

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



**OPTIMASI DAYA REAKTIF FUNGSI NON
LINIER MENGGUNAKAN METODE
EVOLUTIONARY PROGRAMMING AND
QUASY - SIMPLEX (EPQS) PADA SALURAN
TRANSMISI 150 Kv SUB SISTEM PAITON - BALI**

SKRIPSI

**Disusun Oleh :
DIAN ATHO'ILLAH
01.12.145**

MARET 2007



LEMBAR PERSETUJUAN

OPTIMASI DAYA REKTIF FUNGSI NON LINIER MENGUNAKAN METODE EVOLUTIONARY PROGRAMMING AND QUASY – SIMPLEX PADA SALURAN TRANSMISI 150 Kv SUB SISTEM PAITON -BALI

SKRIPSI

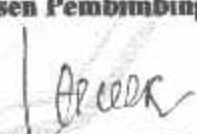
*Disusun Guna Melengkapi dan Memenuhi Syarat-Syarat
Untuk Mencapai Gelar Sarjana Teknik Energi Listrik*

Disusun Oleh :
DIAN ATHO'ILLAH
01.12.145



Mengetahui,
Ketua Jurusan Teknik Elektro (S-1)

Ir. F. Yudi Limpraptono, MT
NIP.Y.103 95 00274

Diperiksa dan Disetujui,
Dosen Pembimbing

Ir. H. CHOIRI
NIP.130703042

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

ABSTRAKSI

OPTIMASI DAYA REAKTIF FUNGSI NON LINIER MENGUNAKAN METODE EVOLUTIONARY PROGRAMMING AND QUASY – SIMPLEX (EPQS) PADA SALURAN TRANSMISI 150 Kv SUB SISTEM PAITON - BALI

(Dian Atho'illah, Nim. 01.12.145, Teknik Elektro Energi Listrik S-1)
(Dosen Pembimbing : Ir.H Choiri)

Kata Kunci: *Evolutionary Programming, Gaussian Mutation, Cauchy Mutation, Quasy-Simplex, Parallel search Algorithm, Non Linier Prigramming.*

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 dengan menggunakan metode *Evolutionary Programming and Quasy-Simplex*. 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.61,453,870.52 atau terjadi optimasi biaya sebesar 24,77%.

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 FUNGSI NON LINIER
MENGUNAKAN METODE EVOLUTIONARY PROGRAMMING AND
QUASY – SIMPLEX (EPQS) PADA SALURAN TRANSMISI 150 kV SUB-
SISTEM PAITON-BALI”.

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

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

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

1. Bapak Prof. Dr. Ir. Abraham Lomi, MSEE, Selaku Rektor Institut Teknologi Nasional Malang
2. Bapak Ir. Mochtar Asroni, MT, Selaku Dekan Fakultas Teknik Industri Institut Teknologi Nasional Malang.
3. Bapak Ir. F Yudi Limpraptono, MT, selaku Ketua Jurusan Teknik Elektro ITN Malang.
4. Bapak H.Ir. CHOIRI, selaku Dosen Pembimbing.

5. Bapak Ir. Totok Prawiro, selaku pembimbing lapangan di PT. PLN (Persero) P3R Jawa-Bali.
6. Bapak Dan Ibu Dosen Teknik Elektro Energi Listrik.
7. Teman Teman Teknik Elektro Energi Listrik, terima Kasih Atas Bantuan Kalian Semua, Serta Semua Pihak Yang turut Membantu Atas Penyelesaian Skripsi Ini.

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

Malang, Maret 2007

Penyusun

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PERSETUJUAN	ii
ABSTRAKSI	iii
KATA PENGANTAR.....	iv
DAFTAR ISI	vi
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
DAFTAR GRAFIK.....	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan	3
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>)	12
2.3.3.	Daya semu (<i>Apparent Power</i>)	12
2.4.	Sistem Per-Unit.....	13
2.4.1.	Mengubah Dasar Sistem Per-Unit	15
2.5.	Sistem Operasi Pada Sistem Tenaga Listrik	16
2.6.	Karakteristik Unit Pembangkit	19
2.6.1.	Karakteristik <i>Input-Output</i>	19
2.6.2.	Karakteristik <i>Heat Rate</i>	21
2.6.3.	Karakteristik <i>Incremental Heat-Rate</i> dan <i>Incremental Fuel Cost</i>	22
2.7.	<i>Economic Dispatch</i>	23
2.7.1.	Fungsi Biaya Bahan Bakar	23
2.7.2.	Fungsi Biaya Daya Reaktif Pembangkit	24
2.7.3.	<i>Economic Dispatch</i> Dengan Mengabaikan Rugi-rugi Transmisi.....	25
2.7.4	<i>Economic Dispatch</i> Dengan Memperhitungkan Rugi-Rugi Transmisi ..	28

BAB III. OPTIMASI DAYA REAKTIF MENGGUNAKAN METODE

EVOLUTIONARY PROGRAMMING AND QUASY - SIMPLEX

(EPQS)..... 30

3.1.	Analisa Aliran Daya	30
3.1.1.	Klasifikasi Bus	31

3.1.2. Metode Newton Raphson	32
3.2. Evolutionary Programming and Quasy - Simplex	34
3.2.1. Evolutionary Programming (EP)	34
3.2.2. Quasy – Simplex(QS)	35
3.3. Parameter Evolutionary Programming and Quasy - Simplex	36
3.3.1. Jumlah Generasi (MAXGEN)	36
3.3.2. Ukuran Populasi (POPSIZE)	37
3.3.3. Probabilitas Crossover (Pc)	38
3.3.4. Probabilitas Mutasi (Pm)	39
3.3.5. Panjang Kromosom (NVAR)	39
3.4. Mekanisme Evolutionary Programming and Quasy - Simplex	39
3.4.1. Pengkodean atau Representasi	39
3.4.2. Fungsi Fitness (Fungsi Evaluasi)	41
3.4.3. Seleksi dan Reproduksi	42
3.4.3.1. Roulette Wheel Selection	42
3.4.3.2. Rank Selection	43
3.4.4. Crossover (Pindah Silang)	44
3.4.5. Mutation (Mutasi)	46
3.4.6. Recombination (Rekombinasi)	47
3.4.7. Competition (Kompetisi)	48
3.4.8. Elitism	48
3.5. Formulasi Masalah Optimasi Daya Reaktif	49
3.6. Adaptasi EPQS Ke Masalah Optimasi Daya Reaktif	50
3.6.1. Representasi Solusi	51

3.6.2. <i>Initialization</i>	51
3.6.3. <i>Evaluasi Fitness</i>	52
3.6.4. <i>Selection dan Reproduction</i>	52
3.6.5. <i>Crossover</i>	53
3.6.6. <i>Statistics</i> (Statistik)	53
3.6.7. <i>Mutation</i>	53
3.6.7.1. <i>Evolutionary Programming</i>	53
3.6.7.2. <i>Quasy - Simplex</i>	54
3.6.8. <i>Competition</i> (Kompetisi)	54
3.6.9. <i>Convergence Test</i> (Konvergensi)	55
 BAB IV. HASIL DAN PEMBAHASAN	 56
4.1. Program Komputer Optimasi Daya Reaktif Menggunakan Metode <i>Evolutionary Programming and Quasy - Simplex</i>	56
4.1.1. Algoritma Program	56
4.1.2. Algoritma Aliran Daya Dengan <i>Newton Rapshon</i>	59
4.2. Validasi Data IEEE 30 Untuk Menyelesaikan Permasalahan Opti- masi Daya Reaktif Menggunakan Metode <i>Evolutionary Programming</i> <i>and Quasy - Simplex</i>	61
4.2.1. Hasil Validasi IEEE 30 Dengan Menggunakan Metode <i>EPQS</i>	63
4.3. Data Pembangkitan Thermal Pada Sub-Sistem Paiton-Bali	65
4.4. Data Pembangkitan dan Pembebanan 150 kV Sub-Sistem Paiton-Bali ,	68
4.5. Data Saluran Transmisi 150 kV Sub-Sistem Paiton-Bali.....	70
4.6. Prosedur Pelaksanaan Program Perhitungan	72

4.7.	Hasil dan Analisis Hasil Perhitungan Optimasi Daya Reaktif- Menggunakan Metode <i>EPQS</i> Pada Saluran Transmisi 150 kV Sub-Sistem Paiton-Bali	79
4.7.1.	Hasil Perhitungan Sebelum Optimasi	79
4.7.2.	Hasil Perhitungan Setelah Optimasi	82
4.8.	Perbandingan Hasil Perhitungan Sebelum dan Setelah Optimasi Daya Reaktif Menggunakan Metode <i>Evolutionary Programming and Quasy - Simplex</i>	85
4.8.1.	Tingkat Optimum Biaya Pembangkitan	85
4.8.2.	Perbandingan Tingkat Tegangan dan Sudut Tegangan Pada Tiap Bus .	86
4.8.3.	Perbandingan Tingkat Rugi-rugi Daya Pada Saluran	87
BAB V KESIMPULAN DAN SARAN		88
5.1.	Kesimpulan.....	88
5.2.	Saran.....	89
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	13
Gambar 2.6.	Unit Boiler-Turbin-Generator	20
Gambar 2.7.	Kurva Karakteristik <i>Input-Output</i> Pembangkit Thermal	21
Gambar 2.8.	Kurva Karakteristik <i>Heat-Rate</i> Unit Pembangkit	21
Gambar 2.9.	Kurva Karakteristik <i>Incremental Heat-Rate/Fuel Cost</i>	22
Gambar 2.10.	N Unit Pembangkit Thermal Melayani Beban P_R	25
Gambar 2.11.	N buah Pembangkit Thermal Melayani Beban P_R Melalui Saluran Transmisi	28
Gambar 3.1.	Jumlah Generasi	37
Gambar 3.2	Ukuran Populasi	38
Gambar 3.3	Pengkodean Representasi	40
Gambar 3.4	<i>Roulette-Wheel</i>	43
Gambar 3.5	Ilustrasi Operator dengan <i>One Point Crossover</i>	44
Gambar 3.6	Ilustrasi Operator dengan <i>Two Point Crossover</i>	45
Gambar 3.7	Ilustrasi Operator Crossover dengan <i>uniform crossover</i>	45
Gambar 3.8	Ilustrasi Operator Mutasi	46
Gambar 3.9	<i>Recombination</i> dari Induk (<i>parent</i>) a) & b) menjadi Anak (<i>offspring</i>) c)	47

Gambar 3.10	Pembentukan Next Generation dalam Genetic Algorithm	49
Gambar 4.1.	Flowchart Program Dasar Permasalahan	59
Gambar 4.2.	Flowchart Program Menggunakan EPQS	59
Gambar 4.3.	Flowchart Perhitungan Aliran Daya Metode Newton Rapshon .	60
Gambar 4.4.	Tampilan Parameter Validasi IEEE 30	63
Gambar 4.5.	Hasil Load Flow	63
Gambar 4.6.	Tampilan Hasil Validasi IEEE 30 Hasil Perhitungan Biaya Pembangkitan	64
Gambar 4.7.	Diagram Segaris Jaringan Sistem Transmisi Listrik 150 kV Sub-Sistem Paiton-Bali	67
Gambar 4.8.	Tampilan Utama Program	72
Gambar 4.9.	Tampilan Masukan Data	72
Gambar 4.10.	Tampilan Data Bus	73
Gambar 4.11.	Tampilan Data Saluran	73
Gambar 4.12.	Tampilan Data Generator	74
Gambar 4.13.	Tampilan Parameter Algoritma Genetika Pada kondisi awal (sebelum Optimasi)	74
Gambar 4.14.	Tampilan Hasil Load flow Pada Kondisi Awal (Sebelum Optimasi)	75
Gambar 4.15.	Tampilan Hasil Aliran Daya Pd Kondisi Awal (Sblm Optimasi)	75
Gambar 4.16.	Tampilan summary Load Flow Sebelum Optimasi.....	76
Gambar 4.17.	Tampilan Hasil Prog. Pd Kondisi Akhir (setelah optimas)	76

Gambar 4.18. Tampilan Hasil <i>Load Flow</i> Pada Kondisi Akhir (Setelah Optimasi)	77
Gambar 4.19. Tampilan Aliran Daya Pada Kondisi Akhir (Setelah Optimasi) ..	77
Gambar 4.20. Tampilan <i>Summary Load Flow</i> Setelah Optimasi.....	78
Gambar 4.21. Tampilan Tegangan Sebelum dan Setelah Optimasi	78

DAFTAR TABEL

Tabel 4.1.	Data Generator Dan Koefisien Biaya IEEE 30	61
Tabel 4.2.	Data Bus IEEE 30	61
Tabel 4.3.	Saluran Impedansi IEEE 30	62
Tabel 4.4.	Perbandingan Hasil Data Referensi Jurnal Dengan Data Optimasi	64
Tabel 4.5.	Parameter Unit Pembangkit Thermal	65
Tabel 4.6.	Persamaan Biaya Pembangkitan Unit Pembangkit Thermal Paiton - Bali	66
Tabel 4.7.	Penomoran Bus 150 kV Sub - Sistem Paiton - Bali	68
Tabel 4.8.	Data Pembangkitan dan Pembebanan 150 kV Sub - Sistem Paiton - Bali.....	69
Tabel 4.9.	Data Saluran Transmisi 150 kV Sub - Sistem Paiton - Bali	71
Tabel 4.10.	Hasil Perhitungan Tegangan, Sudut Tegangan, Pembangkitan dan Pembebanan Sebelum Optimasi	79
Tabel 4.11.	Hasil Perhitungan Aliran Daya Antar Saluran Sebelum Optimasi	80
Tabel 4.12.	Total Pembangkitan, Pembebanan dan Rugi-rugi Saluran Sebelum Optimasi	81
Tabel 4.13.	Hasil Perhitungan Daya Yang Dibangkitkan dan Biaya Operasi Sebelum Optimasi	81
Tabel 4.14.	Hasil Perhitungan Tegangan, Sudut Tegangan, Pembangkitan dan Pembebanan Setelah Optimasi	82
Tabel 4.15.	Hasil Perhitungan Aliran Daya Antar Saluran Setelah Optimasi .	83

Tabel 4.16. Total Peembangkitan, Pembebanan dan Rugi-rugi Saluran	
Setelah Optimasi	84
Tabel 4.17. Hasil Perhitungan Daya Yang Dibangkitkan dan Biaya	
Operasi Setelah Optimasi	84
Tabel 4.18. Perbandingan Tingkat Optimum Biaya Pembangkitan	85
Tabel 4.19. Perbandingan Tingkat Tegangan dan Sudut Tegangan Pada	
Tiap Bus.....	86

DAFTAR GRAFIK

Grafik 4.1. Hasil Perhitungan Daya Yang Dibangkitkan Sebelum Optimasi .	82
Grafik 4.2. Hasil Perhitungan Daya Yang Dibangkitkan Setelah Optimasi	84
Grafik 4.3. Perbandingan Total Pembangkitan, Pembebanan dan Rugi-rugi Saluran Sebelum dan Setelah Optimasi	87

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. 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 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 dengan biaya investasi yang murah, sehingga mampu memperbaiki profil tegangan sistem. Permasalahan besarnya rugi-rugi di saluran telah menjadi perhatian utama khususnya pada saluran transmisi 150 KV.

Masalah-masalah diatas dapat diatasi dan dipecahkan salah satunya dengan menggunakan metode kombinasi *Evolutionary programming* dan *Quasy* -

Simplex. Dengan metode *Evolutionary Programming* dan *Quasy - Simplex* ini akan mengoptimalkan penyaluran daya reaktif untuk mengurangi rugi daya dengan meminimalkan biaya pengoperasian pada jaringan transmisi 150 kV Sub-Sistem Piton-Bali, sehingga di dapat kualitas suatu sistem tenaga listrik yang baik.

1.2. Rumusan Masalah

Berdasarkan latar belakang tersebut diatas permasalahan yang timbul :

- Berapa besar profil tegangan setiap bus sebelum dan sesudah dilakukan optimasi daya reaktif.
- Berapa jumlah pengoptimalan penyaluran daya reaktif sebelum dan sesudah dilakukan optimasi sehingga dapat mengurangi rugi – rugi daya.
- Berapa besar biaya operasi sebelum dan sesudah dilakukan optimasi pada saluran tranmisi 150 Kv sub Piton – Bali.

Berdasarkan pada deskripsi permasalahan dan latar belakang tersebut diatas maka skripsi ini berjudul :

**“OPTIMASI DAYA REAKTIF FUNGSI NON LINIER
MENGUNAKAN METODE EVOLUTIONARY
PROGRAMMING AND QUASY – SIMPLEX (EPQS)
PADA SALURAN TRANSMISI 150 Kv SUB SISTEM
PAITON - BALI”**

1.3. Tujuan

Tujuan dari penelitian skripsi ini adalah untuk memperbaiki profil tegangan dan mengurangi rugi-rugi daya dengan biaya pengoperasian yang murah pada Saluran Transmisi 150 kV Sub-Sistem Paiton-Bali dengan cara mengoptimalkan biaya operasional dengan menggunakan metode *Evolutionary Programming dan Quasy – Simplex*.

1.4. Batasan Masalah

Agar permasalahan mengarah sesuai dengan tujuan, maka pembahasan skripsi ini dibatasi sebagai berikut:

1. Analisis perhitungan optimasi daya reaktif pada saluran transmisi 150 kV Sub-Sistem Paiton-Bali menggunakan metode *Evolutionary Programming dan Quasy - Simplex*
2. Analisa dilakukan dengan asumsi bahwa sistem berada dalam operasi normal.
3. Analisa perhitungan pada pembangkit thermal saja.
4. Tidak membahas kabel laut.
5. Tidak membahas masalah peralatan kompensasi dan penempatannya
6. Tidak membahas masalah kontrol tegangan.

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 obyek penelitian yakni data impedansi saluran transmisi dan data pembebanan saluran transmisi 150 kV Sub-Sistem Palton-Bali.
3. Menganalisa dan menghitung besar tegangan, sudut fasa tegangan, aliran daya pada tiap saluran serta biaya pembangkitan menggunakan metode *Evolutionary Programming dan Quasy – Simplex*.
4. Simulasi dan pembahasan masalah
Simulasi dan pembahasan masalah dengan menggunakan bahasa pemrograman Borland Delphi versi 07.
5. Menarik kesimpulan.

1.6. Sistematika Penulisan

Adapun sistematika pembahasan pada skripsi ini adalah sebagai berikut :

BAB I : PENDAHULUAN

Menguraikan latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, metodologi 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 AND QUASY - SIMPLEX

Menguraikan teori dasar dari aliran daya, metode aliran daya *Newton Raphson*, teori *EPQS*, dan adaptasi *EPQS* ke permasalahan.

BAB IV : HASIL DAN ANALISIS HASIL

Menguraikan alur program, serta hasil perhitungan menggunakan metode *EVOLUTIONARY PROGRAMMING AND QUASY - SIMPLEX (EPQS)*

BAB V : KESIMPULAN DAN SARAN

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

1.7. Kontribusi Penelitian

Adapun kontribusi dari skripsi ini adalah diharapkan penggunaan dari metode *Evolutionary Programming dan Quasy - Simplex* dalam menganalisa penyaluran daya reaktif sehingga dapat menekan rugi-rugi daya dengan biaya operasional 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^{1,31}

Tiga bagian utama sistem tenaga listrik yaitu : pusat pembangkit tenaga listrik, saluran transmisi dan 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).

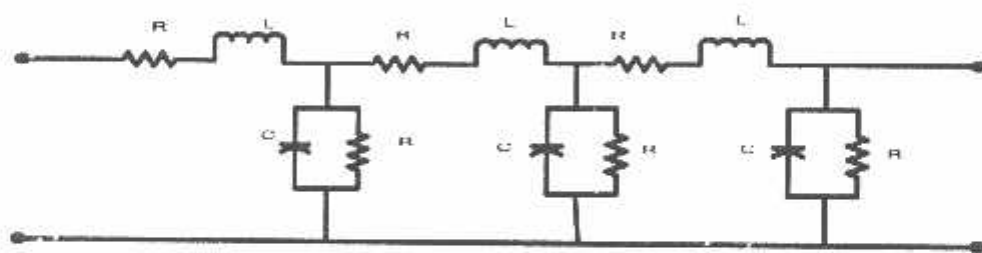
Resistansi umumnya tergantung pada jenis penghantar sedangkan konduktansi menyatakan besarnya arus bocor antar penghantar, antar penghantar dengan tanah, tetapi harganya relatif kecil maka dapat diabaikan. Induktansi adalah parameter rangkaian yang menghubungkan tegangan yang diimbaskan oleh perubahan 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

penghantar-penghantar dari saluran fasa tunggal atau diantara penghantar dengan netral dari suatu saluran berfasa tiga membentuk admitansi paralel.

2.2 Saluran Transmisi^[3]

Tenaga listrik yang dibangkitkan disalurkan melalui saluran transmisi. Saluran-saluran transmisi ini membawa tenaga listrik dari pusat-pusat tenaga listrik ke pusat-pusat beban. Suatu saluran transmisi tenaga listrik mempunyai empat parameter yang mempengaruhi kemampuan untuk berfungsi sebagai bagian dari sistem tenaga, yaitu resistansi, induktansi, kapasitansi, dan konduktansi. Keempat parameter saluran transmisi tersebut merata disepanjang saluran transmisi. Parameter-parameter tersebut sangat berpengaruh terhadap tegangan bus dan aliran daya yang mengalir pada saluran tersebut.



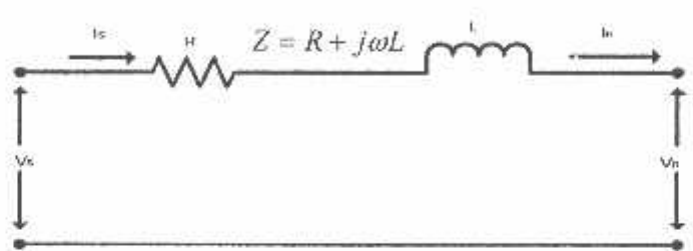
Gambar 2.1
Rangkaian Setara Saluran Transmisi^[3]
Sumber : Stevenson, William D, Jr, 1996 : 86

Menurut panjangnya, saluran transmisi dapat diklasifikasikan menjadi 3 golongan (Stevenson, William D, Jr, 1996 : 86), yaitu:

1. Saluran transmisi pendek, adalah saluran yang panjangnya < 80 km.
2. Saluran transmisi menengah, adalah saluran yang panjangnya 80-240 km.
3. Saluran transmisi panjang, adalah saluran yang panjangnya > 240 km.

2.2.1 Saluran Transmisi Pendek^[3]

Rangkaian ekivalen suatu saluran transmisi pendek diperlihatkan pada gambar 2.2 dimana I_S dan I_R merupakan arus pada ujung pengiriman dan ujung penerimaan, sedangkan V_S dan V_R adalah tegangan saluran terhadap netral pada ujung pengiriman dan ujung penerimaan.



Gambar 2.2
Rangkaian Setara Saluran Transmisi Pendek^[3]
Sumber : Stevenson, William D, Jr, 1996 : 87

Karena tidak terdapat cabang paralel (shunt), arus pada ujung-ujung pengirim dan penerima akan sama besarnya (Stevenson, William D, Jr, 1996 : 87)

: $I_S = I_R$ (2.1)

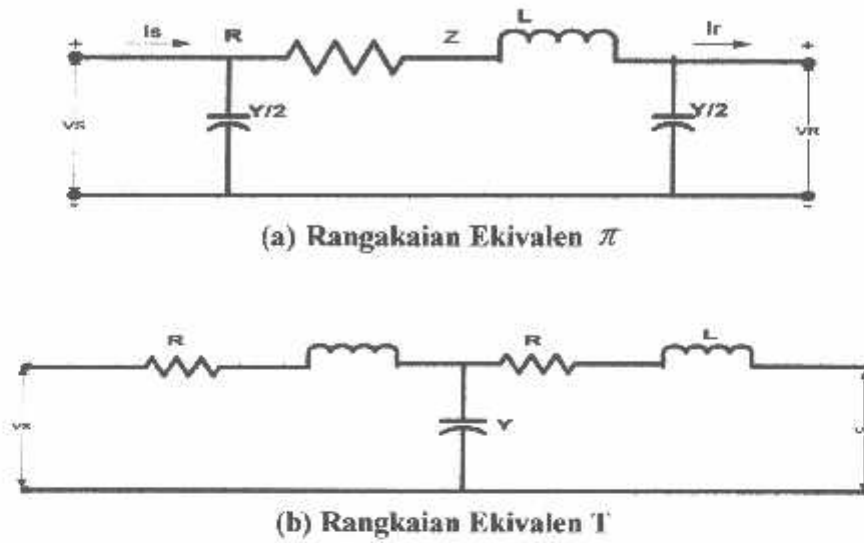
Bila kondisi tegangan pada ujung penerima diketahui, maka tegangan pada ujung sisi kirim adalah (Stevenson, William D, Jr, 1996 : 87) :

$V_S = V_R + I_R Z$ (2.2)

2.2.2 Saluran Transmisi Menengah^[3]

Bertambahnya saluran menyebabkan kapasitansi shunt bertambah besar dan tidak dapat diabaikan. Saluran transmisi jarak menengah pada umumnya digambarkan dengan rangkaian π atau rangkaian T, dapat dilihat pada gambar

2.3(a) dan gambar 2.3(b). Dari dua versi ini rangkaian π lebih umum dipakai dari pada rangkaian T.



