

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



SKRIPSI

**ANALISIS PENEMPATAN KAPASITOR SHUNT PADA
JARINGAN DISTRIBUSI PRIMER 20 KV TYPE RADIAL
DENGAN MENGGUNAKAN METODE NONDOMINATED
SORTING GENETIC ALGORITHM II (NSGA-II) DI
PENYULANG TEJAKULA, BALI**

**Disusun Oleh:
I NYOMAN YUS ANTARA
00.12.129**

SEPTEMBER 2006

LEMBAR PERSETUJUAN

**ANALISIS PENEMPATAN KAPASITOR SHUNT PADA
JARINGAN DISTRIBUSI PRIMER 20 KV TYPE RADIAL
DENGAN MENGGUNAKAN METODE NONDOMINATED
SORTING GENETIC ALGORITHM II (NSGA-II) DI
PENYULANG TEJAKULA, BALI**

SKRIPSI

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

Disusun Oleh :

**I NYOMAN YUS ANTARA
NIM. 00.12.129**



**Mengetahui,
Ketua Jurusan Teknik Elektro**

**Ir. F. Rudi Limpraptono, MT
NIP. Y. 103 950 0274**

**Menyetujui,
Dosen Pembimbing**

**Ir. Teguh Herbasuki, MT
NIP. P. 103 890 0209**

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

ABSTRAKSI

ANALISIS PENEMPATAN KAPASITOR SHUNT PADA JARINGAN DISTRIBUSI PRIMER 20 KV TYPE RADIAL DENGAN MENGUNAKAN METODE *NONDOMINATED SORTING GENETIC ALGORITHM II (NSGA-II)* DI PENYULANG TEJAKULA, BALI

(1 Nyoman Yus Antara, 00.12.129, September 2006, 66 hal)

(Dosen Pembimbing: Ir. Teguh Herbasuki, MT.)

Kata kunci : Penempatan kapasitor, Sistem Distribusi Radial,
Metode Nondominated Sorting Genetic Algorithm
II (NSGA-II).

Jaringan distribusi menengah type radial merupakan jaringan yang banyak digunakan untuk mendistribusikan daya listrik pada tegangan menengah. Dengan meningkatnya beban induktif maka daya reaktif yang ada di jaringan akan semakin besar yang selanjutnya akan memperbesar komponen rugi-rugi yang meliputi rugi-rugi energi dan rugi-rugi daya disamping itu dapat memperburuk kondisi tegangan. Analisa perhitungan penempatan kapasitor pada jaringan 20 kV sistem distribusi radial menggunakan Metode Nondominated Sorting Genetic Algorithm II (NSGA-II) adalah salah satu solusi untuk mengurangi rugi-rugi daya dan memperbaiki profil tegangan. Dengan menggunakan Metode NSGA-II juga dapat menentukan letak dan kapasitas kapasitor yang terpasang secara optimal pada jaringan distribusi tegangan menengah.

KATA PENGANTAR

Puji Syukur dihadapan Ide Sang Hyang Widhi Wace, Tuhan Yang Maha Esa, karena Atas Asung Kertha Ware Nugrahe-Nya penyusun dapat menyelesaikan skripsi yang berjudul

**ANALISIS PENEMPATAN KAPASITOR SHUNT PADA JARINGAN
DISTRIBUSI PRIMER 20 KV TYPE RADIAL DENGAN
MENGUNAKAN METODE *NONDOMINATED SORTING GENETIC
ALGORITHM II (NSGA-II)* DI PENYULANG TEJAKULA, BALI**

Skripsi ini merupakan salah satu persyaratan untuk menempuh ujian tingkat sarjana S-1 di jurusan Teknik Elektro Konsentrasi Teknik Energi Listrik, Fakultas Teknologi Industri, Institut Teknologi Nasional Malang.

Pada kesempatan ini penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Bapak Prof. Dr. Ir. Abraham Lomi, MSEE. selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. Mochtar Asroni, MSME. selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang.
3. Bapak Ir.F.Yudi Limpraptono, MT. selaku Ketua Jurusan Teknik Elektro S-1 Institut Teknologi Nasional Malang.
4. Bapak Ir. Teguh Herbasuki, MT. selaku Dosen Pembimbing.
5. Bapak dan Ibu Dosen Teknik Elektro Energi Listrik, Institut Teknologi Nasional Malang.

Penyusun menyadari bahwa skripsi ini masih jauh dari sempurna dan masih banyak kekurangan-kekurangan baik isi maupun penyajian, hal ini tidak

lain karena keterbatasan ilmu yang penulis miliki, oleh karena itu penulis mengharapkan kritik dan saran yang sifatnya membangun guna kesempurnaan skripsi ini.

Pada akhirnya penulis berharap semoga laporan skripsi ini dapat bermanfaat bagi penulis dan bagi yang membutuhkannya.

Malang, 2006

Penulis

DAFTAR ISI

LEMBAR PERSETUJUAN	i
ABSTRAKSI	ii
KATA PENGHANTAR	iii
DAFTAR ISI	v
DAFTAR GAMBAR	viii
DAFTAR TABEL	ix
DAFTAR GRAFIK	ix
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan Pembahasan.....	3
1.4. Batasan Masalah	3
1.5. Metodologi Pembahasan	4
1.6. Kontribusi	5
1.7. Sistematika Pembahasan	5
BAB II SISTEM JARINGAN DISTRIBUSI PRIMER DAN	
PERBAIKAN FAKTOR DAYA	8
2.1. Sistem Distribusi	8
2.1.1. Sistem Distribusi Primer	9
2.1.2. Sistem Distribusi Sekunder	9
2.2. Struktur Jaringan Distribusi Tenaga Listrik	10
2.3. Sistem Jaringan Distribusi Radial	10
2.3.1. Sistem Radial Pohon	12
2.3.2. Sistem Radial dengan Tie dan Switch Pemisah	12
2.3.3. Sistem Radial dengan Pembagian Phase Area.....	13
2.3.4. Sistem Radial dengan Beban Terpusat	14
2.4. Daya Reaktif dan Faktor Daya	15
2.5. Kapasitor Daya	17
2.5.1. Kapasitor Seri dan Kapasitor Shunt	17

2.5.2. Faktor-Faktor Pemilihan Kapasitor Seri dan Kapasitor Shunt	18
2.6. Pengaruh Pemasangan Kapasitor Shunt	20
2.6.1. Pengurangan Rugi-rugi dengan Kapasitor Shunt	21
2.6.2. Perbaikan Tegangan	22
2.6.3. Perbaikan Faktor Daya dan Kenaikan Kapasitas Sistem	24
2.6.3.1. Perhitungan Pengaruh Perbaikan Faktor Daya	25
2.6.3.2. Penentuan Rating Kapasitor untuk Perbaikan Faktor Daya Beban	27
2.7. Per-unit	28
BAB III METODE NONDOMINATED SORTING GENETIC ALGORITHM II (NSGA-II) UNTUK MENENTUKAN LETAK DAN KAPASITAS KAPASITOR PADA JARINGAN DISTRIBUSI PRIMER 20 KV TYPE RADIAL	
TYPE RADIAL	30
3.1. Analisa Aliran Daya Jaringan Radial	30
3.1.1. Tujuan	30
3.1.2. Metode Newton Raphson	31
3.1.3. Diagram Alir Aliran Daya Newton Raphson	35
3.2. Metode Penempatan Kapasitor	36
3.2.1. Genetic Algoritma	36
3.2.2. Konsep Nondominance Genetic Algorithm	38
3.3. Penentuan Saving Cost	39
3.3.1. Biaya Daya (K_p)	41
3.3.2. Biaya Ekuivalen Instalasi Kapasitor (K^C)	41
BAB IV ANALISA KAPASITAS DAN PENEMPATAN KAPASITOR PADA JARINGAN DISTRIBUSI PRIMER 20 KV PENYULANG TEJAKULA, BALI	42
4.1. Analisis Penempatan Kapasitor Dengan Nondominated Sortig Genetic Algorithm II (NSGA-II)	42
4.1.1. Flow Chart Algoritma Program	43

4.1.2. Algoritma Penyelesaian Masalah	44
4.1.3. Flowchart Nondominated Sorting Genetic Algorithm II (NSGA-II)	45
4.2. Uji Validasi	47
4.3. Data Perhitungan	50
4.3.1. Data Saluran	50
4.3.2. Data Pembebanan	51
4.3.3. Data Kapasitor	52
4.4. Analisa Perhitungan	52
4.5. Prosedur Pelaksanaan Program Perhitungan	53
BAB V KESIMPULAN DAN SARAN	64
5.1 Kesimpulan	64
5.2 Saran.....	65
DAFTAR PUSTAKA	66
LAMPIRAN	

DAFTAR GAMBAR

Gambar :	Halaman :
2-1 Jaringan Distribusi Tegangan Menengah (JTM), Jaringan Tegangan Rendah (JTR) dan Sambungan Rumah ke Pelanggan	8
2-2 Jaringan Distribusi Radial	11
2-3 Jaringan Distribusi Radial dengan Tie dan Switch Pemisah	12
2-4 Jaringan Distribusi Radial dengan Phase Area	14
2-5 Jaringan Distribusi dengan Beban Terpusat	15
2-6 Segitiga Daya	16
2-7 Segitiga Arus	17
2-8 Saluran Primer dengan Beban Terpusat	21
2-9 Gambar Rangkaian Kapasitor Shunt Pada Jaringan Tiga Fasa.....	22
2-10 Bagan Satu Garis dan Fasor Diagram Suatu Penyulang dengan Faktor Daya Tertinggal	23
2-11 Perbaikan Faktor Daya	24
2-12 Diagram Fasor dan Sudut Daya Bebas Distribusi	26
3-1 Pengkodean Kromosom	39
4-1 Single Line Diagram Penyulang Tejakula	46
4-2 Tampilan data IEEE 9 Bus	48
4-3 Tampilan Aliran Daya Newton Raphson	48
4-4 Hasil Perhitungan Metode Nondominated Sorting Genetic Algorithm II (NSGA-II).....	49
4-5 Tampilan Menu Utama	54
4-6 Tampilan Data.....	55
4-7 Tampilan Aliran Daya Newton Raphson	56
4-8 Tampilan Analisis Metode NSGA-II	57
4-9 Hasil Perhitungan Metode Nondominated Sorting Genetic Algorithm II(NSGA-II).....	57
4-10 Penempatan Kapasitor Pada Single Line Diagram Penyulang Tejakula	63

DAFTAR TABEL

Tabel :	Halaman :
2-1 Kapasitor Seri dan Kapasitor Shunt	19
4-1 Data Penyulang Sistem 20 kV GI Pamaran.....	51
4-2 Data Saluran Penyulang Tejakula	51
4-3 Data Pembebanan Penyulang Tejakula	52
4-4 Harga Kapasitor	52
4-5 Data Impedansi Saluran Penyulang Tejakula	54
4-6 Load Flow Penyulang Tejakula Sebelum Pemasangan Kapasitor ...	56
4-7 Aliran Daya Penyulang Tejakula Sebelum Pemasangan Kapasitor..	57
4-8 Load Flow Penyulang Tejakula Sesudah Pemasangan Kapasitor	59
4-9 Aliran Daya Penyulang Tejakula Sebelum Pemasangan Kapasitor..	59
4-10 Perbandingan Kondisi Sebelum dan Sesudah Pemasangan Kapasitor.....	62

DAFTAR GRAFIK

Gambar :	Halaman :
4-1 Tegangan Tiap-Tiap Node Terhadap Asumsi Tegangan Awal Sebelum dan Sesudah Pemasangan Kapasitor	60
4-2 Rugi-Rugi Daya Saluran Sebelum dan Sesudah Pemasangan Kapasitor.....	61
4-3 Perbandingan Daya Pembangkitan Sebelum dan Sesudah Pemasangan Kapasitor.....	61

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dengan meningkatnya usaha di sektor industri dan meningkatnya taraf hidup masyarakat maka kebutuhan akan energi listrik semakin meningkat pula, sehingga diperlukan penyediaan energi listrik beserta jaringan dan penyaluran yang sangat baik. Bertambahnya industri-industri menyebabkan peran penggunaan alat-alat listrik akan semakin luas, misalnya Motor-motor Listrik, Trafo, AC, Lampu-lampu TL dan lain-lain. Beban industri sangat banyak membutuhkan daya reaktif. Dengan meningkatnya beban-beban induktif, maka daya reaktif yang ada di jaringan akan semakin besar yang selanjutnya akan memperbesar rugi-rugi daya, disamping itu dapat memperburuk kondisi profil tegangan.

Alternatif yang sering dipakai untuk memperbaiki kondisi tegangan akibat adanya rugi-rugi tersebut adalah dengan memasang sumber daya reaktif tambahan disisi beban, salah satunya adalah dengan memasang kapasitor.

Ada banyak metode yang sebelumnya digunakan untuk memecahkan masalah optimalisasi penempatan kapasitor, diantaranya Metode Genetika Algoritma, Tabu Search, Fuzzy Logic dan lain-lain.

Pada skripsi ini akan dibahas teknik untuk menempatkan kapasitor pada jaringan distribusi dengan menggunakan metode Nondominated Sorting Genetic Algorithm II (NSGA-II).

1.2 Rumusan Masalah

Bertambahnya usaha-usaha disektor industri menyebabkan beban induktif yang membutuhkan daya reaktif semakin meningkat. Daya reaktif yang banyak digunakan akan dapat memperbesar rugi-rugi daya dan memperburuk kondisi tegangan. Salah satu alternatif yang dipergunakan untuk memperbaiki kondisi tegangan akibat adanya rugi-rugi daya adalah dengan memasang kapasitor.

Pemasangan kapasitor yang tidak optimal akan semakin memperburuk kondisi tegangan dan meningkatkan rugi-rugi daya, selain itu biaya kerugian PLN semakin besar. Karena itu pada pemasangan kapasitor harus ditentukan letak, jumlah dan kapasitas kapasitor yang seoptimal mungkin pada jaringan distribusi primer.

