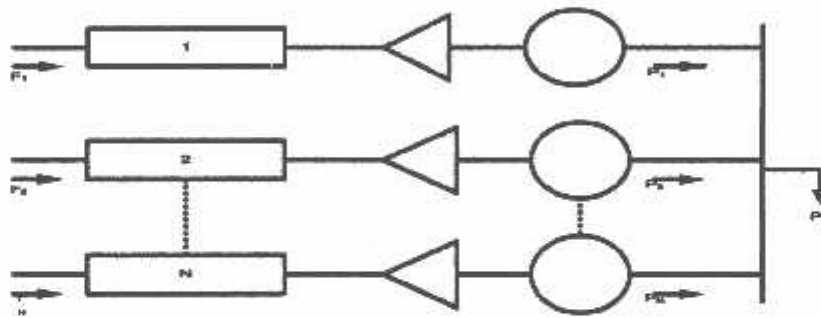
- $C_i(Q_i)$ = Biaya daya reaktif pembangkit
- C_{qi} = Biaya pengoperasian daya reaktif
- \bar{Q}_{qi} = Besar daya reaktif yang terbangkit pada unit generator i

Biaya pengoperasian disini tdak seperti biaya bahan bakar yang merupakan biaya pengoperasian dari pembangkitan daya aktif, tetapi hanya merupakan biaya pengoperasin kecil, seperti biaya pemeliharaan untuk pembangkitan daya reaktif.

2.7.3. *Economic Dispatch Dengan Mengabaikan Rugi-rugi Transmisi*^[2]

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 2.10. Sistem ini terdiri dari N buah pembangkit thermal yang dihubungkan pada *single bus bar* yang melayani beban P_R . *Input* dari masing-masing pembangkit ditunjukkan oleh F_i yang mewakili biaya dari satu unit pembangkit dan *output* dari masing-masing unit P_i adalah daya yang dihasilkan oleh satu unit pembangkit.



Gambar 2.10.^[2]
 N Unit Pembangkit Thermal Melayani Beban P_R

Total biaya rata-rata yang harus ditanggung oleh sistem adalah jumlah biaya dari masing-masing unit pembangkit. Dan pembatas yang paling penting adalah jumlah *output* dari masing-masing unit pembangkit sama dengan beban di konsumen. Yang menjadi permasalahan adalah meminimumkan total biaya F_T dengan memperhatikan pembatas ϕ bahwa daya yang dihasilkan oleh pembangkit

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

$$F_T = F_1 + F_2 + F_3 + \dots + F_N \dots\dots\dots (2.34.)$$

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

$$\phi = 0 = P_R - \sum_{i=1}^N P_i \dots\dots\dots (2.35.)$$

Persamaan di atas adalah pembatas yang merupakan problem dari optimasi dan ini dapat dipecahkan dengan menggunakan kalkulus tingkat lanjut yang melibatkan fungsi La Grange. Dimana fungsi ini didapat dengan cara menambahkan pembatas ϕ yang telah dikalikan dengan faktor pengali La Grange λ pada fungsi tujuan F_T . Fungsi La Grange dapat ditunjukkan dengan persamaan di bawah ini :

$$L = F_T + \lambda \cdot \phi \dots\dots\dots (2.36.)$$

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

$$\nabla L = 0 \dots\dots\dots (2.37.)$$

$$\nabla F_T + \lambda \cdot \phi = 0 \dots\dots\dots (2.38.)$$

$$\frac{\partial L}{\partial P} = \frac{\partial F_i}{\partial P_i} + \lambda \cdot \left(\frac{\partial P_R}{\partial P_i} - \frac{\partial P_i}{\partial P_i} \right) = 0 \dots\dots\dots (2.39.)$$

$$\text{atau } \frac{\partial F_i}{\partial P_i} + \lambda \cdot (0 - 1) = 0 \dots\dots\dots (2.40.)$$

$$\frac{\partial F_i}{\partial P_i} = \lambda \dots\dots\dots (2.41.)$$

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

$$\frac{\partial F_i}{\partial P_i} = \lambda \quad \text{ada N buah persamaan} \dots\dots\dots (2.42.)$$

$$P_{i \text{ min}} \leq P_i \leq P_{i \text{ maks}} \quad \text{ada 2 N buah pertidaksamaan} \dots\dots\dots (2.43.)$$

$$\sum_{i=1}^N P_i = P_R \dots\dots\dots (2.44.)$$

Dari batasan pertidaksamaan pembatas di atas dapat diperluas menjadi :

$$\frac{\partial F_i}{\partial P_i} = \lambda \quad \text{untuk } P_{i \text{ min}} \leq P_i \leq P_{i \text{ maks}} \dots\dots\dots (2.45.)$$

$$\frac{\partial F_i}{\partial P_i} \leq \lambda \quad \text{untuk } P_i = P_{i \text{ maks}} \dots\dots\dots 2.46.)$$

$$\frac{\partial F_i}{\partial P_i} \geq \lambda \quad \text{untuk } P_i = P_{i \text{ min}} \dots\dots\dots (2.47.)$$

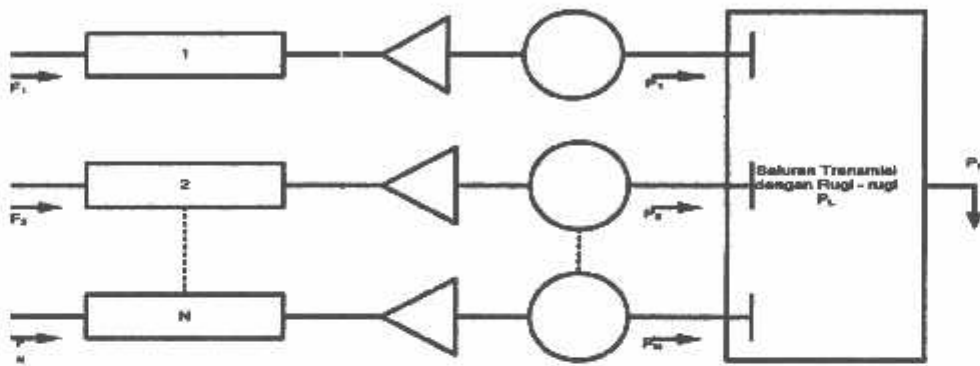
Karena F_i hanya sebagai fungsi P_i maka $\frac{\partial F_i}{\partial P}$ dapat diganti dengan $\frac{dF_i}{dP_i}$.

2.7.4. Economic Dispatch Dengan Memperhitungkan Rugi-Rugi Transmisi^[2]

Sistem dengan memperhitungkan rugi-rugi transmisi dapat dilihat pada gambar 2.11. Sistem ini terdiri dari N buah unit pembangkit thermal dihubungkan

melalui saluran transmisi yang melayani beban P_R . *Input* dari masing-masing unit ditunjukkan oleh F_i yang mewakili biaya dari satu unit pembangkit dan *output* dari masing-masing unit P_i adalah daya yang dihasilkan oleh satu unit pembangkit.

Total biaya rata-rata yang harus ditanggung oleh sistem adalah jumlah dari biaya dari masing-masing unit pembangkit. Dan pembatas yang paling penting adalah bahwa jumlah output dari masing-masing unit pembangkit sama dengan beban di konsumen dan rugi-rugi transmisi.



Gambar 2.11.^[2]

N buah Pembangkit Thermal Melayani Beban P_R Melalui Saluran Transmisi

$$P_R + P_L - \sum_{i=1}^N P_i = \phi = 0 \dots\dots\dots (2.48.)$$

$$L = F_T + \lambda \phi \dots\dots\dots (2.49.)$$

$$\phi = \sum_{i=1}^N P_i - P_R - P_L = 0 \dots\dots\dots (2.50.)$$

Persamaan La Grange nya adalah :

$$L = \sum_{i=1}^N F_i - \lambda \left(\sum_{i=1}^N P_i - P_R - P_L \right) \dots\dots\dots (2.51.)$$

$$\frac{\partial L}{\partial P_i} = \frac{dF_i}{dP_i} - \lambda \left(1 - \frac{\partial P_L}{\partial P_i} \right) = 0 \dots\dots\dots (2.52.)$$

BAB III
OPTIMASI DAYA REAKTIF MENGGUNAKAN METODE
EVOLUTIONARY PROGRAMMING

3.1. Analisa Aliran Daya^{[1][4]}

Dalam melayani beban yang dibutuhkan oleh konsumen dan pengoperasian tenaga listrik perlu dilakukan penganalisaan aliran daya, sehingga system yang dioperasikan dapat memenuhi persyaratan teknis maupun ekonomisnya. Dalam analisa aliran daya dilakukan perhitungan terhadap tegangan, arus, daya aktif dan reaktif, yang terdapat dalam berbagai titik dalam jala-jala jaringan transmisi tenaga listrik.

Tujuan dari analisa daya adalah:

1. Mencari harga magnitude tegangan $|v|$ dan sudut fasa tegangan δ bus beban.
2. Mencari daya reaktif Q dan sudut fasa tegangan δ dari generator bus.
3. Untuk mendapatkan daya aktif dan daya reaktif pada bus slack.
4. Untuk mengetahui apakah semua peralatan pada sistem memenuhi batas – batas yang telah ditetapkan untuk operasi penyaluran daya.
5. Untuk mengetahui kondisi awal pada perencanaan sistem yang baru.
6. Untuk menentukan daya yang mengalir disetiap saluran jaringan tenaga listrik.

3.1.1. Klasifikasi Bus^{[1][4]}

Pada setiap bus dari jaringan terdapat parameter – parameter yaitu : daya aktif (P), daya reaktif (Q), besar tegangan $|v|$ dan sudut fasa tegangan δ .

Dengan melihat parameter diatas, setiap bus dapat diklasifikasikan menjadi 3 bagian:

1. Bus Beban (Load Bus) (PQ)

Pada bus ini hanya terdapat kebutuhan daya untuk beban untuk beban dimana P daya aktif dan Q daya reaktif diketahui, sementara $|v|$ dan δ berubah – ubah menurut kebutuhan. Oleh karena itu, $|v|$ dan δ harus ditentukan (dicari).

2. Bus Generator (PV)

Pada bus ini hanya terdapat daya pembangkitan dimana $|v|$ diatur menggunakan regulator tegangan dan P diatur dengan governor. Sehingga untuk bus ini P dan $|v|$ diketahui. Sementara Q (daya reaktif) dan δ (sudut fasa) dicari.

3. Bus Slack

Pada bus ini $|v|$ dan δ sudah ditentukan besarnya sementara P dan Q dihitung. Biasanya nilai $|v|$ adalah 1 pu, sedangkan sudut fasa tegangan δ berharga nol, karena itu fasor tegangan dari bus dipakai sebagai referensi.

Daya total yang mengalir pada setiap bus dituliskan sebagai berikut :

$$S_k = P_k + jQ_k = V_k^* I_k \dots\dots\dots (3.1)$$

Atau

$$P_k + jQ_k = V_k^* \sum_{n=1}^N Y_{kn} V_n \dots\dots\dots (3.2)$$

Dari persamaan $V_k = v_k + j\delta_k$ dan $Y_{k11} = G_{k11} - jB_{k11}$, maka persamaan 3.2. menjadi :

$$P_k + jQ_k (v_k + j\delta_k) \sum_{n=1}^N (G_{k11} - jB_{k11})(V_n + j\delta_n) \dots\dots\dots (3.3)$$

Bila dituliskan dalam bentuk real dan imajiner maka persamaan di atas menjadi :

$$P_k = \sum_{n=1}^N \{v_k (v_n G_{kn} + \delta_n B_{kn}) + \delta_k (\delta_n G_{kn} - v_n B_{kn})\} \dots\dots\dots (3.4)$$

$$Q_k = \sum_{n=1}^N \{\delta_k (v_n G_{kn} + \delta_n B_{kn}) - v_k (\delta_n G_{kn} - v_n B_{kn})\} \dots\dots\dots (3.5)$$

3.1.2. Metode Newton Rapshon^[5]

Proses yang dilakukan adalah membandingkan antara daya yang ditempatkan berdasarkan data ($P_{k, sched}$ dan $Q_{k, sched}$) dengan daya hasil perhitungan ($P_{k, calc}$ dan $Q_{k, calc}$) menggunakan persamaan (3.4.) dan (3.5.) di atas. Selisih daya yang diterapkan dan perhitungan (ΔP_k dan ΔQ_k) dihitung dengan persamaan :

$$\Delta P_k = P_{k, sched} - P_{k, calc} \dots\dots\dots (3.6)$$

$$\Delta Q_k = Q_{k, sched} - Q_{k, calc} \dots\dots\dots (3.7)$$

Selisih daya dihitung dengan persamaan (3.6.) dan persamaan (3.7.) digunakan untuk menghitung nilai perubahan parameter tegangan bus, yaitu $\Delta|V_k|$ dan $\Delta\delta_k$, yaitu dengan menggunakan elemen Jacobian, sehingga koreksi terhadap nilai parameter tegangan yang telah ditetapkan nilai awal sebelumnya. Elemen Jacobian sendiri merupakan turunan parsial P dan Q terhadap masing-masing

variabel pada persamaan (3.4.) dan (3.5.), yang dalam bentuk matriks dituliskan sebagai :

$$\begin{bmatrix} \Delta P_1 \\ \dots \\ \Delta P_{n-1} \\ \Delta Q_1 \\ \dots \\ \Delta Q_{n-1} \end{bmatrix} = \begin{bmatrix} \frac{\partial P_1}{\partial v_1} & \dots & \frac{\partial P_1}{\partial v_{n-1}} & \frac{\partial P_1}{\partial \delta_1} & \dots & \frac{\partial P_1}{\partial \delta_{n-1}} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \frac{\partial P_{n-1}}{\partial v_1} & \dots & \frac{\partial P_{n-1}}{\partial v_{n-1}} & \frac{\partial P_{n-1}}{\partial \delta_1} & \dots & \frac{\partial P_{n-1}}{\partial \delta_{n-1}} \\ \frac{\partial Q_1}{\partial v_1} & \dots & \frac{\partial Q_1}{\partial v_{n-1}} & \frac{\partial Q_1}{\partial \delta_1} & \dots & \frac{\partial Q_1}{\partial \delta_{n-1}} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \frac{\partial Q_{n-1}}{\partial v_1} & \dots & \frac{\partial Q_{n-1}}{\partial v_{n-1}} & \frac{\partial Q_{n-1}}{\partial \delta_1} & \dots & \frac{\partial Q_{n-1}}{\partial \delta_{n-1}} \end{bmatrix} \begin{bmatrix} \Delta v_1 \\ \dots \\ \Delta v_{n-1} \\ \Delta \delta_1 \\ \dots \\ \Delta \delta_{n-1} \end{bmatrix}$$

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} J_1 & J_2 \\ J_3 & J_4 \end{bmatrix} \begin{bmatrix} \Delta \delta \\ \Delta |v| \end{bmatrix} \dots \dots \dots (3.8.)$$

Dimana elemen-elemen jacobian dapat dihitung dengan menurunkan persamaan-persamaan (3.4.) dan (3.5.)