Gambar 2.3.
Rangkaian Setara Saluran Transmisi Menengah^[3]
 Sumber : *Stevenson, William D, Jr, 1996 : 89*

0

Untuk rangkaian π berlaku (*Stevenson, William D, Jr, 1996 : 89*) :

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

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

Untuk rangkaian T berlaku (*Stevenson, William D, Jr, 1996 : 89*) :

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

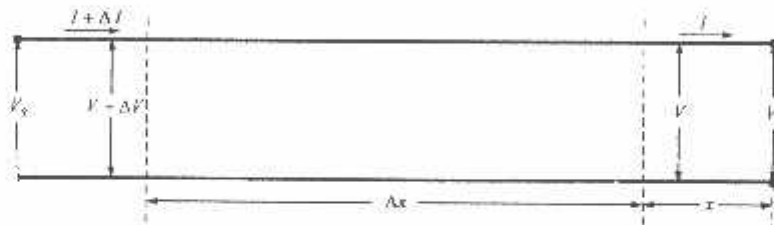
$$I_s = \left[\frac{ZY}{4} + 1 \right] Y V_R + \left[\frac{ZY}{2} + 1 \right] I_R \dots\dots\dots (2.6.)$$

dimana :

- V_S dan I_S = Tegangan dan Arus sisi kirim
- V_R dan I_R = Tegangan dan Arus sisi terima
- Z = Impedansi seri total saluran transmisi
- Y = Admitansi shunt total saluran

2.2.3 Saluran Transmisi Panjang^[3]

Pada saluran panjang parameter-parameter saluran tidak terpusat menjadi satu, melainkan tersebar merata diseluruh panjang saluran, dapat dilihat pada gambar 2.4.



Gambar 2.4.

Diagram Skema Saluran Transmisi Panjang^[3]

Sumber : Stevenson, William D, Jr, 1996 : 90

Persamaan tegangan dan arus pada setiap titik sepanjang saluran transmisi dengan jarak x dari ujung sisi terima dapat ditulis sebagai berikut (Stevenson, William D, Jr, 1996 : 92) :

$$V = \frac{V_R + I_R Z_0}{2} e^{yx} + \frac{V_R - I_R Z_0}{2} e^{-yx} \dots\dots\dots(2.7)$$

$$I = \frac{V_R + I_R}{Z_0} e^{yx} + \frac{V_R - I_R}{Z_0} e^{-yx} \dots\dots\dots(2.8)$$

Persamaan untuk saluran transmisi panjang dapat ditulis dalam bentuk hiperbola sebagai berikut (Stevenson, William D, Jr, 1996 : 95) :

$$V_{(x)} = V_R \cosh \gamma x + I_R Z_c \sinh \gamma x \dots\dots\dots(2.9)$$

$$I_{(x)} = I_R \cosh \gamma x + \frac{V_R}{Z_c} \sinh \gamma x \dots\dots\dots (2.10)$$

dimana :

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

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

2.3 Daya Dalam Sistem Tenaga Listrik^[3]

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

2.3.1 Daya aktif (*Active Power*)^[3]

Secara umum daya aktif dinyatakan oleh persamaan (Stevenson, William D, Jr, 1996 : 16) :

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

dimana :

V dan I nilai efektifnya.

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

2.3.2 Daya Reaktif (*Reactive Power*)^[3]

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

Persamaan daya reaktif adalah(Stevenson, William D, Jr, 1996 : 17) :

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

Dua macam daya reaktif yaitu : daya reaktif kapasitif dan daya reaktif induktif, 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.

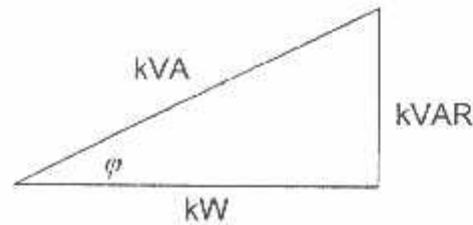
Daya reaktif induktif adalah daya listrik yang dibutuhkan untuk menghasilkan medan magnet yang dibutuhkan oleh alat-alat seperti motor induksi, transformator dan sebagainya.

2.3.3 Daya Semu (*Apparent Power*)^[3]

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(Stevenson, Willism D, Jr, 1996 : 17) :

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



Gambar 2.5.
Representasi Segitiga Daya^[3]
 Sumber : *Stevenson, William D. Jr, 1996 : 19*

dimana :

$\cos \varphi$ = Faktor daya

kW = Daya aktif (P)

kVA = Daya semu (S)

kVAR = Daya reaktif (Q)

2.4 Sistem Per-Unit^[3]

Untuk memudahkan proses perhitungan, dalam sistem tenaga listrik digunakan sistem Per-Unit (pu) (*Stevenson, Willism D, Jr, 1996 : 29*) :

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

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

➤ Untuk data 1 phasa (*Stevenson, William D, Jr, 1996 : 29*)

Arus dasar

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

Impedansi dasar

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

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

➤ Untuk data 3 fasa (*Stevenson, William D, Jr, 1996 : 31*)

Arus dasar

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

Impedansi dasar

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

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

 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 (*Stevenson, William D, Jr, 1996 : 30*) :

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

2.4.1 Mengubah Dasar Sistem Per-Unit^[3]

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 (Stevenson, William D, Jr, 1996 : 32):

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

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 (Stevenson, William D, Jr, 1996 : 32):

$$Z_{\text{baru per-unit}} = Z_{\text{diberikan per-unit}} \left(\frac{\text{kV}_{\text{diberikan dasar}}}{\text{kV}_{\text{baru dasar}}} \right)^2 \times \left(\frac{\text{MVA}_{\text{baru dasar}}}{\text{MVA}_{\text{diberikan dasar}}} \right) \dots\dots\dots (2.23)$$

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^[21]

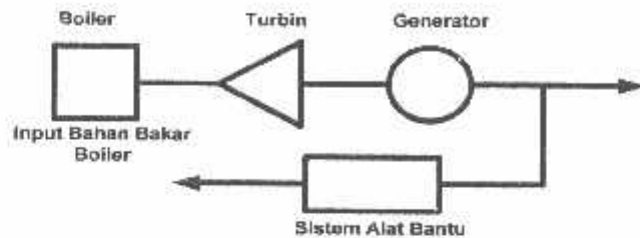
2.6.1 Karakteristik *Input-Output*^[21]

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 *ouiput* 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(Wood, Allan J and B. F. Wollenberg, 1996 : 8) :

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

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



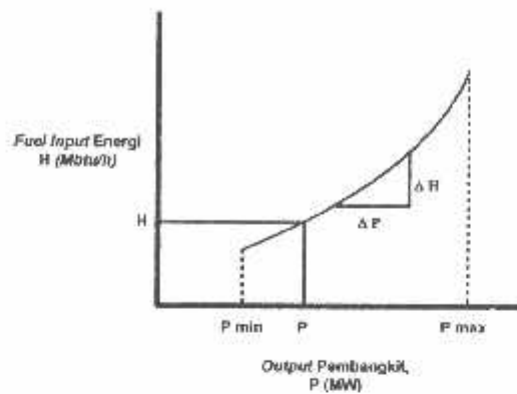
Gambar 2.6.
Unit Boiler-Turbin-Generator^[2]
 Sumber : Wood, Allan J and B. F. Wollenberg, 1996 : 9

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(Wood, Allan J and B. F. Wollenberg, 1996 : 10).

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

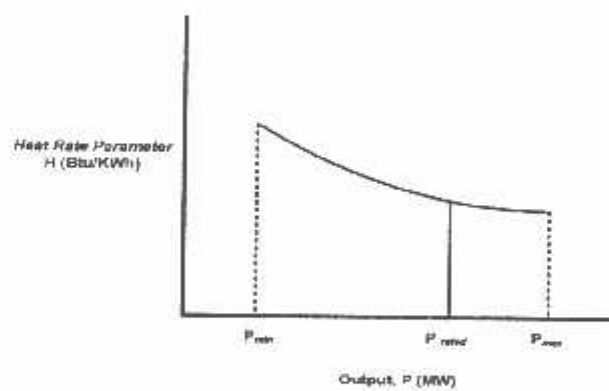
$$F = f(P) \dots\dots\dots (2.26)$$



Gambar 2.7.
Kurva Karakteristik *Input-Output* Pembangkit Thermal^[2]
Sumber : Wood, Allan J and B. F. Wollenberg, 1996 : 9

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.
Kurva Karakteristik *Heat-Rate* Unit Pembangkit^[2]
Sumber : Wood, Allan J and B. F. Wollenberg, 1996 : 11

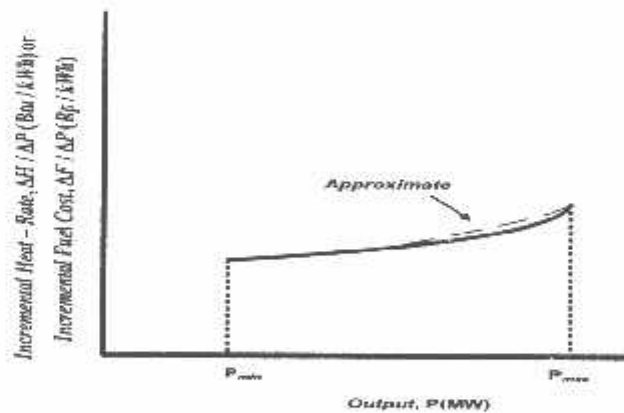
2.6.3 Karakteristik *Incremental Heat-Rate* dan *Incremental Fuel Cost*^[2]

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.) (Wood, Allan J and B. F. Wollenberg, 1996 : 10).

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

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



Gambar 2.9,
Kurva Karakteristik *Incremental Heat-Rate/Fuel Cost*^[2]
Sumber : Wood, Allan J and B. F. Wollenberg, 1996 : 10

Bila harga Δ sangat kecil maka dapat dinyatakan dengan persamaan berikut ini (Wood, Allan J and B. F. Wollenberg, 1996 : 10):

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

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

2.7 Economic Dispatch^[2]

Dalam pembahasan tentang *Optimal Power Flow* 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 (Kwang, Y. Lee, and Frank F. Yang 1998 : 103):

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

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

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 kuadratik.

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^[1]

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(Kwang. Y. Lee, and Frank F. Yang 1998 : 104):

$$C_i(Q_i) = \sum_{j \in N_R} \{C_{ji} | \overline{Q}_{ji} | \} \dots\dots\dots (2.33.)$$

dimana:

- $C_i(Q_i)$ = Biaya daya reaktif pembangkit
- C_{ji} = Biaya pengoperasian daya reaktif

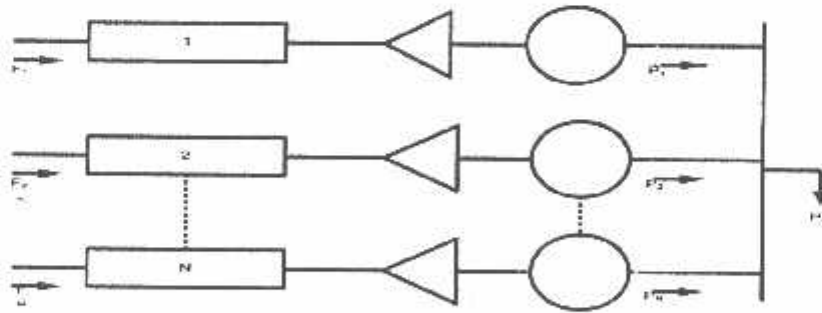
$\bar{Q}_{pi}$ = Besar daya reaktif yang terbangkit pada unit generator i

Biaya pengoperasian disini tidak 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.
N Unit Pembangkit Thermal Melayani Beban P_R ^[2]
Sumber ; Wood, Allan J and B. F. Wollenberg, 1996 : 30

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(Wood, Allan J and B. F. Wollenberg, 1996 : 29) :

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

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

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

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(Wood, Allan J and B. F. Wollenberg, 1996 : 30) :

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

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(Wood, Allan J and B. F. Wollenberg, 1996 : 35).

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

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

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

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

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

Persamaan terakhir ini menunjukkan bahwa bila digunakan biaya bahan bakar F_i 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 :

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

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

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

Dari batasan pertidaksamaan pembatas di atas dapat diperluas menjadi(Wood, Allan J and B. F. Wollenberg, 1996 : 31) :

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

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

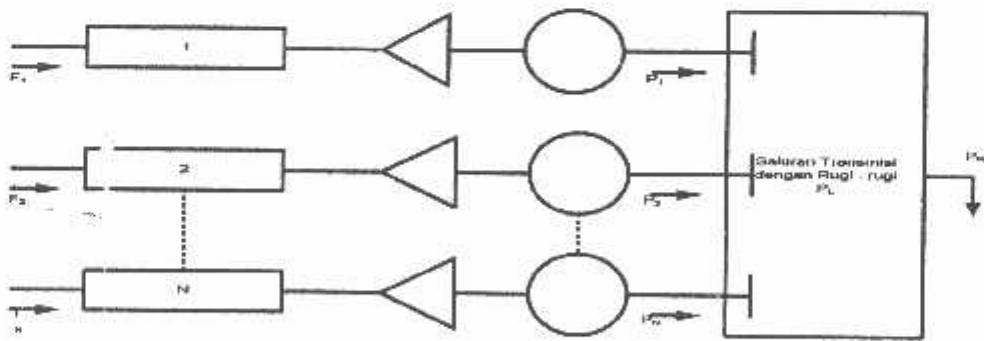
$$\frac{\partial F_i}{\partial P_i} \geq \lambda \quad \text{untuk } P_i = P_{i \text{ min}} \dots\dots\dots (2.47)$$

Karena F_i hanya sebagai fungsi P_i maka $\frac{\partial F_i}{\partial P_i}$ 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(Wood, Allan J and B. F. Wollenberg, 1996 : 31).



Gambar 2.11.
N buah Pembangkit Thermal Melayani Beban P_R Melalui Saluran Transmisi^[2]
 Sumber : Wood, Allan J and B. F. Wollenberg, 1996 : 30

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

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

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

Persamaan La Grange nya adalah(Wood, Allan J and B. F. Wollenberg, 1996 : 31);

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

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

BAB III

OPTIMASI DAYA REAKTIF MENGGUNAKAN METODE *EVOLUTIONARY PROGRAMMING AND QUASY - SIMPLEX*

3.1. Analisa Aliran Daya

Dalam melayani beban yang dibutuhkan oleh konsumen dan pengoperasian tenaga listrik perlu menganalisa aliran daya, sehingga sistem 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 aliran daya adalah :

1. Mencari harga magnitude tegangan $|v|$ dan sudut fasa tegangan δ bus beban.
2. Mencari besar daya reaktif Q dan sudut fasa tegangan δ dari generator bus.
3. Untuk mengetahui besar 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 besar daya yang mengalir disetiap saluran jaringan tenaga listrik.

3.1.1. Klasifikasi Bus

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 (PQ) atau *Load Bus*

Pada bus ini hanya terdapat kebutuhan daya untuk beban dimana P daya aktif dan Q daya reaktif diketahui, sementara $|v|$ dan δ berubah-ubah sesuai dengan kebutuhan. Oleh karena itu $|v|$ dan δ harus 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 itu 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 (Stevenson, Willism D, Jr, 1996 : 187):

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

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 (Stevenson, Willism D, Jr, 1996 : 187) :

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

Bila dituliskan dalam bentuk real dan imajiner maka persamaan di atas menjadi (Stevenson, Willism D, Jr, 1996 : 187) :

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

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

3.1.2. Metode Newton Rapshon.

Proses yang dilakukan adalah membandingkan antara daya yang ditempatkan berdasarkan data ($P_{k, \text{sched}}$ dan $Q_{k, \text{sched}}$) dengan daya hasil perhitungan ($P_{k, \text{calc}}$ dan $Q_{k, \text{calc}}$) menggunakan persamaan (3.4.) dan (3.5.) di atas. Selisih daya yang diterapkan dan perhitungan (ΔP_k dan ΔQ_k) dihitung dengan persamaan (Stevenson, Willism D, Jr, 1996 : 187) :

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

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

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 (Stevenson, Willism D, Jr, 1996 : 187) :

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

$$\Delta Q_{n-1} = \frac{\partial Q_{n-1}}{\partial v_1} \dots \frac{\partial Q_{n-1}}{\partial v_{n-1}} \frac{\partial Q_{n-1}}{\partial \delta_1} \dots \frac{\partial Q_{n-1}}{\partial \delta_{n-1}} = \Delta \delta_{n-1}$$

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

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 dan Quasy – Simplex.*

3.2.1. *Evolutionary Programming.*

Evolutionary programming merupakan metode yang bisa 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 didalam lingkungan yang kompetitif. Dengan meniru proses ini *Evolutionary Programming* dapat digunakan untuk mencari solusi permasalahan-permasalahan dalam hidup nyata skala besar.

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

Sebelum *Evolutionary Programming* dijalankan, maka sebuah kode yang sesuai (representasi) untuk persoalan harus dirancang. Titik solusi dalam ruang permasalahan dikodekan dalam bentuk *kromosom/string* yang terdiri dari

komponen genetik terkecil yaitu gen. pemakaian bilangan real (*floating point*) sebagai *allele* (nilai gen) memungkinkan penerapan operator *Evolusinary Programming* yaitu inisialisasi (*initialization*), statistik (*statistics*), mutasi (*mutation*), dan kompetisi (*competition*) untuk menciptakan himpunan titik –titik solusi. Untuk memeriksa hasil optimasi, kita membutuhkan fungsi *fitnes* yang menandakan gambaran hasil (*solution*) yang sudah dikodekan. Selama proses induk harus digunakan untuk reproduksi, mutasi dan kompetisi untuk menciptakan keturunan (*offspring*).

3.2.2. *Quasy – Simplex*.

Quasy – Simplex merupakan bentuk 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 *Quasy-Simplex* dapat digunakan untuk mencari solusi permasalahan-permasalahan dalam dunia nyata.

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

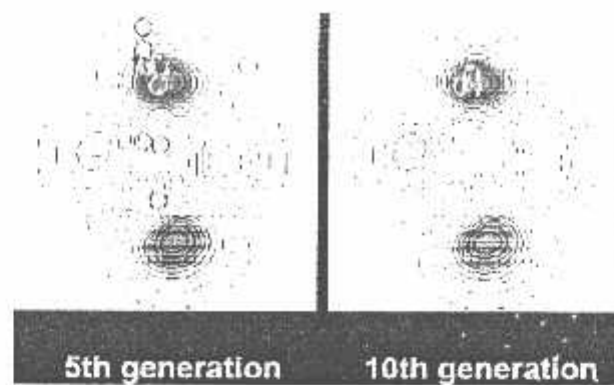
Quasy-Simplex merupakan metode pencarian langsung, dimana diterima secara luas sebagai bentuk pencarian. *Quasy-simplex* tidak hanya mempercepat tingkat konvergensi dari pencarian global atau lokal, tetapi dapat mengurangi probabilitas kehilangan global optimal dalam proses pencarian. sebuah *simplex* dalam ruang n -dimensional didefinisikan menurut polyhedron cembung yang terdiri dari puncak $n+1$. perulangan yang dihasilkan dengan memprediksi satu titik dimana fungsi tersebut mempunyai nilai yang diinginkan, dan kemudian memasukkan titik ini ke dalam *Quasy-Simplex*.

3.3. Parameter Evolutionary Programming and Quasy – Simplex.

Terdapat beberapa parameter yang digunakan dalam *Evolutionary Programming and Quasy-simplex*. Parameter tersebut digunakan untuk melihat kompleksitas dari *Evolutionary Programming and Quasy-simplex*. Parameter yang digunakan tersebut adalah :

3.3.1 Jumlah Generasi (MAXGEN)

Merupakan jumlah perulangan (iterasi) dilakukannya rekombinasi dan seleksi. Jumlah generasi ini mempengaruhi kestabilan output dan lama iterasi (waktu proses *Evolutionary Programming and Quasy-simplex*). 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 dan telah digambarkan seperti gambar 3.1.



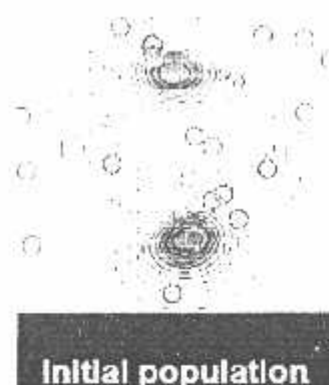
Gambar 3.1 .^[15]

Jumlah Generasi

Sumber : Neuro-Fuzzy and soft Computing” Reference book :Jang et al

3.3.2. Ukuran Populasi (*POPSIZE*)

Ukuran populasi mempengaruhi kinerja dan efektifitas dari *Evolutionary Programming and Quasy-simplex*. Jika ukuran populasi kecil maka populasi tidak menyediakan cukup materi untuk mencakup ruang permasalahan, sehingga pada umumnya kinerja *Evolutionary Programming and Quasy-simplex* menjadi buruk. Dalam hal ini dibutuhkan ruang yang lebih besar untuk mempersentasikan keseluruhan ruang permasalahan. Selain itu penggunaan populasi yang besar dapat mencegah terjadinya konvergensi pada wilayah local dan telah digambarkan seperti gambar 3.2.



Gambar 3.2 .^[15]

Ukuran Populasi

Sumber : Neuro-Fuzzy and soft Computing” Reference book :Jang et al

3.3.3 Probabilitas *Crossover* (P_c)

Probabilitas crossover ini digunakan untuk mengendalikan frekuensi operator crossover. Dalam hal ini, dalam populasi terdapat $P_c \times POPSIZE$ struktur (individu) yang melakukan pindah silang. Semakin besar nilai probabilitas crossover maka semakin cepat struktur baru yang diperkenalkan dalam populasi. Namun jika probabilitas crossover terlalu besar maka struktur dengan nilai fungsi obyektif yang baik dapat hilang dengan lebih cepat dari seleksi. Sebaliknya jika probabilitas terlalu kecil akan menghalangi proses pencarian dalam proses *Evolutionary Programming*.

3.3.4 Probabilitas Mutasi (P_m)

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

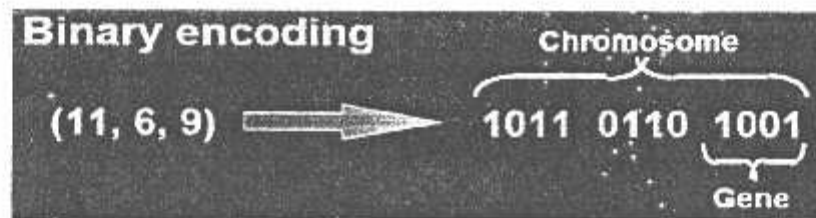
3.4 Mekanisme *Evolutionary Programming and Quasy – Simplex*.

3.4.1 Pengkodean atau Representasi.

Langkah-langkah yang dilakukan dalam penggunaan *Evolutionary Programming and Quasy – Simplex* adalah melakukan Pengkodean atau Representasi terhadap permasalahan yang akan dilakukan.

Secara umum *Evolutionary Programming and Quasy – Simplex* 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* (nilai gen). Variabel-variabel ini dapat dinyatakan dalam bentuk bilangan real (*floating point*).

Selanjutnya beberapa kromosom dibentuk dan berkumpul membentuk populasi. Populasi ini adalah populasi awal bagi *Evolutionary Programming and Quasy – Simplex* untuk awal melakukan pencarian dan digambarkan seperti gambar 3.3.



Gambar 3.3.^[15]

Pengkodean Representasi

Sumber : "Neuro-Fuzzy and soft Computing" Reference book :Jang et al

Ciri-ciri utama dari *Evolutionary Programming and Quasy – Simplex* adalah sebagai berikut :

1. Menggunakan empat individual baik potensial yang dihasilkan berdasarkan ide teknik *simplex* klasik sebagai bagian dari keturunan dalam proses evolusi. ide penting dari teknik *simplex* klasik adalah mempredisi potensial titik bagus di sepanjang garis yang dimulai dari x^H dan kedepan x^L dibawah asumsi bahwa titik terbaik lebih memungkinkan disepanjang arah ini operasi tersebut dapat menjadi refleksi, ekspansi, kompresi, atau kontraksi tergantung pada nilai tengah.
2. Menggunakan informasi populasi bukannya informasi individual terbaik.
3. Menggunakan aturan mutasi *Gaussian* dan mutasi *Cauchy* secara paralel untuk mengambil yang terbaik dari dua mutasi. Mungkin mutasi *Cauchy* dimaksudkan untuk menciptakan langkah-langkah lompatan besar dan dapat meningkatkan kemampuan untuk

menemukan global optimum sementara mutasi *Gaussian* dimaksudkan untuk meningkatkan langkah-langkah lompatan kecil dan dapat meningkatkan kemampuan untuk menemukan local optimum.oleh karena itu mutasi *Gaussian* dan mutasi *Cauchy* digunakan dalam algoritma *EPQS*.

4. Setiap generasi menciptakan keturunan yang ukurannya lebih besar dari induknya.namun demikian hanya indifidual puncak yang dipilih dari seluruh induk dan keturunan dalam pengertian kesesuaian mereka akan menjadi induk pada generasi selanjutnya.*Quasy-Simplex* tidak hanya mempercepat tingkat konvergensi dari pencarian global atau local tetapi dapat mengurangi probabilitas kehilangan global optimal dalam proses pencarian dan digambarkan seperti gambar 3.3.

3.4.2 Fungsi *Fitness* (Fungsi Evaluasi)

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

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

3.4.3 Seleksi dan Reproduksi^[12]

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

3.4.3.1 *Roulette Wheel Selection*

Dimana setiap individual memiliki harga fitness sehingga didapatkan probabilitas individual $(f(i)/\sum f(i))$ tersebut dicopykan pada populasi yang baru. Untuk individual yang memiliki probabilitas 20% untuk jumlah populasi 10 maka kemungkinan individual tersebut dapat terpilih sebanyak dua kali dan digambarkan seperti gambar 3.4.