Dari permasalahan yang timbul diatas, maka skripsi ini diberi judul :

“ANALISIS PENEMPATAN KAPASITOR SHUNT PADA JARINGAN DISTRIBUSI PRIMER 20 KV TYPE RADIAL DENGAN MENGGUNAKAN METODE *NONDOMINATED SORTING GENETIC ALGORITHM II (NSGA-II)* DI PENYULANG TEJAKULA, BALI”

1.3 Tujuan Pembahasan

Tujuan penelitian ini adalah :

1. Menentukan letak kapasitor yang optimal dan besar kapasitasnya.
2. Meningkatkan profil tegangan pada penyulang Tejakula, Bali.
3. Mereduksi rugi-rugi daya pada penyulang Tejakula, Bali.
4. Memaksimalkan total keuntungan (penghematan biaya) yang diperoleh setelah pemasangan kapasitor.

1.4 Batasan Masalah

Untuk menyederhanakan masalah yang akan dibahas, maka diberikan asumsi-asumsi serta batasan-batasan sebagai berikut :

1. Jaringan yang akan dianalisa adalah jaringan distribusi primer type radial 20 kV yang belum dipasang kapasitor di penyulang Tejakula, GI Pamaron – Bali.
 2. Metode yang digunakan dalam perhitungan aliran daya adalah Metode Newton Raphson.
 3. Kapasitor yang dianalisa adalah kapasitor tetap.
 4. Analisa dilakukan pada saat beban puncak.
 5. Analisa penentuan letak dan kapasitas kapasitor shunt menggunakan Metode Nondominated Sorting Genetic Algorithm II (NSGA-II).
 6. Tidak membahas penyebab gangguan.
-

1.5 Metodologi Pembahasan

Metode yang digunakan dalam pembahasan skripsi ini dilakukan dengan langkah-langkah :

1. Studi Literatur

Yaitu kajian pustaka dengan mempelajari teori-teori yang terkait melalui literatur yang ada, yang berhubungan dengan pengaruh yang terjadi akibat penggunaan kapasitor shunt pada sistem distribusi radial.

2. Pengumpulan data

Pengumpulan data lapangan yang dipakai dalam objek penelitian yakni pada PT. PLN (persero) Area Jaringan Bali Utara, penyulang Tejakula GI Pamaran.

- Data kuantitatif yaitu data yang dapat dihitung atau data yang berbentuk angka-angka.
- Data kualitatif yaitu data yang berbentuk diagram. Dalam hal ini adalah single line diagram penyulang.

3. Melakukan analisa dengan menggunakan bahasa program Delphi 7.0.

- a) Analisa Aliran Daya pada penyulang Tejakula sistem 20 kV menggunakan Metode Newton Raphson.
 - b) Analisa penentuan letak, kapasitas dan jumlah kapasitor menggunakan Metode Nondominated Sorting Genetic Algorithm II (NSGA-II).
-

c) Analisa pengaruh pemasangan kapasitor terhadap sistem distribusi radial diantaranya perbaikan profil tegangan, pengurangan rugi-rugi serta penghematan maksimal dan menganalisa aliran daya kembali sistem menggunakan Metode Newton Raphson.

1.6 Kontribusi

Adapun kontribusi dari skripsi ini adalah diharapkan penggunaan dari Metode Nondominated Sorting Genetic Algorithm II (NSGA-II) dalam menganalisa penyaluran daya reaktif sehingga dapat menekan rugi-rugi daya dan dengan biaya operasional yang sangat murah dalam sistem dapat diterapkan oleh pemerintah khususnya PT. PLN (persero), sehingga dapat menambah keuntungan bagi PLN sebagai produsen penyedia energi listrik di Indonesia.

1.7 Sistematika Pembahasan

Untuk mendapat arah yang tepat mengenai hal-hal yang akan dibahas maka dalam skripsi ini disusun sebagai berikut :

BAB I : PENDAHULUAN

Merupakan pendahuluan yang meliputi latar belakang yang melandasi skripsi yang dibahas, rumusan masalah, tujuan yang ingin dicapai, batasan masalah, metodologi penulisan dan sistematika penulisan.

BAB II : SISTEM JARINGAN DISTRIBUSI PRIMER DAN PERBAIKAN FAKTOR DAYA

Disini akan menguraikan mengenai sistem jaringan distribusi, pengertian jatuh tegangan, pengaruh rendahnya faktor daya dan penjelasan teori tentang kapasitor.

BAB III : METODE *NONDOMINATED SORTING GENETIC ALGORITHM II (NSGA-II)* UNTUK MENENTUKAN LETAK DAN KAPASITAS KAPASITOR PADA JARINGAN DISTRIBUSI PRIMER 20 KV

Pada bab ini akan dibahas sedikit metode aliran daya Newton Raphson, teori tentang Metode Nondominated Sorting Genetic Algorithm II (NSGA-II) serta perhitungan optimasi penempatan kapasitor dan harga penghematan biaya.

BAB IV : ANALISA KAPASITAS DAN PENEMPATAN KAPASITOR PADA JARINGAN DISTRIBUSI PRIMER 20 KV PENYULANG TEJAKULA, BALI

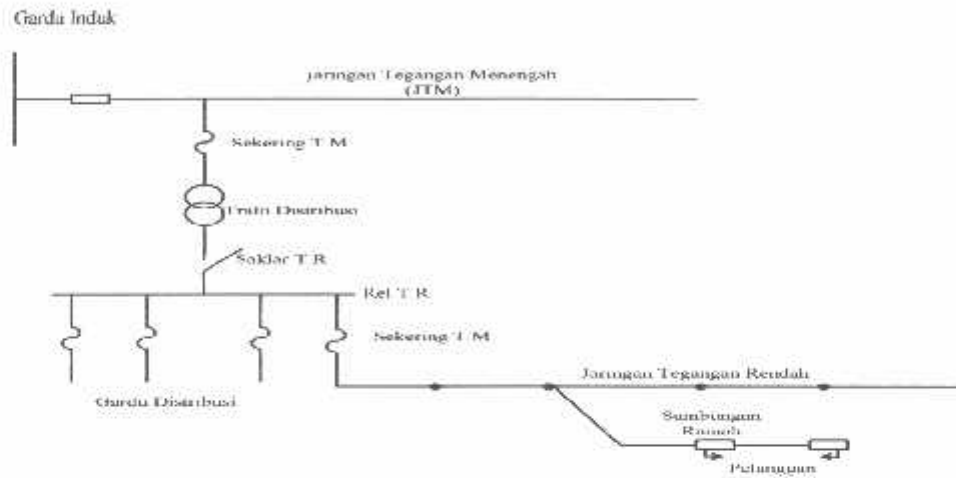
Pada bab ini akan dibahas penempatan kapasitor yang akan dianalisis menggunakan program komputer dengan menggunakan pendekatan Metode Nondominated Sorting Genetic Algorithm II (NSGA-II) serta perbandingan profil tegangan sebelum dan sesudah pemasangan kapasitor.

BAB V : KESIMPULAN DAN SARAN

Merupakan bab terakhir yang merupakan intisari dan hasil pembahasan, berisikan kesimpulan dan saran-saran yang dapat digunakan sebagai pertimbangan untuk pengembangan penulisan selanjutnya.

BAB II
SISTEM JARINGAN DISTRIBUSI PRIMER DAN PERBAIKAN
FAKTOR DAYA

2.1 Sistem Distribusi



Sumber : Djirang Maesudi, 1990, "Operasi Tenaga Listrik", Balai Penerbit dan Humas ISTN.

Gambar 2-1
Jaringan Distribusi Tegangan Menengah (JTM), Jaringan Tegangan Rendah (JTR) dan Sambungan Rumah ke Pelanggan

Jaringan setelah keluar dari G.I. biasa disebut jaringan distribusi. Setelah tenaga listrik disalurkan melalui jaringan distribusi primer maka kemudian tenaga listrik diturunkan tegangannya dalam gardu-gardu distribusi menjadi tegangan rendah, kemudian disalurkan melalui jaringan tegangan rendah untuk selanjutnya disalurkan ke rumah-rumah pelanggan (konsumen) PLN melalui sambungan rumah.

Dalam pendistribusian tenaga listrik ke konsumen, tegangan yang digunakan bervariasi tergantung dari jenis konsumen yang membutuhkan. Untuk konsumen industri biasanya digunakan tegangan menengah 20 kV sedangkan untuk konsumen perumahan digunakan tegangan rendah 220/380 Volt, yang merupakan tegangan siap pakai untuk peralatan-peralatan rumah tangga. Dengan demikian maka sistem distribusi tenaga listrik dapat diklasifikasikan menjadi dua bagian sistem yaitu :

1. Sistem Distribusi Primer.
2. Sistem Distribusi Sekunder.

Pengklasifikasian sistem distribusi tenaga listrik menjadi dua ini berdasarkan tingkat tegangan distribusinya.

2.1.1 Sistem Distribusi Primer

Tingkat tegangan yang digunakan pada sistem distribusi primer adalah meliputi tegangan menengah 20 kV, oleh karena itu sistem distribusi ini sering disebut dengan sistem distribusi tegangan menengah.

2.1.2 Sistem Distribusi Sekunder

Tingkat tegangan yang digunakan pada sistem distribusi sekunder adalah tegangan rendah 127/220 Volt atau 220/380 Volt, oleh karena itu distribusi ini sering disebut dengan jaringan distribusi tegangan rendah.

Sistem jaringan yang digunakan untuk menyalurkan dan mendistribusikan tenaga listrik tersebut dapat menggunakan satu fasa dengan dua kawat maupun dengan sistem tiga fasa dengan empat kawat.

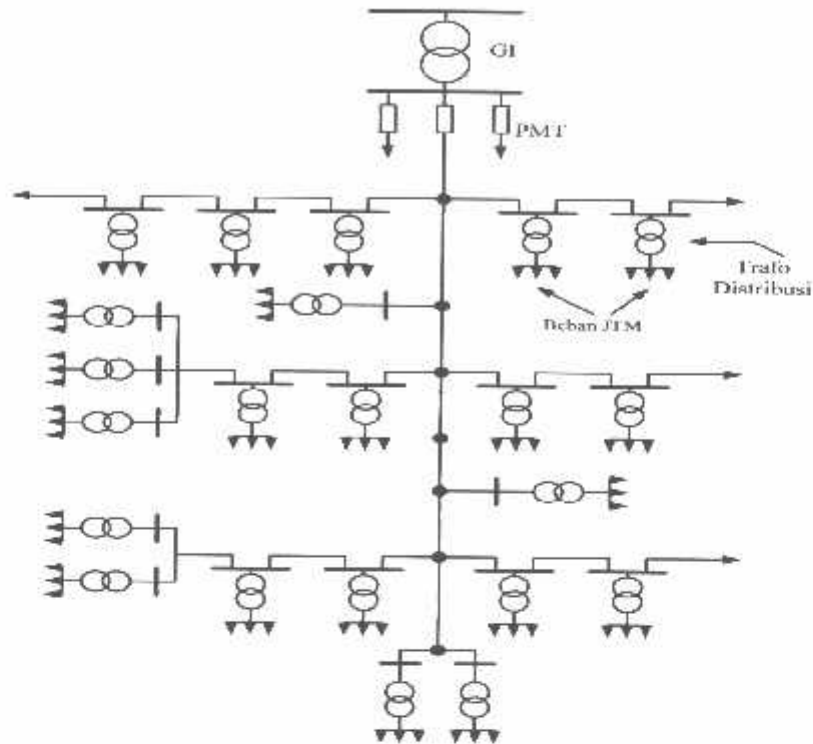
2.2 Struktur Jaringan Distribusi Tenaga Listrik

Ada beberapa bentuk jaringan yang umum dipergunakan untuk menyalurkan dan mendistribusikan tenaga listrik yaitu :

1. Sistem jaringan distribusi radial
2. Sistem jaringan distribusi rangkaian tertutup (loop)
3. Sistem jaringan distribusi mesh

2.3 Sistem Jaringan Distribusi Radial

Bentuk jaringan ini merupakan bentuk dasar yang paling sederhana dan paling banyak digunakan. Ini dikatakan radial karena dari kenyataanya bahwa jaringan ini ditarik secara radial dari gardu induk ke pusat-pusat beban/ konsumen yang dilayaninya. Sistem ini terdiri dari saluran utama (tunk line) dan saluran cabang (lateral) seperti pada gambar 2-2.



Sumber : Turan Gonen, "Electric Power Distribution System Engineering", University of Missouri at Columbia.

Gambar 2-2
Contoh Jaringan Distribusi Radial

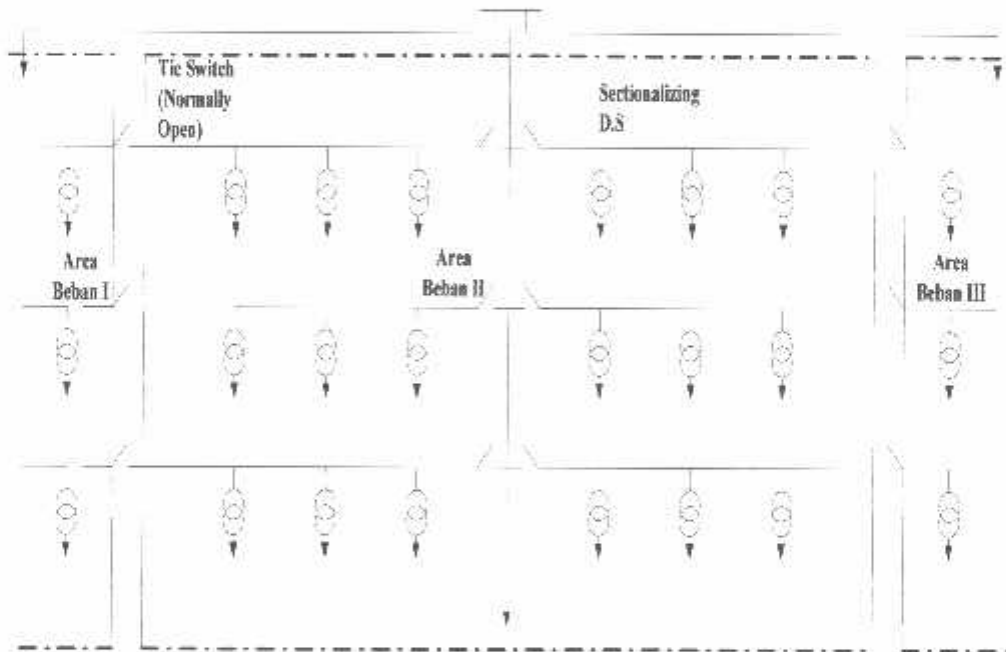
Pelayanan tenaga listrik untuk suatu daerah beban tertentu dilaksanakan dengan memasang transformator pada sembarang titik pada jaringan yang sedekat mungkin dengan daerah beban yang dilayani. Untuk daerah beban yang menyimpang jauh dari saluran utama maupun saluran cabang, maka akan ditarik lagi saluran tambahan yang dicabangkan pada saluran tersebut.