Perubahan nilai tegangan bus dijumlahkan dengan nilai tegangan bus sebelumnya, yang kemudian nilai tegangan bus terbaru ini digunakan untuk menghitung kembali daya $P_{k, calc}$ dan $Q_{k, calc}$ menurut persamaan (3.4.) dan (3.5.) Proses ini terus berulang, yang disebut iterasi hingga mencapai kondisi dimana nilai perubahan daya ΔP dan ΔQ konvergen mencapai suatu nilai minimum yang telah ditentukan (berkisar 0,001 hingga 0,0001 pu)

3.2. *Evolutionary Programming (EP)*^{[6][7][8]}

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

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

EP ditemukan oleh Lawrence.J. Fogel pada tahun 1960 yang dilandasi oleh sifat-sifat evolusi alam. Fogel percaya bahwa ini sangat cocok digabungkan dalam sebuah algoritma komputer, menghasilkan sebuah teknik penyelesaian untuk permasalahan-permasalahan yang sulit dengan langkah alami yaitu melalui evolusi. Fogel mulai bekerja dengan algoritma yang dibentuk dari string-string bilangan real yang disebut *kromosom*.

Sebelum EP dijalankan, maka sebuah kode yang sesuai (representasi) untuk persoalan harus dirancang. Titik solusi dalam ruang permasalahan dikodekan dalam bentuk kromosom/string yang terdiri dari komponen genetik terkecil yaitu gen. Pemakaian bilangan real (*floating point*) sebagai *allele* (nilai gen) memungkinkan penerapan operator EP yaitu inialisasi (*initialization*), statistik (*statistics*), mutasi (*mutation*), dan kompetisi (*competition*) untuk menciptakan himpunan titik-titik solusi. Untuk memeriksa hasil optimasi, kita membutuhkan fungsi *fitness* yang menandakan gambaran hasil (*solution*) yang sudah dikodekan. Selama proses, induk harus digunakan untuk reproduksi, mutasi dan kompetisi untuk menciptakan keturunan (*offspring*).

3.2.1. Parameter *Evolutionary Programming*

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

❖ Jumlah Generasi (MAXGEN)

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

❖ Ukuran Populasi (*POPSIZE*)

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

❖ Probabilitas Mutasi (*Pm*)

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

❖ Panjang Kromosom (*NVAR*)

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

3.2.2. Mekanisme *Evolutionary Programming*

A. Pengkodean

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

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

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

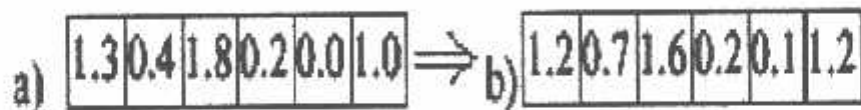
B. Fungsi Evaluasi (*Fungsi Fitness*)

Dalam EP, sebuah fungsi *fitness* $f(x)$ harus dirancang untuk masing-masing permasalahan yang akan diselesaikan. Dengan menggunakan kromosom tertentu, fungsi obyektif atau fungsi evaluasi akan mengevaluasi status masing-masing kromosom. Setiap gen x_i ($i = 1, 2, \dots, N$) dipergunakan untuk menghitung $f_k(x)$ ($k = 1, 2, \dots, \text{POPSIZE}$).

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

C. Mutasi (*Mutation*)^[8]

Operator mutasi digunakan untuk memodifikasi satu atau lebih nilai gen dalam satu individu. Cara kerjanya dengan membangkitkan sebuah nilai random r_k dimana $k = 1, 2, \dots, NVAR$ (panjang kromosom). Probabilitas mutasi (P_m) ditentukan dan digunakan untuk mengendalikan frekuensi operator mutasi. Apabila nilai random r_k, P_m maka gen ke- k kromosom tersebut terpilih untuk mengalami mutasi. Proses mutasi dalam *Evolutionary Programming* sama dengan *Evolutionary Strategies* yaitu menggunakan operator *Gaussian mutation*, dimana setiap individu akan terpilih secara acak untuk mengalami mutasi berdasarkan nomor acak *Gaussian* untuk menciptakan individu baru (*offspring*).



Gambar 3.1.^[10]

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

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

D. Kompetisi (*Competition*)^[5]

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

3.3 Formulasi Masalah Optimasi Daya Reaktif^[3]

Permasalahan perencanaan optimasi daya reaktif adalah untuk meminimalkan biaya operasional, dengan persamaan sebagai berikut :

$$C_Q = C_P + C_i(Q_i) \dots\dots\dots (3.9.)$$

$$C_P = \sum_{l \in N_l} d_l C_F^l \dots\dots\dots (3.10.)$$

Dimana :

$C_i(Q_i)$ = Biaya daya reaktif pembangkitan.

C_F^l = Biaya operasi daya nyata pada tingkatan l beban

d_l = Jangka waktu tingkatan l beban

N_l = Banyaknya tingkatan beban

Dengan subyek :

$$\underline{P}_{g^i} \leq P_{g^i} \leq \overline{P}_{g^i} \quad (\text{Daya nyata generator})$$

$$\underline{Q}_{g^i} \leq Q_{g^i} \leq \overline{Q}_{g^i} \quad (\text{Daya reaktif generator})$$

$$\underline{T}_k \leq T_k \leq \overline{T}_k \quad (\text{Tap transformer})$$

$$\underline{V}_i \leq V_i \leq \overline{V}_i \quad (\text{Tegangan bus beban})$$

Tegangan bus beban dan daya reaktif generator merupakan variabel keadaan, yang dapat dibatasi dengan menambahkannya sebagai bentuk penalti kuadrat ke fungsi obyektif sebagai berikut :

$$C_p = \sum_{i \in N_p} d_i \left\{ C_f^i + \sum_{i \in N_v} \lambda_v (V_i - \text{Sat}(V_i))^2 + \sum_{i \in N_q} \lambda_{g^i} (Q_{g^i} - \text{Sat}(Q_{g^i}))^2 \right\} \dots (3.11.)$$

Dimana; $\text{sat}(x)$ adalah fungsi saturasi sebagai berikut :

$$\text{Sat}(x) = \begin{cases} \underline{x} & \text{if } x < \underline{x} \\ x & \text{if } \underline{x} \leq x \leq \overline{x} \\ \overline{x} & \text{if } x > \overline{x} \end{cases} \dots (3.12)$$

3.4. Adaptasi *Evolutionary Programming* ke Masalah Optimasi Daya Reaktif⁽³¹⁾

Evolutionary Programming adalah suatu mekanisme seleksi alam. Individu dari sebuah populasi dikodekan secara real, populasi pertama dibangkitkan secara random. Generasi baru dibuat dengan mengaplikasikan operator berikut terhadap sebuah populasi yaitu : inialisasi, mutasi dan

kompetisi dimana tingkat pengoptimalan setiap calon solusi atau individu tergantung pada fungsi tujuan (*objective function*) permasalahan.

3.4.1. Representasi Solusi^[3]

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

3.4.2. Inisialisasi (*Initialization*)^[3]

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

3.4.3. Statistik (*Statistics*)^[3]

Fitness maksimum (f_{max}), fitness minimum (f_{min}), Jumlah fitness $\sum f_i$, dan fitness rata-rata (f_{avg}) dari generasi dihitung.

3.4.4. Mutasi (*Mutation*)^[3]

Di dalam proses mutasi, setiap terpilihnya orang tua, misalnya P_i dimutasikan dan ditambahkan ke populasinya, sesuai persamaan berikut :

$$P_{i+m,j} = P_{i,j} + N(0, \beta(\bar{X}_j - X_j) \frac{f_i}{f_{max}}), j = 1, 2, \dots, n \dots \dots \dots (3.13.)$$

Dimana :

n = Banyaknya variabel keputusan di dalam setiap individu.

$P_{i,j}$ = Elemen j^{th} unsur dari I^{th}

$N(\mu, \sigma^2)$ = Variabel acak dengan menggunakan μ dan σ^2

f_{max} = Nilai *fitness* terbesar dari statistik, \bar{x}_j dan \underline{x}_j

β = Skala mutasi, $0 < \beta \leq 1$

3.4.5. Kompetisi (*Competition*)^[3]

Di dalam proses kompetisi, individu dari populasi yang memiliki nilai *fitness* yang terbaik disebut sebagai (k). Kemudian dalam proses selanjutnya untuk mendapatkan individu baru populasi dikombinasikan dengan ukuran $(2m-k)$. Nilai berat W_i dari individu yang ke i dihitung menurut kompetisi berikut ini :

$$W_i = \sum_{i=1}^N W_{i,r} \dots\dots\dots (3.14)$$

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

$$W_{i,r} = \begin{cases} 1 & \text{jika } U_i < \frac{f_r}{f_r + f_i} \\ 0 & \text{jika sebaliknya} \end{cases} \dots\dots\dots (3.15.)$$

Dimana :

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

f_i = *Fitness* dari individu ke i

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

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

3.4.6. Konvergensi (*Convergence*)^[3]

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

$$\{f_{avg} / f_{max}\} \geq \delta \dots\dots\dots (3.16.)$$

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

BAB IV

HASIL DAN ANALISA HASIL

4.1. Program Komputer Optimasi Daya Reaktif Menggunakan Metode *Evolutionary Programming*

Dalam penyelesaian masalah ini maka diperlukan bantuan program komputer dalam perhitungan yang membutuhkan ketelitian dan keakuratan.

Program komputer dalam skripsi ini dijalankan dengan menggunakan bahasa pemrograman Borland Delphi versi 7.0 dan diaplikasikan pada komputer AMD Athlon dengan Prosesor 2 GHz, memori 256 Mb.

4.1.1. Algoritma Program

Algoritma program sebelum optimasi dilakukan dengan langkah-langkah sebagai berikut :

1. Memasukkan inputan data beban yang meliputi tegangan (V), sudut fasa tegangan (δ), daya aktif (P), daya reaktif (Q) dan data impedansi saluran, data pembangkitan P_{maks} , P_{min} , konstanta biaya.
2. Lakukan proses *load flow Newton Raphson*.
3. Hitung total biaya pembangkitan.
4. Berhenti.

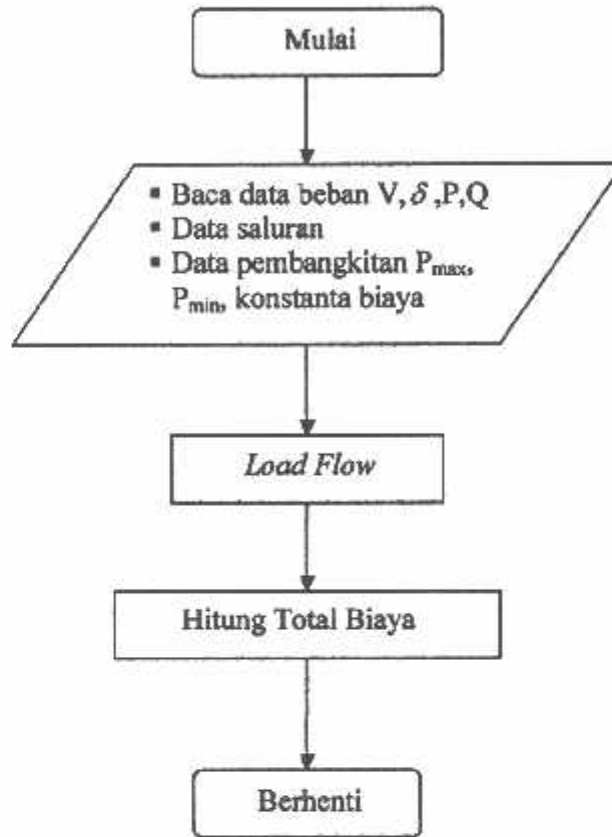
Algoritma Program Perencanaan Optimasi Daya Reaktif dengan menggunakan metode *Evolutionary Programming*

1. Memasukan inputan data beban yang meliputi tegangan (V), sudut fasa tegangan (δ), daya aktif (P), daya reaktif (Q), dan data impedansi saluran, data pembangkitan P_{maks} , P_{min} , konstanta biaya.
2. Menentukan parameter inputan *evolutionary programming* yang meliputi jumlah populasi, maksimum generasi, nilai kemungkinan mutasi, dan panjang kromosom tiap-tiap individu.
3. Generasi = 0, Populasi = 0.
4. Melakukan inisialisasi calon solusi.
5. Menghitung *fitness* dari kromosom tiap-tiap individu.
6. Melakukan proses statistik.
7. Melakukan proses mutasi.
8. Apakah *offspring* yang diinginkan sudah terpenuhi (max pop).
9. Jika “ Tidak “ maka populasi = pop + 1, kembali ke langkah 7.
10. Jika “Ya “ maka lanjutkan ke langkah 11.
11. Menghitung *fitness* dari *offspring*.
12. Melakukan proses kompetisi.
13. Melakukan proses statistik.
14. Apakah generasi yang diinginkan sudah terpenuhi (max Gen).
15. Jika “ Tidak “ maka generasi = gen + 1, kembali ke langkah 7.
16. Jika “ Ya “ maka perhitungan berhenti.

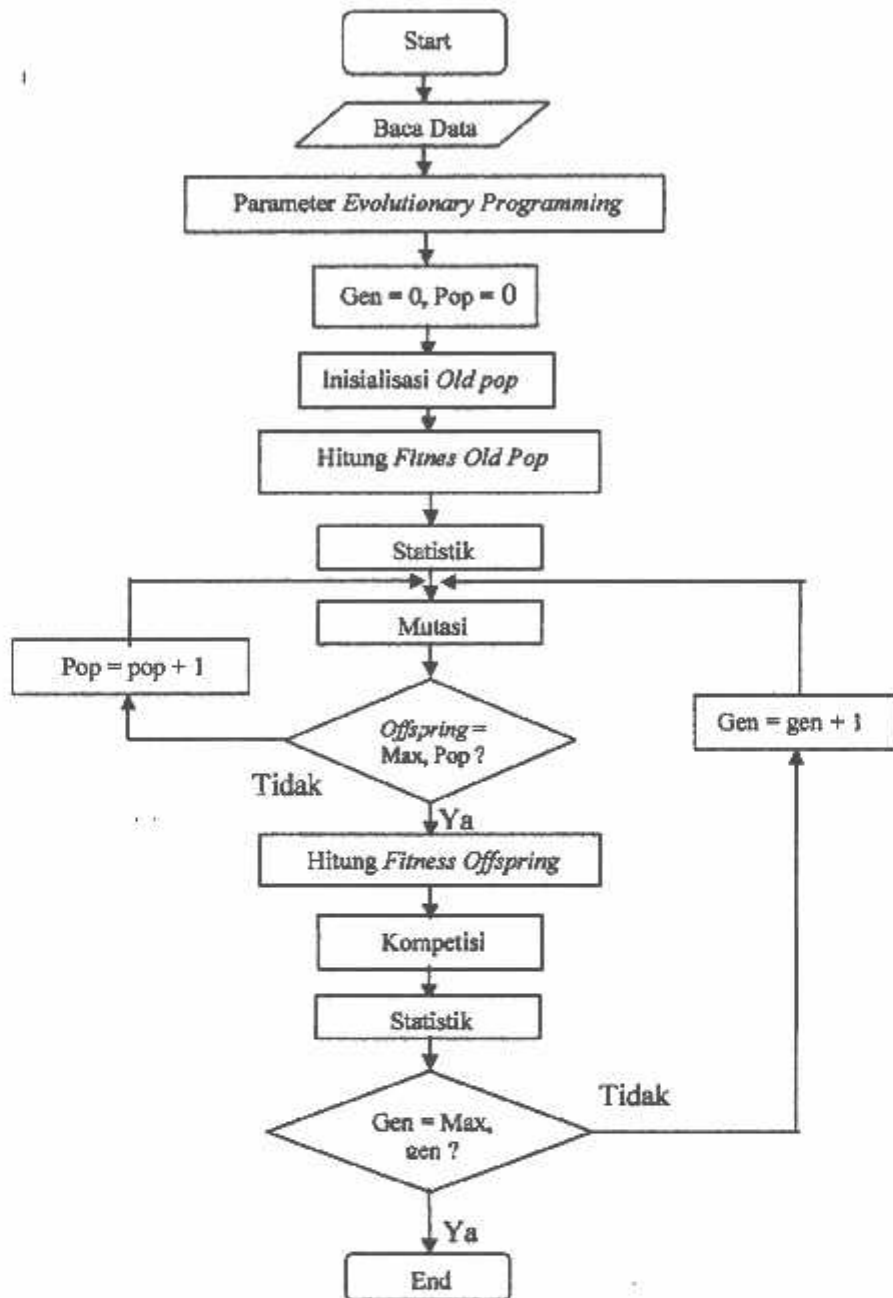
Sedangkan untuk algoritma aliran daya dengan metode *Newton Rapshon* adalah sebagai berikut :