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

1. Menjumlahkan fitness dari seluruh anggota populasi.
2. Membangkitkan nilai k, suatu nilai random antara 0 dan total fitnessnya.
3. Menjumlahkan fitness dari kromosom-kromosom dari populasi mulai 0 hingga total fitness lebih besar atau sama dengan nilai k lalu ambil kromosom tersebut.



Gambar 3.4 .^[15]

Roulette Wheel

Sumber : *Neuro-Fuzzy and soft Computing* Reference book :Jang et al

3.4.3.2 *Rank Selection*

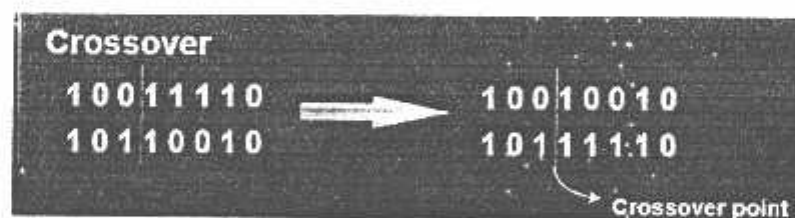
Apabila fitness yang dimiliki oleh suatu kromosom dalam populasi berbeda terlalu jauh dari kromosom lainnya maka hal ini dapat menjadi permasalahan. Misalnya bila kromosom terbaik mempunyai *fitness* yang menyebabkan besarnya tempat yang dimilikinya dalam *roulette wheel* sebesar 90% maka kromosom-kromosom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

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

3.4.4 Crossover (Pindah Silang) ^{[14][12]}

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

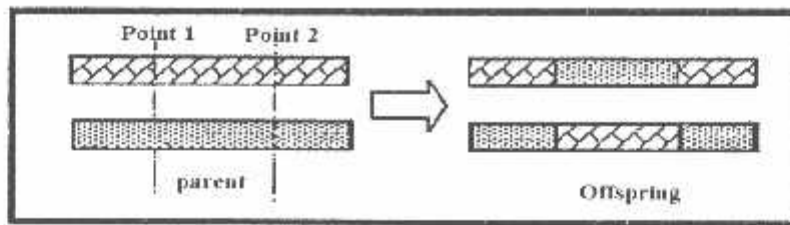
Kemudian ditingkatkan lagi dengan menggunakan *two point crossover*. Penentuan posisi titik persilangan sama seperti sama seperti *one point crossover* sebelumnya. Pemilihan secara random dilakukan 2 kali. Kemudian nilai *offspring* diambil dari dua parent tersebut dengan batas dua titik persilangan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



Gambar 3.5. ^[15]

Ilustrasi operator dengan *One Point Crossover*

Sumber : *Neuro-Fuzzy and soft Computing* Reference book :Jang et al

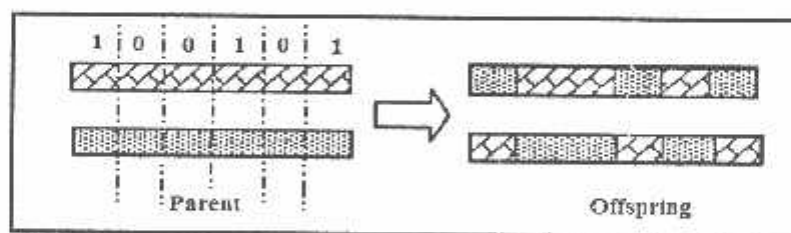


Gambar 3.6.^[11]

Ilustrasi operator dengan *Two Point Crossover*

Sumber : "Artificial Intelligence (teknik dan aplikasinya)" Sri Kusuma dewi

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



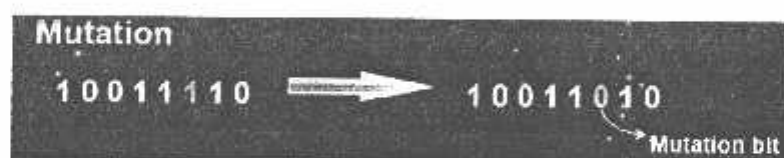
Gambar 3.7.^[11]

Ilustrasi operator *crossover* dengan *uniform crossover*

Sumber : "Artificial Intelligence (teknik dan aplikasinya)" Sri Kusuma dewi

3.4.5 Mutation (Mutasi)^{[14][12]}

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 kombinasi *Evolutionary Programming and Quasy Simplex* menggunakan operator *Cauchy mutation*, *Gaussian mutation*, *Quay-Simplex* dimana setiap individu akan terpilih secara acak untuk mengalami mutasi berdasarkan nomor acak dan menemukan nilai yang terbaik serta menciptakan individu baru (*offspring*) dan ilustrasi mutasi tersebut telah digambarkan seperti gambar 3.8.



Gambar 3.8.^[15]

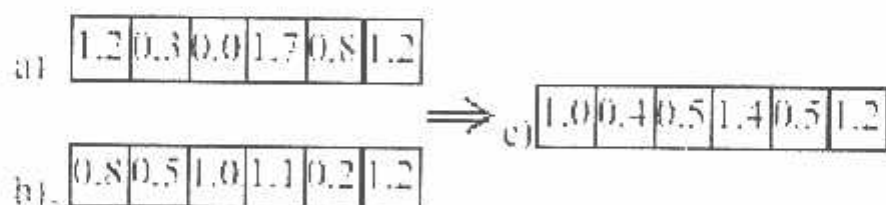
Ilustrasi operasi mutasi

Sumber : Neuro-Fuzzy and soft Computing” Reference book Jang et al

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

3.4.6 *Recombination (Rekombinasi)*^{[11][12]}

Dalam *Evolutionary Programming* setelah melakukan *initialization* dan *evaluation*, individual dipilih secara acak menjadi *parents*. Setelah melakukan seleksi *parent* kemudian melakukan rekombinasi. Rekombinasi merupakan metode *intermediate recombination* (rekombinasi menengah) yang hanya dapat digunakan untuk variabel real (dan variabel yang bukan biner). Nilai variabel *offspring* dipilih disekitar dan antara nilai variabel *parent*, dimana vektor dari dua orang tua (*parent*) dirata-ratakan bersama-sama, unsur demi unsur, untuk membentuk suatu keturunan yang baru dan telah digambarkan seperti gambar 3.9.



Gambar 3.9^[11]

Recombination dari Induk (*parent*) a) & b) menjadi Anak (*offspring*) c)

Seleksi orang tua (*parent*) menjadi turunan (*offspring*) menjadi lebih mudah, sebagai contoh dalam kaitannya dengan sifat alami penyajian, adalah sangat mudah merata-ratakan dari banyak individu menjadi turunan tunggal. Didalam *Evolutionary Programming*, *N parent* terpilih secara acak (tidak didasarkan pada *fitness* terbaik), kemudian *N offspring* dihasilkan melalui penggunaan *intermediate recombination*, dimana variabel pada anak merupakan hasil kombinasi variabel induk,

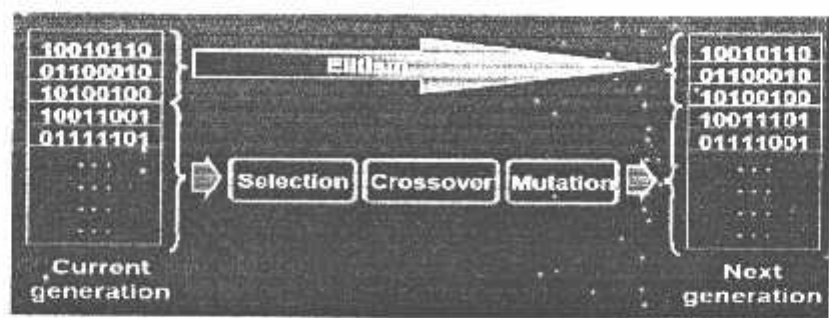
3.4.7 *Competition* (Kompetisi)

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

3.4.8 *Elitism*

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

seleksi, *crossover* dan mutasi. Elitism dapat secara cepat meningkatkan performansi dari *Evolutionary Programming* karena *elitism* menghindarkan hilangnya solusi terbaik (*best / few best*) yang telah ditemukan, pembentukan next generation dalam evolusionary programming digambarkan seperti gambar 3.10.



Gambar 3.10.^[15]

Pembentukan *Next Generation* dalam *Evolutionary Programming*

Sumber : *Neuro-Fuzzy and soft Computing”* Reference book :Jang et al

A. Formulasi Masalah Optimasi Daya Reaktif^[11]

Permasalahan perencanaan optimasi daya reaktif adalah untuk meminimalkan biaya operasional, dengan persamaan sebagai berikut: (Kwang. Y. Lee, and Frank F. Yang 1998 : 104);

$$C_Q = C_P + C_i(Q_i) \dots\dots\dots (3.8.)$$

$$C_P = \sum_{i \in N_r} d_i C_P^i \dots\dots\dots (3.9.)$$

dimana:

- $C_i(Q_i)$ = Biaya daya reaktif pembangkitan.
- C_P^i = Biaya operasi daya nyata pada tingkatan *i* beban

d_i = Jangka waktu tingkatan i beban

N_i = Banyaknya tingkatan beban

dengan subyek:(Kwang, Y. Lee, and Frank F. Yang 1998 : 104):

$$\underline{P}_g \leq P_g \leq \bar{P}_g \quad (\text{Daya nyata generator})$$

$$\underline{Q}_g \leq Q_g \leq \bar{Q}_g \quad (\text{Daya reaktif generator})$$

$$\underline{T}_i \leq T_i \leq \bar{T}_i \quad (\text{Tap-Setting transformer})$$

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

Tegangan bus beban dan daya reaktif generator merupakan variabel keadaan, yang dapat dibatasi dengan menambahkannya sebagai bentuk penalti kuadrat ke fungsi obyektif sebagai berikut(Kwang, Y. Lee, and Frank F. Yang 1998 : 104):

$$C_Q = C_i + \sum_{i \in N_i} d_i \left\{ C_{P_i} + \sum_{i \in N_i} \lambda_{P_i} (V_i - \text{Sat}(V_i))^2 + \sum_{i \in N_i} \lambda_{Q_i} (Q_{P_i} - \text{Sat}(Q_{P_i}))^2 \right\} \quad (3.10.)$$

Dimana; $\text{sat}(x)$ adalah fungsi saturasi sebagai berikut:(Kwang, Y. Lee, and Frank F. Yang 1998 : 104):

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

3.6 Adaptasi *Evolutionary Programming* and *Quasy – Simplex* Ke Masalah Optimasi Daya Reaktif.

Kombinasi *Evolutionary Programming* and *Quasy – Simplex* adalah suatu mekanisme seleksi alam. Individu dari sebuah populasi mutasi secara real,

Populasi pertama dibangkitkan secara random. Generasi baru dibuat dengan mengaplikasikan operator berikut terhadap sebuah populasi yaitu : inisialisasi, rekombinasi, mutasi dan kompetisi dimana tingkat pengoptimalan setiap calon solusi atau individu diukur dengan ketepatannya tergantung pada fungsi tujuan (*objective function*) permasalahan.

3.6.1 Representasi Solusi^[1]

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).

3.6.2 Initial Population Generation (Inisial populasi generasi)^[1]

Di dalam rencana suatu populasi awal bilangan biner diciptakan secara acak dari string menghadirkan satu solusi mungkin kepada rumusan masalah, i.e., di dalam pencarian suatu daerah yang menghasilkan ketidaksesuaian.

3.6.2. Initialization (Inisialisasi)^[1]

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

3.6.3. Evaluasi.Fitness^[1] ;

Berikutnya string solusi diubah jadi sistim desimal dan setiap calon solusi diuji dalam lingkungan nya . kebenaran dari tiap calon dievaluasi melalui/sampai beberapa ukuran yang sesuai, begitu juga sebagai kebalikan menyangkut fungsi biaya : (Kwang, Y. Lee, and Frank F. Yang 1998 : 103).

$$f = 1 / (\alpha + c) \dots\dots\dots (3.11)$$

Jika C adalah biaya fungsional untuk diperkecil dan α adalah suatu fungsi parameter. Algoritma dikemudikan untuk memaksimalkan ukuran fitness ini. Setelah fitness keseluruhan populasi telah ditentukan ya atau tidaknya ukuran penghentian jumlah.dari hal hal manapun. satu. Kemungkinan akan stop algoritma pada beberapa jumlah terbatas dan mengambil hasil sebagai yang terbaik cocok dari populasi itu. Kemungkinan yang lain akan menguji rata-rata fitness dari populasi melebihi beberapa pecahan menyangkut yang terbaik cocok di populasi itu. Jika ukuran adalah satisfiet kemudian kita melanjutkan dengan ke tiga operasi *reproduksi* , *crossorver*, dan *mutasi*

3.6.4 *Selection dan Reproduction*^[1]

Fitness-yang sebanding reproduksinya diakibatkan putaran yang ditirukan dari suatu rolet dihargai kemudi rolet itu. kemudi dibiaskan dengan fitness dari tiap calon solusi. Kemudi adalah spun N di mana N adalah banyaknya string di (dalam) populasi itu. Operasi ini menghasilkan suatu populasi string yang baru yang mencerminkan fitness dari generasi sebelumnya

3.6.5 Crossover^[1]

Berikutnya operasi *crossover* dilakukan pada dua string serentak terpilih dari populasi secara acak. *Crossover* melibatkan suatu posisi acak di dalam kedua string dan menukar bit yang terjadi setelah posisi ini. Di dalam satu generasi *crossover* operasi dilakukan pada persentase yang ditetapkan pada *population*. *Crossover* dapat terjadi pada *single position*, atau pada jumlah posisi berbeda. *crossover* dapat juga dilakukan di dua alat-alat yang berbeda: head-tail *crossover* (10,13) adalah *crossover* yang umum, jika bagian paling akhir dari kedua string berubah. Di dalam *head tail crossover*, pada sisi lain, bagian paling akhir satu *string* menjadi head dan sebaliknya, *string* yang lain. *head-tail crossover* bit berubah semakin sedikit. Dua metoda *crossover* dapat diubah selama perhitungan berulang-ulang; head-tail *crossover* menggunakan ketel masak generasi dan yang *switch* ke *tail-tail crossover* di (dalam) generasi yang kemudian untuk penyetelan yang baik.

3.6.6 Statistics (Statistik)^[1]

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

3.6.7 Mutation^[1],

3.6.7.1 Evolutionary Programming

Di akhir genetika operator dalam *Evolutionary Programming* adalah mutasi. Mutasi dilakukan dengan hemat, secara khas setelah tiap-tiap 100-1000 bit memindahkan dari *crossover*, dan melibatkan pemilihan suatu nilai string

secara acak juga sedikit memposisikan secara acak dan mengubahnya dari 1 ke 0 atau vice-versa. Digunakan untuk tidak mencapai nilai minimum. Setelah mutasi, generasi baru dilengkapi / menyudahi dan prosedur mulai lagi dengan evaluasi fitness dari populasi.

3.6.7.2 *Quasy – Simplex*

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

$$P_{i+m,j} = P_{i,j} + N(0, \beta \nabla_{dev}), \quad j = 1, 2, \dots, n, \dots \dots \dots (3.12)$$

dimana:

- $P_{i,j}$ = Elemen ke j dari individu ke i
- β = Skala mutasi ($0 < \beta < 1$)
- ∇_{dev} = Variabel deviasi

3.6.8 *Competition (Kompetisi)*^[1]

Di dalam proses kompetisi, m *parent* akan bersaing dengan m *offspring*. Dan m individu yang pertama dijaga sebagai orang tua untuk proses generasi selanjutnya. Nilai berat W_i dari individu yang ke i dihitung menurut kompetisi berikut ini:

$$W_i = \sum_{j=1}^N W_{i,j} \dots \dots \dots (3.13)$$

dimana N adalah nomor kompetisi generasi secara acak. $W_{i,j}$ untuk yang manapun, 0 untuk kalah dan 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 :

$$W_{i,r} = \begin{cases} 1 & \text{jika } U_i < \frac{f_r}{f_r + f_i} \\ 0 & \text{jika sebaliknya} \end{cases} \dots\dots\dots (3.14)$$

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_i kompetisinya, mereka akan diatur dalam order turunan menurut nilai W_i 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.

3.6.9 Convergence Test (Konvergensi)^[1]

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

$$\{f_{avg} / f_{max}\} \geq \delta \dots\dots\dots (3.15)$$

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

4.1. Program Komputer Optimasi Daya Reaktif Menggunakan Metode *Evolutionary Programming and Quasy – Simplex*.

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 Pentium 4 dengan Prosesor 1800 MHz, Memori 512 Mb.

4.1.1. Algoritma Program

Algoritma Program Perencanaan Optimasi Daya Reaktif dengan Menggunakan Metode *Evolutionary Programming and Quasy – Simplex* :

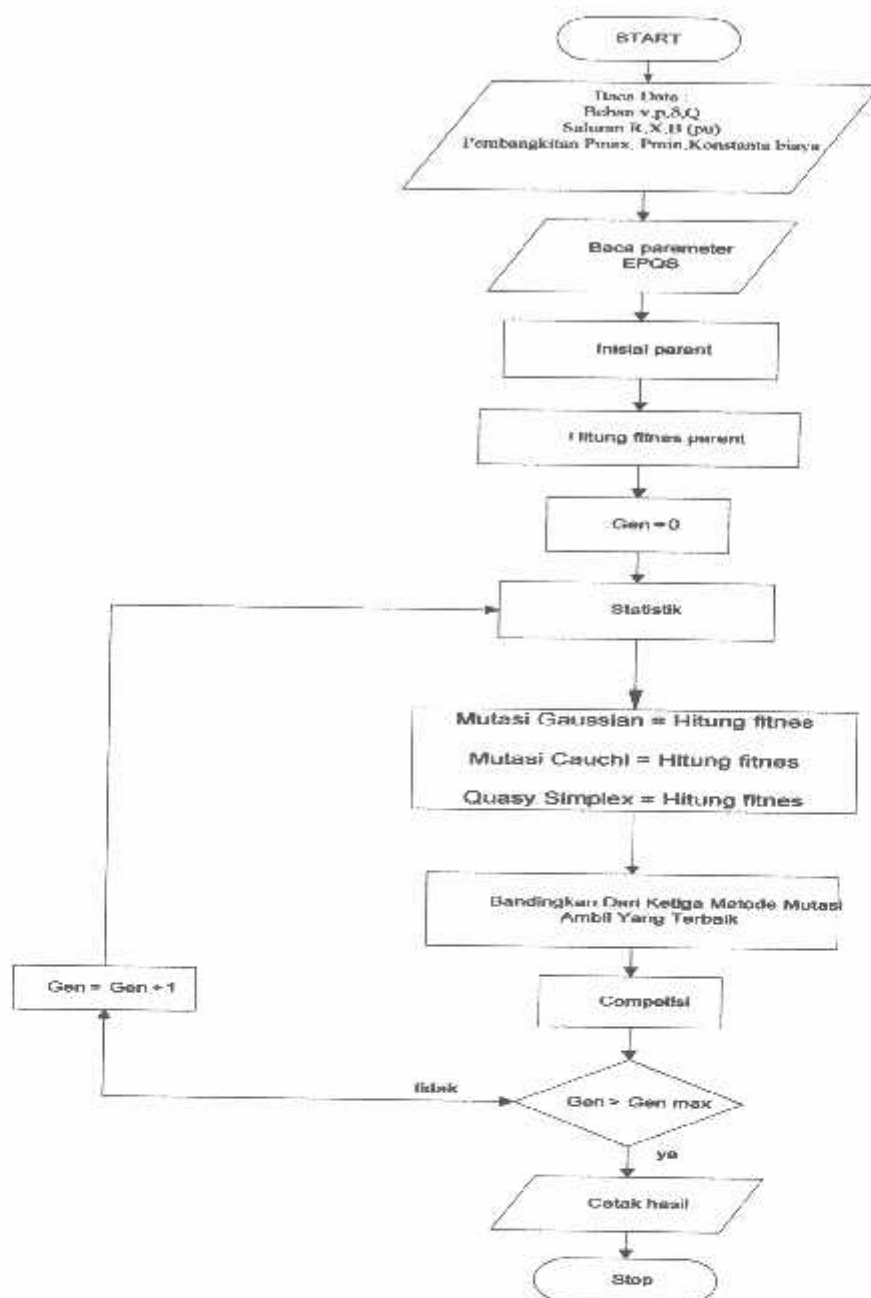
1. Memasukkan inputan data beban yang meliputi tegangan, sudut fasa tegangan, daya aktif (P), daya reaktif (Q) dan data impedansi saluran.
2. Menentukan parameter inputan *Evolutionary Programming and Quasy - Simplex* yang meliputi jumlah populasi, maksimum generasi, nilai kemungkinan *crossover*, nilai kemungkinan mutasi, kompetisi dan panjang kromosom tiap-tiap individu.
3. Melakukan *inisialisasi chromosome* calon solusi
4. menghitung *fitness* dari kromosom tiap-tiap individu.
5. Melakukan proses *statistik*.

6. Melakukan proses *seleksi*.
7. Melakukan proses *crossover*.
8. Melakukan proses *mutasi*.
9. Proses no. 6, 7, 8 diulang sampai *child* sampai jumlah populasi.
10. Menghitung *fitness* dari *child*.
11. Melakukan proses *elitism*.
12. Apakah *generasi* yang diinginkan sudah terpenuhi (*max Gen*).
13. Jika “ tidak “ maka $\text{generasi} = \text{gen} + 1$, kembali ke langkah 5.
14. Jika “ Ya “ cetak hasil
15. Menentukan individu terbaik dari *EP*
16. Melakukan *inisial chromosome* calon solusi
17. Menghitung *fitness* dari *child*.
18. Melakukan proses *statistik*
19. Melakukan proses *mutasi*
20. Proses no. 19 diulang sampai *child* sampai jumlah populasi.
21. Menghitung *fitness* dari *child*
22. Melakukan proses *kompetisi*
23. Apakah generasi yang diinginkan sudah terpenuhi (*max Gen*)
24. Jika “tidak” maka $\text{generasi} = \text{gen} + 1$, kembali ke langkah 18.
25. Jika “ya” maka cetak hasil dan perhitungan berhenti.

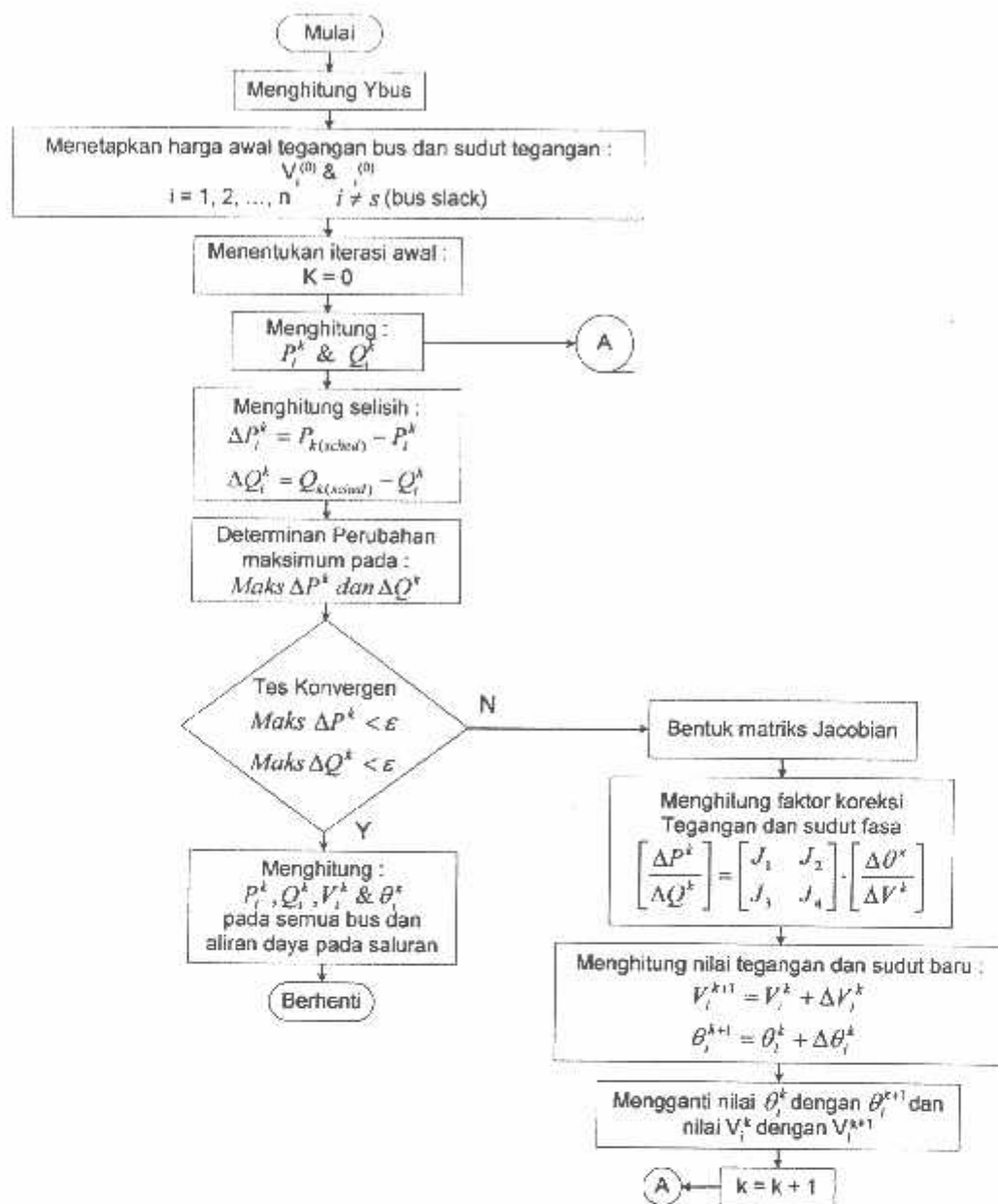
4.1.2. Sedangkan untuk algoritma aliran daya dengan metode *Newton*

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.