Kelemahan yang dimiliki oleh sistem radial adalah jatuh tegangan yang cukup besar dan bila terjadi gangguan pada sistem akan dapat mengakibatkan jatuhnya sebagian atau bahkan keseluruhan beban sistem.

2.3.1 Sistem Radial Pohon

Radial jaringan pohon ini merupakan bentuk yang paling dasar dari sistem jaringan radial. Saluran utama (main feeder) ditarik dari suatu Gardu Induk sesuai dengan kebutuhan kemudian dicabangkan melalui saluran cabang (lateral feeder), selanjutnya dicabangkan lagi melalui saluran anak cabang (sub lateral feeder). Ukuran dari masing-masing saluran tergantung dari kerapatan arus yang ditanggung. Dari gambar 2-2, main feeder merupakan saluran yang dialiri arus terbesar, selanjutnya arus mengecil pada tiap cabang tergantung dari besarnya beban.

2.3.2 Sistem Radial dengan Tie dan Switch Pemisah



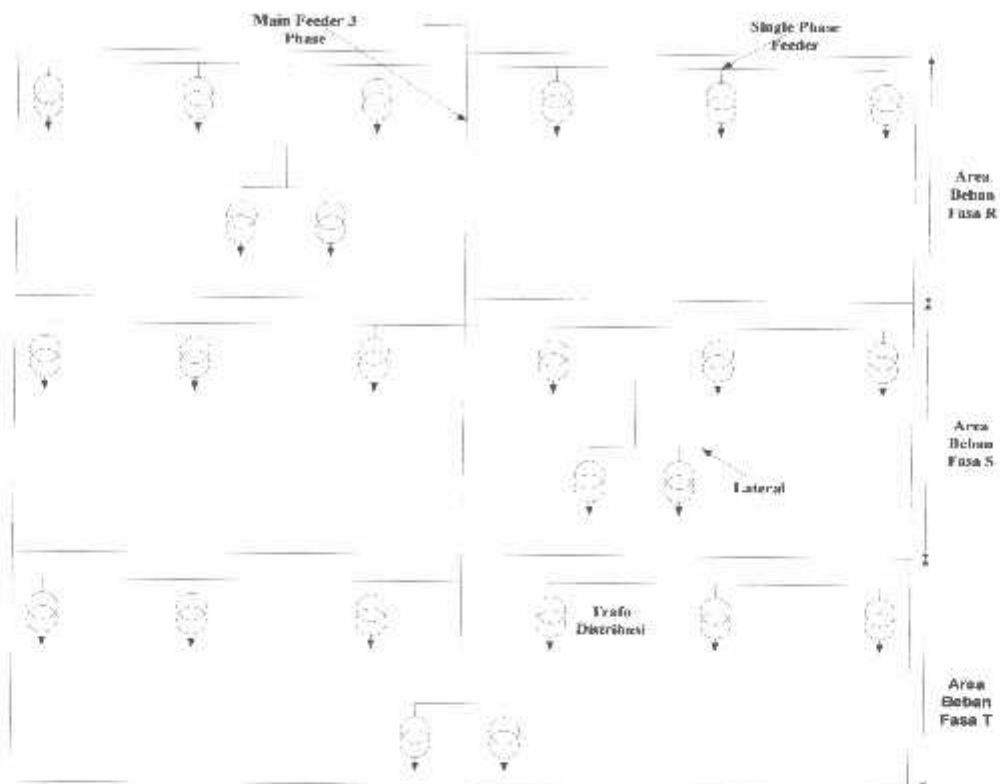
Sumber: Turan Gonen, "Electric Power Distribution System Engineering", University of Missouri at Columbia.

Gambar 2-3
Contoh Jaringan Distribusi Radial dengan Tie dan Switch Pemisah

Sistem ini merupakan pengembangan dari sistem radial pohon, untuk meningkatkan keandalan sistem saat terjadinya gangguan maka feeder yang terganggu akan dilokalisir sedangkan area yang semula dilayani feeder tersebut pelayanannya dialihkan pada feeder yang sehat atau yang tidak terganggu. Sistem radial dengan tie dan switch pemisah dapat dilihat pada gambar 2-3.

2.3.3 Sistem Radial dengan Pembagian Phase Area

Pada bentuk ini masing-masing fasa dari jaringan bertugas melayani daerah yang berlainan. Bentuk ini akan dapat menimbulkan kondisi sistem tiga fasa yang tidak seimbang (simetris), bila digunakan pada daerah beban yang baru dan belum mantap pembagian bebannya. Contoh dari sistem jaringan ini dapat dilihat pada gambar 2-4.

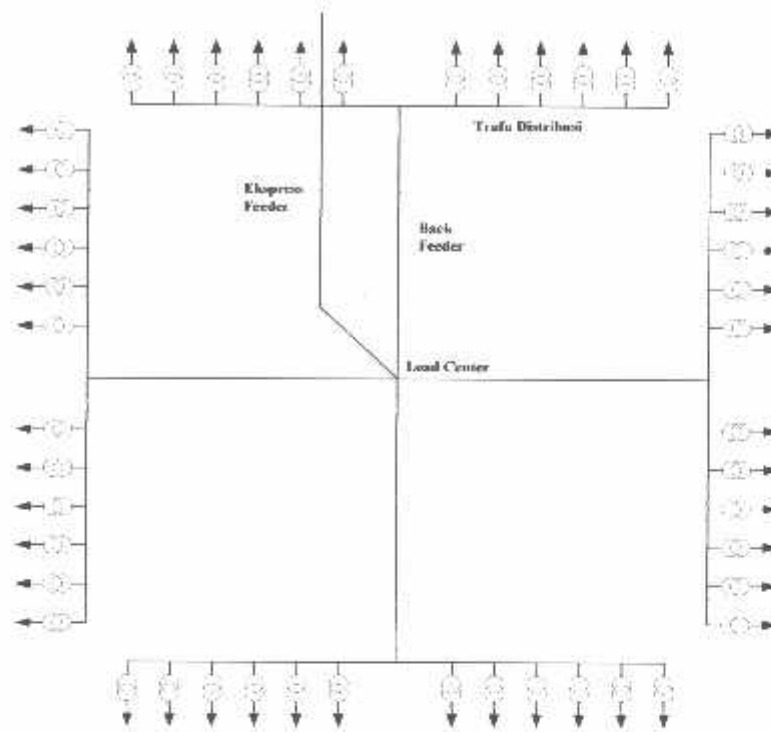


Sumber: Turan Gonen, "Electric Power Distribution System Engineering", University of Missouri at Columbia.

Gambar 2-4
Contoh Jaringan Distribusi Radial dengan Phase Area

2.3.4 Sistem Radial dengan Beban Terpusat

Bentuk dari sistem ini mensuplai daya dengan menggunakan main feeder yang disebut express feeder langsung kepusat beban, dan dari titik pusat beban ini disebut dengan menggunakan back feeder secara radial seperti terlihat pada gambar 2-5.



Sumber : Turan Gunen, "Electric Power Distribution System Engineering", University of Missouri at Columbia.

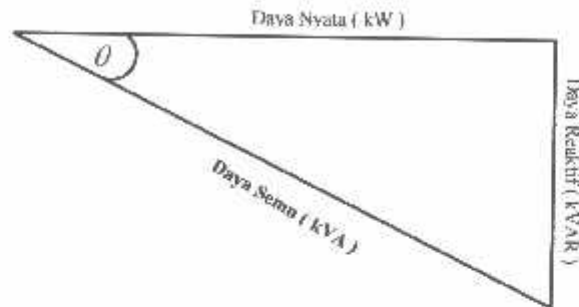
Gambar 2-5
Contoh Jaringan Distribusi dengan Beban Terpusat

2.4 Daya Reaktif dan Faktor Daya

Setiap pemakaian daya reaktif akan menyebabkan turunnya faktor daya yang kemudian menyebabkan memburuknya karakteristik kerja peralatan-peralatan sistem pada umumnya, baik dari segi teknik operasional maupun segi ekonomisnya. Faktor daya adalah perbandingan antara daya nyata dengan daya semu.

$$\text{Faktor daya} = \frac{\text{Daya nyata (kW)}}{\text{Daya semu (kVA)}} \dots\dots\dots(2.1)$$

Untuk daya semu sendiri dibentuk oleh dua komponen daya nyata (kW) dan komponen daya reaktif (kVAR). Hubungan ini dapat digambarkan sebagai berikut :



Sumber : Hasan Basri, "Sistem Distribusi Tenaga Listrik".

Gambar 2-6
Segitiga Daya

$$\text{Dengan faktor daya} = \cos \varphi = \frac{P}{S}$$

P = Daya nyata (kW)

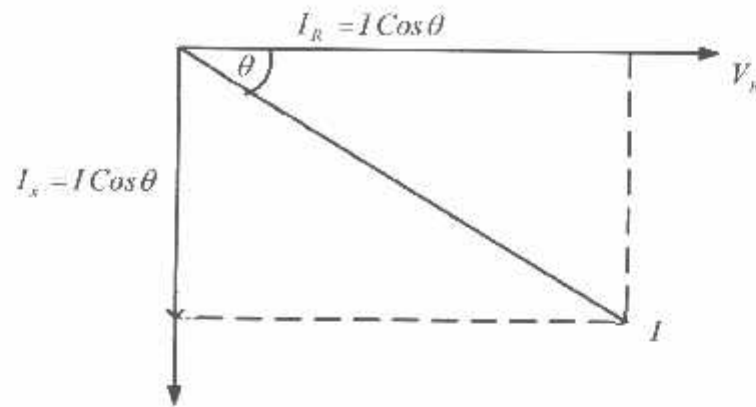
S = Daya Semu (kVA)

Q = Daya Reaktif (kVAR)

φ = Sudut Phase

Faktor daya dapat pula dilihat hubungannya dengan arus nyata dengan arus total. Arus nyata (I_R) adalah arus yang mengalir pada beban yang diubah menjadi tenaga. Sedangkan arus total (I) adalah arus yang mengalir pada jaringan dan merupakan jumlah vektor antara arus magnetisasi (I_X) merupakan arus yang mengalir pada beban dan menimbulkan medan magnet.

Arus yang terbaca dalam ampere meter adalah arus total (I) hubungan ketiga arus tersebut seperti pada gambar 2-7 sebagai berikut :



Sumber: Hasan Basri, "Sistem Distribusi Tenaga Listrik".

Gambar 2-7
Segitiga Arus

2.5 Kapasitor Daya

Secara sederhana kapasitor terdiri dari dua buah pelat logam yang dipisahkan oleh suatu bahan dielektrik dan kapasitor ini mempunyai sifat menyimpan muatan listrik. Pada beberapa tahun lalu kebanyakan kapasitor terbuat dari dua buah pelat aluminium murni yang dipisahkan oleh tiga atau lebih lapisan kertas yang dilapisi oleh bahan kimia. Kapasitor daya telah mengalami perkembangan yang begitu cepat selama 30 tahun terakhir. Karena bahan dielektrik yang digunakan lebih efisien serta teknologi pembuatan kapasitor lebih baik.

2.5.1 Kapasitor Seri dan Kapasitor Shunt

Pemasangan kapasitor daya terdiri atas 2 bagian yaitu kapasitor shunt dan kapasitor seri, dalam sistem tenaga membangkitkan daya reaktif untuk memperbaiki faktor daya dan tegangan, sehingga meningkatkan kapasitas sistem

dan mengurangi rugi-rugi jaringan. Biaya pemasangan kapasitor seri lebih tinggi dari pada pemasangan kapasitor shunt, hal ini disebabkan karena peralatan pelindung untuk kapasitor seri lebih banyak, biasanya kapasitor seri didesain untuk daya yang lebih besar dari pada kapasitor shunt dengan tujuan untuk mengatasi kenaikan beban.

2.5.2 Faktor-Faktor Pemilihan Kapasitor Seri dan Kapasitor Shunt

Faktor- faktor yang mempengaruhi pemilihan antara kapasitor shunt dan kapasitor seri ditabelkan sebagai berikut :

Tabel 2-1
Pemilihan Kapasitor Seri dan Kapasitor Shunt

NO	KEBUTUHAN	Pemilihan Kapasitor	
		Seri	Shunt
1	Memperbaiki Faktor Daya.	Kedua	Pertama
2	Memperbaiki tingkat tegangan pada sistem saluran udara dengan faktor daya normal dan rendah.	Pertama	Kedua
3	Memperbaiki tingkat tegangan pada sistem saluran udara dengan faktor daya yang tinggi.	Tidak dipakai	Pertama
4	Memperbaiki tingkat tegangan pada sistem saluran udara bawah tanah dengan faktor daya tinggi.	Tidak dipakai	Tidak dipakai
5	Memperbaiki tingkat tegangan pada sistem saluran bawah tanah dengan faktor daya normal dan rendah.	Pertama	Tidak dipakai
6	Mengurangi rugi-rugi saluran.	Kedua	Pertama
7	Mengurangi fluktuasi tegangan.	Pertama	Tidak dipakai

2.6 Pengaruh Pemasangan Kapasitor Shunt

Kapasitor shunt adalah kapasitor yang dihubungkan paralel dengan saluran dan secara intensif digunakan pada sistem distribusi. Kapasitor shunt mencatu daya reaktif atau arus yang menentang komponen arus beban induktif. Dengan dipasangnya kapasitor shunt pada jaringan distribusi akan dapat memperbaiki profil tegangan, memperbaiki faktor daya, dan menaikkan kapasitas sistem serta dapat mengurangi rugi saluran.