1. Bentuk matriks admitansi bus : $[Y_{bus}]$
2. Penetapan harga awal tegangan dan sudut fasa untuk semua bus kecuali bus slack $[V_i(0), \delta(0)]$
3. Menentukan nomor iterasi awal, $k=0$
4. Menghitung injeksi daya aktif dan reaktif pada setiap bus dengan persamaan kecuali bus slack
5. Menghitung selisih daya yang dijadwalkan dengan injeksi daya bus dari perhitungan
6. Menentukan perubahan maksimum pada daya aktif dan daya reaktif
7. Membandingkan apakah selisih daya sudah sama atau lebih kecil dari ϵ
8. Jika “ya” hitung daya aktif dan reaktif, tegangan dan sudut fasa tegangan pada setiap bus, serta aliran daya pada saluran dan perhitungan selesai, jika “tidak” lanjutkan ke langkah berikutnya
9. Membentuk elemen matriks Jacobian
10. Menghitung faktor koreksi tegangan dan sudut fasa setiap bus kecuali bus slack dan bus generator
11. Menghitung nilai tegangan dan sudut fasa yang baru
12. Mengganti nilai sudut fasa yang lama dengan sudut fasa yang baru, tegangan yang lama dengan tegangan yang baru
13. Perhitungan dilanjutkan ke langkah 4 dengan nilai iterasi yang baru sampai hasil yang didapatkan konvergen.

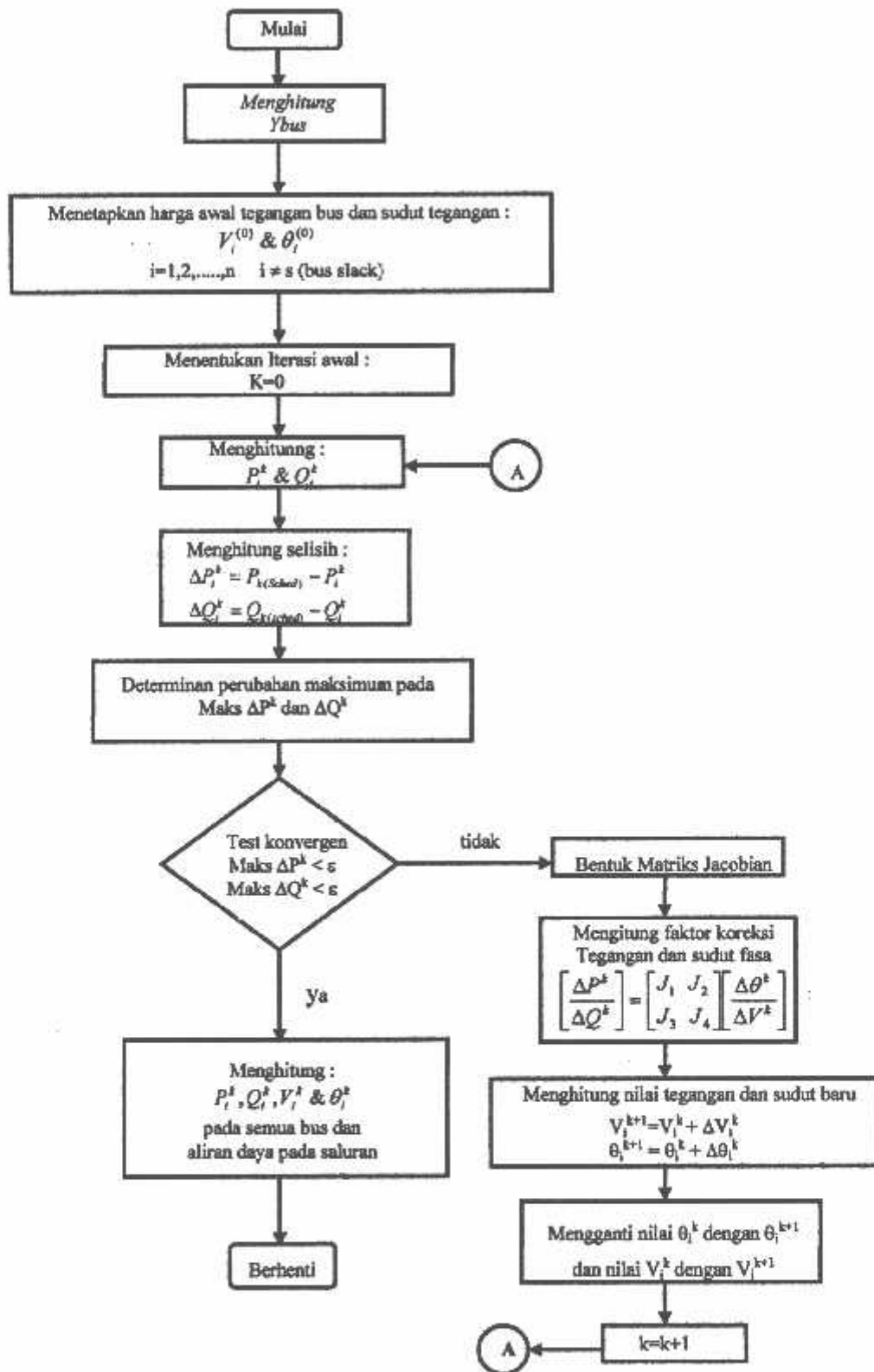
4.1.2. Flowchart Program



Gambar 4.1
Flowchart Perhitungan Sebelum Optimasi



Gambar 4.2
Flowchart Program Menggunakan *Evolutionary Programming*



Gambar 4.3.
Flowchart Perhitungan Aliran Daya Metode Newton Rapshon

22	1	0	0	0	0	0
23	1	0	0	0	3.2	1.6
24	1	0	0	0	8.7	6.7
25	1	0	0	0	0	0
26	1	0	0	0	3.5	2.3
27	1	0	0	0	0	0
28	1	0	0	0	0	0
29	1	0	0	0	0.4	0.9
30	1	0	0	0	10.6	1.9

Tabel 4.3.^[10]
Data Saluran Impedansi IEEE 30

No. Saluran	Hubungan Bus	R (pu)	X (pu)	B (pu)
1	1-2	0.0192	0.0575	0.0264
2	1-3	0.0452	0.1852	0.0204
3	2-4	0.057	0.1737	0.0184
4	3-4	0.0132	0.0379	0.0042
5	3-5	0.0472	0.1983	0.0209
6	2-6	0.0581	0.1763	0.0187
7	4-6	0.0119	0.0414	0.0045
8	5-7	0.046	0.0102	0.0102
9	6-7	0.0267	0.082	0.0085
10	6-8	0.012	0.042	0.0045
11	6-9	0.000	0.208	0.0000
12	6-10	0.000	0.556	0.0000
13	9-11	0.000	0.208	0.0000
14	9-10	0.000	0.11	0.0000
15	4-12	0.000	0.256	0.0000
16	12-13	0.000	0.14	0.0000
17	12-14	0.1231	0.2559	0.0000
18	12-15	0.0602	0.1304	0.0000
19	12-16	0.0945	0.1987	0.0000
20	14-15	0.221	0.1997	0.0000
21	16-17	0.0824	0.1932	0.0000
22	15-18	0.107	0.2185	0.0000
23	18-19	0.0639	0.1292	0.0000
24	19-20	0.034	0.068	0.0000
25	10-20	0.0936	0.209	0.0000
26	10-17	0.0324	0.0845	0.0000
27	10-21	0.0348	0.0749	0.0000
28	10-22	0.0727	0.1499	0.0000
29	21-22	0.0116	0.0236	0.0000
30	15-23	0.1	0.202	0.0000

Tabel 4.5.^[12]
Parameter Unit Pembangkit Thermal

Nama Pembangkit	a2	a1	a0	Pmin (MW)	Pmax (MW)
PLTU Paiton 1&2	276,246	58384,905	17350044,31	150	700
PLTG Gilimanuk	188,355772	214074,12	11705797,8	50	133,8
PLTD Pesanggaran	1913,573616	223579,6	11909964,8	2,1	75
PLTG Pesanggaran	15248,932	709162,36	18684292,8	15	125,5

Dengan memasukkan data-data pada tabel 4.5. ke persamaan fungsi biaya bahan bakar, maka untuk unit pembangkit thermal PLTU Paiton 1&2 diperoleh sebagai berikut :

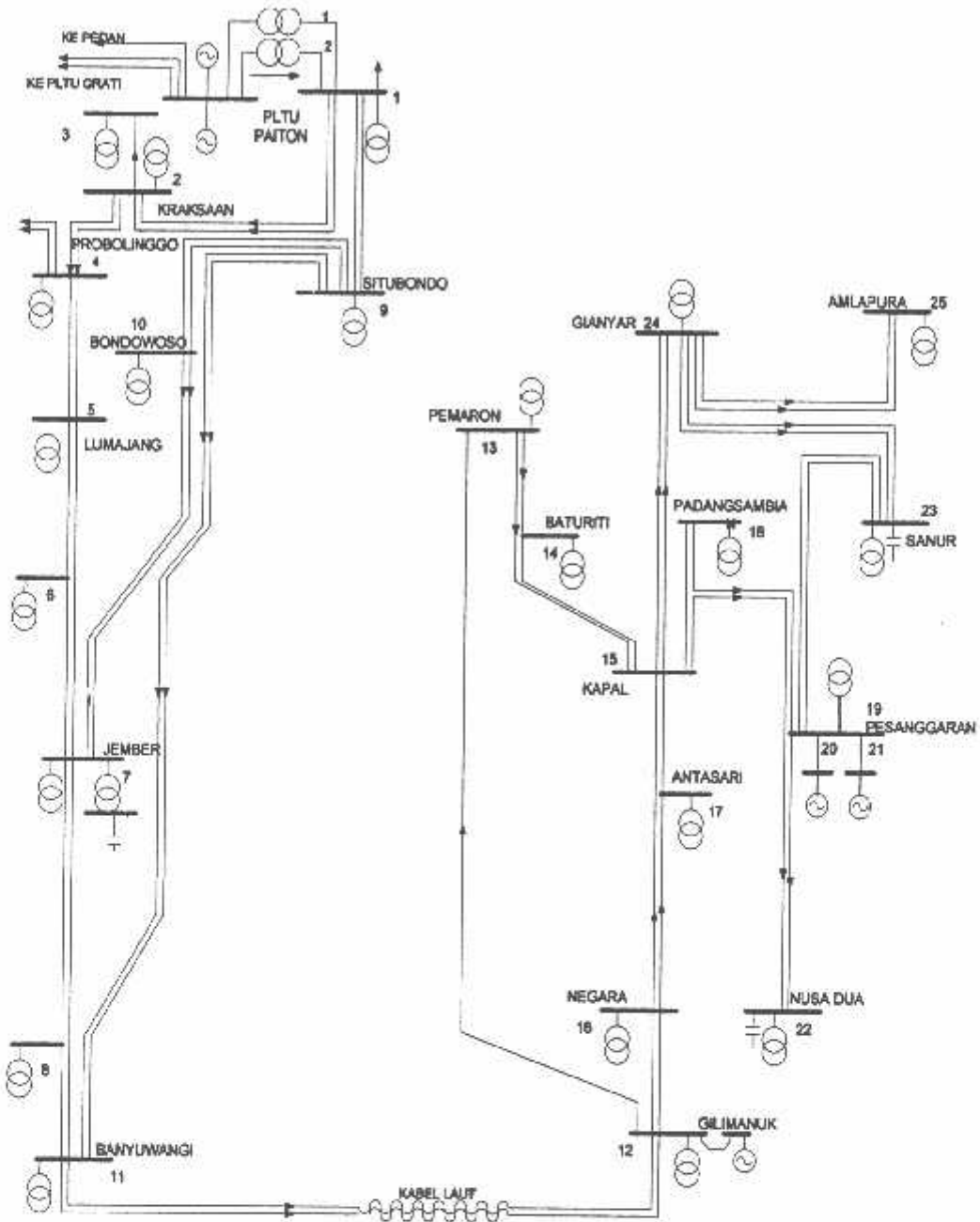
Biaya bahan bakar pembangkit :

$$F_i(G_i) = a_2 P_{Gi}^2 + a_1 P_{Gi} + a_0$$

$$F_1(G_1) = 276,246 P^2 + 58384,905 P + 17350044,31$$

Sedangkan untuk biaya pengoperasian daya reaktif dari setiap pembangkit, dalam skripsi ini berdasarkan referensi maka diset sebesar US \$ 6,77 / Mvar perjam^[15]. Kurs dollar yang dipakai sebesar Rp 9200, jadi biaya pengoperasian daya reaktif dari masing-masing pembangkit sebesar Rp 62284 / MVar perjam.