Gambar 4.1
Flowchart Program Menggunakan
Evolutionary Programming and Quasy-Simplex



Gambar 4.2.
Flowchart Perhitungan Aliran Daya Metode Newton Rapshon

4.2. Validasi Data IEEE 30 Untuk Menyelesaikan Permasalahan Optimasi Daya Reaktif Secara Ekonomis Menggunakan Metode *Evolutionary Programming and Quasy - Simplex*

Tabel 4.1.^[1]
Data Generator Dan Koefisien Biaya IEEE 30

No. Bus	a2	a1	a0	Pmin (MW)	Pmax (MW)	Qmin (MVAR)	Qmax (MVAR)
1	0.00375	2.00	0	50	200	-20	150
2	0.01750	1.75	0	20	80	-20	60
5	0.06250	1.00	0	15	50	-15	62.5
8	0.00834	3.25	0	10	35	-15	50
11	0.02500	3.00	0	10	30	-10	40
13	0.02500	3.00	0	12	40	-15	45

Tabel 4.2.^[16]
Data Bus IEEE 30

Nu. Bus	Tegangan		Pembangkitan		Pembebanan	
	Magnitude (pu)	Sudut (deg)	Aktif (MW)	Reaktif (MVAR)	Aktif (MW)	Reaktif (MVAR)
1	1.06	0	138.63	3.13	0	0
2	1.05	0	57.56	13.75	21.7	12.7
3	1	0	0	0	2.4	1.2
4	1	0	0	0	7.6	1.6
5	1.01	0	24.56	26.88	94.2	19
6	1	0	0	0	0	0
7	1	0	0	0	22.8	10.9
8	1.01	0	35	48.89	30	30
9	1	0	0	0	0	0
10	1	0	0	0	5.8	2
11	1.08	0	17.93	7.53	0	0
12	1	0	0	0	11.2	7.5
13	1.07	0	16.91	14.65	0	0
14	1	0	0	0	6.2	1.6
15	1	0	0	0	8.2	2.5
16	1	0	0	0	3.5	1.8
17	1	0	0	0	9	5.8
18	1	0	0	0	3.2	0.9
19	1	0	0	0	9.5	3.4
20	1	0	0	0	2.2	0.7
21	1	0	0	0	17.5	11.2
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

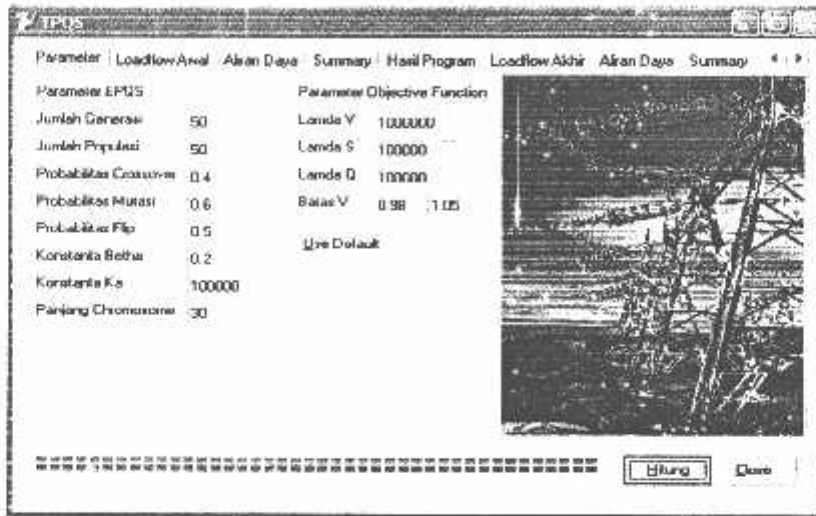
27	1	0	0	0	0	0
28	1	0	0	0	0	0
29	1	0	0	0	2.4	0.9
30	1	0	0	0	10.6	1.9

Tabel 4.3.^[16]
Data Saluran Impedansi IEEE 30

No. Saluran	Hubungan Bus	R (pu)	X (pu)	B (pu)
1	1-2	0.0096	0.0288	0.0264
2	1-3	0.0452	0.1852	0.0408
3	2-4	0.0570	0.1737	0.0368
4	3-4	0.0132	0.0379	0.0084
5	3-5	0.0472	0.1983	0.0418
6	2-6	0.0581	0.1763	0.0374
7	4-6	0.0119	0.0414	0.0090
8	5-7	0.0460	0.0116	0.0204
9	6-7	0.0267	0.0820	0.0170
10	6-8	0.0120	0.0420	0.0090
11	6-9	0.0000	0.2082	0.0000
12	6-10	0.0000	0.5560	0.0000
13	9-11	0.0000	0.2080	0.0000
14	9-10	0.0000	0.1100	0.0000
15	4-12	0.0000	0.2560	0.0000
16	12-13	0.0000	0.1400	0.0000
17	12-14	0.1231	0.2559	0.0000
18	12-15	0.0662	0.1304	0.0000
19	12-16	0.0945	0.1987	0.0000
20	14-15	0.2210	0.1997	0.0000
21	16-17	0.0524	0.1932	0.0000
22	15-18	0.1073	0.2185	0.0000
23	18-19	0.0639	0.1292	0.0000
24	19-20	0.0340	0.0680	0.0000
25	10-20	0.0936	0.2090	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	0.1000	0.2020	0.0000
31	22-24	0.1150	0.1790	0.0000
32	23-24	0.1320	0.2700	0.0000
33	24-25	0.1885	0.3292	0.0000
34	25-26	0.2544	0.3800	0.0000
35	25-27	0.1093	0.2087	0.0000
36	28-27	0.0000	0.3960	0.0000
37	27-29	0.2198	0.4153	0.0000
38	27-30	0.3202	0.6027	0.0000
39	29-30	0.2399	0.4533	0.0000
40	8-28	0.0636	0.2000	0.0428
41	6-28	0.0169	0.0599	0.0130

4.2.1. Hasil Validasi IEEE 30 Dengan Menggunakan Metode *Evolutionary Programming and Quasy - Simplex*

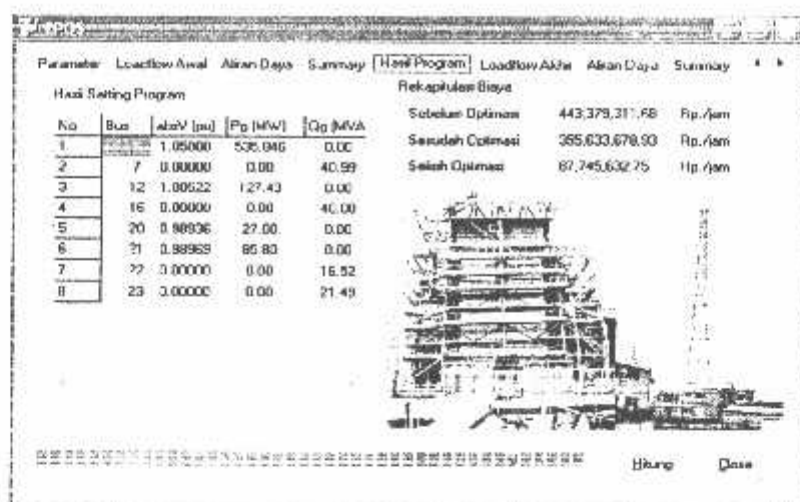
1. Tampilan Parameter Validasi



Gambar 4.4.
Tampilan Parameter Kombinasi *EPQS* Validasi IEEE 30

Bus	U ₀ (pu)	U ₁ (deg)	P ₀ (MW)	Q ₀ (MVAR)	P ₁ (MW)	Q ₁ (MVAR)	S ₀ (pu)	Type Bus
1	1.000000	0.00000	536.846	-39.292	54.600	15.000	0.000	1
2	1.03223	-0.03534	0.000	0.000	14.000	7.000	0.000	3
3	1.02619	-0.04001	0.000	0.000	17.000	8.500	0.000	3
4	1.01323	-0.08424	0.000	0.000	33.000	13.600	0.000	3
5	0.99241	-0.10707	0.000	0.000	40.600	14.600	0.000	3
6	0.98461	-0.14488	0.000	0.000	27.200	9.700	0.000	3
7	0.98564	-0.12510	0.000	40.989	65.200	28.600	0.000	3
8	0.98766	-0.20733	0.000	0.000	40.000	19.000	0.000	3
9	0.97296	-0.12684	0.000	0.000	26.200	10.200	0.000	3
10	0.97707	-0.14697	0.000	0.000	18.400	6.700	0.000	3
11	1.00432	-0.20641	0.000	0.000	38.200	15.600	0.000	3
12	1.00522	-0.20924	127.426	93.268	6.200	2.800	0.000	2
13	0.99616	-0.25621	0.000	0.000	13.700	4.500	0.000	3
14	0.99884	-0.30361	0.000	0.000	5.600	2.600	0.000	3
15	0.98638	-0.33206	0.000	0.000	54.900	16.600	0.000	3

Gambar 4.5.
Hasil *Load Flow*



Gambar 4.6.
Tampilan Hasil Validasi IEEE 30
Hasil Perhitungan Biaya Pembangkitan

Tabel 4.4.^[1]
Perbandingan Hasil Data Referensi Jurnal Dengan Data Optimasi

No.	No. Bus	Data Referensi		Data Optimasi	
		P _{Gen} (MW)	Q _{Gen} (MVar)	P _{Gen} (MW)	Q _{Gen} (MVar)
1	1	106.438	18.966	97.868	18.666
2	2	76.054	28.083	80.000	28.091
3	5	20.830	59.231	20.373	58.971
4	8	33.660	8.635	35.000	8.165
5	11	14.429	13.333	17.090	13.241
6	13	40.043	0.933	40.000	0.926

Setelah dilakukan proses optimasi maka diperoleh biaya pembangkitan dari data IEEE 30 yaitu sebesar 801.95 \$/hr, sedangkan dari data referensi jurnal diperoleh biaya pembangkitan sebesar 802.07 \$/hr. Dari proses validasi di atas maka didapatkan error sebesar 0,01%

4.3. Data Pembangkitan Thermal Pada Sub-Sistem Paiton-Bali

Pada skripsi ini akan membahas pembangkit *thermal* yang berada pada Sub-Sistem Paiton-Bali. Pembangkit *thermal* yang dibahas dalam skripsi ini adalah PLTU Paiton 5-6, PLTG Gilimanuk, PLTD Pesanggaran, dan PLTG Pesanggaran.

Untuk data dari bentuk karakteristik semua unit pembangkit yang dibahas dalam skripsi ini beserta kapasitasnya dan *fuel cost* (biaya bahan bakar) yang digunakan dalam perhitungan adalah berdasarkan data dari PT. Indonesia Power dan PT. PJB.

Tabel 4.5.^[17]
Parameter Unit Pembangkit Thermal

Nama Pembangkit	a2	a1	a0	Pmin (MW)	Pmax (MW)
PLTU Paiton 5&6	276,246	58384,905	17350044,31	150	700
PLTG Gilimanuk	188,355772	214074,120	11705797,80	50	133,8
PLTD Pesanggaran	1913,573616	223579,600	11909964,80	20	75
PLTG Pesanggaran	15248,932	709162,360	18684292,80	40	125

Dengan memasukkan data-data pada tabel 4.5. ke persamaan fungsi biaya bahan bakar, maka untuk unit pembangkit thermal PLTU Paiton 5&6 diperoleh sebagai berikut :

Biaya bahan bakar pembangkit^[1]:

$$F_1(G_1) = a_2 P_{Gi}^2 + a_1 P_{Gi} + a_0$$
$$F_1(G_1) = 276,246 P^2 + 58384,905 P + 17350044,31$$

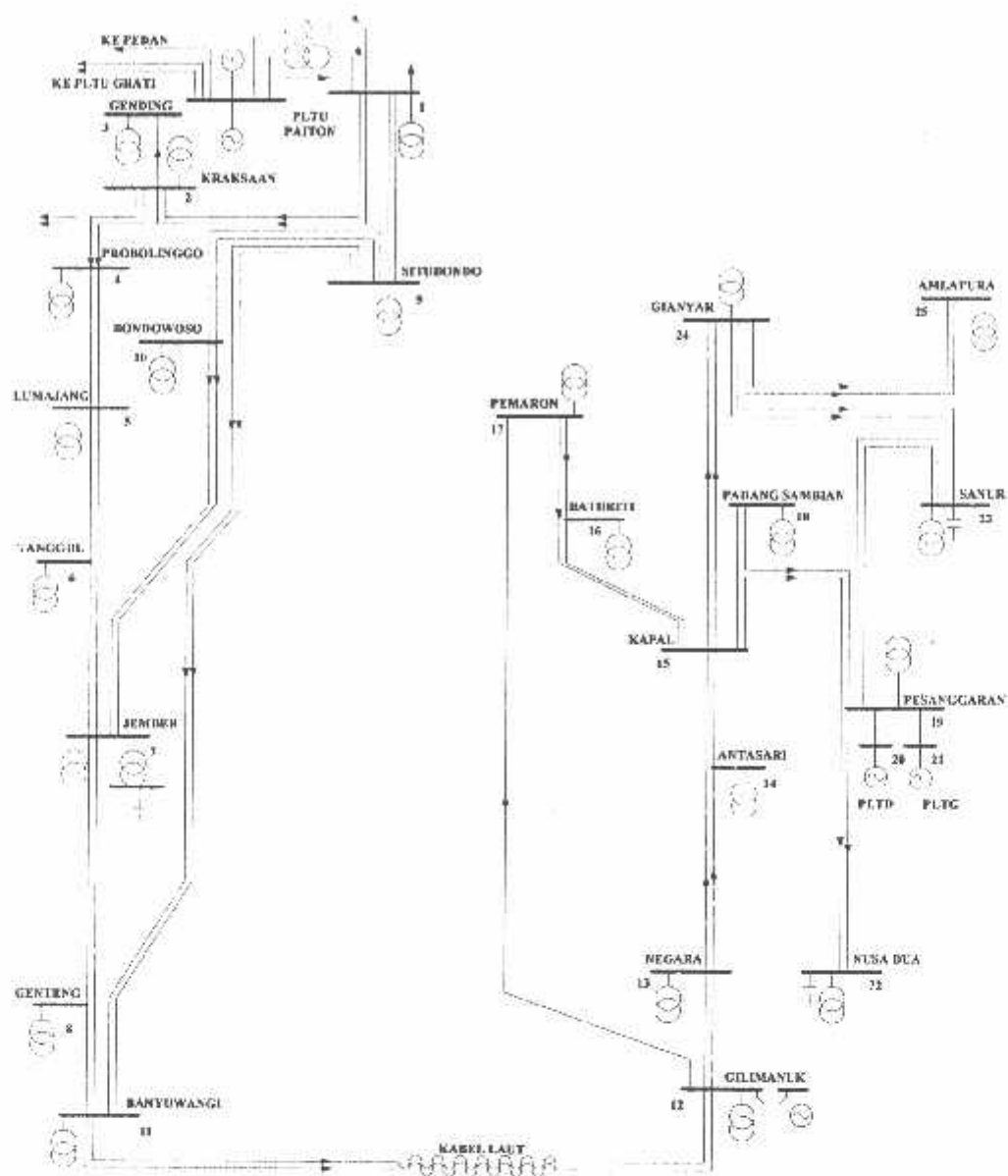
Sedangkan untuk biaya pemeliharaan dari setiap pembangkit yang merupakan biaya pengoperasian daya reaktif, dalam skripsi ini berdasarkan referensi maka diset sebesar U.S \$ 6,77 / Mvar perjam^[2]. Kurs dollar yang dipakai

sebesar Rp 9.500, jadi biaya pengoperasian daya reaktif dari masing-masing pembangkit sebesar Rp 64315 / MVar perjam.

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

Tabel 4.6.^[17]
 Persamaan Biaya Pembangkitan
 Unit Pembangkit Thermal Paiton dan Bali

No.	Nama Pembangkit	Persamaan Biaya Bahan Bakar (Rp/Jam)	Persamaan Biaya Daya Reaktif (Rp/jam)
1	PLTU Paiton 5&6	$F = 276,246 P^2 + 58384,905 P + 17350044,31$	$C_r(Q_r) = 64315 (Q_g)$
2	PLTG Gilimanuk	$F = 188,355772 P^2 + 214074,12 P + 11705797,8$	$C_r(Q_r) = 64315 (Q_g)$
3	PLTDPesanggaran	$F = 1913,573616 P^2 + 223579,6 P + 11909964,8$	$C_r(Q_r) = 64315 (Q_g)$
4	PLTGPesanggaran	$F = 15248,932 P^2 + 709162,36 P + 18684292,8$	$C_r(Q_r) = 64315 (Q_g)$



Gambar 4.7^[13]
Diagram Segaris Jaringan Sistem Tenaga Listrik 150 kV Sub-Sistem
Paiton-Bali

Tabel 4.7.^[13]
 Penomoran Bus 150 kV Sub-Sistem Paiton-Bali

No. Bus	Nama Bus
1	PAITON
2	KRAKSAAN
3	GENDING
4	PROBOLINGGO
5	LUMAJANG
6	TANGGUL
7	JEMBER
8	GENTENG
9	SITUBONDO
10	BONDOWOSO
11	BANYUWANGI
12	GILIMANUK
13	NEGARA
14	ANTASARI
15	KAPAL
16	BATURITI
17	PEMARON
18	PADANG SAMBIAN
19	PESANGGARAN
20	PLTD PESANGGARAN
21	PLTG PESANGGARAN
22	NUSA DUA
23	SANUR
24	GIANYAR
25	AMPLAPURA

4.4. Data Pembangkitan dan Pembebanan 150 kV Sub-Sistem Paiton-Bali

Berdasarkan data pada referensi [13] maka dilakukan pengolahan data untuk mengubah dasar tegangan ke dalam pu. Serta perhitungan P_{load} dan Q_{load} .

Pada bus nomor 1, maka :

$$\begin{aligned}
 \text{Tegangan Dasar} &= 150 \text{ kV} \\
 \text{Tegangan Sebenarnya} &= 154 \text{ kV} \\
 \text{Tegangan (pu)} &= \frac{\text{Tegangan sebenarnya}}{\text{Tegangan dasar}} \\
 \text{Tegangan (pu)} &= \frac{154}{150} = 1.027 \text{ pu} \\
 P_{\text{beban}} &= 24,3 + 30,3 = 54,6 \text{ MW} \\
 Q_{\text{beban}} &= 15 \text{ MVAR}
 \end{aligned}$$

Untuk selanjutnya dilakukan perhitungan seperti langkah di atas, sehingga didapatkan hasil sebagai berikut :

Tabel 4.8.^[13]
 Data Pembangkitan dan Pembebanan
 150 kV Sub-Sistem Paiton-Bali
 Tanggal 27desember 2006, Pukul : 19:00 WIB

No	Nama Bus	Tegangan (pu)	Pgenerator (MW)	Qgenerator (MVAR)	Pload (MW)	Qload (MVAR)	Tipe
1	PAITON	1,027	-	-	54,6	15	1
2	KR/ KSAN	-	-	-	14	7	3
3	GEN DING	-	-	-	17	8,5	3
4	PROBOLINGGO	-	-	-	33	13,6	3
5	LUMAJANG	-	-	-	40,6	14,8	3
6	TANGGUL	-	-	-	27,2	9,7	3
7	JEMBER	-	-	49,0	65,2	28,8	3
8	GENTENG	-	-	-	40,8	19	3
9	SITUBONDO	-	-	-	24,2	10,2	3
10	BONDOWOSO	-	-	-	18,4	6,7	3
11	BANYUWANGI	-	-	-	38,2	15,6	3
12	GILIMANUK	0,985	99,9	-	6,2	2,8	2
13	NEGARA	-	-	-	10,7	4,5	3
14	ANTASARI	-	-	-	5,6	2,6	3
15	KAPAL	-	-	40,0	54,9	16,6	3
16	BATURITI	-	-	-	4,3	0,9	3
17	PEMARON	-	-	-	24,5	7,5	3
18	PDG SAMBIAN	-	-	-	31,2	12,5	3
19	PESANGGARAN	-	-	-	71,8	27,6	3

20	PLTD PSGRAN	0,935	42,5	-	-	-	2
21	PLTG PSGRAN	0,935	67,1	-	-	-	2
22	NUSADUA	-	-	25,0	45,5	16,4	3
23	SANUR	-	-	25,0	45,9	12,7	3
24	GIANYAR	-	-	-	31,7	5,4	3
25	AMLAPURA	-	-	-	13,0	4,7	3

Ket: 1 : bus slack

2 : bus generator

3 : bus beban

4.5. Data Saluran Transmisi 150 kV Sub-Sistem Paiton-Bali

Pada Sub-Sistem Paiton-Bali terdiri dari 25 bus dan 34 saluran transmisi, dalam hal ini saluran transmisi yang dibahas adalah saluran transmisi 150 kV.

Berdasarkan data pada referensi [15], maka dilakukan pengolahan data untuk saluran dengan jumlah sirkuit 2. Sebagai contoh pada hubungan saluran 1-2 dilakukan perhitungan sebagai berikut :

$$R = \frac{R_1 \times R_2}{R_1 + R_2} = \frac{0.0105 \times 0.0105}{0.0105 + 0.0105} = \frac{1.1025 \times 10^{-4}}{0.021} = 0.0053 \text{ pu}$$

$$X = \frac{X_1 \times X_2}{X_1 + X_2} = \frac{0.0359 \times 0.0359}{0.0359 + 0.0359} = \frac{1.2888 \times 10^{-3}}{0.0718} = 0.0180 \text{ pu}$$

$$B = \frac{B_1 \times B_2}{B_1 + B_2} = \frac{0.0130 \times 0.0130}{0.0130 + 0.0130} = \frac{1.69 \times 10^{-4}}{0.026} = 0.0065 \text{ pu}$$

Untuk selanjutnya dilakukan perhitungan seperti langkah di atas, sehingga didapatkan hasil sebagai berikut :

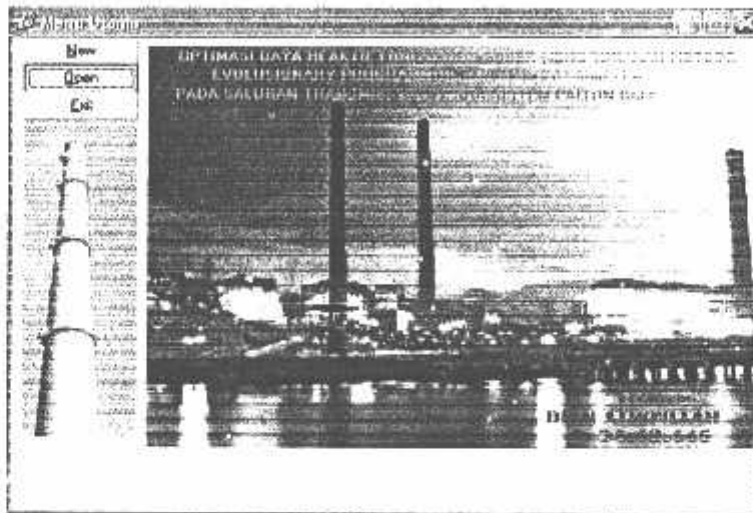
Tabel 4.9.^[15]
Data Saluran Transmisi 150 kV Sub-Sistem Paiton-Bali

No	Hubungan Bus	Jumlah Sirkuit	R (pu)	X (pu)	B (pu)
1	1 – 2	2	0,0053	0,0180	0,0065
2	1 – 9	2	0,0477	0,0346	0,0251
3	2 – 3	1	0,0100	0,0340	0,0123
4	2 – 4	2	0,0079	0,0269	0,0097
5	4 – 5	2	0,0135	0,0145	0,0166
6	5 – 6	1	0,0167	0,0571	0,0206
7	5 – 7	1	0,0315	0,1077	0,0389
8	6 – 7	1	0,0148	0,0506	0,0189
9	7 – 8	1	0,0337	0,1151	0,0415
10	7 – 10	2	0,0104	0,0353	0,0128
11	7 – 11	1	0,0430	0,1469	0,0530
12	8 – 11	1	0,0177	0,0604	0,0218
13	9 – 10	2	0,0091	0,0310	0,0112
14	9 – 11	2	0,0097	0,0464	0,0340
15	11 – 12	2	0,0009	0,0013	0,0364
16	12 – 13	2	0,0116	0,0336	0,0124
17	12 – 17	1	0,0231	0,0690	0,0476
18	13 – 14	1	0,0270	0,0783	0,0288
19	13 – 15	1	0,0412	0,1194	0,0439
20	14 – 15	1	0,0412	0,0411	0,0151
21	15 – 16	1	0,0232	0,0673	0,0247
22	15 – 17	1	0,0556	0,1054	0,0369
23	15 – 18	1	0,0132	0,0383	0,0141
24	15 – 19	1	0,0105	0,0304	0,0112
25	15 – 22	1	0,0199	0,0379	0,0130
26	15 – 24	2	0,0059	0,0170	0,0063
27	16 – 17	1	0,0183	0,0350	0,0120
28	18 – 19	1	0,0046	0,0134	0,0049
29	19 – 20	1	0,0020	0,0044	-
30	19 – 21	1	0,0020	0,0056	-
31	19 – 22	1	0,0127	0,0243	0,0083
32	19 – 23	2	0,0029	0,0069	0,0024
33	23 – 24	2	0,0061	0,0147	0,0052
34	24 – 25	2	0,0103	0,0298	0,0110