Ada dua cara dalam pemakaian kapasitor shunt

- Kapasitor tetap
- Kapasitor saklar

a. Kapasitor Tetap

Adalah kapasitor untuk mengkompensasi daya reaktif yang kapasitasnya tetap dan selalu terpasang di jaringan. Penggunaan kapasitor jenis ini harus memperhatikan kenaikan tegangan yang terjadi pada saat beban ringan agar tidak melebihi batas tegangan yang telah ditetapkan.

b. Kapasitor Saklar

Adalah kapasitor untuk kompensasi daya reaktif yang dapat dihubungkan dan dilepaskan dari jaringan dan dapat diatur besar kapasitasnya sesuai dengan kondisi beban.

Proses membuka-menutup dari saklar kapasitor shunt dapat dilakukan dengan cara manual maupun dengan cara otomatis. Pengendalian secara manual (pada lokasi atau kendali jarak jauh) dapat dilakukan pada GI. Untuk pengendalian secara otomatis, termasuk didalamnya peralatan pengendali tegangan, arus, tegangan-waktu dan suhu. Type yang paling populer adalah pengendali saklar waktu (time-switch control), pengendali tegangan dan pengendali tegangan-arus.

2.6.1 Pengurangan Rugi-Rugi Dengan Kapasitor Shunt

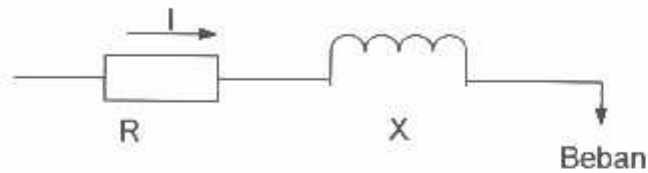
Rugi-rugi saluran per fasa dari 3 fasa scimbang dengan beban terpusat seperti pada gambar 2-8 adalah $I^2 (R-jX)$ atau dapat dibedakan menjadi :

$$- \text{Rugi Daya Aktif } (I^2R) = (I_R^2 + I_X^2)R$$

$$- \text{ Rugi Daya Reaktif } (I^2X) = (I_R^2 + I_X^2)$$

Dimana : I_R adalah komponen arus aktif

I_X adalah komponen arus reaktif



Sumber : Hasan Basri, "Sistem Distribusi Tenaga Listrik".

Gambar 2-8
Saluran Primer Dengan Beban Terpusat

Rugi-rugi daya (I^2R) dapat dibagi menjadi dua komponen yaitu komponen arus aktif dan komponen arus reaktif. Rugi-rugi karena komponen arus aktif tidak akan mempengaruhi penempatan kapasitor shunt pada saluran, hal ini dapat dijelaskan sebagai berikut :

Diasumsikan bahwa rugi-rugi daya (I^2R) disebabkan oleh arus saluran (lagging) I , yang mengalir pada resistansi R , sehingga :

$$I^2R = (I \cos \theta)^2R + (I \sin \theta)^2R \dots \dots \dots (2.2)$$

Setelah dipasang kapasitor shunt dengan arus I_c , didapat arus saluran baru I_1 , dan rugi-rugi daya I_1R sebagai berikut :

$$I^2R = (I \cos \theta)^2R + (I \sin \theta - I_c)^2R \dots \dots \dots (2.3)$$

Sehingga pengurangan rugi daya sebagai akibat pemasangan kapasitor didapat :

$$\Delta p_{ls} = I^2R - I_1^2R$$

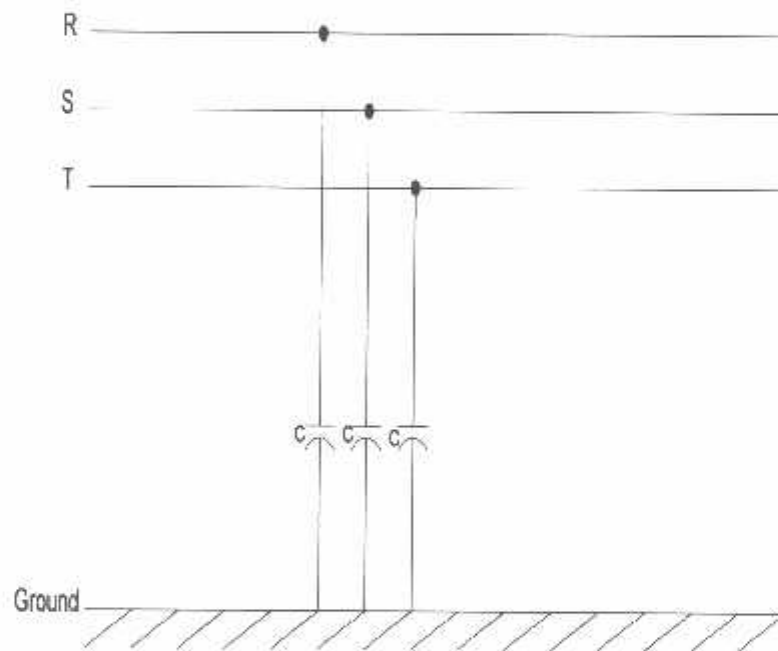
$$\begin{aligned}
 &= (I \cos \theta)^2 R + (I \sin \theta)^2 R - (I \cos \theta)^2 R + (I \sin \theta - I_c)^2 R \\
 &= 2(I \sin \theta) I_c R - I_c^2 R \dots \dots \dots (2.4)
 \end{aligned}$$

Maka hanya komponen arus Reaktif ($I \sin \theta$) saja yang berpengaruh terhadap pengurangan rugi daya $I^2 R$ akibat pemasangan kapasitor shunt pada saluran distribusi. Sehingga pengurangan rugi daya saluran 3 fasa adalah :

$$3R (2 (I \sin \theta) I_c - I_c^2) \text{Watt} \dots \dots \dots (2.5)$$

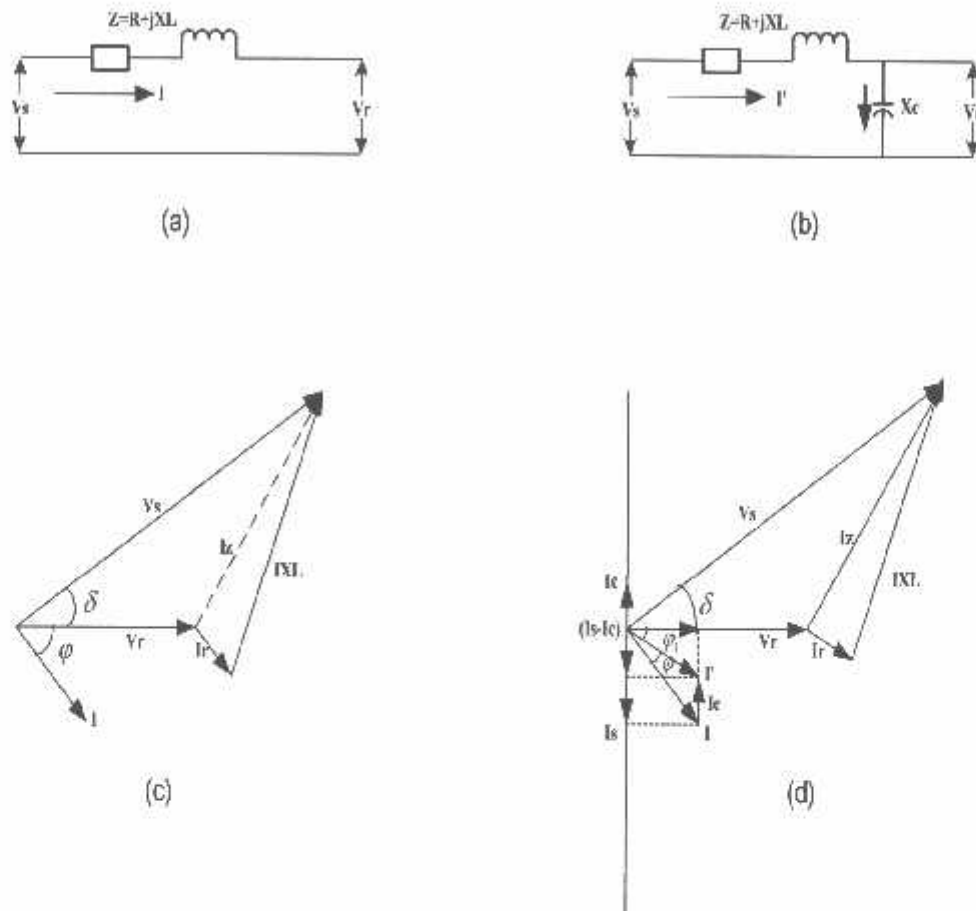
2.6.2 Perbaikan Tegangan

Pemakaian kapasitor shunt dalam sistem tenaga listrik selain untuk memperbaiki faktor daya juga bertujuan menaikkan tegangan. Berikut adalah gambar rangkaian kapasitor shunt pada jaringan tiga fasa :



Gambar 2-9
Gambar Rangkaian Kapasitor Shunt Pada Jaringan Tiga Fasa

Dan secara vektoris dapat digambarkan sebagai berikut :



Sumber : Hasan Basri, "Sistem Distribusi Tenaga Listrik".

Gambar 2-10
Bagan Satu Garis dan Fasor Diagram suatu Penyulang dengan Faktor Daya Tertinggal
 (i) Tanpa Kapasitor, Gambar (a) dan (c)
 (ii) Dengan Kapasitor Shunt, Gambar (b) dan (d)

Kerugian tegangan disebabkan arus beban I sebelum kapasitor dipasang :

$$E = RI_r + XI_L \dots \dots \dots (2.6)$$

Kerugian tegangan setelah kapasitor dipasang :

$$E = I_r + XI_L - X I_c \dots \dots \dots (2.7)$$

$$\cos \theta_1 = \frac{P}{S_1} \dots \dots \dots (2.8)$$

Bila suatu kapasitor Q_c (kVAR) dipasang pada beban, faktor daya dapat diperbaiki dari $\cos \theta_1$ menjadi $\cos \theta_2$ dimana :

$$\begin{aligned} \cos \theta_2 &= \frac{P}{S_2} \\ &= \frac{P}{\sqrt{(P^2 + Q^2)}} \\ &= \frac{P}{\sqrt{P^2 + (Q_1 - Q_c)^2}} \dots \dots \dots (2.9) \end{aligned}$$

Sehingga daya semu dan daya reaktif berkurang dari S_1 (kVA) ke S_2 (kVA) dan dari Q_1 (kVAR) ke Q_2 (kVAR) sehingga kapasitas beban akan meningkat.

Dengan demikian dapat diambil kesimpulan bahwa persentase tegangan rugi-rugi daya jaringan dapat dihitung menggunakan rumus :

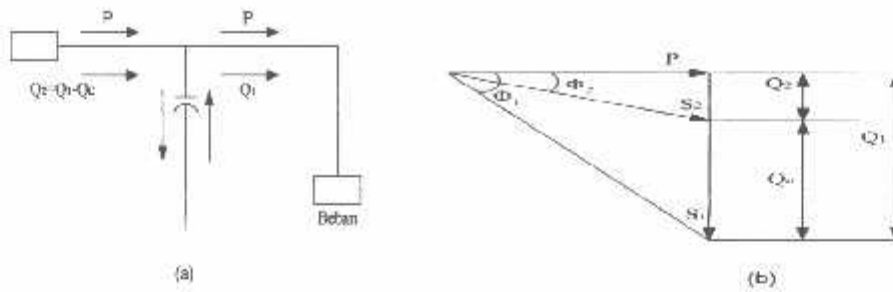
$$\% \text{ Rugi Daya} = 100 \left(\frac{\text{Faktor daya mula - mula } (\cos \theta_1)}{\text{Faktor daya baru } (\cos \theta_2)} \right) \dots \dots \dots (2.10)$$

% Pengurangan rugi daya

$$= 100 \left(1 - \left(\frac{\text{Faktor daya mula - mula } (\cos \theta_1)}{\text{Faktor daya baru } (\cos \theta_2)} \right)^2 \right) \dots \dots \dots (2.11)$$

2.6.3.1 Perhitungan Pengaruh Perbaikan Faktor Daya

Diagram Phase dari dua komponen arus nyata itu arus aktif dan reaktif dapat dilihat pada gambar 2-11 berikut :



Sumber : Hasan Basri, "Sistem Distribusi Tenaga Listrik".

Gambar 2-11
Diagram Fasor dan Sudut Daya Bebas Distribusi

Penjumlahan secara vektor dari arus aktif dan reaktif menghasilkan arus-
arus total yang dapat dinyatakan dengan persamaan :

$$I = \text{Arus Semu} = \sqrt{(\text{Arus aktif})^2 + (\text{Arus reaktif})^2}$$

$$= \sqrt{(I \cos \theta)^2 + (I \sin \theta)^2}$$

Pada suatu tegangan V daya aktif, daya reaktif dan daya nyata adalah
sebanding dengan arus, dimana hubungannya dapat dinyatakan sebagai berikut :

$$\text{Daya semu (kVA)} = \sqrt{(\text{Daya aktif})^2 + (\text{Daya Reaktif})^2}$$

$$(VI) = \sqrt{(VI \cos \theta)^2 + (VI \sin \theta)^2}$$

$$\text{Faktor daya} = \frac{\text{Daya aktif}}{\text{Daya semu}} = \frac{(kW)}{(kVA)}$$

$$\text{Daya aktif} = \text{Daya semu} \times \text{Faktor daya}$$

$$kW = kVA \times \text{Faktor daya}$$

$$kW = kVA \cos \theta$$

2.6.3.2 Penentuan Rating Kapasitor untuk Perbaikan Faktor Daya Beban

Dari hubungan fasor diagram daya aktif dan reaktif dapat ditulis beberapa persamaan matematis sebagai berikut :

$$\cos \theta = \frac{\text{Daya aktif}}{\text{Daya semu}} = \frac{(kW)}{(kVA)}$$

$$\sin \theta = \frac{\text{Daya reaktif}}{\text{Daya semu}} = \frac{(kVAR)}{(kVA)}$$

$$\tan \theta = \frac{\text{Daya reaktif}}{\text{Daya aktif}} = \frac{(kVAR)}{(kW)}$$

Karena komponen daya aktif biasanya konstan, dan daya nyata serta komponen daya reaktif berubah sesuai dengan faktor daya, maka persamaan yang dinyatakan dalam komponen daya aktif yang paling tepat digunakan. Persamaan ini dapat ditulis sebagai berikut :

$$\begin{aligned} \text{Daya reaktif pada faktor daya mula-mula} &= \text{Daya aktif} \times \tan \theta_1 \\ &= (kW) \times \tan \theta_1 \end{aligned}$$

$$\begin{aligned} \text{Daya reaktif pada faktor daya baru} &= \text{Daya aktif} \times \tan \theta_2 \\ &= (kW) \times \tan \theta_2 \end{aligned}$$

Dengan θ_1 = sudut dari faktor daya mula-mula.