Untuk persamaan biaya pembangkitan dari masing-masing unit pembangkit yang berdasarkan data-data diatas adalah sebagai berikut :



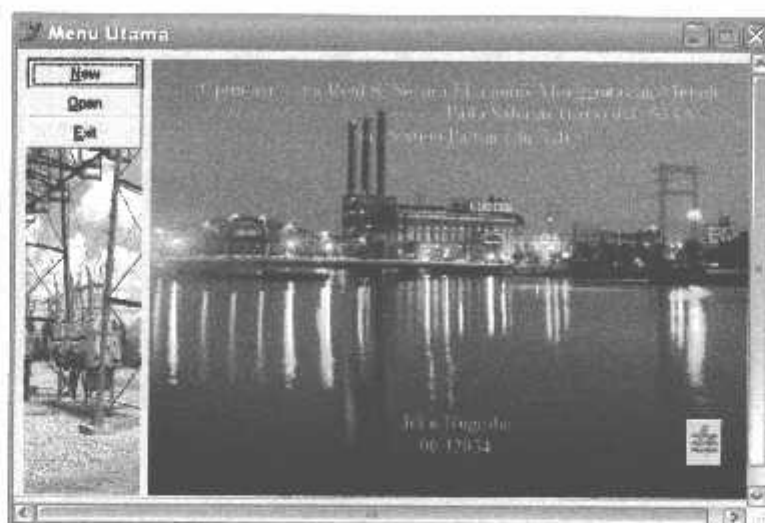
Sumber : PT. PLN (persero) P3B Jawa-Bali

Gambar 4.7.
Diagram Segaris Jaringan Sistem Transmisi Listrik 150 kV Paiton-Bali

4.6. Prosedur Pelaksanaan Program Perhitungan

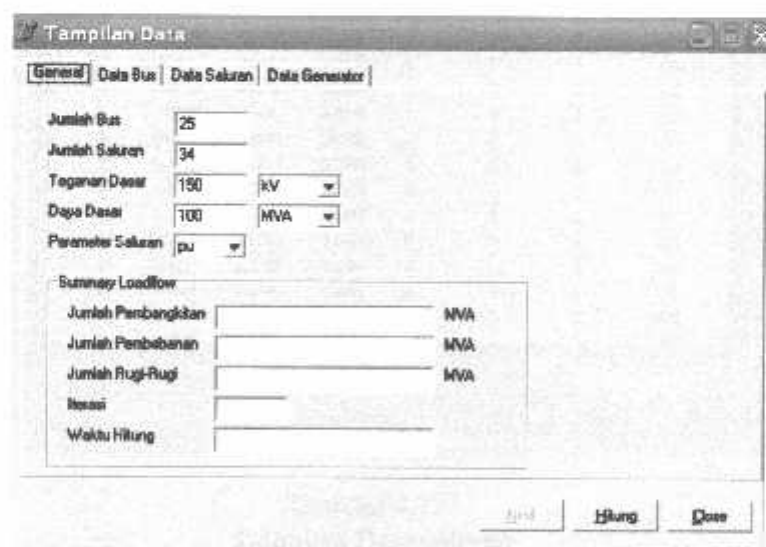
Prosedur menjalankan program perhitungan dengan menggunakan bahasa pemrograman Borland Delphi versi 7.0 dapat dilakukan sebagai berikut :

1. Tampilan utama dari program.



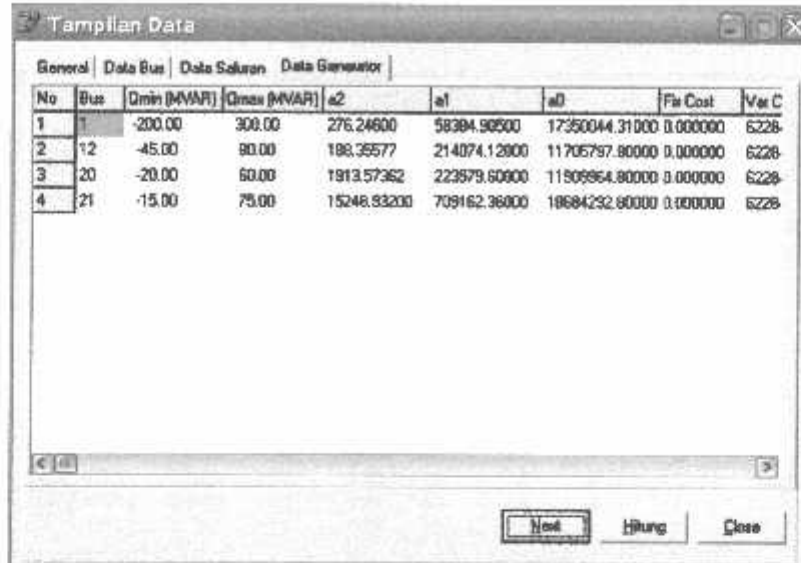
Gambar 4.8.
Tampilan Utama Program

2. Tekan tombol *Open File* untuk membuka data yang sudah tersimpan.



Gambar 4.9.
Tampilan Masukan Data

5. Kemudian tekan data generator



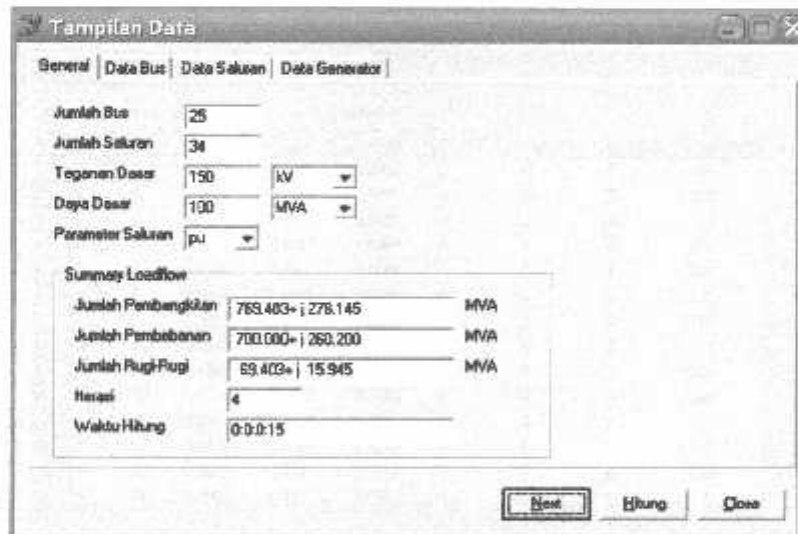
The screenshot shows a window titled 'Tampilan Data' with tabs for 'General', 'Data Bus', 'Data Saluran', and 'Data Generator'. The 'Data Generator' tab is active, displaying a table with the following data:

No	Bus	Qmin (MVAR)	Qmax (MVAR)	a2	a1	a0	Fix Cost	Var C
1	1	-200.00	300.00	276.24600	58394.98500	17350044.31000	0.000000	6228
2	12	-45.00	80.00	188.35577	214074.12000	11705797.80000	0.000000	6228
3	20	-20.00	60.00	1913.57362	223579.50000	11809864.80000	0.000000	6228
4	21	-15.00	75.00	15248.83200	709162.36000	18684292.80000	0.000000	6228

At the bottom of the window, there are buttons for 'Next', 'Hitung', and 'Close'.

Gambar 4.12.
Tampilan Data Generator

6. Tekan tombol hitung untuk melihat hasil perhitungan aliran daya dan *Load Flow* pada kondisi awal (sebelum optimasi)



The screenshot shows the 'Tampilan Data' window with the 'Data Saluran' tab active. It displays various system parameters and a 'Summary Loadflow' section. The parameters are:

- Jumlah Bus: 25
- Jumlah Saluran: 34
- Tegangan Dasar: 150 kV
- Daya Dasar: 100 MVA
- Parameter Saluran: pu

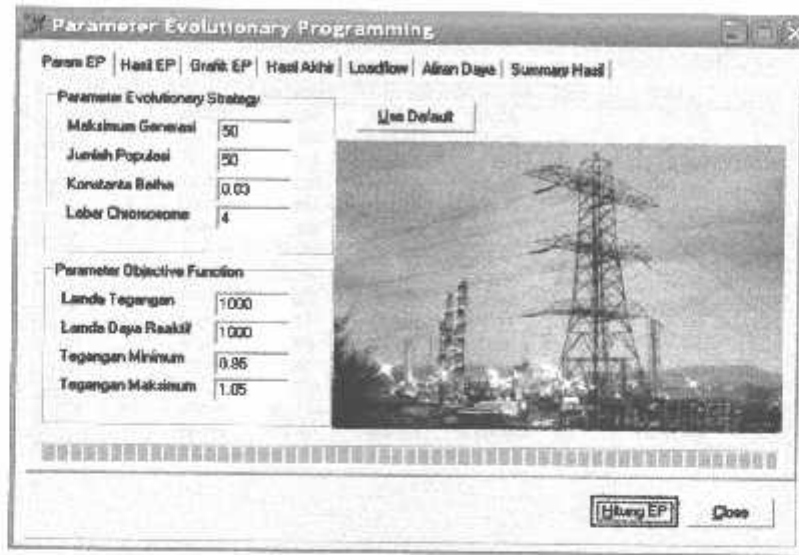
The 'Summary Loadflow' section shows the following results:

Jumlah Pembangkitan	769.483+ j 278.145	MVA
Jumlah Pembebanan	700.000+ j 260.200	MVA
Jumlah Rugi-Rugi	59.403+ j 15.945	MVA
Iterasi	4	
Waktu Hitung	0:0:15	

At the bottom of the window, there are buttons for 'Next', 'Hitung', and 'Close'.

Gambar 4.13.
Tampilan *Summary load flow* pada kondisi awal (Sebelum Optimasi)

9. Tekan tombol next, kemudian tekan tombol *use default*, untuk memasukkan parameter *Evolutionary Programming*, lalu tekan tombol hitung EP.



Gambar 4.16.
Tampilan parameter *Evolutionary Programming*

10. Kemudian tekan tombol *Load Flow*, untuk melihat hasil *Load Flow* pada kondisi akhir (setelah optimasi)

Bus	absV (pu)	sudV (deg)	Pg (MW)	Qg (MVAR)	Pl (MW)	Ql (MVAR)	Sups (pu)	Type Bus
1	1.04721	0.00000	953.297	0.609	39.300	7.900	0.000	1
2	1.03227	-1.81560	0.000	0.000	14.000	6.000	0.000	3
3	1.02913	-2.10713	0.000	0.000	18.000	8.500	0.000	3
4	1.01875	-4.20080	0.000	0.000	36.100	18.500	0.000	3
5	1.00452	-7.56983	0.000	0.000	39.000	16.800	0.000	3
6	1.00252	-8.24792	0.000	0.000	22.500	9.700	0.000	3
7	1.00750	-10.13674	0.000	50.000	62.200	30.000	0.000	3
8	0.97214	-14.11207	0.000	0.000	41.000	21.300	0.000	3
9	1.04964	-5.75079	0.000	0.000	23.000	5.900	0.000	3
10	1.02798	-7.89875	0.000	0.000	17.200	7.000	0.000	3
11	0.97132	-14.93545	0.000	0.000	37.800	18.700	0.000	3
12	0.97376	-15.38903	133.800	47.585	5.400	2.700	0.000	2
13	0.95151	-16.26149	0.000	0.000	11.900	4.400	0.000	3
14	0.95310	-21.48880	0.000	0.000	6.200	2.900	0.000	3
15	0.95738	-24.04035	0.000	0.000	63.600	22.800	0.000	3
16	0.96147	-22.86685	0.000	0.000	4.100	0.800	0.000	3

Gambar 4.17.
Tampilan hasil *Load Flow* pada kondisi akhir (Setelah Optimasi)

7	7	1.00992	-11.03892	0	50	62.2	30
8	8	0.98542	-15.66768	0	0	41	21.3
9	9	1.04858	-6.12896	0	0	23	5.9
10	10	1.02806	-8.52086	0	0	17.2	7
11	11	0.99115	-16.84471	0	0	37.8	18.7
12	12	0.99500	-17.35388	99.9	178.185	5.4	2.7
13	13	0.97345	-17.80185	0	0	11.9	4.4
14	14	0.95514	-22.07474	0	0	6.2	2.9
15	15	0.94802	-24.20512	0	0	63.8	22.8
16	16	0.95799	-23.40665	0	0	4.1	0.8
17	17	0.96324	-22.86039	0	0	23.6	4.3
18	18	0.94687	-24.73008	0	0	16.4	7
19	19	0.94794	-24.83452	0	0	73.3	28.9
20	20	0.95000	-24.88366	2.2	43.456	0	0
21	21	0.95000	-24.49084	96.7	0.678	0	0
22	22	0.94614	-25.21577	0	25	45.5	13.6
23	23	0.94630	-25.01627	0	25	50.3	13.5
24	24	0.94507	-24.86328	0	0	35.4	7.9
25	25	0.94277	-25.12002	0	0	14.2	3.3

Tabel 4.11.
Hasil Perhitungan Aliran Daya Antar Saluran Sebelum Optimasi

No.	Saluran		Daya	
	Dari	Ke	P(MW)	Q(MVAR)
1	1	2	206.547	19.32
2	1	9	324.756	-73.394
3	2	3	18.036	6.072
4	2	4	172.376	1.41
5	4	5	134.022	-20.771
6	5	6	51.524	-18.72
7	5	7	41.061	-23.608
8	6	7	28.535	-25.964
9	7	8	70.823	-0.655
10	7	10	-132.2	-11.407
11	7	11	67.91	-8.883
12	8	11	28.162	-19.367
13	9	10	153.194	26.248
14	9	11	143.602	-135.85
15	11	12	166.275	-184.01
16	12	13	188.069	-11.388
17	12	17	71.704	0.669
18	13	14	87.105	-6.698
19	13	15	84.907	-7.905
20	14	15	78.739	-10.524

29	19	20	-34.919	50.585	20	19	35.000	-50.406
30	19	21	-43.208	-174.242	21	19	43.900	176.179
31	19	22	20.744	5.000	22	19	-20.681	-6.438
32	19	23	36.55	22.169	23	19	-36.482	-22.496
33	23	24	-13.818	33.99	24	23	13.935	-34.372
34	24	25	14.223	1.353	25	24	-14.200	-3.300

Tabel 4.16.
Total Pembangkitan, Pembebanan dan Rugi-rugi Saluran Setelah Optimasi

Pembangkitan		Pembebanan		Rugi-rugi		Waktu Komputasi
Aktif (MW)	Reaktif (MVAR)	Aktif (MW)	Reaktif (MVAR)	Aktif (MW)	Reaktif (MVAR)	
765,987	273,966	700	260,200	65,987	13,766	0:0:16:203

Tabel 4.17.
Hasil Perhitungan Daya Yang Dibangkitkan dan Biaya Operasi Setelah Optimasi

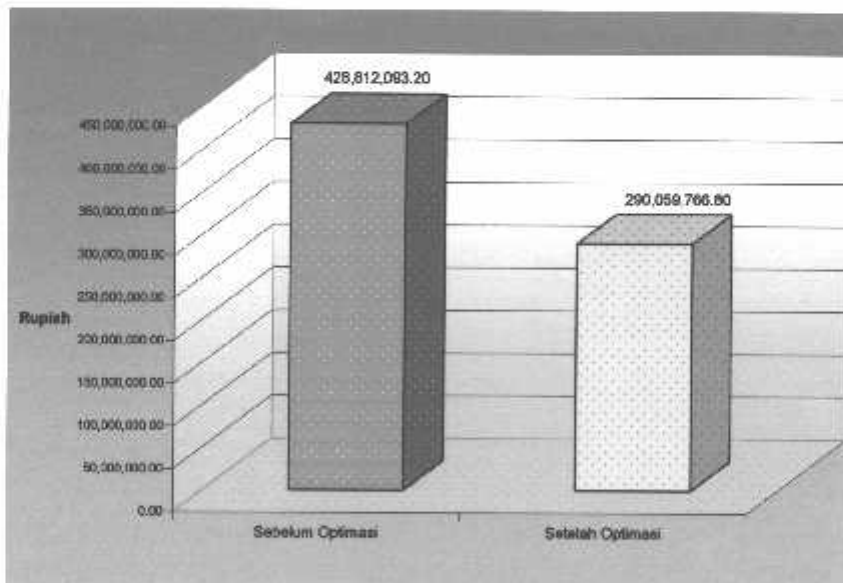
No	Nama Pembangkit	Daya Yang Dibangkitkan		Biaya Operasi (Rupiah/jam)
		(MW)	(MVAR)	
1	PLTU Paiton 1 & 2	553.287	0.609	Rp 134.257.806,6
2	PLTG Gilimanuk	133.8	47.585	Rp 46.684.727,10
3	PLTD Pesanggaran	35	-50.406	Rp 18.939.891,18
4	PLTG Pesanggaran	43.9	176.179	Rp 90.177.547,48
	Total	765,987	173,966	Rp 290.059.766,8

Tabel 4.18.
Perbandingan Tingkat Optimum Biaya Pembangkitan

No	Nama Pembangkit	Sebelum Optimasi	Setelah Optimasi
1	PLTU Paiton 1 & 2	Rp 137.668.784,2	Rp 134.257.806,6
2	PLTG Gilimanuk	Rp 46.069.669,42	Rp 46.684.727,10
3	PLTD Pesanggaran	Rp 15.117.727,12	Rp 18.939.891,18
4	PLTG Pesanggaran	Rp 229.893.607,4	Rp 90.177.547,48
	Total	Rp 428.812.093,2	Rp 290.059.766,8

Dari tabel tingkat optimum diatas dapat dilihat bahwa besarnya biaya produksi pembangkitan sebelum optimasi adalah Rp 428.812.093,2 sedangkan besarnya biaya produksi pembangkitan setelah optimasi adalah Rp 290.059.766,8 Jadi selisih antara biaya produksi pembangkitan sebelum dengan setelah optimasi adalah Rp 138.752.326,4 atau terjadi optimasi biaya sebesar 32 %.