4.6. Prosedur Pelaksanaan Program Perhitungan

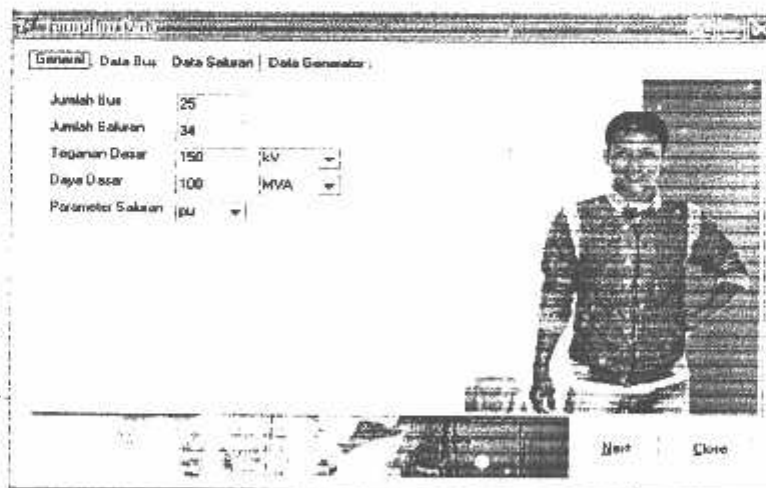
Prosedur menjalankan program perhitungan dengan menggunakan bahasa pemrograman Borland Delphi versi 7.0 dapat dilakukan sebagai berikut :

1. Tampilan utama dari program.



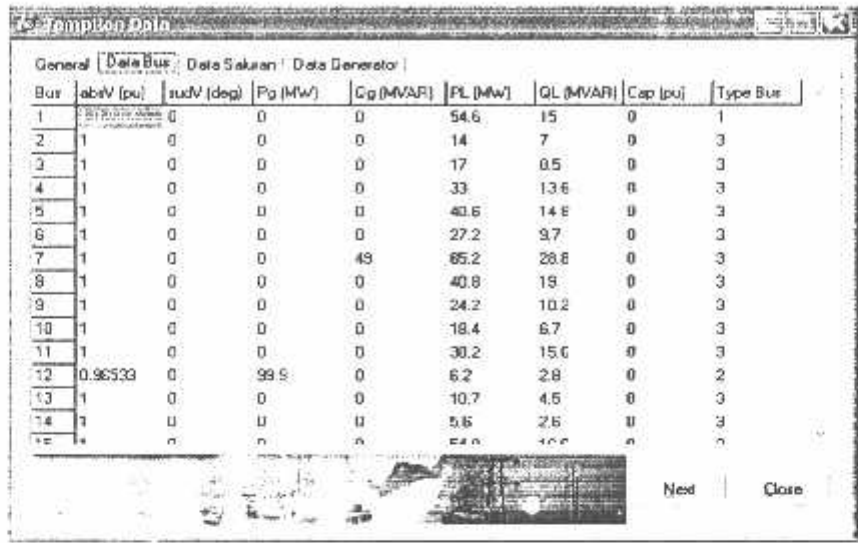
Gambar 4.8.
Tampilan Utama Program

2. Tekan tombol *General* untuk membuka data yang sudah tersimpan.



Gambar 4.9.
Tampilan Masukan Data

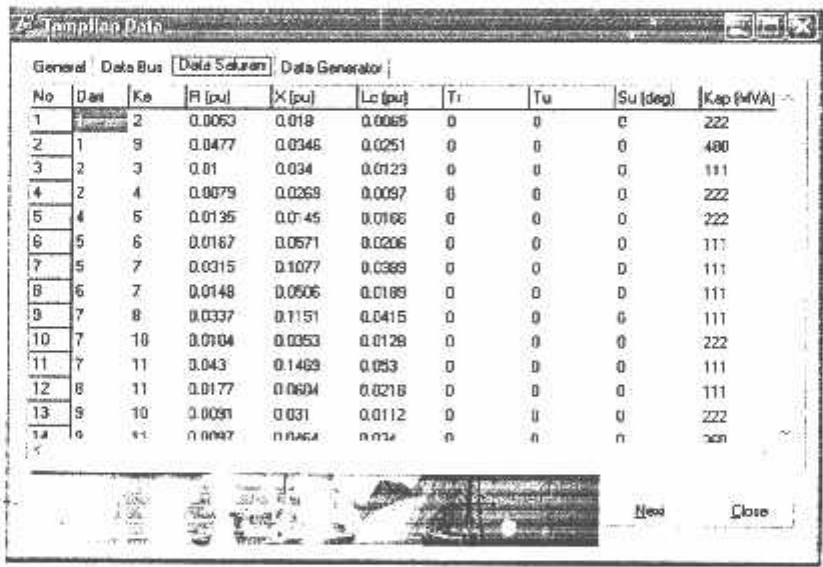
3. Kemudian tekan tombol data bus



Bus	absV (pu)	sudV (deg)	Pg (MW)	Qg (MVAR)	PL (MW)	QL (MVAR)	Cap (pu)	Type Bus
1	1	0	0	0	54.6	15	0	1
2	1	0	0	0	14	7	0	3
3	1	0	0	0	17	0.5	0	3
4	1	0	0	0	33	13.6	0	3
5	1	0	0	0	40.6	14.8	0	3
6	1	0	0	0	27.2	9.7	0	3
7	1	0	0	49	65.2	28.8	0	3
8	1	0	0	0	40.8	19	0	3
9	1	0	0	0	24.2	10.2	0	3
10	1	0	0	0	18.4	6.7	0	3
11	1	0	0	0	30.2	15.0	0	3
12	0.96539	0	99.9	0	6.2	2.8	0	2
13	1	0	0	0	10.7	4.5	0	3
14	1	0	0	0	5.6	2.6	0	3
15	1	0	0	0	54.6	15	0	3

Gambar 4.10.
Tampilan Data Bus

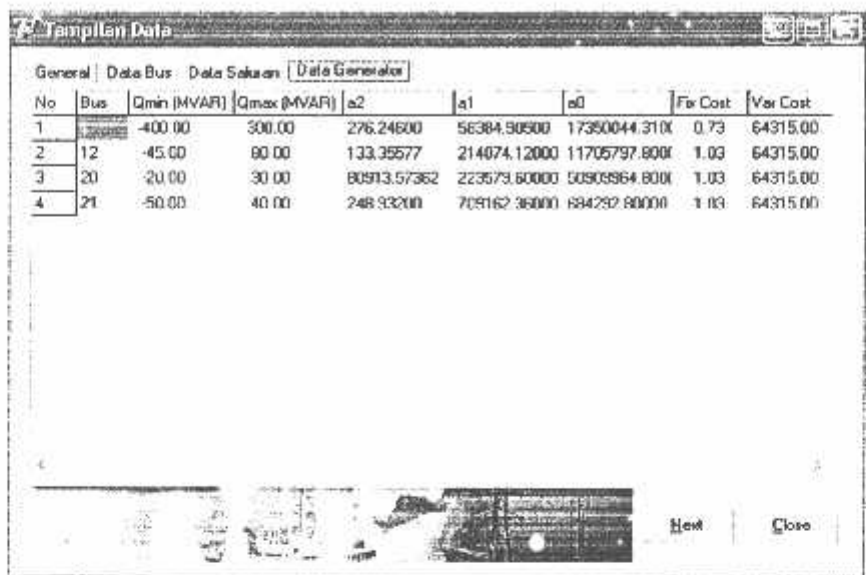
4. Kemudian tekan tombol data saluran



No	Das	Ka	R (pu)	X (pu)	G (pu)	B (pu)	Ti	Tu	Su (deg)	Kap (MVA)
1	1	2	0.0053	0.018	0.0065	0	0	0	0	222
2	1	9	0.0477	0.0348	0.0251	0	0	0	0	480
3	2	3	0.01	0.034	0.0123	0	0	0	0	111
4	2	4	0.0079	0.0269	0.0097	0	0	0	0	222
5	4	5	0.0135	0.0145	0.0166	0	0	0	0	222
6	5	6	0.0167	0.0571	0.0206	0	0	0	0	111
7	5	7	0.0315	0.1077	0.0389	0	0	0	0	111
8	6	7	0.0148	0.0506	0.0189	0	0	0	0	111
9	7	8	0.0337	0.1151	0.0415	0	0	0	0	111
10	7	10	0.0104	0.0353	0.0128	0	0	0	0	222
11	7	11	0.043	0.1469	0.053	0	0	0	0	111
12	8	11	0.0177	0.0604	0.0216	0	0	0	0	111
13	9	10	0.0091	0.031	0.0112	0	0	0	0	222
14	9	11	0.0097	0.034	0.012	0	0	0	0	222

Gambar 4.11.
Tampilan Data Saluran

5. Kemudian tekan data generator



No	Bus	Qmin (MVAR)	Qmax (MVAR)	a2	a1	a0	Fir Cost	Var Cost
1	1	-400.00	300.00	276.24600	56384.90500	17350044.3100	0.73	64315.00
2	12	-45.00	90.00	133.35577	214074.12000	11705797.8000	1.03	64315.00
3	20	-20.00	30.00	80913.57362	223573.60000	50903964.8000	1.03	64315.00
4	21	-50.00	40.00	248.93200	709162.36000	684292.80000	1.03	64315.00

Gambar 4.12.
Tampilan Data Generator

6. Tekan tombol Parameter untuk melihat Parameter *Evolutionary Programming* dan *Quasy – Simplex* Pada kondisi awal (Sebelum Optimasi)

TPQS

Parameter | Loadflow Awal | Aliran Daya | Summary | Hasil Program | Loadflow Akhir | Aliran Daya | Summary | < >

Parameter EPQS

Jumlah Generasi: 100

Jumlah Populasi: 50

Probabilitas Crossover: 0.4

Probabilitas Mutasi: 0.6

Probabilitas Flip: 0.5

Konstanta Delta: 0.2

Konstanta Ka: 100000

Panjang Chromosome: 30

Parameter Objective Function

Lambda V: 1000000

Lambda S: 100000

Lambda Q: 100000

Batas V: 0.98 1.05

Use Default

Hitung Close

Gambar 4.13.
Tampilan Parameter EPQS pada kondisi awal (Sebelum Optimasi)

7. Tekan *load flow* awal bus untuk melihat hasil perhitungan *Load Flow* pada kondisi awal (sebelum optimasi).

Bus	absV (pu)	angV (deg)	Pg (MW)	Qg (MVAR)	PL (MW)	QL (MVAR)	Supr (pu)	Type Bus
1	1.00000	0.00000	582.157	57.473	54.600	15.000	0.000	1
2	1.00793	-0.04049	0.000	0.000	14.000	7.000	0.000	3
3	1.00377	-0.04548	0.000	0.000	17.000	8.500	0.000	3
4	0.98625	-0.03690	0.000	0.000	33.000	13.600	0.000	3
5	0.96561	-0.12304	0.000	0.000	40.600	14.800	0.000	3
6	0.95843	-0.16809	0.000	0.000	27.200	9.700	0.000	3
7	0.96056	-0.19413	0.000	49.000	65.200	28.900	0.000	3
8	0.96539	-0.24639	0.000	0.000	40.800	18.000	0.000	3
9	0.94313	-0.16640	0.000	0.000	24.200	10.200	0.000	3
10	0.94942	-0.17680	0.000	0.000	18.400	6.700	0.000	3
11	0.96422	-0.24965	0.000	0.000	38.200	15.600	0.000	3
12	0.98533	-0.25335	90.000	170.909	6.200	2.800	0.000	2
13	0.96951	-0.28915	0.000	0.000	10.700	4.500	0.000	3
14	0.95827	-0.34956	0.000	0.000	5.600	2.600	0.000	3
15	0.93900	-0.37591	0.000	0.000	54.900	16.600	0.000	3

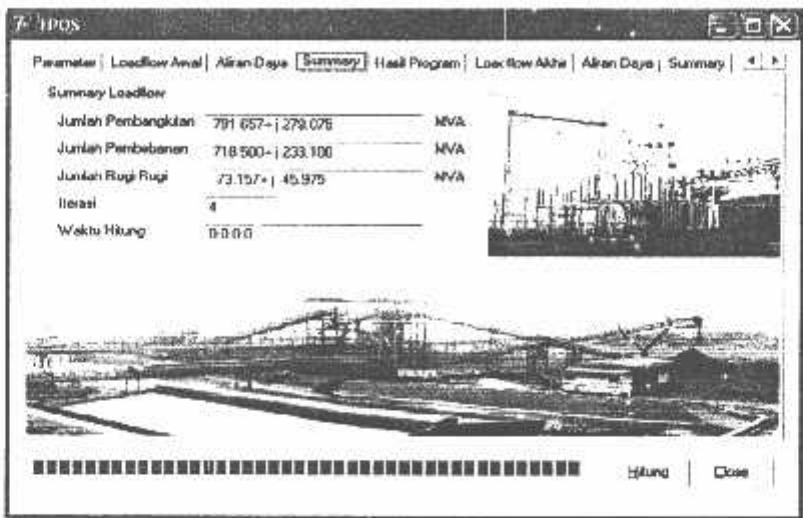
Gambar 4.14.
Tampilan Hasil *Load Flow* Pada Kondisi Awal (Sebelum Optimasi)

8. Tekan aliran daya saluran untuk melihat kondisi aliran daya pada kondisi awal (sebelum optimasi).

No	Dari	Ke	P (MW)	Q (MVAR)	Asus ke (A)	Asus in (A)	Dari	Ke	P (MW)	Q
1	1	2	244.370	30.934	1586.874	252.948	2	1	-241.297	-
2	1	9	283.177	-111.427	1838.607	-723.559	9	1	-241.533	1
3	2	3	17.034	6.127	110.533	45.049	3	2	-17.000	-
4	2	4	210.264	16.706	1385.115	166.704	4	2	-206.801	-
5	4	5	173.801	-6.751	1171.428	66.934	5	4	-169.522	-
6	5	6	70.877	-8.897	493.147	-0.632	6	5	69.968	-
7	5	7	58.144	-14.014	410.273	-46.753	7	5	-56.966	-
8	6	7	42.769	-12.950	314.068	-72.858	7	6	-42.432	-
9	7	8	37.986	-12.795	282.554	-70.303	8	7	-37.390	-
10	7	10	-32.806	39.448	-276.201	224.893	10	7	33.114	-
11	7	11	29.008	-27.799	234.743	-150.460	11	7	-28.371	-
12	8	11	-3.402	-31.141	28.698	-214.290	11	8	3.565	-
13	8	10	51.000	-34.710	400.457	-165.611	10	8	51.514	-
14	9	11	165.420	-112.246	1278.066	602.876	11	9	-161.142	1

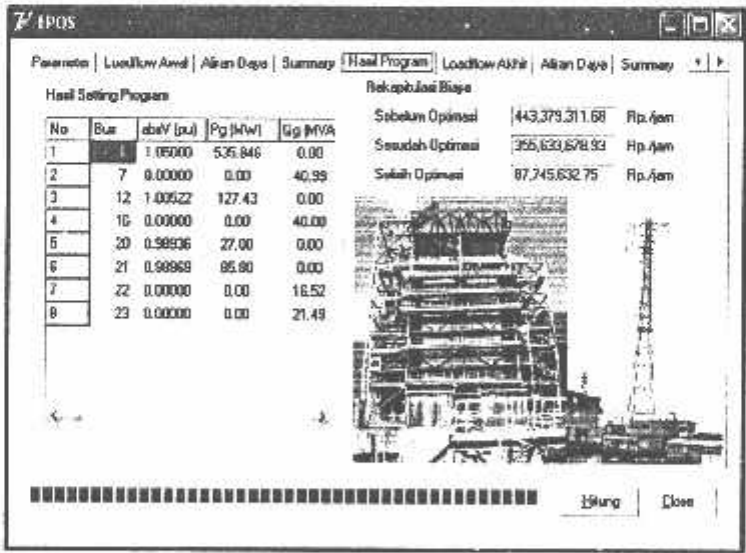
Gambar 4.15.
Tampilan Hasil Aliran Daya Pada Kondisi Awal (Sebelum Optimasi)

9. Tekan tombol *summary*, untuk mengetahui jumlah pembangkitan dan jumlah pembebanan serta rugi – rugi sebelum optimasi.



Gambar 4.16.
Tampilan *summary load flow* sebelum optimasi

10. Kemudian tekan tombol Hasil Program, untuk melihat Hasil Program pada kondisi akhir (setelah optimasi).



Gambar 4.17.
Tampilan Hasil Program pada kondisi akhir (Setelah Optimasi)

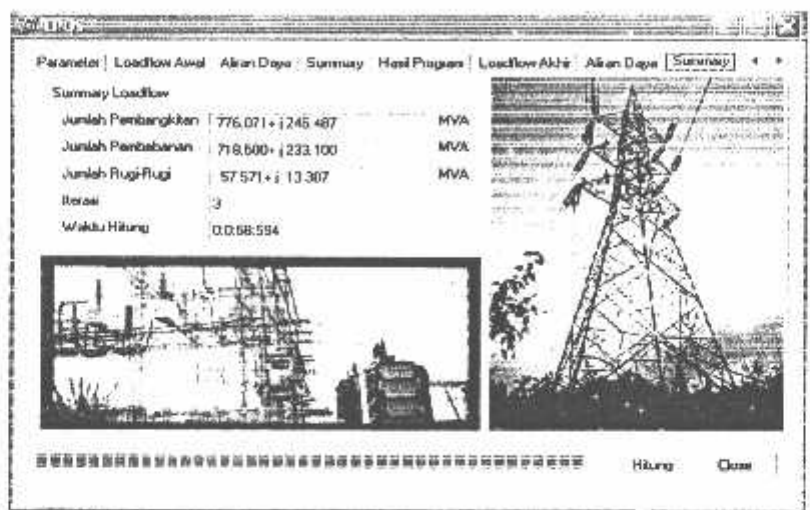
9. Kemudian tekan tombol *load flow* akhir untuk melihat kondisi *load flow* pada kondisi akhir (setelah optimasi).

Gambar 4.18.
Tampilan hasil *Load Flow* pada kondisi akhir (Setelah Optimasi).

10. Kemudian tekan tombol aliran daya, untuk melihat aliran daya (setelah optimasi).

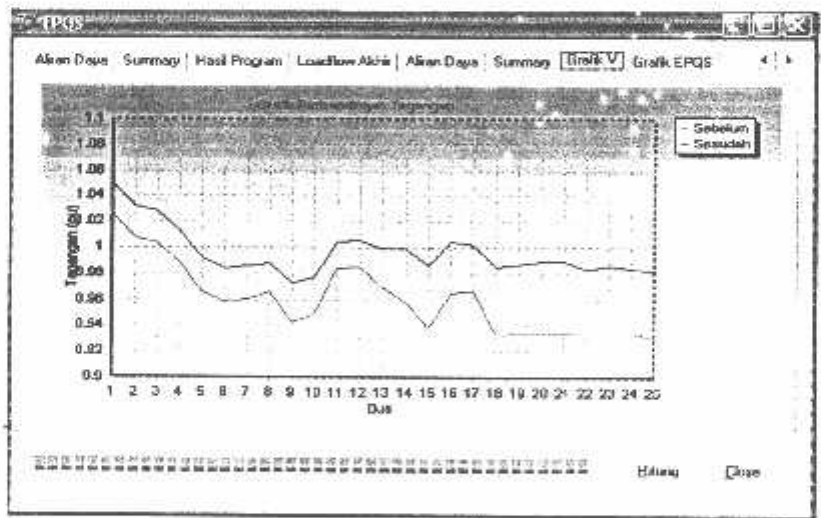
Gambar 4.19.
Tampilan aliran daya pada kondisi akhir (Setelah Optimasi)

13. Kemudian tekan tombol *summary*, untuk melihat jumlah pembangkitan dan jumlah pembebanan serta rugi - rugi setelah optimasi.



Gambar 4. 20.
Tampilan *Summary Load Flow* setelah optimasi

14. Kemudian tekan tombol Grafik V, untuk melihat nilai tegangan sebelum dan setelah optimasi.



Gambar 4. 21.
Tampilan nilai tegangan sebelum dan setelah optimasi

4.7. Hasil dan Analisis Hasil Perhitungan Optimasi Daya Reaktif
Menggunakan Metode *Evolutionary Programming dan Quasy – Simplex*
Pada Saluran Transmisi 150 kV Sub-Sistem Paiton-Bali

4.7.1. Hasil Perhitungan Sebelum Optimasi

Tabel 4.10,
Hasil Perhitungan Tegangan, Sudut Tegangan, Pembangkitan
dan Pembebanan Sebelum Optimasi

No. Bus	Tegangan (pu)	Sudut Tegangan (deg)	Pembangkitan		Pembebanan	
			Aktif (MW)	Reaktif (MVAR)	Aktif (MW)	Reaktif (MVAR)
1	1,02667	0,00000	582.157	-57.473	54,6	15,0
2	1,00793	-0,04049	0,000	0,000	14,0	7,0
3	1,00377	-0,04548	0,000	0,000	17,0	8,5
4	0,98825	-0,09590	0,000	0,000	33,0	13,6
5	0,96561	-0,12304	0,000	0,000	40,6	14,8
6	0,9843	-0,16803	0,000	0,000	27,2	9,7
7	0,96066	-0,19413	0,000	49,000	65,2	28,8
8	0,96413	-0,24639	0,000	0,000	40,8	19,0
9	0,94313	-0,15540	0,000	0,000	24,2	10,2
10	0,94942	-0,17680	0,000	0,000	18,4	6,7
11	0,98422	-0,24965	0,000	0,000	38,2	15,6
12	0,98533	-0,25335	99.900	178.988	6,2	2,8
13	0,96951	-0,29915	0,000	0,000	10,7	4,5
14	0,95827	-0,34956	0,000	0,000	5,6	2,6
15	0,93900	-0,37591	0,000	0,000	54,9	16,6
16	0,96483	-0,35385	0,000	40,000	4,3	0,9
17	0,96645	-0,33396	0,000	0,000	24,5	7,5
18	0,93307	-0,38481	0,000	0,000	31,2	12,5
19	0,93410	-0,38372	0,000	0,000	71,8	27,6
20	0,93533	-0,38174	42.500	6.847	0	0
21	0,93533	-0,37934	67,100	-3.287	0	0
22	0,93385	-0,38929	0,000	25,000	45,5	16,4
23	0,93386	-0,42498	0,000	25,000	45,9	12,7
24	0,93435	-0,38571	0,000	0,000	31,7	5,4
25	0,93171	-0,38971	0,000	0,000	13,0	4,7

Tabel 4.11.
 Hasil Perhitungan Aliran Daya Antar Saluran Sebelum Optimasi

Saluran		Daya		Saluran		Daya	
Dari	Ke	P (MW)	Q (Mvar)	Dari	Ke	P (MW)	Q (Mvar)
1	2	244.379	38.954	2	1	-241.297	-29.833
1	9	283.177	-111.427	9	1	-241.533	136.756
2	3	17.034	6.127	3	2	-17.000	-8.500
2	4	210.264	16.706	4	2	-206.801	-6.850
4	5	173.801	-6.751	5	4	-169.622	8.070
5	6	70.877	-8.857	6	5	-69.969	8.150
5	7	58.144	-14.014	7	5	-56.966	10.826
6	7	42.769	-17.850	7	6	-42.432	15.520
7	8	37.996	-17.795	8	7	-37.398	12.141
7	10	-32.806	39.448	10	7	33.114	-40.740
7	11	29.008	-27.799	11	7	-28.371	19.949
8	11	-3.402	-31.141	11	8	3.565	27.555
9	10	51.906	-34.710	10	9	-51.514	34.040
9	11	165.428	-112.246	11	9	-161.142	126.426
11	12	147.749	-189.529	12	11	-147.225	183.227
12	13	131.532	2.760	13	12	-129.463	0.863
12	17	109.392	-9.798	17	12	-106.539	9.255
13	14	58.188	-7.350	14	13	-57.209	4.837
13	15	60.576	1.987	15	13	-58.951	-5.275
14	15	51.609	-7.437	15	14	-50.398	5.927
15	16	-37.858	-24.832	16	15	38.371	21.840
15	17	-38.024	-6.887	17	15	38.944	1.931
15	18	22.716	5.579	18	15	-22.632	-7.805
15	19	24.834	5.668	19	15	-24.755	-7.405
15	22	29.613	-3.721	22	15	-29.414	1.821
15	24	53.167	6.941	24	15	-52.974	-7.491
16	17	-42.671	17.260	17	16	43.095	-18.686
18	19	-8.568	-4.695	19	18	8.573	3.854
19	20	-42.458	-6.754	20	19	42.500	6.847
19	21	-66.997	3.576	21	19	67.100	-3.287
19	22	16.132	-8.139	22	19	-16.086	6.779
19	23	37.705	-12.733	23	19	-37.652	12.439
23	24	-8.248	-0.139	24	23	8.252	-0.757
24	25	13.022	2.848	25	24	-13.000	-4.700