θ_2 = sudut dari faktor daya yang telah diperbaiki.

Rating kapasitor yang dibutuhkan perbaikan faktor daya :

$$\text{Daya reaktif (kVAR)} = \text{Daya aktif} \times (\tan \theta_1 - \theta_2)$$

$$(kVAR) = (kW) \times (\tan \theta_1 - \theta_2)$$

Untuk menyederhanakan $(\tan \theta_1 - \theta_2)$ sering ditulis $\Delta \tan$, yang merupakan suatu faktor pengali untuk menentukan daya reaktif.

$$\begin{aligned} \text{Daya reaktif (kVAR)} &= \text{Daya aktif} \times \Delta \tan \\ (\text{kVAR}) &= (\text{kW}) \times \Delta \tan \end{aligned}$$

2.7 Per- Unit

Untuk memudahkan perhitungan-perhitungan dalam sistem tenaga listrik digunakan sistem p.u. (per unit) yang didefinisikan sebagai perbandingan harga yang sebenarnya dengan harga dasar (base value), sehingga dapat dirumuskan sebagai berikut :

$$\text{Besaran per-unit} = \frac{\text{Besaran sesungguhnya}}{\text{Besaran dasar dengan dimensi yang sama}} \dots\dots\dots(2.12)$$

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

❖ Untuk data 1 fasa

Arus dasar

$$I_d = \frac{\text{kVA dasar 1 fasa}}{\text{kV dasar L - N}} \dots\dots\dots(2.13)$$

Impedansi dasar

$$\begin{aligned} Z_d &= \frac{(\text{kV dasar L - N})^2 \times 1000}{\text{kVA dasar 1 fasa}} \\ &= \frac{(\text{kV dasar L - N})^2}{\text{MVA dasar 1 fasa}} \dots\dots\dots(2.14) \end{aligned}$$

Dalam persamaan diatas 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 ini

dinormalisasikan terhadap besaran dasar. Dengan demikian impedansi persatuan didefinisikan sebagai berikut :

$$Z = \frac{\text{impedansi sebenarnya } Z (\Omega)}{\text{impedansi dasar } Z_d} \text{ pu} \dots\dots\dots(2.15)$$

BAB III

**METODE NONDOMINATED SORTING GENETIC ALGORITHM II
(NSGA-II) UNTUK MENENTUKAN LETAK DAN KAPASITAS
KAPASITOR PADA JARINGAN DISTRIBUSI PRIMER 20 KV TYPE
RADIAL**

3.1 Analisa Aliran Daya Sistem Distribusi Radial

Dalam analisis dengan menggunakan metode NSGA-II diperlukan suatu proses aliran daya untuk mengetahui rugi-rugi daya pada saluran. Oleh karena itu pembahasan selanjutnya sebelum proses menggunakan metode NSGA-II adalah analisis aliran daya dengan menggunakan Metode Newton Raphson.

3.1.1 Tujuan

1. Untuk mengetahui keadaan tegangan pada setiap bus dari sistem jaringan.
2. Untuk mengetahui besarnya daya yang mengalir pada setiap cabang dari struktur jaringan.
3. Untuk mengetahui besar rugi-rugi daya aktif dan daya reaktif pada setiap cabang jaringan.

3.1.2 Metode Newton Raphson

Secara matematis persamaan aliran daya Newton Raphson dapat diselesaikan dengan menggunakan koordinat rektangular, koordinat polar atau bentuk hibrid (gabungan antara bentuk kompleks dengan bentuk polar). Dalam pembahasan skripsi ini menggunakan bentuk polar.

Hubungan antara arus simpul I_p dengan tegangan V_q pada suatu jaringan dengan n simpul dapat dituliskan :

$$I_p = \sum_{q=1}^n Y_{pq} V_{pq} \dots\dots\dots(3.1)$$

Injeksi daya pada simpul p adalah :

$$S_p = P_p - jQ_p = V_p^* \cdot I_p \dots\dots\dots(3.2)$$

$$= V_p^* \sum_{q=1}^n Y_{pq} V_{pq} \dots\dots\dots(3.3)$$

Dalam penyelesaian aliran daya dengan Newton Raphson bentuk persamaan aliran daya yang dipilih adalah polar, dimana tegangan dinyatakan dalam bentuk polar, yaitu :

$$V_p^* = |V_p| e^{-j\delta_p}$$

$$V_q = |V_q| e^{j\delta_q}$$

$$Y_{pq}^* = |Y_{pq}| e^{-j\theta_{pq}}$$

Maka persamaan dapat ditulis :

$$P_p - jQ_p = \sum_{q=1}^n |V_p| |V_q| |Y_{pq}| e^{-j(\delta_p - \delta_q + \theta_{pq})} \dots\dots\dots(3.4)$$

Dengan memisahkan bagian riil dan bagian imajiner maka diperoleh :

$$P_p = \sum_{q=1}^n |V_p V_q Y_{pq}| \cos(\delta_p - \delta_q + \theta_{pq}) \dots \dots \dots (3.5)$$

$$Q_p = \sum_{q=1}^n |V_p V_q Y_{pq}| \sin(\delta_p - \delta_q + \theta_{pq}) \dots \dots \dots (3.6)$$

Kedua persamaan diatas akan menghasilkan suatu kumpulan persamaan serempak (simultan) yang tidak linier untuk setiap simpul sistem tenaga listrik. Untuk mengetahui magnitudo tegangan (V) dan sudut fasa (δ) disetiap simpul dapat diselesaikan dengan menggunakan persamaan (3.5) dan (3.6) yang dilinierkan dengan metode Newton Raphson yang dapat dilihat dari persamaan dibawah ini :

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} H & N \\ M & L \end{bmatrix} \begin{bmatrix} \Delta \delta \\ \Delta |V| \end{bmatrix} \dots \dots \dots (3.7)$$

Dimana :

ΔP = Selisih injeksi bersih daya nyata dengan penjumlahan aliran daya nyata tiap saluran yang menghubungkan simpul dengan V yang didapat dari perhitungan iterasi ke-k

ΔQ = Selisih injeksi bersih daya reaktif dengan penjumlahan aliran daya reaktif tiap saluran yang menghubungkan simpul dengan V yang didapat dari perhitungan iterasi ke-k

$\Delta \delta$ = Vektor koreksi sudut fasa tegangan

$\Delta |V|$ = Vektor koreksi magnitudo tegangan

H, L, M, N merupakan elemen-elemen off diagonal dari sub matriks Jaqobian yang dibentuk dengan mendefinisikan persamaan (3.5) dan (3.6), dimana :

$$H_{pq} = \frac{\partial P_p}{\partial \delta_q} \qquad N_{pq} = \frac{\partial P_p}{\partial |V_q|}$$

$$M_{pq} = \frac{\partial Q_p}{\partial \delta_q} \qquad L_{pq} = \frac{\partial Q_p}{\partial |V_q|}$$

Persamaan diselesaikan untuk menghitung vektor koreksi magnitudo tegangan $\Delta(|V|)$ dan sudut fasa tegangan ($\Delta\delta$) yang baru. Sehingga diperoleh harga magnitudo tegangan dan sudut fasa yang baru, yaitu :

$$|V|^{k+1} = |V|^k + \Delta|V|^k$$

$$\delta^{k+1} = \delta^k + \Delta\delta^k$$

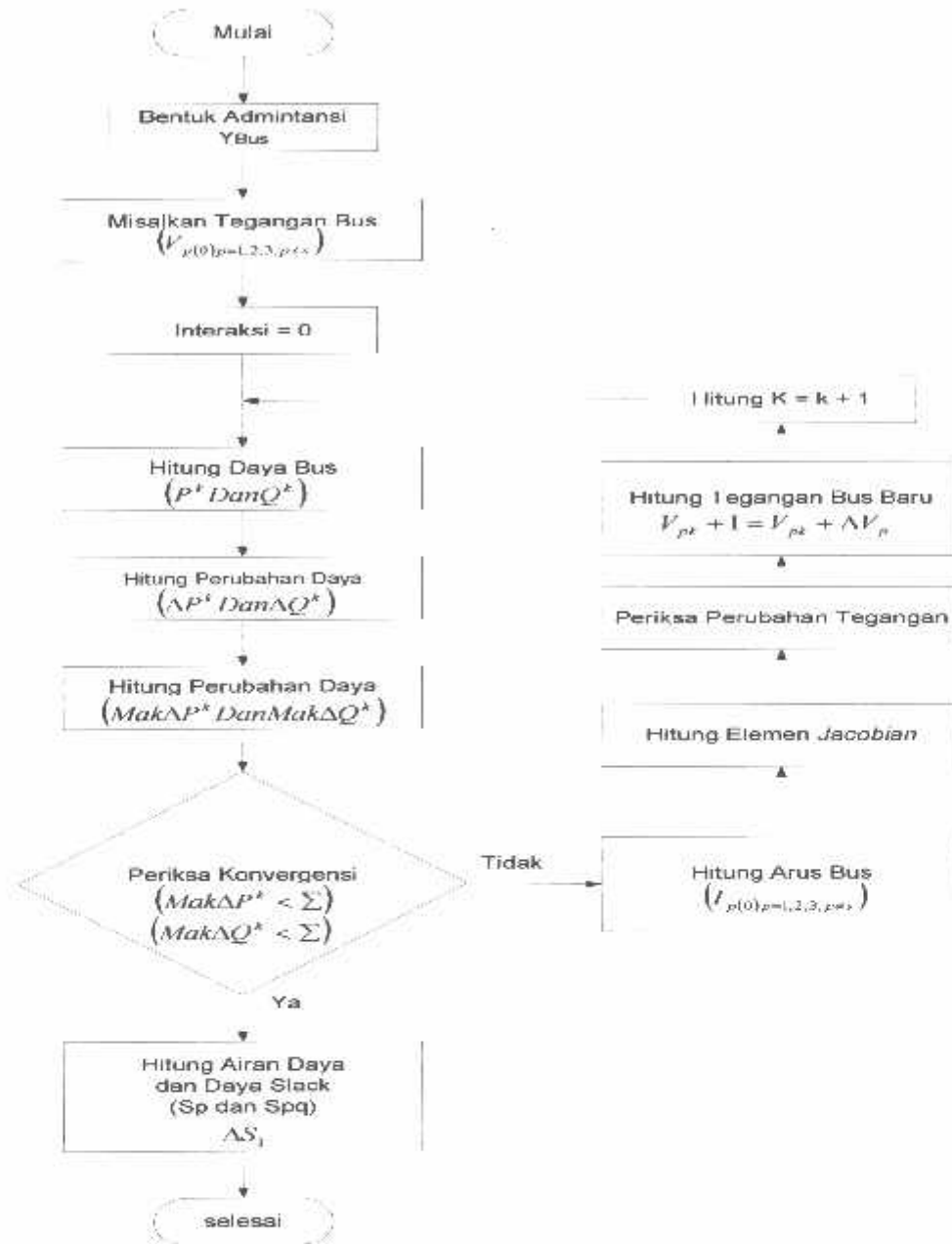
proses perhitungan akan berulang sampai selisih daya nyata dan daya reaktif antara yang dijadwalkan dengan yang dihitung, yaitu ΔP dan ΔQ untuk semua simpul mendekati nilai toleransi atau proses perhitungn iterasi mencapai konvergen.

Algoritma Aliran Daya Newton Raphson

1. Tentukan nilai $P_{p(\text{ditetapkan})}$ dan $Q_{p(\text{ditetapkan})}$ yang mengalir kedalam sistem pada setiap rel untuk nilai yang ditentukan atau perkiraan dari besar dan sudut tegangan untuk iterasi pertama atau tegangan yang ditentukan paling akhir untuk iterasi berikutnya.
2. Hitung ΔP pada setiap rel.
3. Hitung nilai-nilai matriks Jaqobian dengan menggunakan nilai-nilai perkiraan atau yang ditentukan dari besar dan sudut tegangan dalam persamaan untuk turunan parsial yang ditentukan dengan differensiasi persamaan (3.5) dan (3.6).
4. Balikkan Jaqobian itu dan hitung koreksi-koreksi tegangan $\Delta\delta_q$ dan $\Delta|V_q|$ pada nilai sebelumnya.

5. Hitung nilai baru dari δ_q dan $|V_q|$ dengan menambah $\Delta\delta_p$ dan $\Delta|V_q|$ pada nilai sebelumnya.
 6. Kembali kelangkah pertama dan ulangi proses itu dengan menggunakan nilai untuk besar dan sudut tegangan yang ditentukan paling akhir sehingga semua nilai ΔP dan ΔQ atau semua nilai $\Delta\delta$ dan $\Delta|V|$ lebih kecil dari suatu indeks ketetapan yang dipilih.
-

3.1.3. Diagram Alir Aliran Daya Newton Rapshon



3.2 Metode Penempatan Kapasitor

3.2.1 Genetic Algoritma

Algoritma Genetika (GA) adalah algoritma pencarian heuristic yang didasarkan atas mekanisme evolusi biologis. Keberagaman pada evolusi biologis adalah variasi dari kromosom antar individu organisme. Variasi kromosom ini akan mempengaruhi laju reproduksi dan tingkat kemampuan organisme untuk tetap hidup. Pada dasarnya ada 4 kondisi yang sangat mempengaruhi proses evolusi, yaitu :

1. Kemampuan organisme untuk melakukan reproduksi.
2. Keberadaan populasi organisme yang bisa melakukan reproduksi.
3. Keberagaman organisme dalam suatu populasi.
4. Perbedaan kemampuan untuk survive.

Individu yang lebih kuat (fit) akan memiliki tingkat survival dan tingkat reproduksi yang lebih tinggi jika dibandingkan dengan individu yang kurang fit (sering dikenal dengan istilah generasi), populasi secara keseluruhan akan lebih banyak memuat organisme fit.