Grafik 4.3.
Perbandingan Biaya Pembangkitan Sebelum dan Setelah Optimasi



BAB V KESIMPULAN DAN SARAN

5.1. Kesimpulan

Setelah dilakukan analisa perhitungan optimasi daya reaktif dengan menggunakan metode *Evolutionary Programming* (EP) maka dapat diambil kesimpulan sebagai berikut :

1. Dari hasil perhitungan *load flow* setelah optimasi daya reaktif menggunakan metode *Evolutionary Programming* profil tegangan masih dalam batas-batas yang diijinkan yaitu antara -5 % sampai +5 % dari tegangan nominal 150 kV, dan menjadi lebih baik dibandingkan dengan kondisi awal.
2. Biaya produksi pembangkitan berhasil diminimalisasi. Biaya produksi pembangkitan sebelum optimasi adalah Rp. 428.812.093,295 sedangkan besarnya biaya produksi pembangkitan setelah optimasi adalah Rp 290.059.766,8 Jadi selisih antara biaya produksi pembangkitan sebelum dengan setelah optimasi adalah Rp 138.752.326,4 atau terjadi optimasi biaya sebesar 32 %.
3. Dengan mengoptimasi daya reaktif menggunakan metode *Evolutionary Programming* ternyata rugi-rugi daya sistem turun dari 69,403 MW dan 15,945 MVAR menjadi 65,987 MW dan 13,766 MVAR terjadi optimasi sebesar 3,416 MW dan 2,179 MVAR.

5.2. Saran

Penggunaan metode EP untuk menyelesaikan permasalahan optimasi daya reaktif masih perlu dikembangkan lagi terhadap sistem jaringan yang lebih luas dan penggunaan waktu komputasi yang lebih singkat sehingga nantinya dapat digunakan lebih efektif dan efisien dalam pemecahan permasalahan optimasi biaya pembangkitan.

DAFTAR PUSTAKA

- [1]. Stevenson, William D., Jr, "*Analisa Sistem Tenaga Listrik*", Penerbit Erlangga edisi keempat, 1996.
 - [2]. Wood, Allan J. and B. F. Wollenberg., "*Power Generation, Operation, and Control*", John Willey & Sons, Inc., 1996.
 - [3]. Kwang. Y. Lee, and Frank F. Yang, "*Optimal Reactive Power Planning Using Evolutionary Algorithms: A Comparative Study for Evolutionary Programming, Evolutionary Strategy, Genetic Algorithm, and Linier Programming,*" IEEE Trans on Power System, Vol 13, No 1, February 1998.
 - [4]. Zuhail, "*Dasar Teknik Tenaga Listrik dan Elektronika Daya*", PT. Gramedia Utama, Jakarta, 1995.
 - [5]. Stagg, Glenn W. and Ahmed H. El-Abiad "*Computer Methods In Power System Analysis*".
 - [6]. Yuryevich, J. and K.P. Wong, "*Evolutionary Programming Based Optimal Power Flow algorithm*", IEEE Trans. on Power Systems, Vol. 14, No. 4, pp.1245-1250, Nov. 1999.
 - [7]. <http://zooland.alife.org/hhg2ec/>, "*What's Evolutionary Programming (EP)?*".
 - [8]. Luke, Brian T., "*Overview of Evolutionary Programming Methods*", Learning From The Web.net.
 - [9]. Spears, William M., "*An Overview of Evolutionary Computation*"..
 - [10]. Hussain, Talib s., "*An Introduction to Evolutionary Computation*", Department of Computing and Information Science Queen's University, Kingston.
 - [11]. Pai, M. A., "*Computer Techniques in Power System Analysis*", Tata Mc Graw-Hill Publishing Company Limited.
 - [12]. Syafena, Jatri, Skripsi, "*Economic Dispatch Dengan Optimasi Daya Aktif dan Reaktif Pada Pembangkit Thermal di Area IV*", 2002.
 - [13]. Data "*Load Flow Region Jawa Timur dan Bali*", Sub Sistem 150 kV Paiton dan Bali Senin 19 Januari 2004, Pukul 19:00 WIB, PT. PLN (Persero) P3B Region Jawa dan Bali, Waru.
-

- [14]. Data Saluran Transmisi 150 kV Sub Sistem Paiton dan Bali.
- [15]. Lamont. John W and Fu Jian, "*Cost Analysis Of Reactive Power Support*," IEEE Trans on Power System, Vol. 14, No. 2, August. 1999.
-

LAMPIRAN




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

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

Nama Mahasiswa : JOKO NUGROHO
N.I.M. : 00.12.034
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Energi Listrik
Judul Skripsi : OPTIMASI DAYA REAKTIF SECARA
EKONOMIS MENGGUNAKAN METODE
EVOLUTIONARY PROGRAMMING PADA
SALURAN TRANSMISI 150 KV SUB SISTEM
PAITON DAN BALI

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

Hari : Rabu
Tanggal : 30 Maret 2005
Dengan Nilai : 81,85 (A) 



(Ir. Mochtar Asroni, MSME)
Ketua

Panitia Ujian

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

Anggota Penguji

(Ir. Choirul Saleh, MT.)
Penguji Pertama

(Ir. Eko Nurcahyo.)
Penguji Kedua



LEMBAR BIMBINGAN SKRIPSI

1. Nama : JOKO NUGROHO
2. NIM : 00.12.034
3. Jurusan : Teknik Elektro S-1
4. Konsentrasi : Teknik Energi Listrik
5. Judul Skripsi : OPTIMASI DAYA REAKTIF SECARA EKONOMIS MENGGUNAKAN METODE *EVOLUTIONARY PROGRAMMING* PADA SALURAN TRANSMISI 150 KV SUB SISTEM PAITON DAN BALI
6. Tanggal Mengajukan Skripsi : 25 September 2004
7. Tanggal Menyelesaikan Skripsi : 28 Maret 2005
8. Dosen Pembimbing : Ir. Yusuf Ismail Nakhoda, MT.
10. Telah dievaluasi dengan nilai : 88,00 (Delapan Puluh Delapan Koma Nol) 8_

Malang, 28 Maret 2005

Mengetahui,
Ketua Jurusan Teknik Elektro S-1

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

Disetujui,
Dosen Pembimbing

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



LEMBAR PENGAJUAN JUDUL SKRIPSI
JURUSAN TEKNIK ELEKTRO S-1

Konsentrasi : Teknik Energi Listrik/Teknik Elektronika

1	Nama Mahasiswa : ...	JOKO WURHO	Nim : 00 12034
2	Waktu pengajuan	Tanggal :	Bulan : Tahun :
3	Spesifikasi judul (berilah tanda silang)		
	a. Sistem Tenaga Elektrik	e. Elektronika & Komponen	
	b. Energi & Konversi Energi	f. Elektronika Digital & Komputer	
	c. Tegangan Tinggi & Pengukuran	g. Elektronika Komunikasi	
	d. Sistem Kendali Industri	h. lainnya	
4	Konsultasikan judul sesuai materi bidang ilmu kepada Dosen *) :		Ketua Jurusan.
	Ir. Yurut Ismail N. MT,		 Ir. I Made Warta, MT Nip: 131 991 182
5	Judul yang diajukan mahasiswa :	ANALISA PENGGUNAAN METODE EVOLUTONARY PROGRAMMING UNTUK OPTIMASI DAYA REAKTIF SECARA EKONOMIS PADA SALURAN TRANSMISI 150 KV SUB SISTEM BALI	
6	Perubahan Judul yang disetujui Dosen sesuai materi bidang ilmu	Optimasi Daya Reaktif Secara Ekonomis Menggunakan Metode Evolutionary Programming pada Saluran Transmisi 150 kv Sub Sistem Bali	
7	Catatan :		
		
		
		
		
		
	Persetujuan Judul Skripsi yang dikonsultasikan kepada Dosen materi bidang ilmu		Disetujui, 30-9-, 2004. Dosen Ir. Yusuf Ismail Nakhoda, MT

Perhatian :

1. Formulir Pengajuan ini harap dikembalikan kepada jurusan paling lambat satu minggu setelah disetujui kelompok dosen keahlian dengan dilampirkan proposal skripsi beserta persyaratan skripsi sesuai form S-1
2. Keterangan : *) coret yang tidak perlu
**) dilingkari a, b, c, atau g. sesuai bidang keahlian

Form.S-2

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

Lampiran : 1 (satu) berkas
Pembimbing Skripsi

Kepada : Yth. Bapak/Ibu Ir. Gusut Ismail N. MT,
Dosen Institut Teknologi Nasional
MALANG

Yang bertanda tangan di bawah ini :

Nama : JOKO WUBROHO
Nim : 00.12034
Jurusan : Teknik Elektro S-1
Konsentrasi : Energi Listrik

Dengan ini mengajukan permohonan, kiranya Bapak/Ibu bersedia menjadi Dosen Pembimbing Utama / Pendamping ^{*}, untuk penyusunan Skripsi dengan judul (proposal terlampir): OPTIMASI DAYA REAKTIF SECARA EKONOMIS MENGGUNAKAN METODE EVOLUTIONARY PROGRAMMING PADA SALURAN TRANSMISI 150 KV SUB SISTEM BALI

Adapun tugas tersebut sebagai salah satu syarat untuk menempuh Ujian Akhir Sarjana Teknik.

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

Malang, OKTOBER 2004

Mengetahui
Ketua Jurusan Teknik Elektro



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

Hormat kami,



JOKO WUBROHO

^{*}) coret yang tidak perlu

PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa/i :

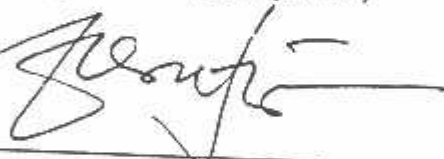
Nama : JOKO WUROTO
Nim : 0012034
Semester : IX (sembilan)
Jurusan : Teknik Elektro S-1
Konsentrasi : Energi Listrik

Dengan ini Menyatakan bersedia / tidak bersedia *) Membimbing Skripsi dari mahasiswa tersebut, dengan judul : OPTIMASI DAYA REAKTIF SECARA EKONOMIS MENGGUNAKAN METODE EVOLUTIONARY PROGRAMMING PADA SALURAN TRANSMISI 150 KV GUB KUSTEM BALI

Demikian surat Pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Malang, 4-10-2004

Kami yang Membuat pernyataan,



NIP. P. 1018800189

Catatan:

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



PT (PERSERO) MALANG
BANK NIAGA MALANG

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

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

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

Malang, 13 Des. 2004

Nomor : ITN-1041/I.SKP /2/04
Lampiran : satu lembar
Perihal : **BIMBINGAN SKRIPSI**

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

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

Nama : JOKO NUGROHO
Nim : 0012034
Fakultas : Teknologi Industri
Jurusan : Teknik Elektro
Konsentrasi : T. Energi Listrik (S-1)

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

10 Des. 2004 s/d 10 Juni 2005

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



Ketua
Jurusan Teknik Elektro
[Signature]
Ir. I Made Wartana, MT
Nip. 131 991 182

Tindakan :

1. Mahasiswa yang bersangkutan
2. Arsip.

Form. S-4a



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

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

Jl. PERSERO, MALANG
KORPORASI PERUSAHAAN
KONVENSIONAL NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting) Fax. (0341) 533015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Nomor : ITN-599/III.TA/2/2004
Tempat : -
Perihal : SURVEY / Permintaan Data

Malang, 03 Desember 2004

Keperihal : Yth. Bapak Pimpinan
PT. PLN (Persero) P3B Jawa Bali
Region Jawa Timur dan Bali
Jl. Suningrat No.45

di -

Taman Sidoarjo 651256

Bersama ini dengan hormat kami mohon kebijaksanaan Bapak agar mahasiswa/i kami dari **Fakultas Teknologi Industri Jurusan Teknik Elektro Konsentrasi Teknik Energi Listrik S-1** dapat diijinkan untuk melaksanakan survey pada perusahaan yang Bapak pimpin untuk mendapatkan data – data guna menyusun skripsi dengan judul :

OPTIMALISASI DAYA REAKTIF SECARA EKONOMIS MENGGUNAKAN METODE EVOLUTIONARY PROGRAMMING PADA SALURAN TRANSMISI 150 KV SISTEM BALI

Mahasiswa tersebut adalah :

1. Joko Nugroho

Nim. 00.12.034

Adapun lama survey 1 (Satu) Bulan

Demikian agar maklum dan atas perhatian serta bantuannya kami ucapkan terima kasih.



DEKAN

Fakultas Teknologi Industri,
U.B. Pembantu Dekan I

(Signature)
Ir. Mochtar Asroni, MSME
NIP. P. 1018100036



PT PLN (PERSERO)
PENYALURAN DAN PUSAT PENGATUR BEBAN JAWA BALI
REGION JAWA TIMUR & BALI

Jl. Suningrat No. 45 Taman Sidoarjo 61257

Telepon : (031) 7882113, 7882114
Facsimile : (031) 7882578, 7881024

Kotak Pos : 4119 SBS
Bank : Bank Mandiri

Nomor : *deb 2 / 334 / RJTB / 2004.*
Surat Sdr. No. : ITN-189/III.TA/2/2004.
ITN-190/III.TA/2/2004.
Lampiran : 1 (satu) lampiran.
Perihal : Ijin Survey/ Pengambilan Data.

07 DEC 2004

Kepada

Yth. Dekan Fakultas Teknik,
Institut Teknologi Nasional Malang
Di
MALANG

Menunjuk surat Saudara nomor : ITN-189/III.TA/2/2004 dan ITN-190/III.TA/2/2004 tanggal 11 Oktober 2004 perihal : Survey/ Permintaan Data. dengan ini diberitahukan bahwa kami tidak keberatan untuk memberikan ijin kepada Mahasiswa Saudara. bernama :

- JOKO NUGROHO NIM : 00.12.034.
- DWI ARIES TIRTA WINARTA NIM : 00.12.091.