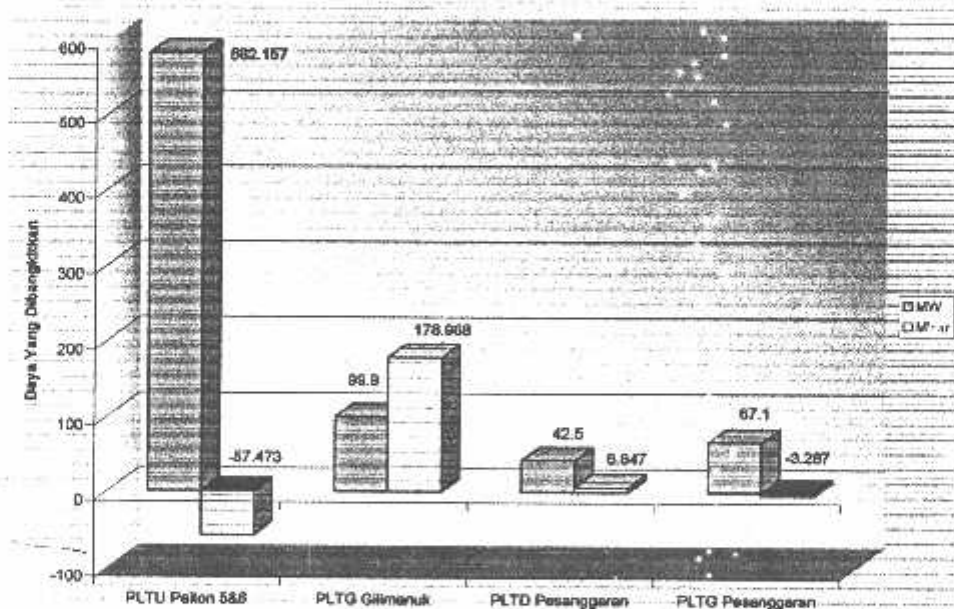
Tabel 4.12.
Total Pembangkitan, Pembebanan dan Rugi-rugi Saluran Sebelum Optimasi

Pembangkitan		Pembebanan		Rugi-rugi		Waktu Komputasi
Aktif (MW)	Reaktif (MVAR)	Aktif (MW)	Reaktif (MVAR)	Aktif (MW)	Reaktif (MVAR)	
791.657	264.075	718.500	246.170	73.157	17.905	0:0:20

Tabel 4.13.
Hasil Perhitungan Daya Yang Dibangkitkan dan Biaya Operasi Sebelum Optimasi

No	Nama Pembangkit	Daya Yang Dibangkitkan		Biaya Operasi (Rupiah/jam)
		(MW)	(MVar)	
1	PLTU Paiton 5 & 6	582.157	-57.473	Rp 144.960.865,77
2	PLTG Gilimanuk	99.900	178.988	Rp 37.503.754,08
3	PLTD Pesanggaran	42.500	6.847	Rp 27.616.539,30
4	PLTG Pesanggaran	67.100	-3.287	Rp 137.689.453,08
Total				Rp 347.770.612,23

Grafik 4.1
Hasil Perhitungan Daya Yang Dibangkitkan Sebelum Optimasi



4.7.2. Hasil Perhitungan Setelah Optimasi

Tabel 4.14.
Hasil Perhitungan Tegangan, Sudut Tegangan, Pembangkitan Dan
Pembebanan Setelah Optimasi

No. Bus	Tegangan (pu)	Sudut Tegangan (deg)	Pembangkitan		Pembebanan	
			Aktif (MW)	Reaktif (MVAR)	Aktif (MW)	Reaktif (MVAR)
1	1.05000	0.00000	561.471	-50.303	54.6	15.0
2	1.03212	-0.03746	0.000	0.000	14.0	7.0
3	1.02808	-0.04223	0.000	0.000	17.0	8.5
4	1.01318	-0.08836	0.000	0.000	33.0	13.6
5	0.99170	-0.11291	0.000	0.000	40.6	14.8
6	0.98435	-0.15352	0.000	0.000	27.2	9.7
7	0.98593	-0.17641	0.000	49.000	65.2	28.8
8	0.98778	-0.22262	0.000	0.000	40.8	19.0
9	0.96989	-0.13928	0.000	0.000	24.2	10.2
10	0.97556	-0.15975	0.000	0.000	18.4	6.7
11	1.00445	-0.22386	0.000	0.000	38.2	15.6
12	1.00534	-0.22705	133.800	102.052	6.2	2.8
13	0.99878	-0.27811	0.000	0.000	10.7	4.5
14	0.99899	-0.33244	0.000	0.000	5.6	2.6
15	0.98541	-0.36463	0.000	0.000	54.9	16.6
16	1.00277	-0.33758	0.000	39.229	4.3	0.9
17	1.00107	-0.31529	0.000	0.000	24.5	7.5
18	0.98445	-0.37635	0.000	0.000	31.2	12.5
19	0.98705	-0.37665	0.000	0.000	71.8	27.6
20	0.98995	-0.37391	75.000	31.255	0	0
21	0.98082	-0.37704	15.000	61.303	0	0
22	0.98107	-0.37802	0.000	3.042	45.5	16.4
23	0.98501	-0.37849	0.000	12.693	45.9	12.7
24	0.98344	-0.37541	0.000	0.000	31.7	5.4
25	0.98096	-0.37904	0.000	0.000	13.0	4.7

Tabel 4.15.
Hasil Perhitungan Aliran Daya Antar Saluran Setelah Optimasi

Saluran		Daya		Saluran		Daya	
Dari	Ke	P (MW)	Q (Mvar)	Dari	Ke	P (MW)	Q (Mvar)
1	2	236.897	38.033	2	1	-234.127	-30.034
1	9	269.974	-103.336	9	1	-234.256	124.256
2	3	17.032	5.999	3	2	-17.000	-8.500
2	4	203.095	17.036	4	2	-200.012	-8.567
4	5	167.013	-5.031	5	4	-163.344	5.636
5	6	67.747	-7.669	6	5	-66.926	6.330
5	7	54.995	-12.769	7	5	-54.001	8.562
6	7	39.762	-16.030	7	6	-39.489	13.293
7	8	35.811	-15.199	8	7	-35.324	8.782
7	10	-33.825	38.066	10	7	34.113	-39.552
7	11	26.305	-24.521	11	7	-25.833	21.378
8	11	-5.476	-27.782	11	8	5.601	28.696
9	10	52.887	-33.698	10	9	-52.513	-32.852
9	11	156.977	-100.758	11	9	-153.455	110.979
11	12	135.487	-166.091	12	11	-135.088	159.316
12	13	143.530	-27.265	13	12	-141.088	31.848
12	17	119.158	-32.799	17	12	-115.734	33.445
13	14	62.338	-22.756	14	13	-61.179	20.370
13	15	68.050	-13.592	15	13	-66.102	10.594
14	15	55.579	-22.970	15	14	-54.114	21.458
15	16	-43.155	-12.409	16	15	43.624	-8.888
15	17	-41.685	4.904	17	15	42.722	-10.220
15	18	27.358	-8.157	18	15	-27.250	5.735
15	19	32.790	-17.508	19	15	-32.644	15.750
15	22	31.511	-6.292	22	15	-31.302	4.176
15	24	58.497	-9.191	24	15	-58.284	8.528
16	17	-47.923	29.442	17	16	48.512	-30.725
18	19	-3.950	-18.235	19	18	3.966	17.329
19	20	-74.866	-30.959	20	19	75.000	31.255
19	21	-14.919	-61.076	21	19	15.000	61.303
19	22	14.262	16.049	22	19	-14.198	-17.535
19	23	32.402	15.307	23	19	-32.363	-15.683
23	24	-13.537	15.676	24	23	13.565	-16.616
24	25	13.020	2.634	25	24	-13.000	-4.700

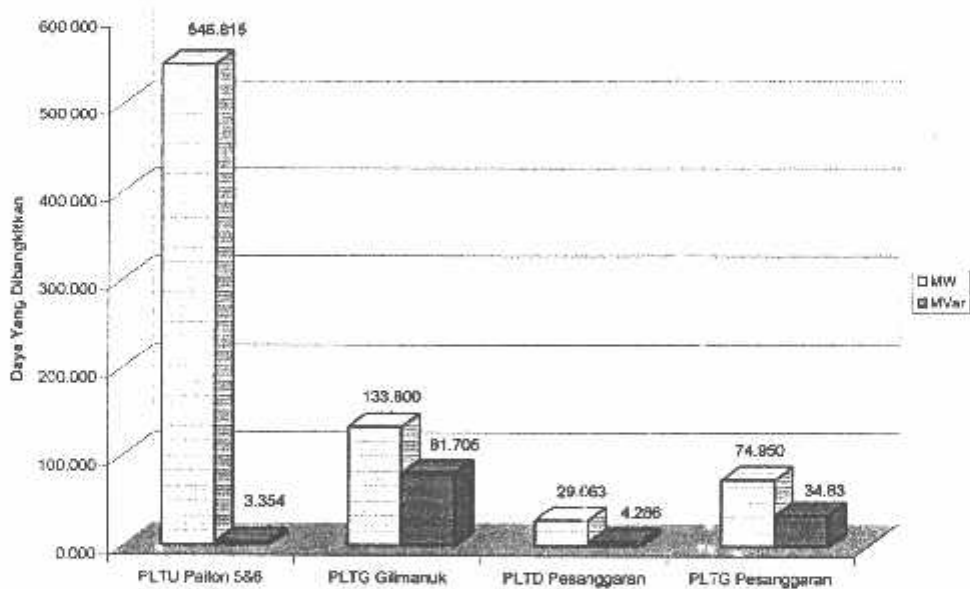
Tabel 4.16.
Total Pembangkitan, Pembebanan dan Rugi-rugi Saluran Setelah Optimasi

Pembangkitan		Pembebanan		Rugi-rugi		Waktu Komputasi
Aktif (MW)	Reaktif (MVAR)	Aktif (MW)	Reaktif (MVAR)	Aktif (MW)	Reaktif (MVAR)	
785.880	254.453	718.500	246.170	66.380	8.283	0:3:44:844

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 5 & 6	546.815	3.354	Rp 131,875,175.58
2	PLTG Gilimanuk	133.800	81.705	Rp 41,920,467.57
3	PLTD Pesanggaran	29.063	4.286	Rp 15,024,173.25
4	PLTG Pesanggaran	74.950	34.830	Rp 97,496,925.31
	Total			Rp 268.316.741,71

Grafik 4.2.
Hasil Perhitungan Daya Yang Dibangkitkan Setelah Optimasi



4.8. Perbandingan Hasil Perhitungan Sebelum dan Setelah Optimasi Daya Reaktif Menggunakan Metode *Evolutionary Programming* dan *Quasy - Simplex*.

Dari semua hasil perhitungan diatas maka bisa dibuat tabel-tabel perbandingan sesuai dengan tujuan yang ingin dicapai.

4.8.1. Tingkat Optimum Biaya Pembangkitan

Berikut ini adalah tabel yang berisi perbandingan biaya optimum pembangkitan antara hasil perhitungan sebelum dengan sesudah optimasi.

Tabel 4.18.
Perbandingan Tingkat Optimum Biaya Pembangkitan

No	Nama Pembangkit	Sebelum Optimasi	Setelah Optimasi
1	PLTU Paiton 5& 6	Rp 144.960.865,77	Rp 131,875,175.58
2	PLTG Gilimanuk	Rp 37.503.754,08	Rp 41,920,467.57
3	PLTD Pesanggaran	Rp 27.616.539,30	Rp 15,024,173.25
4	PLTG Pesanggaran	Rp 137.689.453,08	Rp 97,496,925.31
	Total	Rp 347.770.612,23	Rp 268.316.741,71

Dari tabel tingkat optimum diatas dapat dilihat bahwa besarnya biaya produksi pembangkitan sebelum optimasi adalah Rp 347.770.612,23 sedangkan besarnya biaya produksi pembangkitan setelah optimasi adalah Rp 268.316.741,71 Jadi selisih antara biaya produksi pembangkitan sebelum dengan sesudah optimasi adalah Rp 61,453,870.52 atau terjadi optimasi biaya sebesar 24,77%

4.8.2. Perbandingan Tingkat Tegangan dan Sudut Tegangan Pada Tiap Bus

Berikut ini adalah tabel perbandingan tingkat tegangan dan sudut tegangan pada tiap bus.

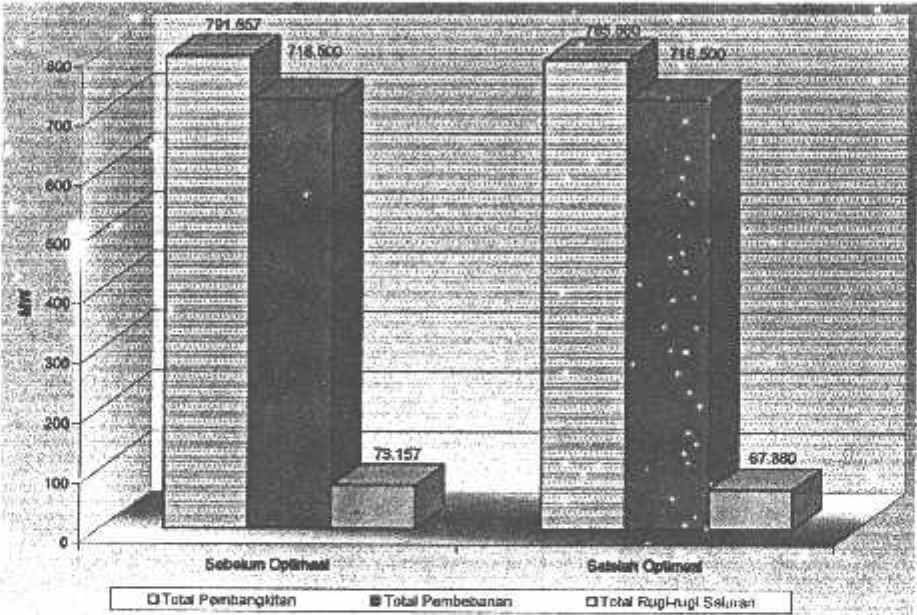
Tabel 4.19
Perbandingan Tingkat Tegangan dan Sudut Tegangan Pada Tiap Bus

No.	Sebelum Optimasi		Setelah Optimasi	
	Tegangan (pu)	Sudut Tegangan (deg)	Tegangan (pu)	Sudut Tegangan (deg)
1	1,02667	0,00000	1,05000	0,00000
2	1,00793	-0,04049	1,03212	-0,03746
3	1,00377	-0,04548	1,02808	-0,04223
4	0,98825	-0,09590	1,01318	-0,08836
5	0,96561	-0,12304	0,99170	-0,11291
6	0,9843	-0,16803	0,98435	-0,15352
7	0,96066	-0,19413	0,98593	-0,17641
8	0,96413	-0,24639	0,98778	-0,22262
9	0,94313	-0,15540	0,96989	-0,13928
10	0,94942	-0,17680	0,97556	-0,15975
11	0,98422	-0,24965	1,00445	-0,22386
12	0,98533	-0,25335	1,00534	-0,22705
13	0,96951	-0,29915	0,99878	-0,27811
14	0,95827	-0,34956	0,99899	-0,33244
15	0,93900	-0,37591	0,98541	-0,36463
16	0,96483	-0,35385	1,00277	-0,33758
17	0,96645	-0,33396	1,00107	-0,31529
18	0,93307	-0,38481	0,98445	-0,37635
19	0,93410	-0,38372	0,98705	-0,37665
20	0,93533	-0,38174	0,98995	-0,37391
21	0,93533	-0,37934	0,98082	-0,37704
22	0,93385	-0,38929	0,98107	-0,37802
23	0,93386	-0,42498	0,98501	-0,37849
24	0,93435	-0,38571	0,98344	-0,37541
25	0,93171	-0,38971	0,98096	-0,37904

4.8.3. Perbandingan Tingkat Rugi-rugi Daya Pada Saluran

Untuk rugi-rugi daya pada saluran sebelum dioptimasi adalah 73,157 MW dan 17,905 MVAR, sedangkan perhitungan rugi-rugi daya setelah optimasi adalah 67,380 MW dan 8,283 MVAR. Dari hasil perhitungan dapat dilihat bahwa rugi-rugi daya yang ada sebelum dan sesudah optimasi terjadi penurunan yaitu sebesar 5.777 MW dan 9.622 MVAR.

Grafik 4.3.
Perbandingan Total Pembangkitan, Pembebanan dan Rugi-rugi Total
Sebelum dan Setelah Optimasi



BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Setelah dilakukan analisa perhitungan optimasi daya reaktif dengan menggunakan metode *Evolutionary Programming dan Quasy - Simplex* maka dapat diambil kesimpulan sebagai berikut :

1. Penyelesaian permasalahan optimasi daya reaktif menggunakan metode *Evolutionary Programming dan Quasy - Simplex* pada Sub-Sistem 150 kV Paiton-Bali menghasilkan biaya operasi pembangkitan yang lebih optimum. Biaya produksi pembangkitan sebelum optimasi adalah Rp 347.770.612,23 per jam sedangkan besarnya biaya produksi pembangkitan setelah optimasi adalah Rp.268.316.741,71 per jam. Jadi selisih antara biaya produksi pembangkitan sebelum dengan sesudah optimasi adalah Rp. Rp 61,453,870.52 per jam atau terjadi optimasi biaya sebesar 24,77%.
2. Dengan mengoptimasi daya reaktif menggunakan metode *Evolutionary Programming dan Quasy - Simplex* ternyata rugi-rugi daya sistem turun dari $73.157 + j 45,975$ MVA menjadi $55,520 + j 9,902$ MVA.
3. Untuk tingkat tegangan didapatkan hasil bahwa profil tegangan setelah optimasi daya reaktif menggunakan metode *Evolutionary Programming dan Quasy - Simplex* masih dalam batas-batas yang diijinkan yaitu antara -5 % sampai +5 % dari tegangan nominal.

5.2. Saran

Penggunaan metode *Evolutionary Programming* dan *Quasy - Simplex* 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 secara lebih efektif dan efisien dalam pemecahan permasalahan optimasi biaya pembangkitan.

DAFTAR PUSTAKA

- [1]. Nidul Sinha, R.Cakrabarti, And P.K.Chattopadhyay, *Evolutionary Programming Techniques for Economic Load Dispatch*, IEEE Trans. Evolutionary Computation, Vol.7, No 1 Feb. 2003.
 - [2]. Xin Yao and Young Liu, *Evolutionary Programming Made Fast*, IEEE Trans. Evolutionary Computation, Vol.3, No. 2, pp.82-102, July, 1999.
 - [3]. Nelder, J.A. and MEAD, R.A *Simplex method for function minimization*, the computer journal, 5, 1965
 - [4]. Stevenson, William D., Jr, "*Analisa Sistem Tenaga Listrik*", Penerbit Erlangga edisi keempat, 1996.
 - [5]. Wood, Allan J. and B. F. Wollenberg, "*Power Generation, Operation, and Control*", John Willey & Sons, Inc., 1996.
 - [6]. Zuhail, "*Dasar Teknik Tenaga Listrik dan Elektronika Daya*", PT. Gramedia Utama, Jakarta, 1995.
 - [7]. Stagg, Glenn W. and Ahmed H. El-Abiad "*Computer Methods In Power System Analysis*".
 - [8]. 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.
 - [9]. <http://zoeland.alife.org/hhg2ec/>, "*What's Evolutionary Programming (EP)?*".
 - [10]. Kusumadewi, sri, "*artificial intelligence (Teknik dan Aplikasinya)*", Graha Ilmu, Yogyakarta, 2003.
 - [11]. Syafena, Jatri, Skripsi, "*Economic Dispatch Dengan Optimasi Daya Aktif dan Reaktif Pada Pembangkit Thermal di Area IV*", 2002.
 - [12]. Data "*Load Flow Region Jawa Timur dan Bali*", Sub Sistem 150 kV Paiton dan Bali Rabu 27 Desember 2006, Pukul 19:30 WIB, PT. PLN (Persero) P3B Region Jawa dan Bali, Waru.
 - [13]. Data Saluran Tranmisi 150 kV Sub Sistem Paiton - Bali
-

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 : DIAN ATHO'ILLAH
N.I.M. : 01.12.145
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Energi Listrik
Judul Skripsi : OPTIMASI DAYA REAKTIF FUNGSI NON
LINIER MENGGUNAKAN METODE
EVOLUTIONARY PROGRAMMING AND
QUASY - SIMPLEX PADA SALURAN
TRANSMISI 150 KV SUB SISTEM PAITON-BALI.

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



Hari : Kamis
Tanggal : 23 MARET 2007
Dengan Nilai : 75,5 (B+) *B+*

(Ir. Mochtar Asroni, MSME.)

Ketua

Panitia Ujian

(Ir. F. Yudi Limpraptono, MT.)

Sekretaris

Anggota Penguji

(Ir. Eko nurcahyo.MT)

Penguji Pertama

(Ir.Bambang Triatmojo.MT)

Penguji Kedua



PERMOHONAN PERSETUJUAN SKRIPSI

Yang betanda tangan dibawah ini :

Nama : DIAN ATHO'ILLAH
IM : 01.12.145
Semester : 8
Fakultas : Teknologi Industri
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Elektronika / Teknik Energi Listrik
Alamat :

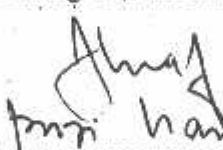
Dengan ini kami mengajukan permohonan untuk mendapatkan persetujuan untuk membuat **SKRIPSI Tingkat Sarjana**. Untuk melengkapi permohonan tersebut, bersama ini lampirkan persyaratan-persyaratan yang harus dipenuhi.

Apapun persyaratan-persyaratan pengambilan **SKRIPSI** adalah sebagai berikut :

Telah melaksanakan semua praktikum sesuai dengan konsentrasinya (.....)
Telah lulus dan menyerahkan Laporan Praktek Kerja (.....)
Telah lulus seluruh mata kuliah keahlian (MKB) sesuai konsentrasinya (.....)
Telah menempuh mata kuliah ≥ 134 sks dengan IPK ≥ 2 dan tidak ada nilai E (.....)
Telah mengikuti secara aktif kegiatan seminar skripsi yang diadakan Jurusan (.....)
Memenuhi persyaratan administrasi (.....)

Dengan permohonan ini untuk mendapatkan penyelesaian lebih lanjut dan atas perhatiannya kami ucapkan terima kasih.

Yang diteliti kebenaran data tersebut diatas
Recording Teknik Elektro

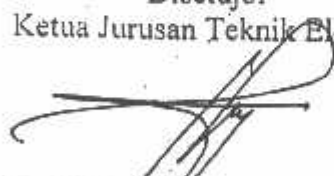

(.....Purni Handayani.....)

Malang,200

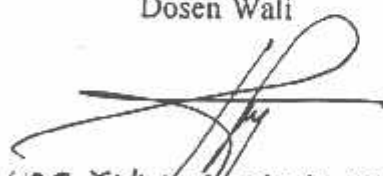
Pemohon


(.....DIAN ATHO'ILLAH.....)

Disetujui
Ketua Jurusan Teknik Elektro


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

Mengetahui
Dosen Wali


(Ir. F. Yudi Limpraptono, MT.)

Atas nama :

Yang mahasiswa yang telah memenuhi persyaratan mengambil SKRIPSI agar membuat proposal dan mendapat persetujuan dari Ketua Jurusan/Sekretaris Jurusan T. Elektro S-1

IPK 2.27 / 2.21
137 / 137



Malang, 06 september 2006

Lampiran : 1 (Satu) Berkas
Perihal : **Kesediaan Sebagai
Dosen Pembimbing**

Kepada : Yth. Bapak/Ibu. *Ir. Churri, MT*
Dosen Jurusan Elektro S-1/ T.Energi Listrik
Institut Teknologi Nasional Malang
di-
Malang

Yang bertandatangan di bawah ini :

Nama	:	Dian Atho'illah
Nim	:	01.12.145
Semester	:	XI (Sebelas)
Jurusan	:	TEKNIK ELEKTRO
Konsentrasi	:	ENERGI LISTRIK S-1

Dengan ini mengajukan permohonan, kiranya Bapak/Ibu bersedia menjadi Dosen Pembimbing Utama, untuk penyusunan Skripsi dengan judul :

**OPTIMASI DAYA REAKTIF FUNGSI NON LINIER
MENGUNAKAN METODE PROGRAMMING
EVOLUTIONARY AND QUASY-SIMPLEX (EPQS) PADA
SALURAN TRANSMISI 150 KV SUB SYSTEM PAITON - BALI**

Seperti proposal terlampir.

Adapun Tugas tersebut sebagai salah satu syarat untuk mendapatkan gelar Sarjana Teknik pada jurusan Teknik Elektro.

Demikian permohonan kami, atas kesediaan Bapak / ibu kami ucapkan terimakasih.

Mengetahui,
Ketua Jurusan
Teknik Elektro

Malang, 06 september 2006

Pemohon,

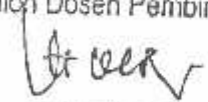
Ir.F.Yudi Limpraptono,MT
Nip.P.1039500274

Dian Atho'illah
Nim : 01.12. 145

Form S-3a

BERITA ACARA SEMINAR PROPOSAL SKRIPSI JURUSAN TEKNIK ELEKTRO S-1

(konsentrasi : Teknik Energi Listrik/Teknik Elektronika*)

1.	Nama Mahasiswa: <u>DIAN ATHO'ILLA H</u>	Nim: <u>01.12.145</u>		
2.	Keterangan	Tanggal	Waktu	Tempat
	Pelaksanaan	<u>04-12-2006</u>		Ruang:
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.	Judul Proposal yang diseminarkan Mahasiswa	<u>Optimalisasi daya Reaktif fungsi dan Linker</u> <u>hingga dengan metode ANALISIS ARBY</u> <u>PROGAMMING and QUASY-SIMPLE X</u>		
5.	Perubahan Judul yang diusulkan oleh Kelompok Dosen Keahlian			
6.	Catatan:			
				
7.	Persetujuan Judul Skripsi			
	Disetujui, Dosen Keahlian I		Disetujui, Dosen Keahlian II	
				
	Mengetahui, Ketua Jurusan.  <u>Ir. F. Yudi Limoraptono MT</u> NIP. P. 1029500274		Disetujui, Calon Dosen Pembimbing ybs  <u>(Ir. H. Choiri</u>)	

Perhatian:

1. Keterangan: *) Coret yang tidak perlu

**) dilingkari a, b, c,atau g sesuai bidang keahlian

Form S-3c



PERSERO) MALANG
: 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

Nomor : ITN-033/I.TA/2/07
Lampiran : -
Perihal : **BIMBINGAN SKRIPSI**

Malang, 12 Januari 2007

Kepada : Yth. Sdr. **Ir. H. CHOIRI**
Dosen Pembimbing
Jurusan T. Elektro S-1
di
Malang

Dengan hormat

Sesuai dengan permohonan dan persetujuan dalam proposal skripsi untuk mahasiswa :

Nama : DIAN ATHO"ILLAH
Nim : 0112145
Fakultas : Teknologi Industri
Jurusan : Teknik Elektro S-1
Konsentrasi : T. Energi Listrik

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

04 Desember 2006 s/d 04 Mei 2007

Sebagai satu syarat untuk menempuh Ujian Sarjana Teknik,
Jurusan Teknik Elektro S-1
Demikian atas perhatian serta kerjasamanya yang baik kami sampaikan terima kasih.