Tahap 1. Seleksi

Pada algoritma genetika terdapat proses seleksi yaitu proses pemilihan kromosom yang akan di-crossover-kan dengan kromosom dari individu lain. Masalah yang paling mendasar pada proses ini adalah bagaimana proses penyeleksiannya. Menurut teori evolusi Darwin proses seleksi individu adalah :

“Individu terbaik akan tetap hidup dan menghasilkan keturunan”. Pada proses seleksi ini dapat menggunakan banyak metode seperti Reulette Wheel Selection.

Tahap 2. Eletism

Selama membuat populasi baru dengan crossover dan mutasi, kemungkinan akan terjadi kehilangan kromosom terbaik (best/few best). Elitism adalah nama metode yang pertama kali mengkopikan kromosom terbaik (best/few best) ke dalam populasi baru. Sisanya dikerjakan dengan cara biasa, yaitu melalui seleksi, crossover dan mutasi. Elitism dapat secara cepat meningkatkan performasi dari algoritma genetika karena elitism menghindarkan hilangnya solusi terbaik (best/few best) yang telah ditemukan.

Tahap 3. Crossover (pindah silang)

Fungsi dari cadalah 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 frekwensi crossover. Apabila nilai $r_k < p_c$ maka kromosom ke-k terpilih untuk mengalami crossover.

Tahap 4. Mutation (mutasi)

Operasi mutasi digunakan untuk memodifikasi satu atau lebih nilai gen dalam satu individu. Cara kerjanya dengan membangkitkan sebuah nilai random r_k dimana $k = 1, 2, \dots, n\text{VAR}$ (panjang kromosom). Probabilitas mutasi (p_m)

ditentukan dan digunakan untuk mengendalikan frekwensi mutasi. Apabila nilai random $r_k < p_m$ maka gen ke- k kromosom tersebut terpilih untuk mengalami mutasi. Mutasi dengan menggantikan 0 dengan 1 atau sebaliknya gen 1 ke 0. Biasanya disebut dengan proses flip yaitu dengan membalik nilai 0 ke 1 atau 1 ke 0.

Proses optimasi dasar adalah mengkombinasi atau menggabungkan bagian populasi yang sehat untuk menghasilkan individu yang sehat sebagai kemajuan dari pencarian tersebut.

Langkah kerja di dalam penerapan metode algoritma genetika adalah :

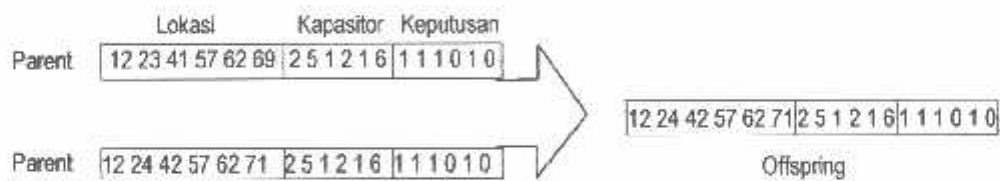
1. Indikasi parameter yang berpengaruh.
2. Konversi parameter ke dalam substring.
3. Menyusun substring tersebut (gen) menjadi string (kromosom, individu).
4. Melakukan ranking terhadap nilai kecocokan masing-masing individu.
5. Melakukan reproduksi, crossover dan mutasi.
6. Mengambil string yang terpilih menjadi parameter.
7. Mengulangi proses tersebut sampai diperoleh suatu nilai parameter yang konvergen.

3.2.2. Konsep Nondominance Genetic Algorithm

Permasalahan multiobjektive terdiri dari k objective functions bertujuan untuk memperkecil secara keseluruhan. Solusion x dikatakan mendominasi solusi yang lain y jika lebih baik dari y untuk akhir dari suatu sasaran f_j tidak lebih buruk dari penyelesaian yang lain, dimana $j = 1, \dots, k$ dan $j \neq i$.

Pengkodean

Pengkodean digunakan sebagai gambaran dari masing-masing bagian kromosom dalam algoritma genetik. kromosom pada bagian ini terdiri dari 3 bagian. Bagian pertama diasumsikan sebagai lokasi yang bernilai integer yang menyatakan nomor bus. Bagian kedua dari kromosom tersusun atas nilai integer yang berisikan no. Urut kapasitas kapasitor. Bagian ketiga diasumsikan keputusan yang bernilai biner. Jika 1, maka kapasitor harus ditempatkan pada bus tersebut dan jika 0, maka sebaliknya panjang kromosom menyatakan jumlah bus.



Sumber : Sri Kusuma Dewi, 2003, "Artificial Intelligence", Penerbit: Graha Ilmu.

Gambar 3-1
Pengkodean Kromosom

3.3. Penentuan *Saving Cost* (S)

Objektivitas pada penempatan kapasitor adalah untuk mengurangi rugi-rugi daya dan menjaga tegangan dengan batas yang diijinkan dengan biaya yang minimum. Rugi-rugi daya P_{Loss} diberikan rumus :

$$P_{Loss}(i, i+1) = R_{i+1} \frac{P_i^2 - Q_i^2}{|V_i|^2} \dots\dots\dots(3.8)$$

Dimana :

P_i = Daya Nyata yang keluar dari Bus i

Q_i = Daya Reaktif yang keluar dari Bus i

$R_{i,i+1}$ = Resistansi yang ada pada saluran Bus i dan $i + 1$

$X_{i,i+1}$ = Reaktansi yang ada pada saluran Bus i dan $i + 1$

Sedangkan untuk rugi-rugi daya total $P_{T, Loss}$ diberikan rumus :

$$P = \sum_{i=0}^{n-1} P_{Loss(i,i+1)} \dots\dots\dots(3.9)$$

Pertimbangan biaya investasi, terdapat nomor bilangan terbatas pada ukuran kapasitor standart yaitu perkalian integer pada ukuran Q_c yang paling kecil. *Cost* per kV berpariasi dari ukuran satu keukuran yang lain. Umumnya ukuran yang lebih besar lebih murah dari ukuran yang lebih kecil, memasukkan ukuran maksimum kapasitor yang diijinkan menjadi batasan untuk :

$$Q_{max}^c = L \cdot Q_0 \dots\dots\dots(3.10)$$

Dimana L adalah integer, pada lokasi yang telah dipilih lainnya, terdapt L ukuran sebagai pilihan. Masukkan $K_1^c, K_2^c, \dots, K_L^c$ menjadi modal infestasi yang sesuai kVar. Mengasumsikan bahwa hanya kapasitor bank digunakan untuk perbaikan tegangan, fungsi *cost* S sebagai Fungsi *fitness* bisa diperoleh dengan :

$$Min\ fc = MKi + Kc \sum_{i=1}^M CI \dots\dots\dots(3.11)$$

Min fc = biaya rugi daya nyata + biaya penempatan

$$S = K_{\rho} * P_{loss} + \sum_{j=1}^k K_j^c Q_j^c \dots\dots\dots(3.12)$$

$$\Delta V_j^p = V_{Min}^p \leq V_j \leq V_{Max}^p \quad j = 1, \dots, N \dots\dots\dots(3.13)$$

$$\left. \begin{aligned} gl &= \max\{0, pf_{min}^p - pf\} \\ gl &= \max_j \{\Delta V_j^p\} \end{aligned} \right\} \dots\dots\dots(3-14)$$

Dimana :

S : Nilai penghematan (Saving)(\$/th)

K_p : Adalah biaya daya yaitu yaitu biaya pembelian, adalah cost per rugi daya (\$/kW/Tahun), nilai K_p sudah ditentukan yaitu U.S.\$ 168/kW/Tahun.

P_{loss} : Rugi-rugi daya Aktif.....(kW)

K^c : Biaya kapasitor(\$/kVar/th)

Q^c : Kapasitas kapasitor(kVar)

$j = 1,2,3,\dots,k$ menggambarkan bus yang telah dipilih. Fungsi objektif adalah untuk meminimalisasi subject ke batas tegangan.

$V_{min} = 0.95$ p.u dan $V_{max} = 1.05$ p.u.

3.3.1. Biaya Daya (K_p)

Biaya Daya (K_p) adalah biaya investasi per-kW ditambah dengan biaya operasi dan pemeliharaan. Dalam skripsi ini berdasarkan referensi dari jurnal maka biaya daya (K_p) diset yaitu sebesar US\$ 168/kW/tahun.

3.3.2. Biaya Ekuivalen Instalasi Kapasitor (K^c)

Harga kapasitor mempunyai fungsi yang diskrit, karena kapasitor merupakan kelipatan bulat dari unit kapasitor standart terkecil, biaya ekuivalen instalasi kapasitor dengan biaya kapasitor. Biaya instalasi kapasitor meliputi harga pembelian kapasitor dan biaya pemeliharaan serta biaya operasinya.

BAB IV

ANALISA KAPASITAS DAN PENEMPATAN KAPASITOR PADA JARINGAN DISTRIBUSI PRIMER 20 KV PENYULANG TEJAKULA, BALI

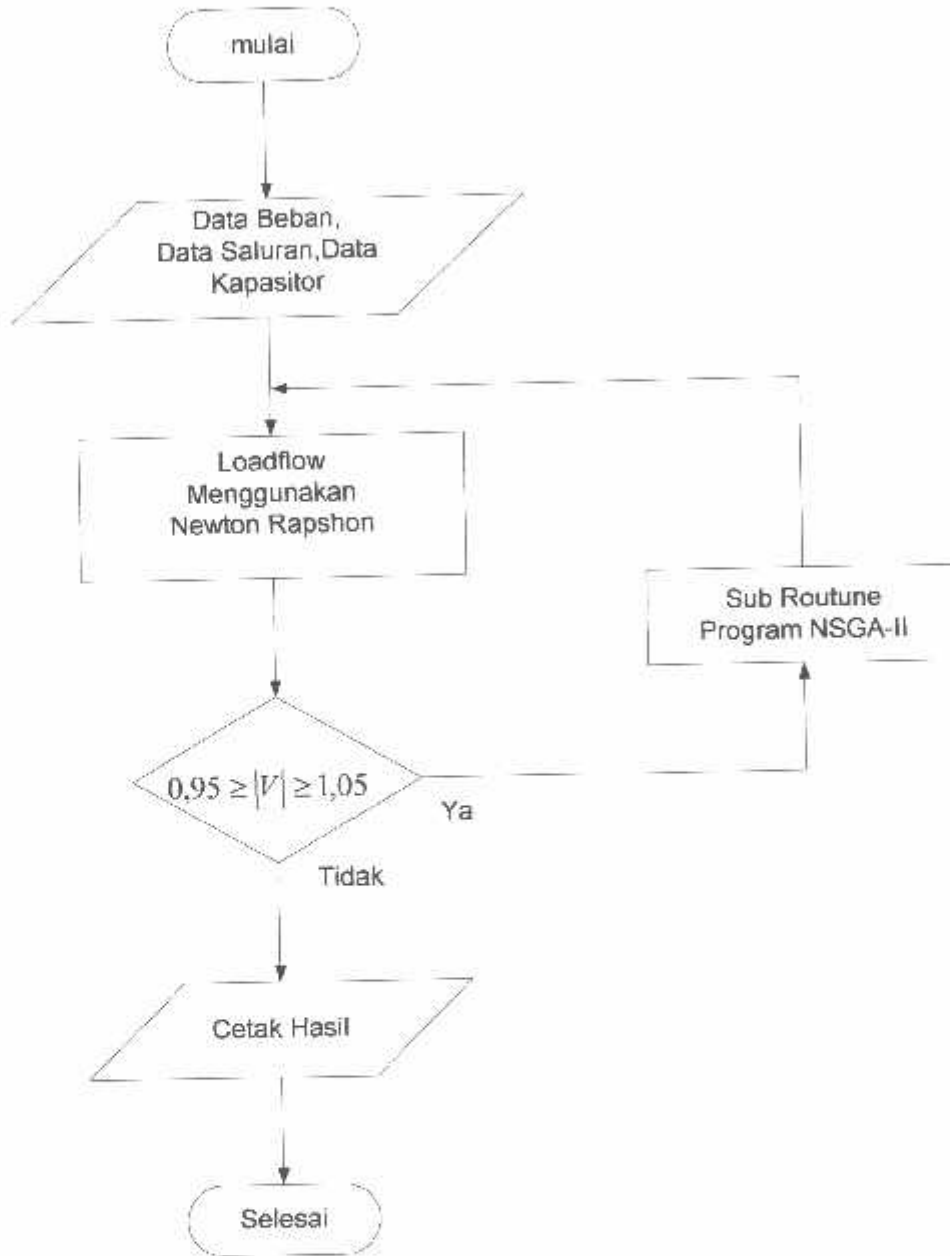
4.1. Analisis Penempatan Kapasitor Dengan Nondominated Sorting Genetic Algorithm-II (NSGA-II)

Pilihan suatu metode solusi untuk aplikasi praktis sering sulit, pilihan itu memerlukan analisis yang cermat atas kelebihan-kelebihan dan kekurangan-kekurangan pada banyak metode yang tersedia. Untuk solusi dari permasalahan-permasalahan diatas maka digunakan suatu alternatif metode Nondominated Sorting Genetic Algorithm-II (NSGA-II) menyediakan cara yang lebih baik dalam menganalisis penempatan kapasitor. Analisisnya menggunakan metode pencarian stokastik dan metode optimalisasi, diharapkan agar perbaikan profil tegangan mencapai nilai maksimum.

Dalam menganalisa penempatan kapasitor, dengan menggunakan program komputer. Pada prinsipnya penggunaan program komputer dapat menganalisa sebuah jaringan distribusi radial dengan jumlah cabang atau jumlah bus yang tidak terbatas tergantung dari tersedianya jumlah memori yang tersedia pada komputer yang digunakan.

4.1.1 Flowchart Algoritma Program

Flowchart Algoritma Pemecahan Masalah Penempatan Kapasitor Pada Jaringan Distribusi Primer 20 kV GI Pamaron Penyulang Tejakula.