Untuk melakukan Praktek Kerja pada PT. PLN (Persero) P3B Region Jawa Timur dan Bali Bidang OPHAR, dengan persyaratan sebagai berikut :

1. Mahasiswa tersebut diatas supaya mengisi dan menanda tangani Surat Pernyataan 1 (satu) lembar bernomorerai Rp. 6.000,-
2. Mahasiswa yang bersangkutan agar mematuhi peraturan/ketentuan yang berlaku di PT. PLN (PERSERO) sehingga faktor-faktor kerahasiaan harus benar-benar diutamakan.
3. Semua biaya perjalanan, penginapan, makan dan lain sebagainya tidak menjadi tanggungan PT. PLN (Persero) P3B Region Jawa Timur dan Bali.
4. Buku Laporan Kerja Praktek Mahasiswa tersebut agar dikirimkan kepada PT. PLN (Persero) P3B Region Jawa Timur dan Bali 1 (satu) buah.
5. Untuk informasi lebih lanjut dapat menghubungi PT. PLN (Persero) P3B Region Jawa Timur dan Bali Cq. Bidang Enjiniring.

Demikian harap maklum dan terima kasih atas perhatian saudara.

a.n. MANAGER
IH MANAGER BIDANG ENJINIRING,



ZATMOKO

Tembusan Yth. :

1. ALSDM PLN P3B
2. Sdr. Joko Nugroho
3. Sdr. Dwi Aries Tirta Winarta



SURAT PERNYATAAN

Yang bertanda tangan dibawah ini, saya :

Nama : JOKO HUGROHO
 Pria/ Wanita : PRIA
 Tempat / Tanggal lahir : MADIUN 14 MEI 1982
JAWA TIMUR
 Alamat / no telepon : Jl. SUMBERSARI 14 / 69 MALANG
081334647900
 Pekerjaan : MAHASISWA

Dengan ini saya menerangkan bahwa :

1. Saya bersedia dan setuju menanggung semua akibat yang ditimbulkan karena kesalahan maupun kelalaian saya dan semua akibat lainnya yang terjadi paca instalasi peralatan milik PLN selama melakukan Training/ Praktek Kerja/ Riset pada PT PLN (Persero) P3B Region Jawa Timur dan Bali, yang telah mendapat ijin dari PT PLN (Persero) P3B region Jawa Timur dan Bali :
2. Saya atas peringatan pertama akan membayar sepenuhnya , semua biaya yang langsung menimbulkan kerugian atau kecelakaan , karena kelalaian saya ;
3. Saya akan segera mematuhi semua petunjuk –petunjuk yang diberikan oleh Petugas PT PLN (Persero) P3B Region Jawa Timur dan Bali ;
4. Saya sanggup tidak membocorkan hal – hal yang bersifat rahasia perusahaan PT PLN (Persero) P3B Region Jawa Timur dan Bali dan bahan yang saya peroleh dalam Training/ Praktek Kerja/ Riset, dan tidak saya pergunakan untuk hal – hal yang dapat merugikan PT PLN (Persero) P3B Region Jawa Timur dan Bali
5. Saya sanggup menanggung sendiri segala sesuatu untuk keperluan Training/ Praktek Kerja/ Riset termasuk biaya perjalanan , penginapan makan dan sebagainya ;
6. Saya sanggup menyerahkan 1 (satu) buah buku laporan Training/ Praktek Kerja/ Riset kepada PT PLN (Persero) P3B Region Jawa Timur dan Bali, setelah saya presentasikan kepada Manager Bidang Enjiniring PT PLN (Persero) P3B Region Jawa Timur dan Bali mengenai tugas Training/ Praktek Kerja/ Riset.
7. Saya tunduk dan akan mentaati semua peraturan yang berlaku di PT PLN (Persero) P3B Region Jawa Timur dan Bali, dan saya sanggup tidak meninggalkan tugas kedinasan selama Training/ Praktek Kerja/ Riset.



Mengetahui
 Manager Bidang SDM & AD

HAENAL ARIFIN

Surabaya, 2 - 12 - 2009
 Yang membuat pernyataan

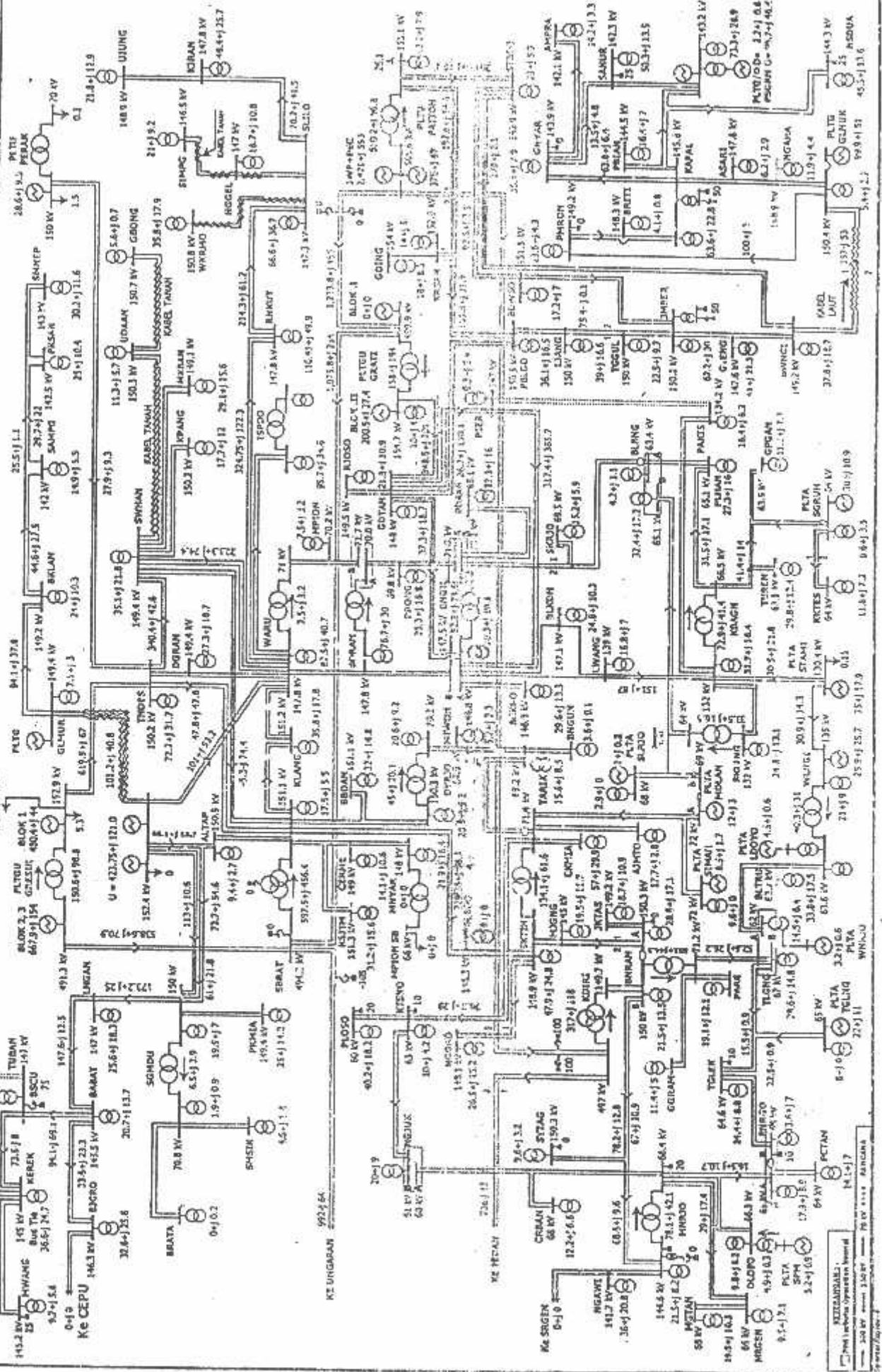


Lampiran-1

LOAD FLOW REGION JAWA TIMUR DAN BALI

Hari Senin, 19 Januari 2004 Pukul : 19:00 WIB

1. Rencana :
 Pemadatan Region-1 : 48753.41 MVA
 Total Transfer : 17280.41 MVA
 Distribusi Region-4 : 3247.34 MVA
 : Normal Operasi



LEGENDA:
 --- Pemadatan
 --- Transfer
 --- Distribusi

100 MW
 100 MVA
 70 MW
 70 MVA
 50 MW
 50 MVA
 30 MW
 30 MVA
 20 MW
 20 MVA
 10 MW
 10 MVA

DATA KARAKTERISTIK TRANSMISI UPI & SRB

Salah	Daerah	Kc	No. Salib	Teg (kV)	Jarak (ft/m)	Jenis	1 Hour (amp)	1 Day (amp)	1 Year (amp)	10 Year (amp)	15 Year (amp)	20 Year (amp)	30 Year (amp)	40 Year (amp)	50 Year (amp)	60 Year (amp)	70 Year (amp)	80 Year (amp)	90 Year (amp)	100 Year (amp)
-------	--------	----	-----------	----------	--------------	-------	--------------	-------------	--------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	----------------

Salah	Daerah	Kc	No. Salib	Teg (kV)	Jarak (ft/m)	Jenis	1 Hour (amp)	1 Day (amp)	1 Year (amp)	10 Year (amp)	15 Year (amp)	20 Year (amp)	30 Year (amp)	40 Year (amp)	50 Year (amp)	60 Year (amp)	70 Year (amp)	80 Year (amp)	90 Year (amp)	100 Year (amp)		
1	PITON7	GRAT17	1	500	55.433	GANNET	2800	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	
1	PITON7	GRAT17	2	500	55.433	GANNET	2800	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000
1	BNGILS	GOTANS	1	150	16.605	ACSR, AW	740	1000	1000	1250	1500	1500	1500	1500	1500	1500	1500	1500	1500	1500	1500	1500
1	BNGILS	GOTANS	2	150	16.605	ACSR, AW	740	1000	1000	1250	1500	1500	1500	1500	1500	1500	1500	1500	1500	1500	1500	1500
1	BNGILS	BDRANS	1	150	9.800	ACSR, AW	740	1000	1000	1600	1600	1600	1600	1600	1600	1600	1600	1600	1600	1600	1600	1600
1	BNGILS	WAR J5	1	150	9.800	ACSR, AW	740	1000	1000	1600	1600	1600	1600	1600	1600	1600	1600	1600	1600	1600	1600	1600
1	BNGILS	PIER5	1	150	5.150	ACSR, ZEERA	1620	2000	1000	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250
1	BNGILS	PIER5	2	150	5.150	ACSR, ZEERA	1620	2000	1000	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250
1	BNGILS	BCKR05	1	150	6.200	ACSR, ZEBRA	1620	1000	1000	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250
1	BNGILS	BCKR05	2	150	6.200	ACSR, ZEBRA	1620	1000	1000	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250
1	BNGILS	PNM55	1	150	27.770	ACSR, AW	740	1000	1000	1250	1500	1500	1500	1500	1500	1500	1500	1500	1500	1500	1500	1500
1	BNGILS	LWANG5	1	150	34.660	ACSR, AW	740	1000	800	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250
1	GOTANS	RJOS05	1	150	10.487	ACSR, AW	740	1000	800	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250
1	GOTANS	RJOS05	2	150	10.487	ACSR, AW	740	1000	800	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250
1	PIER5	GRAT15	1	150	31.935	TACSR	1620	1000	1000	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250
1	PIER5	GRAT15	2	150	31.935	TACSR	1620	1000	1000	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250
1	PBLG05	GOTANS	1	150	33.827	ACSR, AW	740	800	1000	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250
1	PBLG05	GOTANS	2	150	33.827	ACSR, AW	740	800	1000	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250
1	PBLG05	LJANG5	1	150	51.692	ACSR, AW	740	600	600	800	800	800	800	800	800	800	800	800	800	800	800	800
1	PBLG05	LJANG5	2	150	51.692	ACSR, AW	740	600	600	800	800	800	800	800	800	800	800	800	800	800	800	800
1	KRSANS	PBLG05	1	150	30.238	ACSR, AW	740	1000	1000	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250
1	KRSANS	PBLG05	2	150	30.238	ACSR, AW	740	1000	1000	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250
1	KRSANS	GJING5	1	150	19.137	ACSR, AW	740	1000	400	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250
1	KRSANS	KRSANS	1	150	20.204	ACSR, AW	740	1000	1000	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250	1250

6. UPT Probelinggo

(15)=11x10^17.722/1000
(14)=10x10^17.722/1000

(13)=7/14

(5)

(3)

(2)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

DATA KAKAKI ELEKTRIK TRANSMISI UPT & SR8

Jml. Sirkuit	Dari	Kc	No. Sirkuit	Jarak (Km)	Jenis		Jarak (Km)		Jarak (Km)		R (Ohm-km)	X (Ohm-km)	G (Ohm-km)	K INU (Ohm)	K DMU (Ohm)	K INU (Ohm)	K DMU (Ohm)	K INU (Ohm)	K DMU (Ohm)
					1	2	1	2	1	2									

No. Sirkuit	Kc	No. Sirkuit	Jarak (Km)	Jenis	Jarak (Km)		Jarak (Km)		R (Ohm-km)	X (Ohm-km)	G (Ohm-km)	K INU (Ohm)	K DMU (Ohm)	K INU (Ohm)	K DMU (Ohm)	K INU (Ohm)	K DMU (Ohm)
					1	2	1	2									

6. UPT Probolinggo

No. Sirkuit	Kc	No. Sirkuit	Jarak (Km)	Jenis	1	2	1	2	R (Ohm-km)	X (Ohm-km)	G (Ohm-km)	K INU (Ohm)	K DMU (Ohm)	K INU (Ohm)	K DMU (Ohm)	K INU (Ohm)	K DMU (Ohm)			
1	PITONS	KRSANS	2	150	20.204	ACSR AW	330 mm ²	7.10	1070	1000	1250	1250	0.1172	0.4003	2.8530	0.0105	0.0359	0.0130	0.0359	192.2520
1	PITONS	STBDO5	1	150	55.433	ACSR ZEBRA	2 x 435 mm ²	1600	2090	2000	1250	1250	0.0387	0.2807	4.0230	0.0095	0.0692	0.0502	0.0692	415.6850
1	PITONS	STBDO5	2	150	55.433	ACSR ZEBRA	2 x 435 mm ²	1600	2090	2000	1250	1250	0.0387	0.2807	4.0230	0.0095	0.0692	0.0502	0.0692	415.6850
1	BNGIL4	GOTAN4	1	70	4.452	ACSR OSTRICH	300 MCM	440						#DIV/0!	#DIV/0!					53.3456
1	BNGIL4	GOTAN4	2	70	4.452	ACSR OSTRICH	300 MCM	440						#DIV/0!	#DIV/0!					48.4960
1	BNGIL4	PDAAN4	1	70	9.700	ACSR OSTRICH	300 MCM	440	250	400			0.2180	0.3850	2.972	0.0432	0.0762	0.0014	0.0762	53.3456
1	BNGIL4	PDAAN4	2	70	9.700	ACSR OSTRICH	300 MCM	440	250	400	800	800	0.2180	0.3850	2.972	0.0432	0.0762	0.0014	0.0762	53.3456
1	BNGIL4	BDRAN4	1	70	11.700	ACSR PIPER	300 MCM	440	400	400	400	400	0.2180	0.3850	2.972	0.0521	0.0919	0.0017	0.0919	53.3456
1	BNGIL4	PRONG4	1	70	1.700	ACSR PIPEK	300 MCM	440	400	400	400	400	0.2180	0.3850	2.972	0.0521	0.0919	0.0017	0.0919	53.3456
1	BNGIL4	BLBNG4	1	70	40.000	ACSR PIGEON	30 AWG	300	200	200	800	800	0.3053	0.3584	3.202	0.2492	0.2926	0.0063	0.2926	36.3720
1	BNGIL4	BLBNG4	2	70	40.000	ACSR PIGEON	30 AWG	300	200	200	800	800	0.3053	0.3584	3.202	0.2492	0.2926	0.0063	0.2926	36.3720