Ketua Jurusan
Teknik Elektro S-1

Ir. F. Yudi, Limpraptono, MT
NIP. Y. 1029500274

Tembusan Kepada Yth:

1. Mahasiswa Yang Bersangkutan
2. Arsip

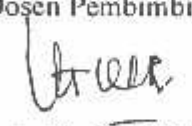
Form S-4a

FORMULIR BIMBINGAN SKRIPSI

Nama : DIAN ATHO'ILLAH
Nim : 01.12.145
Masa Bimbingan : 04 DESEMBER 2006 s/d 04 MEI 2007
Judul Skripsi : OPTIMASI DAYA REAKTIF FUNGSI NON LINIER
MENGUNAKAN METODE EVOLUTIONARY
PROGRAMMING AND QUASY - SIMPLEX (EPQS) PADA
SALURAN TRANSMISI 150 Kv SUB PAITON - BALI

lo.	Tanggal	Uraian	Paraf Pembimbing
1.	10-12-06	Konsultasi BAB. I.	JH
2.	25-12-06	Konsultasi BAB. II (cek rangk. pengantar II).	JH
3.		Konsultasi BAB. III	JH
4.	21-2-07	- setiap gambar harus ada penjelasan - No. gambar & cek 3.4.5 & 3.4.6.	JH
5.	28-2-07	Konsultasi BAB. IV	JH
5.	4-3-07	Konsultasi BAB. V	JH
7.		(cek & seminar team)	JH
3.			
3.			
0.			

Malang, 4-3-07
Dosen Pembimbing,


Ir.H. CHOIRI

SURAT PUAS TUGAS / PRAKTIKUMNomor : ITN 013/PK/D/X / SPT / 2025

Menerangkan bahwa mahasiswa :

Nama : DIAN ATHA'ILLAH Nrp. : 0112145Jurusan : Teknik Elektro / Energi ListrikTingkat : — (—)Telah menyelesaikan tugas : Laporan Pk di PLN. Area Malang
GA UPJ Batu.Semester : — (—)Dengan hasil : Baik (75)Malang, 25 Oktober 2025

Dosen / Asisten ybs.

Colinda
Ir. Made Wiantana, MT

SURAT PUAS TUGAS / PRAKTIKUM

Nomor : ITN 1/TA/III / SPT / 200 7.

erangkan bahwa mahasiswa :

Nama : Dian Atha'illah NIM : 01.12.145
Jurusan : Teknik Elektro Sistem Tenaga
Angkatan : 2001
Telah menyelesaikan tugas : Skripsi "Optimasi Daya Reaktif
Fungsi Non Linier dgn. metode EPQS"
Semester :
Dengan Hasil : 80

Malang, 20 - 3 - 2007.

Dosen / Asisten-ybs.

Choiri

Choiri

DIAGRAM SEGARIS SISTEM TRANSMISI 150kV

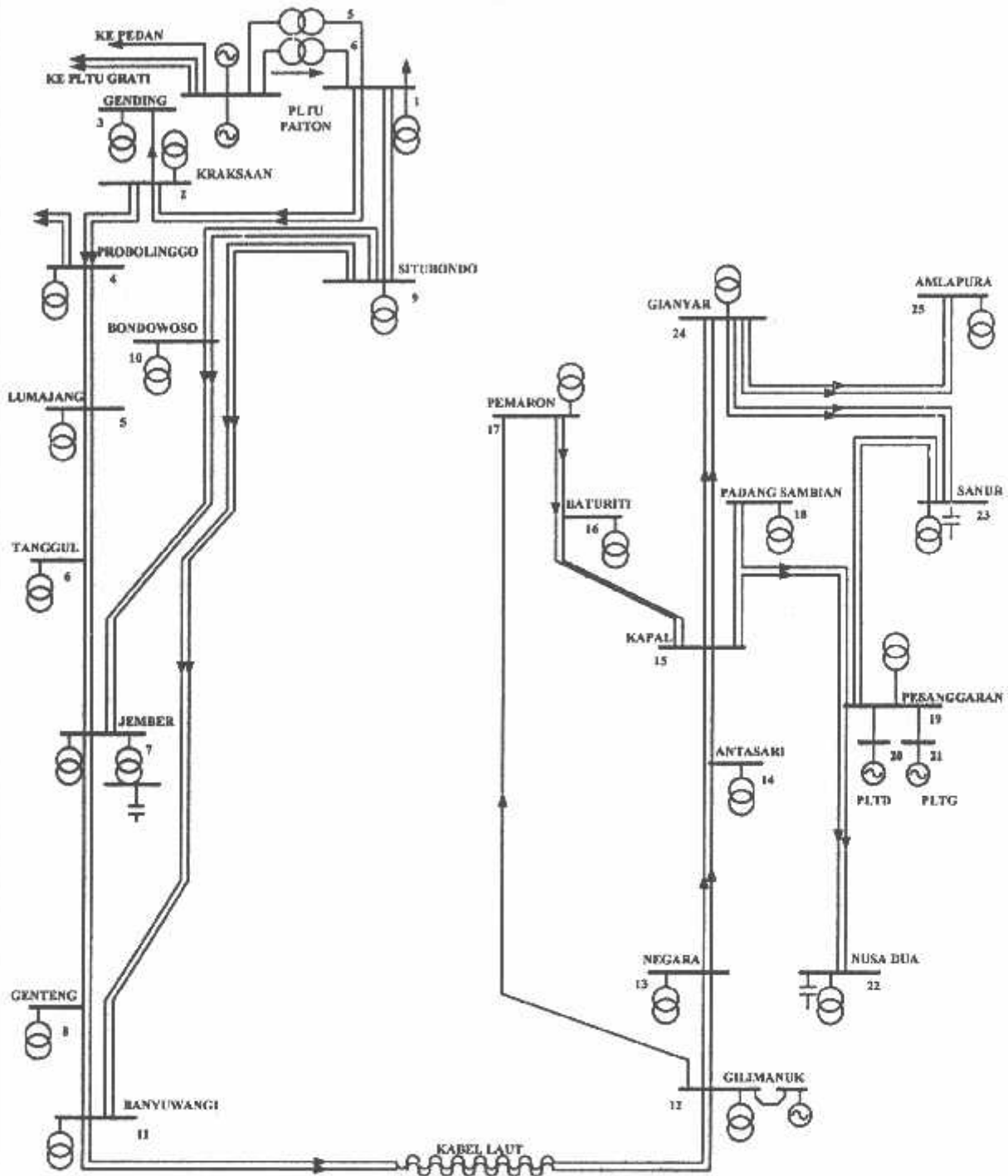


DIAGRAM SEGARIS JARINGAN SISTEM TRANSMISI LISTRIK 150kV SUB-SISTEM PAITON-BALI

DATA

DATA PEMBANGKITAN DAN PEMBEBANAN 150 KV SUB-SISTEM PAITON-BALI HARI SELASA, 27 DESEMBER 2006

No	Nama Bus	Tegangan (pu)	Pgenerator (MW)	Qgenerator (MVAR)	Pload (MW)	Qload (MVAR)	Tipe
1	PAITON	1,027	-	-	54,6	15	1
2	KRAKSAN	-	-	-	14	7	3
3	GENDING	-	-	-	17	8,5	3
4	PROBOLINGGO	-	-	-	33	13,6	3
5	LUMAJANG	-	-	-	40,6	14,8	3
6	TANGGUL	-	-	-	27,2	9,7	3
7	JEMBER	-	-	49,0	65,2	28,8	3
8	GENTENG	-	-	-	40,8	19	3
9	SITUBONDO	-	-	-	24,2	10,2	3
10	BONDOWOSO	-	-	-	18,4	6,7	3
11	BANYUWANGI	-	-	-	38,2	15,6	3
12	GILIMANUK	0,985	99,9	-	6,2	2,8	2
13	NEGARA	-	-	-	10,7	4,5	3
14	ANTASARI	-	-	-	5,6	2,6	3
15	KAPAL	-	-	40,0	54,9	16,6	3
16	BATURITI	-	-	-	4,3	0,9	3
17	PEMARON	-	-	-	24,5	7,5	3
18	PDG SAMBIAN	-	-	-	31,2	12,5	3
19	PESANGGARAN	-	-	-	71,8	27,6	3
20	PLTD PSGRAN	0,935	42,5	-	-	-	2
21	PLTG PSGRAN	0,935	67,1	-	-	-	2
22	NUSADUA	-	-	25,0	45,5	16,4	3
23	SANUR	-	-	25,0	45,9	12,7	3
24	GIANYAR	-	-	-	31,7	5,4	3
25	AMLAPURA	-	-	-	13,0	4,7	3

Ket : 1 : bus slack

2 : bus generator

3 : bus beban

DATA SALURAN TRANSMISI 150 KV SUB-SISTEM PAITON-BALI

No	Hubungan Bus	Jumlah Sirkuit	R (pu)	X (pu)	B (pu)
1	1 – 2	2	0,0053	0,0180	0,0065
2	1 – 9	2	0,0477	0,0346	0,0251
3	2 – 3	1	0,0100	0,0340	0,0123
4	2 – 4	2	0,0079	0,0269	0,0097
5	4 – 5	2	0,0135	0,0145	0,0166
6	5 – 6	1	0,0167	0,0571	0,0206
7	5 – 7	1	0,0315	0,1077	0,0389
8	6 – 7	1	0,0148	0,0506	0,0189
9	7 – 8	1	0,0337	0,1151	0,0415
10	7 – 10	2	0,0104	0,0353	0,0128
11	7 – 11	1	0,0430	0,1469	0,0530
12	8 – 11	1	0,0177	0,0604	0,0218
13	9 – 10	2	0,0091	0,0310	0,0112
14	9 – 11	2	0,0097	0,0464	0,0340
15	11 – 12	2	0,0009	0,0013	0,0364
16	12 – 13	2	0,0116	0,0336	0,0124
17	12 – 17	1	0,0231	0,0690	0,0476
18	13 – 14	1	0,0270	0,0783	0,0288
19	13 – 15	1	0,0412	0,1194	0,0439
20	14 – 15	1	0,0412	0,0411	0,0151
21	15 – 16	1	0,0232	0,0673	0,0247
22	15 – 17	1	0,0556	0,1054	0,0369
23	15 – 18	1	0,0132	0,0383	0,0141
24	15 – 19	1	0,0105	0,0304	0,0112
25	15 – 22	1	0,0199	0,0379	0,0130
26	15 – 24	2	0,0059	0,0170	0,0063
27	16 – 17	1	0,0183	0,0350	0,0120
28	18 – 19	1	0,0046	0,0134	0,0049
29	19 – 20	1	0,0020	0,0044	-
30	19 – 21	1	0,0020	0,0056	-
31	19 – 22	1	0,0127	0,0243	0,0083
32	19 – 23	2	0,0029	0,0069	0,0024
33	23 – 24	2	0,0061	0,0147	0,0052
34	24 – 25	2	0,0103	0,0298	0,0110

HASIL PERHIT. SEBELUM OPTIMASI

HASIL PERHITUNGAN TEGANGAN, SUDUT TEGANGAN, PEMBANGKITAN & PEMBEBANAN SEBELUM OPTIMASI

No. Bus	Tegangan (pu)	Sudut Tegangan (deg)	Pembangkitan		Pembelian	
			Aktif (MW)	Reaktif (MVAR)	Aktif (MW)	Reaktif (MVAR)
1	1,02667	0,00000	582.157	-57.473	54,6	15,0
2	1,00793	-0,04049	0,000	0,000	14,0	7,0
3	1,00377	-0,04548	0,000	0,000	17,0	8,5
4	0,98825	-0,09590	0,000	0,000	33,0	13,6
5	0,96561	-0,12304	0,000	0,000	40,6	14,8
6	0,9843	-0,16803	0,000	0,000	27,2	9,7
7	0,96066	-0,19413	0,000	49,000	65,2	28,8
8	0,96413	-0,24639	0,000	0,000	40,8	19,0
9	0,94313	-0,15540	0,000	0,000	24,2	10,2
10	0,94942	-0,17680	0,000	0,000	18,4	6,7
11	0,98422	-0,24963	0,000	0,000	38,2	15,6
12	0,98533	-0,25335	99,900	178,988	6,2	2,8
13	0,96951	-0,29915	0,000	0,000	10,7	4,5
14	0,95827	-0,34956	0,000	0,000	5,6	2,6
15	0,93900	-0,37591	0,000	0,000	54,9	16,6
16	0,96483	-0,35385	0,000	40,000	4,3	0,9
17	0,96645	-0,33396	0,000	0,000	24,5	7,5
18	0,93307	-0,38481	0,000	0,000	31,2	12,5
19	0,93410	-0,38372	0,000	0,000	71,8	27,6
20	0,93533	-0,38174	42,500	6,847	0	0
21	0,93533	-0,37934	67,100	-3,287	0	0
22	0,93385	-0,38929	0,000	25,000	45,5	16,4
23	0,93386	-0,42498	0,000	25,000	45,9	12,7
24	0,93435	-0,38571	0,000	0,000	31,7	5,4
25	0,93171	-0,38971	0,000	0,000	13,0	4,7

TOTAL PEMBANGKITAN, PEMBEBANAN DAN RUGI-RUGI SALURAN SEBELUM OPTIMASI

Pembangkitan		Pembelian		Rugi-rugi		Waktu Komputasi
Aktif (MW)	Reaktif (MVAR)	Aktif (MW)	Reaktif (MVAR)	Aktif (MW)	Reaktif (MVAR)	
791.657	264.075	718.500	246.100	73.157	17.905	0:0:20

HASIL PERHIT. SETELAH OPTIMASI

HASIL PERHITUNGAN TEGANGAN, SUDUT TEGANGAN, PEMBANGKITAN & PEMBEBANAN SETELAH OPTIMASI

No. Bus	Tegangan (pu)	Sudut Tegangan (deg)	Pembangkitan		Pembelian	
			Aktif (MW)	Reaktif (MVAR)	Aktif (MW)	Reaktif (MVAR)
1	1.05000	0.00000	561.471	-50.303	54.6	15.0
2	1.03212	-0.03746	0.000	0.000	14.0	7.0
3	1.02808	-0.04223	0.000	0.000	17.0	8.5
4	1.01318	-0.08836	0.000	0.000	33.0	13.6
5	0.99170	-0.11291	0.000	0.000	40.6	14.8
6	0.98435	-0.15352	0.000	0.000	27.2	9.7
7	0.98593	-0.17641	0.000	49.000	65.2	28.8
8	0.98778	-0.22262	0.000	0.000	40.8	19.0
9	0.96989	-0.13928	0.000	0.000	24.2	10.2
10	0.97556	-0.15975	0.000	0.000	18.4	6.7
11	1.00445	-0.22386	0.000	0.000	38.2	15.6
12	1.00534	-0.22705	133.800	102.052	6.2	2.8
13	0.99878	-0.27881	0.000	0.000	10.7	4.5
14	1.99899	-0.33244	0.000	0.000	5.6	2.6
15	0.98541	-0.36463	0.000	0.000	54.9	16.6
16	1.00271	-0.33758	0.000	39.229	4.3	0.9
17	1.00107	-0.31529	0.000	0.000	24.5	7.5
18	0.98445	-0.37635	0.000	0.000	31.2	12.5
19	0.98705	-0.37665	0.000	0.000	71.8	27.6
20	0.98995	-0.37391	75.000	31.225	0	0
21	0.98082	-0.37704	15.000	61.003	0	0
22	0.98107	-0.37802	0.000	3.042	45.5	16.4
23	0.98501	-0.37849	0.000	12.693	45.9	12.7
24	0.98334	-0.37541	0.000	0.000	31.7	5.4
25	0.98096	-0.37904	0.000	0.000	13.0	4.7

TOTAL PEMBANGKITAN, PEMBEBANAN DAN RUGI-RUGI SALURAN SETELAH OPTIMASI

Pembangkitan		Pembelian		Rugi-rugi		Waktu Komputasi
Aktif (MW)	Reaktif (MVAR)	Aktif (MW)	Reaktif (MVAR)	Aktif (MW)	Reaktif (MVAR)	
785.880	254.453	718.500	246.170	66.380	8.283	0:3:44:844

unit uAbout;

interface

uses

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

type

TfrmAbout = class(TForm)

private

{ Private declarations }

public

{ Public declarations }

end;

var

frmAbout: TfrmAbout;

implementation

{\$R *.dfm}

end.

unit uHasil;

interface

uses

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

type

TfrmHasil = class(TForm)
 TabSheet2: TTabSheet;
 fgBus: TStringGrid;
 TabSheet3: TTabSheet;
 fgBranch: TStringGrid;
 TabSheet4: TTabSheet;
 GroupBox6: TGroupBox;
 Label8: TLabel;
 Label9: TLabel;
 Label10: TLabel;
 Label11: TLabel;
 Label12: TLabel;
 lblGen: TLabel;
 lblLoad: TLabel;
 lblLoss: TLabel;
 edtSumGen: TEdit;
 edtSumLoad: TEdit;
 edtSumLoss: TEdit;
 edtIterasi: TEdit;
 edtTime: TEdit;
 TabSheet9: TTabSheet;
 Panel1: TPanel;
 btnClose: TButton;
 btnLFAwal: TButton;
 TabSheet1: TTabSheet;
 TabSheet5: TTabSheet;
 TabSheet6: TTabSheet;
 TabSheet7: TTabSheet;
 Chart1: TChart;
 Series1: TLineSeries;
 Series2: TLineSeries;
 fgBus2: TStringGrid;
 fgBranch2: TStringGrid;
 GroupBox1: TGroupBox;
 Label1: TLabel;
 Label2: TLabel;

Label3: TLabel;
Label4: TLabel;
Label5: TLabel;
lblGen2: TLabel;
lblLoad2: TLabel;
lblLoss2: TLabel;
edtSumGen2: TEdit;
edtSumLoad2: TEdit;
edtSumLoss2: TEdit;
edtIterasi2: TEdit;
edtTime2: TEdit;
pbGen: TProgressBar;
GroupBox2: TGroupBox;
Label6: TLabel;
Label7: TLabel;
Label14: TLabel;
Label18: TLabel;
edtMaxGen: TEdit;
edtPopSize: TEdit;
edtPMutasi: TEdit;
edtLength: TEdit;
GroupBox3: TGroupBox;
TabSheet8: TTabSheet;
PageControl2: TPageControl;
TabSheet10: TTabSheet;
Label28: TLabel;
fgHasilSPEA: TStringGrid;
TabSheet11: TTabSheet;
Chart2: TChart;
Series3: TLineSeries;
Series4: TLineSeries;
Series5: TLineSeries;
btnHitung: TButton;
btnUseDefault: TButton;
Label15: TLabel;
edtCountCap: TEdit;
Label19: TLabel;
edtUmlCapBank: TEdit;
Label20: TLabel;
edtKap: TEdit;
Label21: TLabel;
Label22: TLabel;
edtKc: TEdit;
Label23: TLabel;
Label24: TLabel;
edtKa: TEdit;

```

Label25: TLabel;
edtLamda: TEdit;
GroupBox4: TGroupBox;
GroupBox5: TGroupBox;
Label27: TLabel;
Label29: TLabel;
Label32: TLabel;
Label33: TLabel;
Label16: TLabel;
edtKp: TEdit;
Label17: TLabel;
Label26: TLabel;
edtR: TEdit;
Label13: TLabel;
edtBetha: TEdit;
procedure btnCloseClick(Sender: TObject);
procedure btnLFAwalClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure btnHitungClick(Sender: TObject);
procedure btnUseDefaultClick(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  frmHasil: TfrmHasil;

implementation

uses uUtils, uLoadflow, uNewtonRaphson, uComplex, uTopology,
  uVarGenetic, uObjFuncOps, uEvoPro;

{$R *.dfm}

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

procedure TfrmHasil.btnLFAwalClick(Sender: TObject);
var i:integer;
    mulai,selesai,selang:TDateTime;
    jam,menit,detik,mdetik:word;
begin

```

```

mulai:=time;
NewtonRaphson(gBus,gBranch,gParamLF);
selesai:=time;
selang:=selesai-mulai;
Series1.Clear;
Series2.Clear;
for i:=0 to high(gBus) do
begin
  fgBus.Cells[0,i+1]:=IntToStr(i+1);
  fgBus.Cells[1,i+1]:=RealToStr(gBus[i].absV,5);
  fgBus.Cells[2,i+1]:=RealToStr(gBus[i].sudV*
    ANGLE_OF_DEGREES/ANGLE_OF_RADIAN,5);
  fgBus.Cells[3,i+1]:=RealToStr(gBus[i].Pgen,3);
  fgBus.Cells[4,i+1]:=RealToStr(gBus[i].Qgen,3);
  fgBus.Cells[5,i+1]:=RealToStr(gBus[i].PL,3);
  fgBus.Cells[6,i+1]:=RealToStr(gBus[i].QL,3);
  fgBus.Cells[7,i+1]:=RealToStr(gBus[i].Cap,3);
  fgBus.Cells[8,i+1]:=IntToStr(gBus[i].typeBus);
  Series1.Add(gBus[i].absV,IntToStr(i+1));
end;
for i:=0 to high(gBranch) do
begin
  fgBranch.Cells[0,i+1]:=IntToStr(i+1);
  fgBranch.Cells[1,i+1]:=IntToStr(gBranch[i].dari);
  fgBranch.Cells[2,i+1]:=IntToStr(gBranch[i].ke);
  fgBranch.Cells[3,i+1]:=RealToStr(gBranch[i].Sij.real,3);
  fgBranch.Cells[4,i+1]:=RealToStr(gBranch[i].Sij.imag,3);
  fgBranch.Cells[5,i+1]:=RealToStr(gBranch[i].Aij.real,3);
  fgBranch.Cells[6,i+1]:=RealToStr(gBranch[i].Aij.imag,3);
  fgBranch.Cells[7,i+1]:=IntToStr(gBranch[i].ke);
  fgBranch.Cells[8,i+1]:=IntToStr(gBranch[i].dari);
  fgBranch.Cells[9,i+1]:=RealToStr(gBranch[i].Sji.real,3);
  fgBranch.Cells[10,i+1]:=RealToStr(gBranch[i].Sji.imag,3);
  fgBranch.Cells[11,i+1]:=RealToStr(gBranch[i].Aji.real,3);
  fgBranch.Cells[12,i+1]:=RealToStr(gBranch[i].Aji.imag,3);
end;
edtSumGen.Text:=toStringJ(gParamLF.SumGen,3);
edtSumLoad.Text:=toStringJ(gParamLF.SumLoad,3);
edtSumLoss.Text:=toStringJ(gParamLF.SumLoss,3);
edtIterasi.Text:=IntToStr(gParamLF.Iterasi);
DecodeTime(selang,jam,menit,detik,mdetik);
edtTime.Text:=IntToStr(jam)+':'+'+IntToStr(menit)+':'+'+
  IntToStr(detik)+':'+'+IntToStr(mdetik);
btnUseDefault.Enabled:=true;
end;

```

```

procedure TfrmHasil.FormCreate(Sender: TObject);
begin
  fgHasilSPEA.Cells[0,0]:='No';
  fgHasilSPEA.Cells[1,0]:='Bus';
  fgHasilSPEA.Cells[2,0]:='Cap (kVAR)';
end;

procedure TfrmHasil.btnHitungClick(Sender: TObject);
var BatasCap:TBatas;
    Ncap,sa,r,i,j,param,length,maxgen,popsize,CountCap:integer;
    betha,pmutasi,Kp,Kc,Ka,CapBank,Lamda,CostCap:double;
    Min,Avg,Max,Cap:dArr1;
    BestChrom:iArr1;
    Nbus,Nsal,ia,ja:integer;
    V,Sg,SL:CxArr1;
    Lc,Tr,HasilCap:dArr2;
    TypBus:iArr1;
    Z,Tp,Alir,Arus:CxArr2;
    mulai,selesai,selang:TDateTime;
    jam,menit,detik,mdetik:word;
    ELCSebelum,ELCSesudah,CostLoss1,CostLoss2:double;
    ep:TEvoPro11;
    LBus:TBusArr1;
begin
  Kp:=StrToFloat(edtKp.Text);
  Kc:=StrToFloat(edtKc.Text);
  Ka:=StrToFloat(edtKa.Text);
  r:=StrToInt(edtR.Text);
  Lamda:=StrToFloat(edtLamda.Text);
  CountCap:=StrToInt(edtCountCap.Text);
  Ncap:=StrToInt(edtJmlCapBank.Text);
  CapBank:=StrToFloat(edtKap.Text);
  SetLength(Cap,Ncap);
  for i:=0 to Ncap-1 do
  begin
    Cap[i]:=(i+1)*CapBank;
  end;
  gObjFuncOps:=TObjFuncOps.Create(Kp,Kc,Ka,Lamda,r,CountCap,Cap);
  CostLoss1:=gObjFuncOps.HitungObjFuncAwal;
  PopSize:=StrToInt(edtPopSize.Text);
  MaxGen:=StrToInt(edtMaxGen.Text);
  pbGen.Max:=MaxGen;
  Length:=StrToInt(edtLength.Text);
  betha:=StrToFloat(edtBetha.Text);
  Pmutasi:=StrToFloat(edtPmutasi.Text);
  ep:=TEvoPro11.Create(MaxGen,PopSize,Length,Betha,Pmutasi);