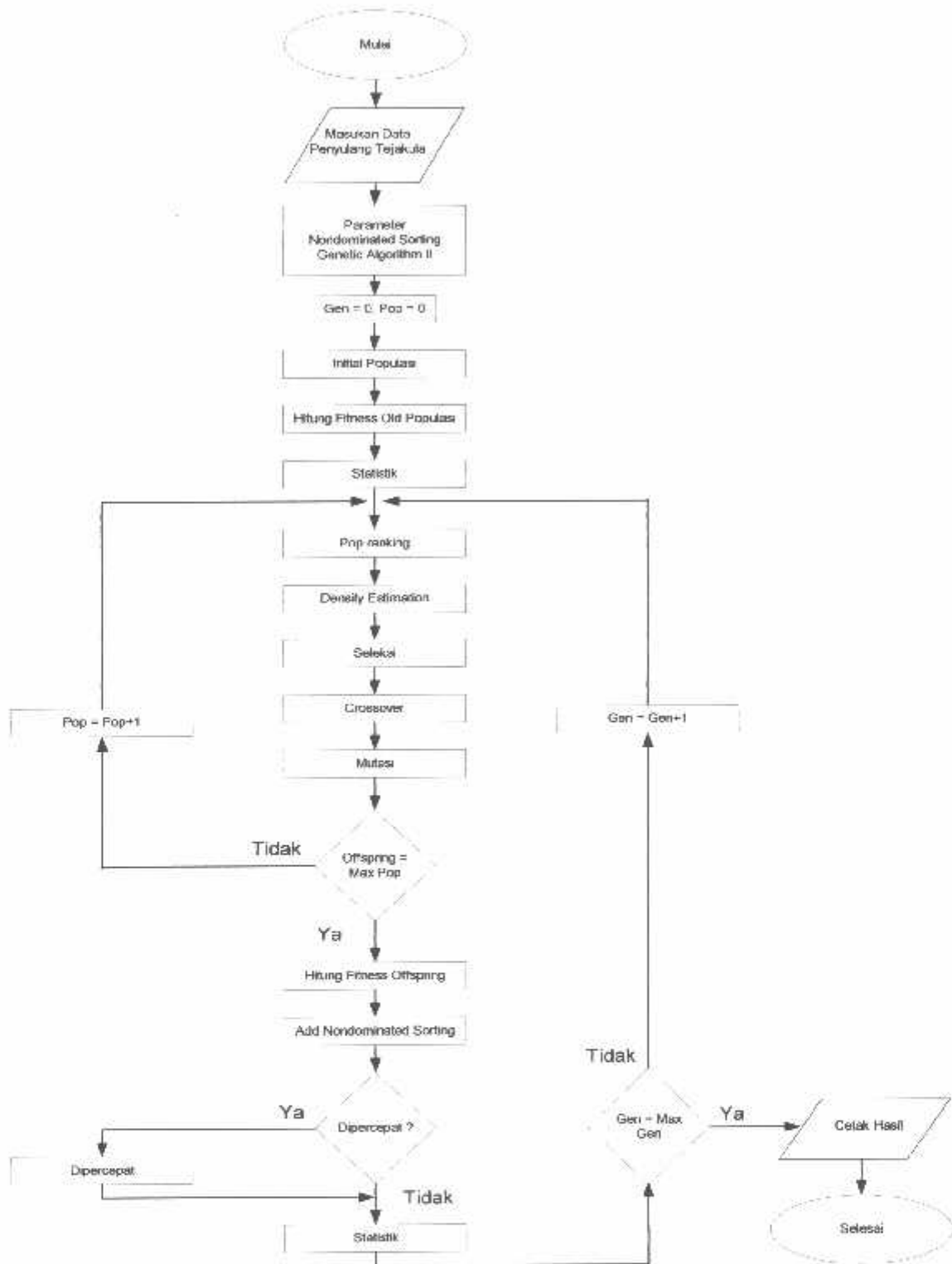


4.1.2. Algoritma Penyelesaian Masalah

Algoritma Program Penyelesaian Masalah pemasangan kapasitor dengan metode *Nondominated Sorting Genetic Algorithm II* (NSGA-II).

1. Memasukkan inputan data beban, yang meliputi tegangan, sudut fasa tegangan (δ), daya aktif (P), daya reaktif (Q) dan data impedansi saluran.
 2. Menentukan parameter inputan *Nondominated Sorting Genetic Algorithm II* (NSGA-II) yang meliputi jumlah populasi, maksimum generasi, nilai kemungkinan crossover (Pc), nilai kemungkinan mutasi (Pm) dan panjang kromosom tiap-tiap individu.
 3. Generasi = 0, Populasi = 0
 4. Melakukan proses *fitness* dari kromosom tiap-tiap individu.
 5. Melakukan proses statistik.
 6. Melakukan proses Populasi Ranking.
 7. Melakukan proses *Density Estimation*.
 8. Melakukan proses seleksi.
 9. Melakukan proses crossover.
 10. Melakukan proses mutasi.
 11. Proses no : 6,7,8,9,10 diulang sampai jumlah keturunan (*offspring*) sama dengan jumlah populasi.
 12. Melakukan proses *Additional Nondominated Sorting*.
 13. Apakah proses perlu dipercepat ? Jika “Ya” langkah 14, Jika “Tidak” langkah 15.
 14. Langkah dipercepat.
 15. Melakukan proses statistik.
 16. Apakah jumlah generasi sudah terpenuhi (max.gen).
 17. Jika “Tidak” maka $gen = gen+1$, kembali ke langkah 6.
 18. Jika “Ya” perhitungan selesai.
-

4.1.3. Flowchart Nondominated Sorting Genetic Algorithm II (NSGA-II)



4.2. Uji Validasi

Uji validasi dilakukan untuk mengetahui apakah program tersebut layak atau tidak. Salah satu cara untuk mengujinya menggunakan program tersebut untuk mengolah data dari jurnal, dan hasilnya dibandingkan dengan hasil jurnal. Apabila hasilnya serupa atau mendekati dengan jurnal, maka program dinyatakan layak digunakan.

Karena jurnal yang digunakan dalam penyusunan skripsi ini (Kalyanmoy Deb, "**A Fast And Elitist Multiobjective Genetic Algorithm: NSGA-II**", IEEE Transactions on Evolutionary computation, Vol.6, No. 2, April 2002) tidak menampilkan / mencantumkan data parameter penyulang secara lengkap dan data pembebanan tiap node, maka hal tersebut menjadi kendala dalam melakukan uji validasi program Nondominated Sorting Genetic Algorithm-II (NSGA-II) yang dibuat. Namun sebagai upaya lain untuk melakukan uji validasi program adalah dengan memasukkan data dari (Ji-Pyng Chiou, Chung-Fu Chang, and Ching-Tzong Su, "**Ant Direction Hybrid Differential Evolution for Solving Large Capacitor Placement Problems**," *IEEE Transactions on Power System*, Vol. 19, No. 4, November 2004). Berdasarkan data pada jurnal tersebut, maka didapatkan hasil sebagai berikut

The screenshot shows a window titled "Tampilan Data" with four tabs: "Parameter Operasional", "Parameter Sakuran", "Parameter Pembebanan", and "Atribut Kapasitor". The "Parameter Sakuran" tab is active, displaying a table with the following data:

No	Dari	Ke	R (pu)	X (pu)	Lc (pu)	Ti
1	1	2	0.014	0.0051	0	0
2	2	3	0.7463	1.295	0	0
3	3	4	0.0384	0.6384	0	0
4	4	5	1.5831	1.7276	0	0
5	5	6	0.5053	0.7886	0	0
6	6	7	2.0552	1.164	0	0
7	7	8	4.7953	2.716	0	0
8	8	9	5.3434	3.0264	0	0

At the bottom of the window, there are buttons for "Click for next" and "Close".

Gambar 4.2
Tampilan Data IEEE 9 Bus

The screenshot shows a window titled "Result Viewer" with several tabs: "LoadFlow Awal", "Aliran Daya", "LoadFlow Akhir", "Aliran Daya", "NSGA II Analis", "NSGA Decision", and "Grafik Daya". The "Aliran Daya" tab is active, displaying a table with the following data:

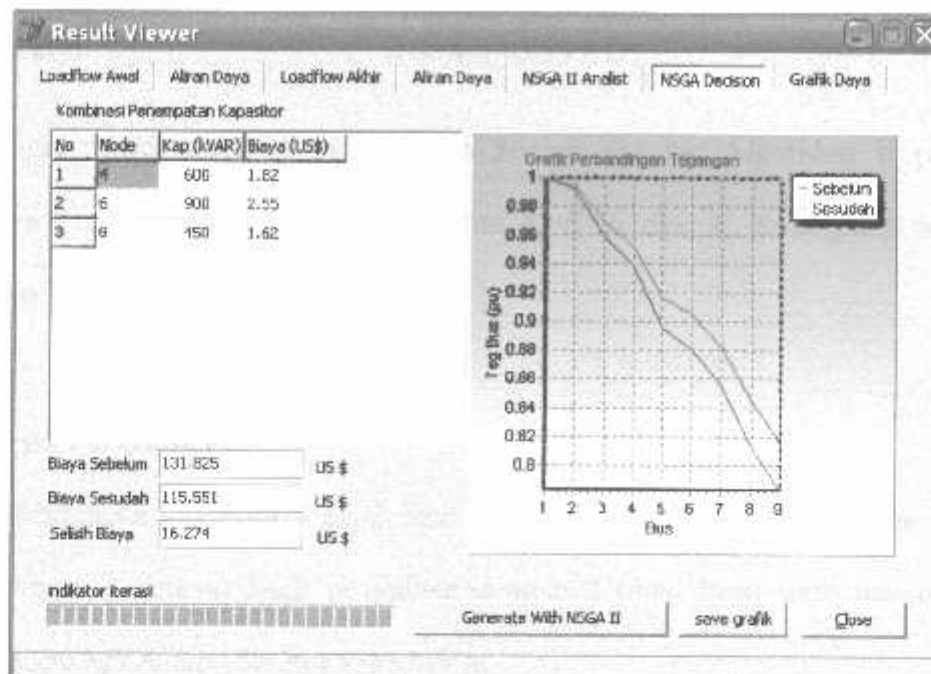
No	Dari	Ke	P (kWatt)	Q (KVAR)	Arus re (A)	AL
1	1	2	11596.642	4986.417	0.000	
2	2	1	-11591.065	-4745.365	0.000	
3	2	3	10611.065	4405.365	0.000	
4	3	2	-10360.897	-4001.435	0.000	
5	3	4	8570.897	3555.435	0.000	
6	4	3	-8407.493	-3413.089	0.000	
7	4	5	6809.493	1573.089	0.000	
8	5	4	-6534.245	-1333.303	0.000	
9	5	6	4924.245	733.303	0.000	
10	6	5	-4854.217	-672.302	0.000	
11	6	7	4074.217	562.302	0.000	
12	7	6	-3962.240	-498.882	0.000	

To the right of the table is a "Summary Result" box containing the following information:

- Total Pembangkitan: $13436.642 + 25446.417$ KVA
- Total Pembebanan: $12368.000 + 4186.000$ KVA
- Total Rugi-Rugi Sakuran: $783.700 + 1588.412$ KVA
- Iterasi: 4
- Waktu Komputasi (jam:menit:detik:ms): 0:0:0:0
- Absolut Total Rugi-Rugi Daya AMF: KW

At the bottom of the window, there is an "Indikator Iterasi" field, a "save grafik" button, and a "Close" button.

Gambar 4-3
Tampilan Aliran daya Newton-Raphson



Gambar 4-4
Hasil Perhitungan Metode Nondominated Sorting Genetic Algorithm II (NSGA-II)

Sebelum dilakukan penempatan kapasitor maka diperoleh rugi daya aktif dari data IEEE 9 Bus sebesar 783.700 kW dan setelah dipasang kapasitor rugi daya aktif sebesar 675.380 kW sedangkan dari data Ji-Pyng Chiou, Chung-Fu Chang, and Ching-Tzong Su, “Ant Direction Hybrid Differential Evolution for Solving Large Capacitor Placement Problems,” *IEEE Transactions on Power System*, Vol. 19, No. 4, November 2004. diperoleh rugi daya aktif sebelum dipasang kapasitor sebesar 783.778 kW dan setelah dipasang kapasitor daya aktif sebesar 675.3746 kW. Maka persentase error rata – rata setelah dijalankan dengan program Delphi 7.0 didapatkan % error rata – rata sebesar :

$$\% \text{ error} = \frac{\text{hasil jurnal} - \text{hasil pengujian}}{\text{hasil jurnal}} \times 100\%$$

Tabel 4-1
Data Penyulang Sistem 20 kV GI Pamaran

Jenis Konduktor	Penampang Nominal (mm ²)	Resistansi Konduktor (Ω/km)	GMR (mm)	Kuat Hantar (A)
AAAC	150	0,2126	5,2365	425
AAAC	120	0.2688	4.6837	365

Adapun data gardu dan pembebanan penyulang Tejakula seperti pada tabel 4-2.

Tabel 4-2
Data Gardu Dan Pembebanan Penyulang Tejakula

No. Node	Chassis	Kapasitas (kVA)	Beban Gardu	Beban Aktif (kW)	Beban Reaktif (kVAr)
1	-	0	0	0.0000	0.0000
2	-	0	0	0.0000	0.0000
3	TJ11	100	65	55.9000	33.1691
4	-	0	0	0.0000	0.0000
5	TJ1	250	83	178.4500	105.8860

Tabel 4-2 selengkapnya dapat dilihat pada lampiran

4.3.2. Data Pembebanan

Data pembebanan diperoleh dengan mengambil data dari masing-masing trafo distribusi, dimana besarnya beban pada masing-masing fasa diasumsikan seimbang. Jika besarnya pembebanan adalah nol, maka pada node tidak terdapat trafo distribusi tetapi hanya merupakan simpul. Pada tahap ini rugi-rugi yang terjadi pada trafo distribusi diabaikan, adapun data-data pembebanan seperti pada tabel 4-3.

Tabel 4-3
Data Pembebanan Penyulang Tejakula

No Node	Terdistribusi P (W)		Pembebanan		
	P (W)	Q (VAr)	P (W)	Q (VAr)	SPF
0	1.0000	0.0000	0.0000	0.0000	Slack
1	1.0000	0.0000	0.0000	0.0000	Load
2	1.0000	0.0000	55.9000	33.1691	Load
3	1.0000	0.0000	0.0000	0.0000	Load
4	1.0000	0.0000	178.4500	105.8860	Load
5	1.0000	0.0000	64.6720	38.3741	Load

Tabel 4-3 Selengkapnya dapat dilihat pada lampiran

4.3.3. Data kapasitor

Berdasarkan referensi, tabel 4-4 berikut menunjukkan 6 macam ukuran kapasitas kapasitor beserta biaya ekivalen instalasi.