Modifikasi TL Bay Grati - Pier 1 sirkit menjadi (Grafir - Gdtan - Pier) dan Probolinggo - Gondangwetan 1 sirkit menjadi (Pblngo - Grafir - Gdtan)

No. Sirkuit	Kc	No. Sirkuit	Jarak (Km)	Jenis	1	2	1	2	R (Ohm-km)	X (Ohm-km)	G (Ohm-km)	K INU (Ohm)	K DMU (Ohm)	K INU (Ohm)	K DMU (Ohm)	K INU (Ohm)	K DMU (Ohm)			
1	GRATIS	GDTANS	1	150	10.743	TACSR	410 mm ²	2800	1000	1000	1250	2000	0.0387	0.2807	4.0230	0.0018	0.0134	0.0097	0.0134	727.4400
1	GRATIS	GDTANS	2	150	10.743	TACSR	410 mm ²	2800	1000	1000	1250	2000	0.0387	0.2807	4.0230	0.0018	0.0134	0.0097	0.0134	727.4400
1	GOTANS	PIER5	1	150	21.183	TACSR	410 mm ²	2800	1000	1000	1250	2000	0.0387	0.2807	4.0230	0.0036	0.0264	0.0192	0.0264	727.4400
1	GOTANS	PIER5	2	150	21.183	TACSR	410 mm ²	2800	1000	1000	1250	2000	0.0387	0.2807	4.0230	0.0036	0.0264	0.0192	0.0264	727.4400

No. Urut	Daerah	Ke. O.	No. Sinyal	Teg (kV)	Jarak (Km)	Udls Pengantar	Tipe		R (Ohm/Km)	X (Ohm/Km)	B (MVA)	R (Ohm)	X (Ohm)	E (MVA)	R (Ohm)	X (Ohm)	E (MVA)
							Lari	Fe									

No. Urut	Daerah	Ke. O.	No. Sinyal	Teg (kV)	Jarak (Km)	Udls Pengantar	Tipe		R (Ohm/Km)	X (Ohm/Km)	B (MVA)	R (Ohm)	X (Ohm)	E (MVA)	R (Ohm)	X (Ohm)	E (MVA)
							Lari	Fe									

Sub Region Bali

1	GLNUKS	CHGLNUK	1	150	1720	ACSR HAWK	477 MCM	645	800	800	0.3966	0.1370	0.0030	0.0051	0.0030	0.0051	167.5710
1	GLNUKS	CHGLNUK	2	150	1720	ACSR HAWK	477 MCM	645	800	800	0.3966	0.1370	0.0030	0.0051	0.0030	0.0051	167.5710
1	CHGLNUK	CHKTPNG	1	150	4471	OFC	3 x 300 mm ²	452	800	800	0.3482	0.1613	0.0032	0.0069	0.0032	0.0069	120.0270
1	CHGLNUK	CHKTPNG	2	150	4389	OFC	3 x 300 mm ²	452	800	800	0.3548	0.1643	0.0032	0.0069	0.0032	0.0069	120.0270
1	CHKTPNG	BWNGIS	1	150	7990	ACSR HAWK	477 MCM	645	800	800	0.3966	0.1370	0.0049	0.0141	0.0141	0.0052	157.5710
1	CHKTPNG	BWNGIS	2	150	7990	ACSR HAWK	477 MCM	645	800	800	0.3966	0.1370	0.0049	0.0141	0.0141	0.0052	157.5710
1	GLNUKS	NGARAS	1	150	38.040	ACSR HAWK	477 MCM	645	800	800	0.3966	0.1370	0.0232	0.0671	0.0671	0.0247	157.5710
1	GLNUKS	NGARAS	2	150	38.040	ACSR HAWK	477 MCM	645	800	800	0.3966	0.1370	0.0232	0.0671	0.0671	0.0247	157.5710
1	GLNUKS	PIMONS	1	150	75.900	ACSR HAWK	2 x 477 MCM	1290	800	800	0.3966	0.1183	0.0399	0.1314	0.1314	0.0502	335.1420
1	NGARAS	KAPALS	1	150	67.710	ACSR HAWK	477 MCM	645	800	800	0.3966	0.1370	0.0412	0.1194	0.1194	0.0439	167.5710
1	NGARAS	ASARIS	1	150	44.400	ACSR HAWK	477 MCM	645	800	800	0.3966	0.1370	0.0270	0.0783	0.0783	0.0288	167.5710
1	KAPALS	ANSRI5	1	150	23.310	ACSR HAWK	477 MCM	645	800	800	0.3966	0.1370	0.0142	0.0411	0.0411	0.0151	167.5710
1	KAPALS	PIMONS	1	150	58.580	ACSR PARTRIDGE	135 MCM	400	400	400	0.4046	0.2136	0.0556	0.1054	0.1054	0.0369	103.9200
1	KAPALS	GNYARS	1	150	19.210	ACSR HAWK	477 MCM	645	800	800	0.3966	0.1370	0.0117	0.0335	0.0335	0.0125	167.5710
1	KAPALS	GNYARS	2	150	19.210	ACSR HAWK	477 MCM	645	800	800	0.3966	0.1370	0.0117	0.0335	0.0335	0.0125	167.5710
1	KAPALS	NSDUAS	1	150	30.910	ACSR PARTRIDGE	135 mm ²	400	400	400	0.4064	0.2136	0.0293	0.0561	0.0561	0.0195	103.9200
1	KAPALS	PGRANS	1	150	17.220	ACSR HAWK	477 mm ²	645	1000	1000	0.3966	0.1370	0.0165	0.0304	0.0304	0.0112	157.5710
1	KAPALS	BRITIS	1	150	38.170	ACSR PARTRIDGE	135 mm ²	400	800	800	0.4084	0.2136	0.0362	0.0683	0.0683	0.0240	103.9200
1	GNYARS	SANURS	1	150	16.500	ACSR PARTRIDGE	135 mm ²	400	200	1000	0.4084	0.2136	0.0157	0.0299	0.0299	0.0104	103.9200
1	GNYARS	SANURS	2	150	16.380	ACSR PARTRIDGE	135 mm ²	400	800	1000	0.4084	0.2136	0.0156	0.0297	0.0297	0.0103	103.9200
1	GRAGE	EMF22E	1	150	32.750	ACSR HAWK	477 MCM	645	400	500	0.3966	0.1370	0.0206	0.0595	0.0595	0.0219	157.5710
1	GNYARS	AMPRA5	2	150	33.760	ACSR HAWK	477 MCM	645	1000	1000	0.3966	0.1370	0.0206	0.0595	0.0595	0.0219	157.5710
1	SANURS	PGRANS	1	150	7.740	ACSR PARTRIDGE	135 mm ²	400	400	400	0.4084	0.2136	0.0073	0.0140	0.0140	0.0049	103.9200
1	SANURS	PGRANS	2	150	7.750	ACSR PARTRIDGE	135 mm ²	400	1000	1000	0.4084	0.2136	0.0074	0.0141	0.0141	0.0049	103.9200


```

unit uGenetic;

interface

uses uUtils,uRandom;

type
  TGenetic=class
  private
    FMaxGen,FPopSize,FLength:integer;
    function GetMin:dArr1;
    function GetAvg:dArr1;
    function GetMax:dArr1;
  protected
    FRandom:TRandomu;
    FMin,FAvg,FMax:dArr1;
    procedure InitParent;virtual;abstract;
    procedure Statistik;virtual;abstract;
    procedure Generasi;virtual;abstract;
  public
    constructor Create;overload;
    constructor Create(const aMaxGen,aPopSize,aLength:integer);overload;
    destructor Destroy;override;
    property MaxGen:integer read FMaxGen write FMaxGen;
    property PopSize:integer read FPopSize write FPopSize;
    property LengthX:integer read FLength write FLength;
    property Min:dArr1 read GetMin;
    property Avg:dArr1 read GetAvg;
    property Max:dArr1 read GetMax;
  end;

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

  TPopBin1=array of TIndiBin1;

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

  TPopBin2=array of TIndiBin2;

```

```
TIndiDouble1=record
  chrom:dArr1;
  fitness:double;
end;
```

```
TPopDouble1=array of TIndiDouble1;
```

```
TIndiDouble2=record
  chrom:dArr2;
  fitness:double;
end;
```

```
TPopDouble2=array of TIndiDouble2;
```

```
TIndiEP1=record
  chrom:TChromEP1;
  fitness:double;
end;
```

```
TPopEP1=array of TIndiEP1;
```

```
TIndiUpfr1=record
  chrom:TChromUpfr1;
  fitness:double;
end;
```

```
TPopUpfr1=array of TIndiUpfr1;
```

implementation

```
constructor TGenetic.Create;
begin
  inherited Create;
  FMaxGen:=50;
  FPopSize:=50;
  FLength:=1;
  SetLength(FMin,FMaxGen);
  SetLength(FAvg,FMaxGen);
  SetLength(FMax,FMaxGen);
  FRandom:=TRandomu.Create;
end;
```

```
constructor TGenetic.Create(const aMaxGen,aPopSize,aLength:integer);
begin
  inherited Create;
```

```

FMaxGen:=aMaxGen;
FPopSize:=aPopSize;
FLength:=aLength;
SetLength(FMin,FMaxGen);
SetLength(FAvg,FMaxGen);
SetLength(FMax,FMaxGen);
FRandom:=TRandomu.Create;
end;

destructor TGenetic.Destroy;
begin
  try
    FRandom.Free;
  finally
    inherited Destroy;
  end;
end;

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

function TGenetic.GetAvg:dArr1;
var i:integer;
begin
  SetLength(result,FMaxGen);
  for i:=1 to FMaxGen do
  begin
    result[i-1]:=FAvg[i-1];
  end;
end;

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

```

```

end;
end;

end.

unit uLoadflow;

interface

uses uUtils,uComplex,SysUtils;

type
  TParam=(psPu,psOhm);

  TGeneral=record
    MaxIterasi,Iterasi:integer;
    Vbase,VKonst,Pbase,PKonst,Zbase,Ibase,Toleransi:double;
    Param:TParam;
    sumGen,sumLoad,sumLoss:TComplex;
  end;

var gGeneral:TGeneral;
    gBus,gBranch,gGen:dArr2;

procedure DecodeData(const rBus,rBranch:dArr2;
  var rGeneral:TGeneral;
  var rNbus,rNsal:integer;
  var rE,rF,rPg,rQg,rPL,rQL,rCap:dArr1;
  var rTyp:iArr1;
  var rR,rX,rLc,rTr,rTu,rSu:dArr2);

function FindSumGen(const rNbus:integer;const rTyp:iArr1):integer;
procedure Admitansi(const rNbus:integer;
  const rR,rX,rLc,rTr,rTu,rSu:dArr2;
  const rCap:dArr1;
  var rG,rB:dArr2);
procedure RecToPolar(const rRecRe,rRecIm:dArr1;
  var rPolRe,rPolIm:dArr1);
procedure PolarToRec(const rPolRe,rPolIm:dArr1;
  var rRecRe,rRecIm:dArr1);
proccdure AliranDaya(const rNbus:integer;const rE,rF:dArr1;
  const rG,rB,rLc:dArr2;
  var rAlirP,rAlirQ:dArr2);
procedure DayaGen(const rNbus:integer;

```

```

SetLength(rPL,rNbus);
SetLength(rQL,rNbus);
SetLength(rCap,rNbus);
SetLength(rTyp,rNbus);
SetLength(rR,rNbus,rNbus);
SetLength(rX,rNbus,rNbus);
SetLength(rLc,rNbus,rNbus);
SetLength(rTr,rNbus,rNbus);
SetLength(rTu,rNbus,rNbus);
SetLength(rSu,rNbus,rNbus);
for i:=0 to rNbus-1 do
begin
  rE[i]:=rBus[i,0];
  rF[i]:=rBus[i,1];
  rPg[i]:=rBus[i,2]/rGeneral.Pbase;
  rQg[i]:=rBus[i,3]/rGeneral.Pbase;
  rPL[i]:=rBus[i,4]/rGeneral.Pbase;
  rQL[i]:=rBus[i,5]/rGeneral.Pbase;
  if rGeneral.Param=psPu then
  begin
    rCap[i]:=rBus[i,6];
  end
  else
  begin
    rCap[i]:=rBus[i,6]/rGeneral.Zbase;
  end;
  rTyp[i]:=round(rBus[i,7]);
  for j:=0 to rNbus-1 do
  begin
    rR[i,j]:=0.0;
    rX[i,j]:=0.0;
    rLc[i,j]:=0.0;
    rTr[i,j]:=0.0;
    rTu[i,j]:=0.0;
    rSu[i,j]:=0.0;
  end;
end;
for i:=0 to rNsal-1 do
begin
  dari:=round(rBranch[i,0])-1;
  ke:=round(rBranch[i,1])-1;
  if rGeneral.Param=psPu then
  begin
    rR[dari,ke]:=rBranch[i,2];
    rX[dari,ke]:=rBranch[i,3];
  end;
end;

```



```

rLc[dari,ke]:=rBranch[i,4];
rR[ke,dari]:=rBranch[i,2];
rX[ke,dari]:=rBranch[i,3];
rLc[ke,dari]:=rBranch[i,4];
end
else if rGeneral.Param=psOhm then
begin
rR[dari,ke]:=rBranch[i,2]/rGeneral.Zbase;
rX[dari,ke]:=rBranch[i,3]/rGeneral.Zbase;
rLc[dari,ke]:=rBranch[i,4]/rGeneral.Zbase;
rR[ke,dari]:=rBranch[i,2]/rGeneral.Zbase;
rX[ke,dari]:=rBranch[i,3]/rGeneral.Zbase;
rLc[ke,dari]:=rBranch[i,4]/rGeneral.Zbase;
end;
rTr[dari,ke]:=rBranch[i,5];
rTu[dari,ke]:=rBranch[i,6];
rSu[dari,ke]:=rBranch[i,7];
end;
end;

```

```

function FindSumGen(const rNbus:integer;
const rTyp:iArr1):integer;
var i:integer;
begin
result:=0;
for i:=0 to rNbus-1 do
begin
if rTyp[i]=2 then
begin
inc(result);
end;
end;
end;
end;

```

```

procedure Admitansi(const rNbus:integer;
const rR,rX,rLc,rTr,rTu,rSu:dArr2;
const rCap:dArr1;
var rG,rB:dArr2);
var i,j,k:integer;
a,b:double;
Cr,Ci:dArr2;
sum,Ca,Za:TComplex;
begin
SetLength(Cr,rNbus,rNbus);
SetLength(Ci,rNbus,rNbus);