```

```

BestChrom:=ep.BestCrom;
Min:=ep.Min;
Avg:=ep.Avg;
Max:=ep.Max;
gObjFuncOps.HitungObjFuncAkhir(BestChrom,LBus,CostCap,CostLoss2);
sa:=0;
for i:=0 to high(gBus) do
begin
  if BestChrom[i]<>0 then
  begin
    fgHasilSPEA.Cells[0,sa+1]:=IntToStr(sa+1);
    fgHasilSPEA.Cells[1,sa+1]:=IntToStr(i+1);
    fgHasilSPEA.Cells[2,sa+1]:=RealToStr(BestChrom[i]*CapBank,0);
    inc(sa);
  end;
end;
Series3.Clear;
Series4.Clear;
Series5.Clear;
for i:=0 to high(Min) do
begin
  Series3.Add(Min[i],IntToStr(i+1));
  Series4.Add(Avg[i],IntToStr(i+1));
  Series5.Add(Max[i],IntToStr(i+1));
end;
//
DecodeCommDataToLFData(LBus,Nbus,Nsal,V,Sg,SL,Cap,TypBus,
gBranch,Z,Tp,Lc,Tr);
mulai:=time;
NewtonRaphson(gParamLF,V,Sg,SL,Cap,TypBus,Z,Tp,Alir,Arus,
Lc,Tr);
selesai:=time;
selang:=selesai-mulai;
Series2.Clear;
for i:=0 to high(gBus) do
begin
  fgBus2.Cells[0,i+1]:=IntToStr(i+1);
  fgBus2.Cells[1,i+1]:=RealToStr(V[i].real,5);
  fgBus2.Cells[2,i+1]:=RealToStr(V[i].imag*
    ANGLE_OF_DEGREES/ANGLE_OF_RADIAN,5);
  fgBus2.Cells[3,i+1]:=RealToStr(Sg[i].real,3);
  fgBus2.Cells[4,i+1]:=RealToStr(Sg[i].imag,3);
  fgBus2.Cells[5,i+1]:=RealToStr(SL[i].real,3);
  fgBus2.Cells[6,i+1]:=RealToStr(SL[i].imag,3);
  fgBus2.Cells[7,i+1]:=RealToStr(Cap[i],3);
  fgBus2.Cells[8,i+1]:=IntToStr(TypBus[i]);

```

```

    Series2.Add(V[i].real,IntToStr(i+1));
end;
for i:=0 to high(gBranch) do
begin
    ia:=gBranch[i].dari-1;
    ja:=gBranch[i].ke-1;
    fgBranch2.Cells[0,i+1]:=IntToStr(i+1);
    fgBranch2.Cells[1,i+1]:=IntToStr(gBranch[i].dari);
    fgBranch2.Cells[2,i+1]:=IntToStr(gBranch[i].ke);
    fgBranch2.Cells[3,i+1]:=RealToStr(Alir[ia,ja].real,3);
    fgBranch2.Cells[4,i+1]:=RealToStr(Alir[ia,ja].imag,3);
    fgBranch2.Cells[5,i+1]:=RealToStr(Arus[ia,ja].real,3);
    fgBranch2.Cells[6,i+1]:=RealToStr(Arus[ia,ja].imag,3);
    fgBranch2.Cells[7,i+1]:=IntToStr(gBranch[i].ke);
    fgBranch2.Cells[8,i+1]:=IntToStr(gBranch[i].dari);
    fgBranch2.Cells[9,i+1]:=RealToStr(Alir[ja,ia].real,3);
    fgBranch2.Cells[10,i+1]:=RealToStr(Alir[ja,ia].imag,3);
    fgBranch2.Cells[11,i+1]:=RealToStr(Arus[ja,ia].real,3);
    fgBranch2.Cells[12,i+1]:=RealToStr(Arus[ja,ia].imag,3);
end;
edtSumGen2.Text:=toStringJ(gParamLF.SumGen,3);
edtSumLoad2.Text:=toStringJ(gParamLF.SumLoad,3);
edtSumLoss2.Text:=toStringJ(gParamLF.SumLoss,3);
edtIterasi2.Text:='3'//IntToStr(gParamLF.Iterasi);
DecodeTime(selang,jam,menit,detik,mdetik);
edtTime2.Text:=IntToStr(jam)+':' +IntToStr(menit)+':' +
    IntToStr(detik)+':' +IntToStr(mdetik);
//
ep.Free;
gObjFuncOps.Free;
end;

```

```

procedure TfrmHasil.btnUseDefaultClick(Sender: TObject);
begin
    //input parameter Genetic Algorithm
    edtMaxGen.Text:='50';
    edtPopSize.Text:='10';
    edtBetha.Text:='0.3';
    edtPMutasi.Text:='0.004';
    edtLength.Text:=IntToStr(high(gBus)+1);
    //input parameter objective function
    edtCountCap.Text:='4';
    edtJmlCapBank.Text:='4';
    edtKap.Text:='300';
    edtKc.Text:='0.5';
    edtKp.Text:='120';

```

```
edtKa.Text:='100000';  
edtLamda.Text:='100000';  
edtR.Text:='1';  
btnHitung.Enabled:=true;  
end;  
  
end.
```

```

unit uInputLFChild;

interface

uses uInputLF,uHasil,SysUtils,Forms;

type
  TfrmInputLFChild=class(TfrmInputLF)
  protected
    procedure ShowHasil;override;
  end;

var frmInput:TfrmInputLFChild;

implementation

uses uLoadflow;

procedure TfrmInputLFChild.ShowHasil;
begin
  try
    if frmHasil=nil then
      begin
        frmHasil:=TfrmHasil.Create(Application);
      end;
      frmHasil.fgBus.Cells[0,0]:='Bus';
      frmHasil.fgBus.Cells[1,0]:='absV (pu)';
      frmHasil.fgBus.Cells[2,0]:='sudV (deg)';
      frmHasil.fgBus.Cells[7,0]:='Sups (pu)';
      frmHasil.fgBus.Cells[8,0]:='Type Bus';
      frmHasil.fgBranch.Cells[0,0]:='No';
      frmHasil.fgBranch.Cells[1,0]:='Dari';
      frmHasil.fgBranch.Cells[2,0]:='Ke';
      frmHasil.fgBranch.Cells[5,0]:='Arus re (A)';
      frmHasil.fgBranch.Cells[6,0]:='Arus im (A)';
      frmHasil.fgBranch.Cells[7,0]:='Dari';
      frmHasil.fgBranch.Cells[8,0]:='Ke';
      frmHasil.fgBranch.Cells[11,0]:='Arus re (A)';
      frmHasil.fgBranch.Cells[12,0]:='Arus im (A)';
      //
      frmHasil.fgBus2.Cells[0,0]:='Bus';
      frmHasil.fgBus2.Cells[1,0]:='absV (pu)';
      frmHasil.fgBus2.Cells[2,0]:='sudV (deg)';
      frmHasil.fgBus2.Cells[7,0]:='Sups (pu)';
      frmHasil.fgBus2.Cells[8,0]:='Type Bus';
      frmHasil.fgBranch2.Cells[0,0]:='No';

```

```

frmHasil.fgBranch2.Cells[1,0]:='Dari';
frmHasil.fgBranch2.Cells[2,0]:='Ke';
frmHasil.fgBranch2.Cells[5,0]:='Arus re (A)';
frmHasil.fgBranch2.Cells[6,0]:='Arus im (A)';
frmHasil.fgBranch2.Cells[7,0]:='Dari';
frmHasil.fgBranch2.Cells[8,0]:='Ke';
frmHasil.fgBranch2.Cells[11,0]:='Arus re (A)';
frmHasil.fgBranch2.Cells[12,0]:='Arus im (A)';
//
if gParamLF.PKonst=1 then
begin
  frmHasil.fgBus.Cells[3,0]:='Pg (W)';
  frmHasil.fgBus.Cells[4,0]:='Qg (VAR)';
  frmHasil.fgBus.Cells[5,0]:='PL (W)';
  frmHasil.fgBus.Cells[6,0]:='QL (VAR)';
  frmHasil.fgBranch.Cells[3,0]:='P (W)';
  frmHasil.fgBranch.Cells[4,0]:='Q (VAR)';
  frmHasil.fgBranch.Cells[9,0]:='P (W)';
  frmHasil.fgBranch.Cells[10,0]:='Q (VAR)';
  frmHasil.lblGen.Caption:='VA';
  frmHasil.lblLoad.Caption:='VA';
  frmHasil.lblLoss.Caption:='VA';
  //
  frmHasil.fgBus2.Cells[3,0]:='Pg (W)';
  frmHasil.fgBus2.Cells[4,0]:='Qg (VAR)';
  frmHasil.fgBus2.Cells[5,0]:='PL (W)';
  frmHasil.fgBus2.Cells[6,0]:='QL (VAR)';
  frmHasil.fgBranch2.Cells[3,0]:='P (W)';
  frmHasil.fgBranch2.Cells[4,0]:='Q (VAR)';
  frmHasil.fgBranch2.Cells[9,0]:='P (W)';
  frmHasil.fgBranch2.Cells[10,0]:='Q (VAR)';
  frmHasil.lblGen2.Caption:='VA';
  frmHasil.lblLoad2.Caption:='VA';
  frmHasil.lblLoss2.Caption:='VA';
  //
end
else if gParamLF.PKonst=1000 then
begin
  frmHasil.fgBus.Cells[3,0]:='Pg (kW)';
  frmHasil.fgBus.Cells[4,0]:='Qg (kVAR)';
  frmHasil.fgBus.Cells[5,0]:='PL (kW)';
  frmHasil.fgBus.Cells[6,0]:='QL (kVAR)';
  frmHasil.fgBranch.Cells[3,0]:='P (kW)';
  frmHasil.fgBranch.Cells[4,0]:='Q (kVAR)';
  frmHasil.fgBranch.Cells[9,0]:='P (kW)';
  frmHasil.fgBranch.Cells[10,0]:='Q (kVAR)';

```

```

frmHasil.lblGen.Caption:='kVA';
frmHasil.lblLoad.Caption:='kVA';
frmHasil.lblLoss.Caption:='kVA';
//
frmHasil.fgBus2.Cells[3,0]:='Pg (kW)';
frmHasil.fgBus2.Cells[4,0]:='Qg (kVAR)';
frmHasil.fgBus2.Cells[5,0]:='PL (kW)';
frmHasil.fgBus2.Cells[6,0]:='QL (kVAR)';
frmHasil.fgBranch2.Cells[3,0]:='P (kW)';
frmHasil.fgBranch2.Cells[4,0]:='Q (kVAR)';
frmHasil.fgBranch2.Cells[9,0]:='P (kW)';
frmHasil.fgBranch2.Cells[10,0]:='Q (kVAR)';
frmHasil.lblGen2.Caption:='kVA';
frmHasil.lblLoad2.Caption:='kVA';
frmHasil.lblLoss2.Caption:='kVA';
//
end
else if gParamLF.PKonst=1000000 then
begin
frmHasil.fgBus.Cells[3,0]:='Pg (MW)';
frmHasil.fgBus.Cells[4,0]:='Qg (MVAR)';
frmHasil.fgBus.Cells[5,0]:='PL (MW)';
frmHasil.fgBus.Cells[6,0]:='QL (MVAR)';
frmHasil.fgBranch.Cells[3,0]:='P (MW)';
frmHasil.fgBranch.Cells[4,0]:='Q (MVAR)';
frmHasil.fgBranch.Cells[9,0]:='P (MW)';
frmHasil.fgBranch.Cells[10,0]:='Q (MVAR)';
frmHasil.lblGen.Caption:='MVA';
frmHasil.lblLoad.Caption:='MVA';
frmHasil.lblLoss.Caption:='MVA';
//
frmHasil.fgBus2.Cells[3,0]:='Pg (MW)';
frmHasil.fgBus2.Cells[4,0]:='Qg (MVAR)';
frmHasil.fgBus2.Cells[5,0]:='PL (MW)';
frmHasil.fgBus2.Cells[6,0]:='QL (MVAR)';
frmHasil.fgBranch2.Cells[3,0]:='P (MW)';
frmHasil.fgBranch2.Cells[4,0]:='Q (MVAR)';
frmHasil.fgBranch2.Cells[9,0]:='P (MW)';
frmHasil.fgBranch2.Cells[10,0]:='Q (MVAR)';
frmHasil.lblGen2.Caption:='MVA';
frmHasil.lblLoad2.Caption:='MVA';
frmHasil.lblLoss2.Caption:='MVA';
//
end;
frmHasil.fgBus.RowCount:=StrToInt(edtNbus.Text)+1;
frmHasil.fgBranch.RowCount:=StrToInt(edtNsal.Text)+1;

```

```
//
frmHasil.fgBus2.RowCount:=StrToInt(edtNbus.Text)+1;
frmHasil.fgBranch2.RowCount:=StrToInt(edtNsal.Text)+1;
//
frmHasil.ShowModal;
finally
    frmHasil.Free;
end;
end;

end.
```

```

unit uMenu;

interface

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

type
  TfrmMenu = class(TForm)
    Panel1: TPanel;
    btnNew: TButton;
    btnOpen: TButton;
    btnExit: TButton;
    StatusBar1: TStatusBar;
    Panel2: TPanel;
    OpenFileDialog1: TOpenDialog;
    procedure btnExitClick(Sender: TObject);
    procedure btnNewClick(Sender: TObject);
    procedure btnOpenClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmMenu: TfrmMenu;

implementation

uses uInputLFChild, uComplex, uUtils, uLoadflow;

{$R *.dfm}

procedure TfrmMenu.btnExitClick(Sender: TObject);
begin
  Application.Terminate;
end;

procedure TfrmMenu.btnNewClick(Sender: TObject);
begin
  try
    if frmInput=nil then
      begin
        frmInput:=TfrmInputLFChild.Create(Application);
      end
    end
  except
  end
end;

```

```

end;
frmInput.Caption:='Input Data';
frmInput.btnNext.Caption:='&Save';
frmInput.ShowModal;
finally
    frmInput.Free;
end;
end;

procedure TfrmMenu.btnOpenClick(Sender: TObject);
var NamaFile,Nama:string;
    output:TextFile;
    i,j,Typ,dari,ke,Nbus,Nsal,Param,Ngen,NCable:integer;
    Cap,absV,sudV,Pg,Qg,PL,QL,CapSal,Pmin,Pmax,Harga,Length:double;
    R,X,Lc,Tr,Tu,Su,VKonst,PKonst,Pbase,Vbase:double;
begin
    try
        if OpenFileDialog1.Execute then
            begin
                NamaFile:=OpenFileDialog1.FileName;
                AssignFile(output,NamaFile);
                Reset(output);
                Readln(output,Nbus);
                Readln(output,Nsal);
                Readln(output,Vbase);
                Readln(output,VKonst);
                Readln(output,Pbase);
                Readln(output,PKonst);
                Readln(output,param);
                gParamLF.Vbase:=Vbase;
                gParamLF.VKonst:=VKonst;
                gParamLF.Pbase:=Pbase;
                gParamLF.PKonst:=PKonst;
                if Param=1 then
                    begin
                        gParamLF.ParamBranch:=pbPu;
                    end
                else if Param=2 then
                    begin
                        gParamLF.ParamBranch:=pbOhm;
                    end;
                gParamLF.MaxIterasi:=15;
                gParamLF.Toleransi:=0.0001;
                try
                    frmInput:=TfrmInputLFChild.Create(Application);
                    frmInput.edtNbus.Text:=IntToStr(Nbus);

```

```

frmInput.edtNsal.Text:=IntToStr(Nsal);
frmInput.edtVbase.Text:=FloatToStr(Vbase);
if VKonst=1 then
begin
  frmInput.cmbVKonst.Text:='V';
end
else if VKonst=1000 then
begin
  frmInput.cmbVKonst.Text:='kV';
end
else if VKonst=1000000 then
begin
  frmInput.cmbVKonst.Text:='MV';
end;
frmInput.edtPbase.Text:=FloatToStr(Pbase);
if PKonst=1 then
begin
  frmInput.cmbPKonst.Text:='VA';
  frmInput.fgBus.Cells[3,0]:='Pg (W)';
  frmInput.fgBus.Cells[4,0]:='Qg (VAR)';
  frmInput.fgBus.Cells[5,0]:='PL (W)';
  frmInput.fgBus.Cells[6,0]:='QL (VAR)';
  frmInput.fgBranch.Cells[9,0]:='Kap (VA)';
  frmInput.fgBranch.Cells[10,0]:='P (W)';
  frmInput.fgBranch.Cells[11,0]:='Q (VAR)';
  frmInput.fgBranch.Cells[16,0]:='P (W)';
  frmInput.fgBranch.Cells[17,0]:='Q (VAR)';
end
else if PKonst=1000 then
begin
  frmInput.cmbPKonst.Text:='kVA';
  frmInput.fgBus.Cells[3,0]:='Pg (kW)';
  frmInput.fgBus.Cells[4,0]:='Qg (kVAR)';
  frmInput.fgBus.Cells[5,0]:='PL (kW)';
  frmInput.fgBus.Cells[6,0]:='QL (kVAR)';
  frmInput.fgBranch.Cells[9,0]:='Kap (kVA)';
  frmInput.fgBranch.Cells[10,0]:='P (kW)';
  frmInput.fgBranch.Cells[11,0]:='Q (kVAR)';
  frmInput.fgBranch.Cells[16,0]:='P (kW)';
  frmInput.fgBranch.Cells[17,0]:='Q (kVAR)';
end
else if PKonst=1000000 then
begin
  frmInput.cmbPKonst.Text:='MVA';
  frmInput.fgBus.Cells[3,0]:='Pg (MW)';
  frmInput.fgBus.Cells[4,0]:='Qg (MVAR)';

```

```

frmInput.fgBus.Cells[5,0]:='PL (MW)';
frmInput.fgBus.Cells[6,0]:='QL (MVAR)';
frmInput.fgBranch.Cells[9,0] :='Kap (MVA)';
frmInput.fgBranch.Cells[10,0]:='P (MW)';
frmInput.fgBranch.Cells[11,0]:='Q (MVAR)';
frmInput.fgBranch.Cells[16,0]:='P (MW)';
frmInput.fgBranch.Cells[17,0]:='Q (MVAR)';
end;
if param=1 then
begin
frmInput.cmbParam.Text:='pu';
frmInput.fgBranch.Cells[3,0]:='R (pu)';
frmInput.fgBranch.Cells[4,0]:='X (pu)';
frmInput.fgBranch.Cells[5,0]:='Lc (pu)';
frmInput.fgBus.Cells[7,0]:='Cap (pu)';
end
else if param=2 then
begin
frmInput.cmbParam.Text:='ohm';
frmInput.fgBranch.Cells[3,0]:='R (ohm)';
frmInput.fgBranch.Cells[4,0]:='X (ohm)';
frmInput.fgBranch.Cells[5,0]:='Lc (ohm)';
frmInput.fgBus.Cells[7,0]:='Cap (ohm)';
end;
SetLength(gBus,Nbus);
for i:=0 to Nbus-1 do
begin
Readln(output,absV,sudV,Pg,Qg,PL,QL,Cap,Typ);
gBus[i].absV:=absV;
gBus[i].sudV:=sudV;
gBus[i].Pgen:=Pg;
gBus[i].Qgen:=Qg;
gBus[i].PL:=PL;
gBus[i].QL:=QL;
gBus[i].Cap:=Cap;
gBus[i].typeBus:=Typ;
frmInput.fgBus.Cells[0,i+1]:=IntToStr(i+1);
frmInput.fgBus.Cells[1,i+1]:=FloatToStr(absV);
frmInput.fgBus.Cells[2,i+1]:=FloatToStr(sudV);
frmInput.fgBus.Cells[3,i+1]:=FloatToStr(Pg);
frmInput.fgBus.Cells[4,i+1]:=FloatToStr(Qg);
frmInput.fgBus.Cells[5,i+1]:=FloatToStr(PL);
frmInput.fgBus.Cells[6,i+1]:=FloatToStr(QL);
frmInput.fgBus.Cells[7,i+1]:=FloatToStr(Cap);
frmInput.fgBus.Cells[8,i+1]:=IntToStr(typ);
end;

```

```

SetLength(gBranch,Nsal);
for i:=0 to Nsal-1 do
begin
  Readln(output,dari,ke,R,X,Lc,Tr,Tu,Su,CapSal);
  gBranch[i].dari:=dari;
  gBranch[i].ke:=ke;
  gBranch[i].R:=R;
  gBranch[i].X:=X;
  gBranch[i].Lc:=Lc;
  gBranch[i].Tr:=Tr;
  gBranch[i].Tu:=Tu;
  gBranch[i].Su:=Su;
  gBranch[i].KapSal:=CapSal;
  frmInput.fgBranch.Cells[0,i+1]:=IntToStr(i+1);
  frmInput.fgBranch.Cells[1,i+1]:=IntToStr(dari);
  frmInput.fgBranch.Cells[2,i+1]:=IntToStr(ke);
  frmInput.fgBranch.Cells[3,i+1]:=FloatToStr(R);
  frmInput.fgBranch.Cells[4,i+1]:=FloatToStr(X);
  frmInput.fgBranch.Cells[5,i+1]:=FloatToStr(Lc);
  frmInput.fgBranch.Cells[6,i+1]:=FloatToStr(Tr);
  frmInput.fgBranch.Cells[7,i+1]:=FloatToStr(Tu);
  frmInput.fgBranch.Cells[8,i+1]:=FloatToStr(Su);
  frmInput.fgBranch.Cells[9,i+1]:=FloatToStr(CapSal);
end;
Readln(output,Ngen);
if Ngen<>0 then
begin
  frmInput.fgGen.RowCount:=Ngen+1;
  SetLength(gGenLF,Ngen);
  for i:=0 to Ngen-1 do
  begin
    Readln(output,dari,R,X,Lc,Tr,Tu,Su,CapSal,Pmin,Pmax);
    gGenLF[i].bus:=dari;
    gGenLF[i].Qmin:=R;
    gGenLF[i].Qmax:=X;
    gGenLF[i].a2:=Lc;
    gGenLF[i].a1:=Tr;
    gGenLF[i].a0:=Tu;
    gGenLF[i].FixCost:=Su;
    gGenLF[i].VarCost:=CapSal;
    gGenLF[i].Pmin:=Pmin;
    gGenLF[i].Pmax:=Pmax;
    frmInput.fgGen.Cells[0,i+1]:=IntToStr(i+1);
    frmInput.fgGen.Cells[1,i+1]:=IntToStr(gGenLF[i].bus);
    frmInput.fgGen.Cells[2,i+1]:=RealToStr(gGenLF[i].Qmin,2);
    frmInput.fgGen.Cells[3,i+1]:=RealToStr(gGenLF[i].Qmax,2);
  end;
end;

```

```

    frmInput.fgGen.Cells[4,i+1]:=RealToStr(gGenLF[i].a2,5);
    frmInput.fgGen.Cells[5,i+1]:=RealToStr(gGenLF[i].a1,5);
    frmInput.fgGen.Cells[6,i+1]:=RealToStr(gGenLF[i].a0,5);
    frmInput.fgGen.Cells[7,i+1]:=RealToStr(gGenLF[i].FixCost,2);
    frmInput.fgGen.Cells[8,i+1]:=RealToStr(gGenLF[i].VarCost,2);
    frmInput.fgGen.Cells[9,i+1]:=RealToStr(gGenLF[i].Pmin,2);
    frmInput.fgGen.Cells[10,i+1]:=RealToStr(gGenLF[i].Pmax,2);
  end;
end
else
begin
  frmInput.fgGen.RowCount:=2;
end;
CloseFile(output);
frmInput.Caption:='Tampilan Data';
frmInput.btnNext.Caption:='&Next';
frmInput.ShowModal;
finally
  frmInput.Free;
end;
end;
except
  MessageDlg('File Corrupt atau Error Program!',mtWarning,[mbOK],0);
end;
end;

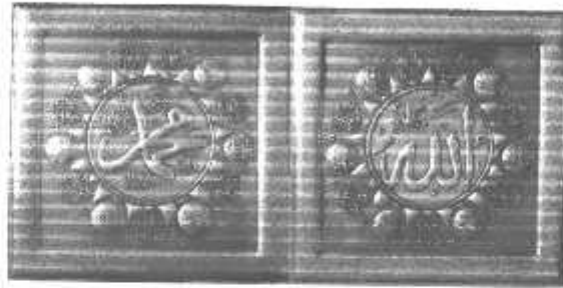
end.

```

Lembar Persembahan



Kupersembahkan skripsiku ini special untuk



Syukur koe ya alloh

Atas segala rahmat yang telah Engkau berikan kepadaku, Hanya keridhoan-Mulah yang kuharapkan dalam hidupku ini, Agar aku bisa menjalani perjalanan hidupku ini dengan penuh keikhlasan dan ketabahan dalam keimanan dan ketaqwaan Nikmat yang engkau berikan begitu sempurna kepada semua umat manusia sesungguhnya engkau maha pengasih lagi maha penyayang sujud syukur kucapkan kepadamu ya alloh

serta

Tunjungan , Nabi besar Muhammad SAW karena beliau lah nabi yang diutus Allah ke muka bumi dengan membawa al-Qur'an dan agama islam untuk mengajak manusia dari kegelapan menuju jalan yang penuh barokah dan hanya dengan sunnahmu manusia dapat menjalani hidup dengan bahagia ,hanya engkau lah suri tauladan bagi

kami ya rosullulloh