```

```

sum:=TComplex.Create(1.0,0.0);
for i:=0 to rNbus-1 do
begin
  for j:=0 to rNbus-1 do
  begin
    if rX[i,j]<>0 then
    begin
      Za:=TComplex.Create(rR[i,j],rX[i,j]);
      Ca:=sum.Divide(Za);
      Cr[i,j]:=Ca.Real;
      Ci[i,j]:=Ca.Imag;
      Ca.Free;
      Za.Free;
    end
    else
    begin
      Cr[i,j]:=0.0;
      Ci[i,j]:=0.0;
    end;
  end;
end;
sum.Free;
SetLength(rG,rNbus,rNbus);
SetLength(rB,rNbus,rNbus);
for i:=0 to rNbus-1 do
begin
  for j:=0 to rNbus-1 do
  begin
    if j=i then
    begin
      rG[i,j]:=0.0;
      rB[i,j]:=0.0;
      for k:=0 to rNbus-1 do
      begin
        rG[i,j]:=rG[i,j]+Cr[i,k];
        rB[i,j]:=rB[i,j]+Ci[i,k]+rLc[i,k];
      end;
    end
    else
    begin
      rG[i,j]:=-Cr[i,j];
      rB[i,j]:=-Ci[i,j];
    end;
  end;
end;
end;

```

```

for i:=0 to rNbus-1 do
begin
  for j:=0 to rNbus-1 do
  begin
    if rTr[i,j]>0 then
    begin
      rG[i,i]:=rG[i,i]-Cr[i,j];
      rB[i,i]:=rB[i,i]-Ci[i,j]-rLc[i,j];
      rG[i,i]:=rG[i,i]+Cr[i,j]/sqr(rTr[i,j]);
      rB[i,i]:=rB[i,i]+Ci[i,j]/sqr(rTr[i,j]);
      rG[j,j]:=rG[j,j]-Cr[i,j];
      rB[j,j]:=rB[j,j]-Ci[i,j]-rLc[i,j];
      rG[j,j]:=rG[j,j]+Cr[i,j];
      rB[j,j]:=rB[j,j]+Ci[i,j];
      rG[i,j]:=-1 * Cr[i,j]/rTr[i,j];
      rB[i,j]:=-1 * Ci[i,j]/rTr[i,j];
      rG[j,i]:=rG[i,j];
      rB[j,i]:=rB[i,j];
      {CLc:=TComplex.Create(0.0,aLc[i,j]);
      result[i,i]:=result[i,i]-Cx[i,j]-CLc;
      result[i,i]:=result[i,i]+Cx[i,j]/sqr(at);
      result[j,j]:=result[j,j]-Cx[i,j]-CLc;
      result[j,j]:=result[j,j]+Cx[i,j];
      result[i,j]:=-Cx[i,j]/at;
      result[j,i]:=result[i,j];
      CLc.Free;}
    end;
  end;
end;
for i:=0 to rNbus-1 do
begin
  for j:=0 to rNbus-1 do
  begin
    if rTu[i,j]>0 then
    begin
      a:=rTu[i,j]*cos(rSu[i,j]);
      b:=rTu[i,j]*sin(rSu[i,j]);
      rG[i,i]:=rG[i,i]-Cr[i,j];
      rB[i,i]:=rB[i,i]-Ci[i,j]-rLc[i,j];
      rG[i,i]:=rG[i,i]+Cr[i,j]/(sqr(a)+sqr(b));
      rB[i,i]:=rB[i,i]+Ci[i,j]/(sqr(a)+sqr(b))+rLc[i,j];
      Za:=TComplex.Create(Cr[i,j],Ci[i,j]);
      Za:=Za.Negative;
      Ca:=TComplex.Create(a,b);
      sum:=Za.Divide(Ca);
    end;
  end;
end;

```

```

    rG[j,i]:=sum.Real;
    rB[j,i]:=sum.Imag;
    sum.Free;
    Ca:=Ca.Conj;
    sum:=Za.Divide(Ca);
    rG[i,j]:=sum.Real;
    rB[i,j]:=sum.Imag;
    sum.Free;
    Ca.Free;
    Za.Free;
    {sUpfr:=TComplex.Create(aUpfr[i,j].tap*cos(aUpfr[i,j].sudut),
        aUpfr[i,j].tap*sin(aUpfr[i,j].sudut));
    CLc:=TComplex.Create(0.0,aLc[i,j]);
    result[i,i]:=result[i,i]-Cx[i,j]-CLc;
    result[i,i]:=result[i,i]+Cx[i,j]/sqr(sUpfr.Abs)+CLc;
    result[i,j]:=-Cx[i,j]/conj(sUpfr);
    result[j,i]:=-Cx[i,j]/sUpfr;
    CLc.Free;
    sUpfr.Free;}
    end;
end;
end;
for i:=0 to rNbus-1 do
begin
    if rCap[i]<>0 then
    begin
        rB[i,i]:=rB[i,i]+rCap[i];
    end;
end;
end;
end;

```

```

procedure RecToPolar(const rRecRe,rRecIm:dArr1;
    var rPolRe,rPolIm:dArr1);
var i:integer;
begin
    if high(rRecRe)<>high(rRecIm) then
    begin
        raise Exception.Create('Dimensi kedua Vector tidak sama!');
    end;
    SetLength(rPolRe,high(rRecRe)+1);
    SetLength(rPolIm,high(rRecRe)+1);
    for i:=0 to high(rRecRe) do
    begin
        rPolRe[i]:=sqrt(sqr(rRecRe[i])+sqr(rRecIm[i]));
        rPolIm[i]:=arctan(rRecIm[i]/rRecRe[i]);
    end;
end;

```

```

end;
end;

procedure PolarToRec(const rPolRe,rPolIm:dArr1;
    var rRecRe,rRecIm:dArr1);
var i:integer;
begin
if high(rRecRe) < high(rRecIm) then
begin
raise Exception.Create('Dimensi kedua Vector tidak sama!');
end;
SetLength(rRecRe,high(rPolRe)+1);
SetLength(rRecIm,high(rPolRe)+1);
for i:=0 to high(rPolRe) do
begin
rRecRe[i]:=rPolRe[i]*cos(rPolIm[i]);
rRecIm[i]:=rPolRe[i]*sin(rPolIm[i]);
end;
end;

procedure AliranDaya(const rNbus:integer;const rE,rF:dArr1;
    const rG,rB,rLc:dArr2;
    var rAlirP,rAlirQ:dArr2);
var i,j:integer;
    Vi,Vj,Ya,Lca,Sa,tmp1,tmp2:TComplex;
begin
SetLength(rAlirP,rNbus,rNbus);
SetLength(rAlirQ,rNbus,rNbus);
for i:=0 to rNbus-1 do
begin
Vi:=TComplex.Create(rE[i],rF[i]);
for j:=0 to rNbus-1 do
begin
rAlirP[i,j]:=0.0;
rAlirQ[i,j]:=0.0;
if j < i then
begin
if rB[i,j] < 0 then
begin
Vj:=TComplex.Create(rE[j],rF[j]);
Ya:=TComplex.Create(rG[i,j],rB[i,j]);
Lca:=TComplex.Create(0.0,rLc[i,j]);
tmp1:=Vi.Conj.Multiply(Vi.Subtract(Vj)).Multiply(Ya.Negative);
tmp2:=Vi.Conj.Multiply(Vi).Multiply(Lca);
Sa:=tmp1.Add(tmp2);

```

```

rAlirP[i,j]:=Sa.Real;
rAlirQ[i,j]:=-Sa.Imag;
//result[i,j]:=conj(aV[i])*(aV[i]-aV[j])*(-aY[i,j])+
//      conj(aV[i])*aV[i]*dLc;
//result[i,j]:=conj(result[i,j]);
Sa.Free;
tmp2.Free;
tmp1.Free;
Lca.Free;
Ya.Free;
Vj.Free;
end;
end;
end;
Vi.Free;
end;
end;

```

```

procedure DayaGen(const rNbus:integer;
  const rE,rF,rPL,rQL:dArr1;
  const rG,rB:dArr2;const rTyp:iArr1;
  var rQg:dArr1);
var i,j:integer;
  sum:double;
begin
  for i:=0 to rNbus-1 do
  begin
    sum:=0.0;
    if rTyp[i]=2 then
    begin
      for j:=0 to rNbus-1 do
      begin
        //sum:=sum+(Fi*(Ej*Gij+Fj*-Bij)-Ei*(Fj*Gij-Ej*-Bij));
        sum:=sum+(rF[i]*(rE[j]*rG[i,j]+rF[j]*-rB[i,j])-
          rE[i]*(rF[j]*rG[i,j]-rE[j]*-rB[i,j]));
      end;
      rQg[i]:=sum+rQL[i];
    end;
  end;
end;
end;

```

```

procedure DayaSlack(const rNbus:integer;
  const rAlirP,rAlirQ:dArr2;
  const rTyp:iArr1;
  const rPL,rQL:dArr1;

```

```

    var rPg,rQg:dArr1);
var ij:integer;
    sumP,sumQ:double;
begin
    for i:=0 to rNbus-1 do
    begin
        if rTyp[i]=1 then
        begin
            sumP:=0.0;
            sumQ:=0.0;
            for j:=0 to rNbus-1 do
            begin
                if rAlirQ[i,j] <> 0 then
                begin
                    sumP:=sumP+rAlirP[i,j];
                    sumQ:=sumQ+rAlirQ[i,j];
                end;
            end;
            rPg[i]:=sumP+rPL[i];
            rQg[i]:=sumQ+rQL[i];
        end;
    end;
end;

procedure ArusBranch(const rNbus:integer;
    const rE,rF:dArr1;
    const rLc,rG,rB:dArr2;
    var rArusRe,rArusIm:dArr2);
var ij:integer;
    Vi,Vj,Ya,xLc,Arus,tmp1,tmp2:TComplex;
begin
    SetLength(rArusRe,rNbus,rNbus);
    SetLength(rArusIm,rNbus,rNbus);
    for i:=0 to rNbus-1 do
    begin
        Vi:=TComplex.Create(rE[i],rF[i]);
        for j:=0 to rNbus-1 do
        begin
            if rB[i,j] <> 0 then
            begin
                Vj:=TComplex.Create(rE[j],rF[j]);
                Ya:=TComplex.Create(rG[i,j],rB[i,j]);
                xLc:=TComplex.Create(0,rLc[i,j]);
                tmp1:=Ya.Negative.Multiply(Vi.Subtract(Vj));
                tmp2:=Vi.Multiply(xLc);
            end;
        end;
    end;
end;

```

```

    Arus:=tmp1.Add(tmp2);
    rArusRe[i,j]:=Arus.Real;
    rArusIm[i,j]:=-Arus.Imag;
    Arus.Free;
    tmp2.Free;
    tmp1.Free;
    xLc.Free;
    Ya.Free;
    Vj.Free;
    //result[i,j]:=(aV[i]-aV[j])*(-aY[i,j])+aV[i]*xLc;
    //result[i,j]:=Conj(result[i,j]);
end
else
begin
    rArusRe[i,j]:=0.0;
    rArusIm[i,j]:=0.0;
end;
end;
Vi.Free;
end;
end;

```

```

procedure Summary(const rNbus:integer;
    const rPg,rQg,rPL,rQL:dArr1;
    var rSumGen,rSumLoad,rSumLoss:TComplex);
var i:integer;
    sumPg,sumPL,sumQg,sumQL:double;
begin
    sumPg:=0.0;
    sumQg:=0.0;
    sumPL:=0.0;
    sumQL:=0.0;
    for i:=0 to rNbus-1 do
    begin
        sumPg:=sumPg+rPg[i];
        sumQg:=sumQg+rQg[i];
        sumPL:=sumPL+rPL[i];
        sumQL:=sumQL+rQL[i];
    end;
    rSumGen:=TComplex.Create(sumPg,sumQg);
    rSumLoad:=TComplex.Create(sumPL,sumQL);
    rSumLoss:=rSumGen.Subtract(rSumLoad);
end;

```

```

function MaxArray(const rData:dArr1):double;

```

```

var i:integer;
begin
  result:=abs(rData[0]);
  for i:=1 to high(rData) do
  begin
    if result<abs(rData[i]) then
    begin
      result:=abs(rData[i]);
    end;
  end;
end;

procedure UpdateBusGen(const rNbus:integer;
  const rE,rF,rPg,rQg:dArr1;
  const rGeneral:TGeneral;
  var rBus:dArr2);
var i:integer;
begin
  for i:=0 to rNbus-1 do
  begin
    rBus[i,0]:=rE[i];
    rBus[i,1]:=rF[i];
    rBus[i,2]:=rPg[i]*rGeneral.Pbase;
    rBus[i,3]:=rQg[i]*rGeneral.Pbase;
  end;
end;

procedure UpdateBranch(const rNsal:integer;
  const rAlirRe,rAlirIm,rArusRe,rArusIm:dArr2;
  const rGeneral:TGeneral;
  var rBranch:dArr2);
var i,dari,ke:integer;
begin
  for i:=0 to rNsal-1 do
  begin
    dari:=round(rBranch[i,0])-1;
    ke:=round(rBranch[i,1])-1;
    rBranch[i,9]:=rAlirRe[dari,ke]*rGeneral.Pbase;
    rBranch[i,10]:=rAlirIm[dari,ke]*rGeneral.Pbase;
    rBranch[i,11]:=rAlirRe[ke,dari]*rGeneral.Pbase;
    rBranch[i,12]:=rAlirIm[ke,dari]*rGeneral.Pbase;
    rBranch[i,13]:=rArusRe[dari,ke]*rGeneral.Ibase;
    rBranch[i,14]:=rArusIm[dari,ke]*rGeneral.Ibase;
    rBranch[i,15]:=rArusRe[ke,dari]*rGeneral.Ibase;
    rBranch[i,16]:=rArusIm[ke,dari]*rGeneral.Ibase;
  end;
end;

```

end;
end;
end.



unit uFitness;

interface

uses uUtils,uObjFunc,uRandom;

```
function CalcFitness(var aChrom:bArr1):double;overload;  
function CalcFitness(var aChrom:bArr2):double;overload;  
function CalcFitness(var aChrom:dArr1):double;overload;  
function CalcFitness(var aChrom:dArr2):double;overload;  
function CalcFitness(var aChrom:TChromEP1):double;overload;  
function CalcFitness(var aChrom:TChrom1):double;overload;  
function MutasiEvoPro(const rSfun,rSmin,rBeta,rValue:double;  
    var rRand:TRandomu):double;overload;  
function MutasiEvoPro(const rSfun,rSmin,rBeta,rValue:double;  
    const ip:integer;  
    var rRand:TRandomu):double;overload;  
function MutasiEvoPro(const rSfun,rSmin,rBeta,rValue:double;  
    const ip,jp:integer;  
    var rRand:TRandomu):double;overload;
```

implementation

```
function CalcFitness(var aChrom:bArr1):double;  
var aX:double;  
begin  
    aX:=DecodeBinToFloat1(aChrom);  
    //insert code obj function here  
    result:=aX;  
end;
```

```
function CalcFitness(var aChrom:bArr2):double;  
var aX:dArr1;  
begin  
    SetLength(aX,high(aChrom));  
    aX:=DecodeBinToFloat2(aChrom);  
    //insert code obj function here  
    result:=aX[1];  
end;
```

```
function CalcFitness(var aChrom:dArr1):double;  
begin  
    //insert code obj function here  
    result:=aChrom[1];  
end;
```