Tabel 4-4
Harga Kapasitor

No	Kapasitas (kVAR)	Harga (Rp)
1	150	1,498
2	300	1,604
3	450	1,620
4	600	1,823
5	900	2,550
6	1200	2,955

Sumber : A. Mendes, P.M. Franca, C. Pissarra and C.Cavellucci, "Capacitor Placement in Large-Sized Radial Distribution Networks", *IEE Proc-Gen. Transm. Distrib.*, Vol. 152, No. 4, July 2005

4.4. Analisa Perhitungan

Perhitungan penempatan kapasitor diawali dengan melakukan studi aliran daya dengan metode Newton Raphson. Studi aliran daya dilakukan untuk mengetahui harga tegangan dan sudut fasa di tiap-tiap bus dan arus yang mengalir pada saluran antar bus.

Setelah studi aliran daya dilakukan, barulah dilakukan perhitungan aliran daya dan rugi-rugi daya pada saluran.

Untuk memudahkan perhitungan dan analisa pada sistem tenaga, biasanya dipakai harga-harga dalam persatuan. Harga persatuan adalah harga yang sebenarnya dibagi dengan harga dasar, harga dasar ini dapat dipilih sembarang. Harga yang dipilih pada studi ini adalah 20 kV dan 100 kVA sebagai harga tegangan dasar dan daya dasar. Mengingat bahwa pada jaringan tidak dilakukan pengukuran faktor daya maka pada perhitungan ini diambil harga faktor daya diasumsikan 0,86.

Perhitungan diawali dengan menampilkan single line diagram dari penyulang yang mewakili keadaan sistem yang sesungguhnya. Dari gambar 4-1 diperlihatkan single line diagram dari penyulang tejakula. Pada penyulang ini jumlah bus dan saluran masing-masing adalah :

- ❖ Slack bus = 1
- ❖ Load bus = 70
- ❖ Jumlah saluran = 70

Dengan mengacu pada gambar single line diagram penyulang tejakula, maka dapat diperoleh data impedansi saluran seperti tabel 4-5 dibawah ini. Setelah ini baru kita lakukan perhitungan aliran daya dengan menggunakan metode Newton Raphson,

Tabel 4-5
Data Impedansi Saluran Penyulang Tejakula

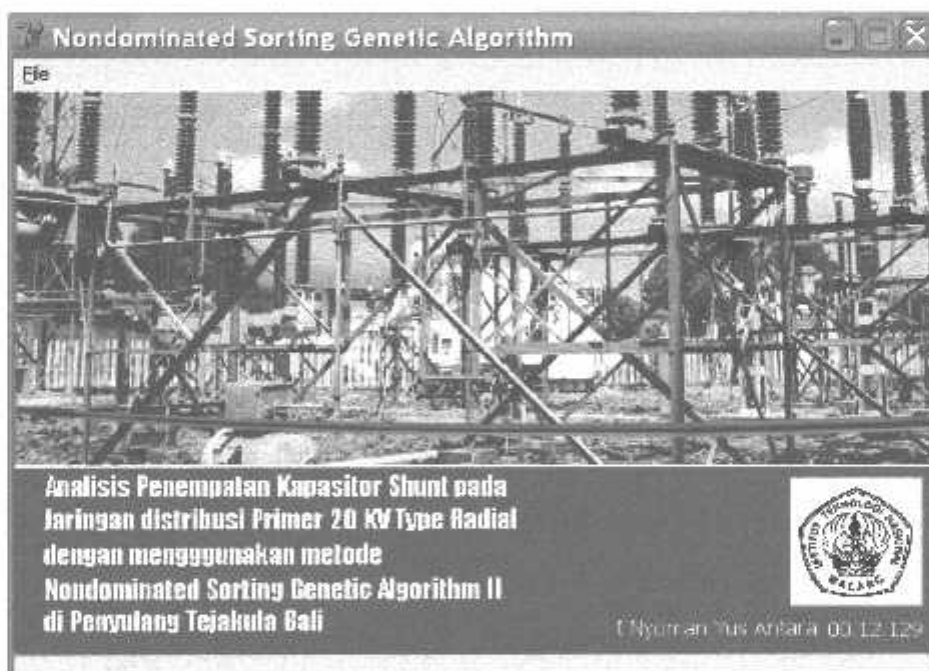
No. Saluran	Tipe Saluran	No. Saluran	Panjang (km)	Impedansi Saluran	
				R (Ohm)	X (jika)
1	1	2	8,776.50	1.8974793	2.90063325
2	2	3	650.1111	0.140554022	0.214861722
3	2	4	7,765.22	1.67883971	2.566403904
4	4	5	2,311.51	0.499747635	0.76395279
5	4	20	794.5802	0.171788249	0.262608772

Tabel 4-5 selengkapnya dapat dilihat pada lampiran

4.5. Prosedur Pelaksanaan Program Perhitungan

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

1. Tampilan Menu Utama



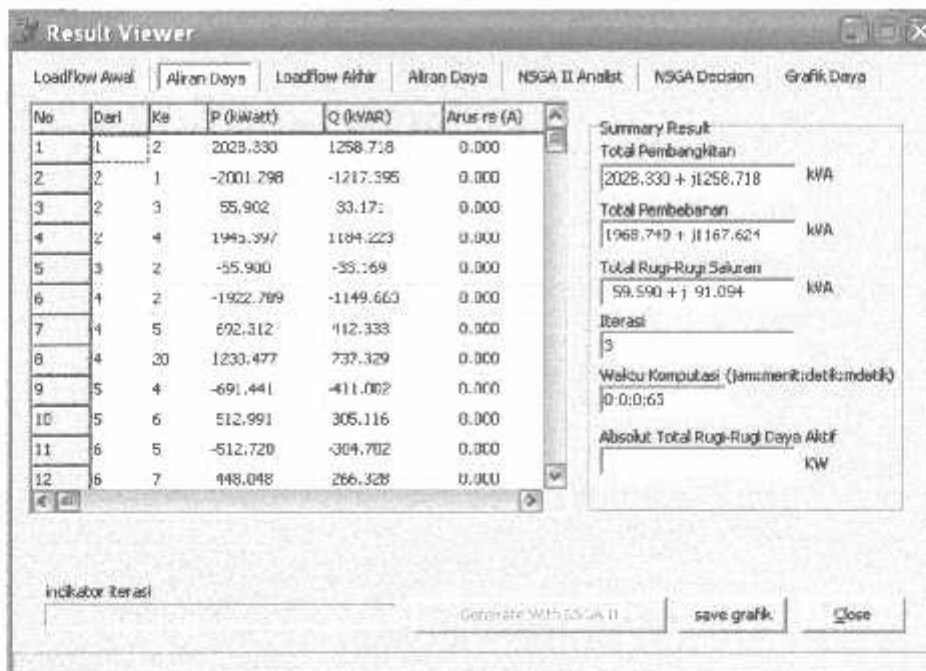
Gambar 4-5
Tampilan Menu Utama

2. Tampilan Data

No	Dari	Ke	R (pu)	X (pu)	Lo (pu)	Tr
1	1	2	1.9974793	2.90063325	0	0
2	2	3	0.140554022	0.214961722	0	0
3	2	4	1.67983971	2.566403904	0	0
4	4	5	0.499747635	0.76395279	0	0
5	4	20	0.171798249	0.262608772	0	0
6	5	6	0.282669756	0.432110797	0	0
7	6	7	0.074962145	0.114592519	0	0
8	7	8	0.640301657	0.978814512	0	0
9	7	12	0.702770111	1.074308671	0	0
10	8	9	0.121813486	0.186213493	0	0
11	9	10	0.135868886	0.207639665	0	0
12	10	11	0.21863959	0.334229346	0	0
13	12	13	0.118690063	0.181438789	0	0
14	13	14	0.075523857	0.116960271	0	0
15	14	15	0.19521392	0.298419059	0	0

Gambar 4-6
Tampilan Data

3. Kemudian Klik next untuk menjalankan program, sehingga didapatkan tampilan sebagai berikut :



Gambar 4-7
Tampilan aliran daya Newton-Raphson

Setelah dilakukan analisa aliran daya dengan menggunakan metode Newton Raphson maka diperoleh besarnya tegangan tiap-tiap bus dan rugi-rugi saluran seperti pada tabel 4-6 sampai 4-7.

Tabel 4-6
Load Flow Penyulang Tejakula Sebelum Pemasangan Kapasitor

Bus	Abs V (pu)	Angle (MVA)	P (kW)	Q (KVAR)	P load (kW)	Q load (KVAR)
1	1.00000	0.00000	2028.330	1228.718	0.000	0.000
2	0.98129	-0.51018	0.000	0.000	0.000	0.000
3	0.98125	-0.51127	0.000	0.000	55.900	33.169
4	0.96526	-0.96455	0.000	0.000	0.000	0.000
5	0.96354	-1.01427	0.000	0.000	178.450	105.886

Tabel 4-6 selengkapnya dapat dilihat pada lampiran

Tabel 4-7
Aliran Daya Penyulang Tejakula Sebelum Pemasangan Kapasitor

No	From	To	Real (MW)	Reactive (MVar)	Loss (MW)	Loss (MVar)
1	1	2	2028.330	1258.718	0.000	0.000
2	2	1	-2001.298	-1217.395	0.000	0.000
3	2	3	55.901	33.171	0.000	0.000
4	2	4	1945.397	1184.223	0.000	0.000
5	3	2	-55.900	-33.169	0.000	0.000

Tabel 4-7 selengkapnya dapat dilihat pada lampiran

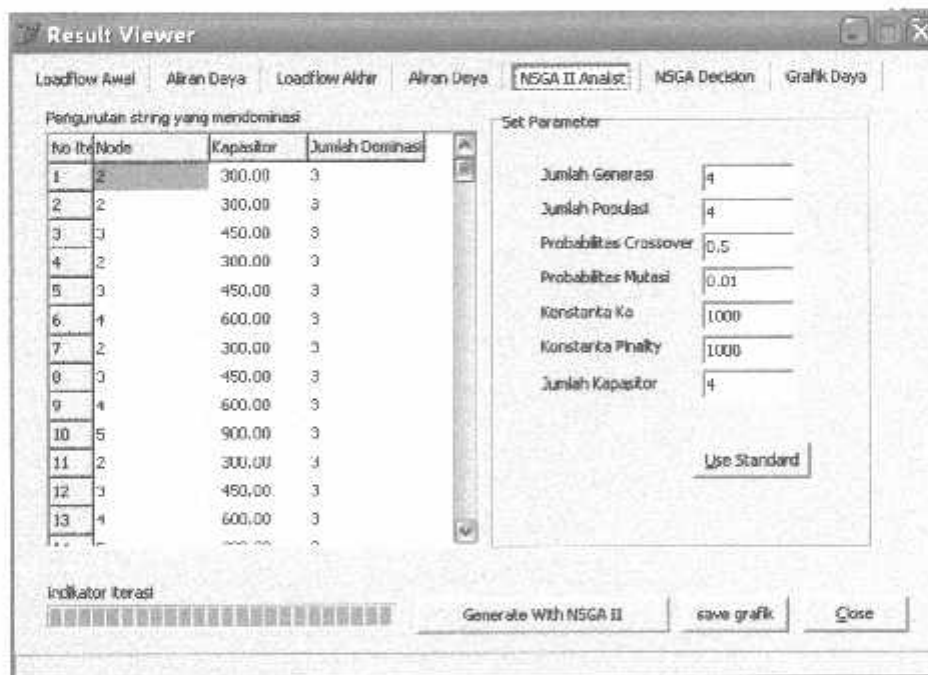
Banyaknya Saluran = 70

Banyaknya Node = 71

Banyaknya Iterasi = 3

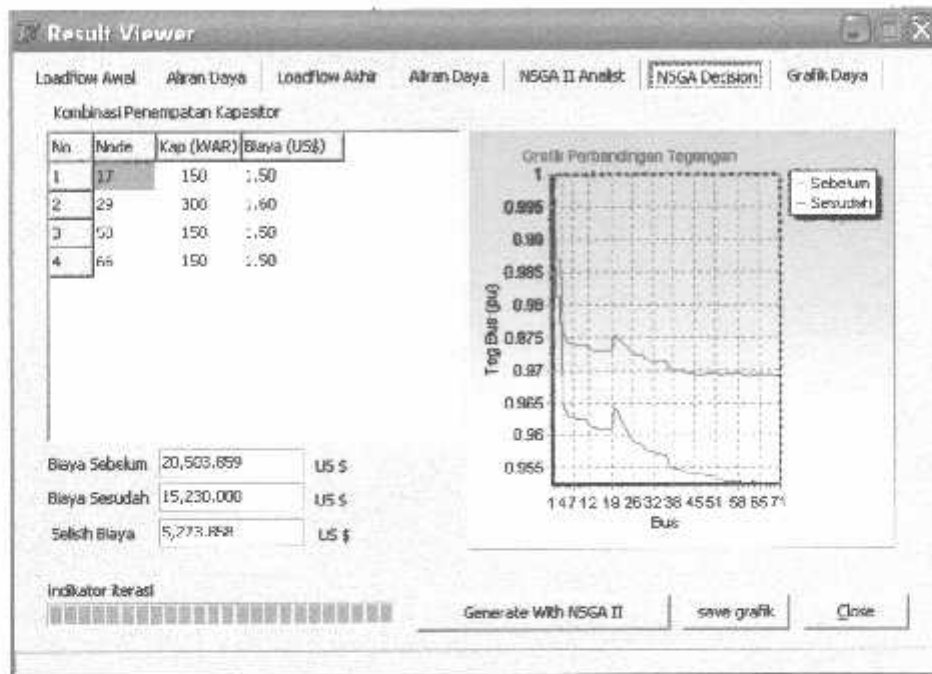
Rugi Total Saluran = $59.590 + j 91.094$ kVA

4. Kemudian Klik NSGA-II Analist untuk menjalankan program, sehingga didapatkan tampilan sebagai berikut :



Gambar 4-8
Tampilan Analisis Metode NSGA-II

5. Kemudian Klik NSGA Decision untuk menjalankan program, sehingga didapatkan tampilan sebagai berikut :



Gambar 4-9
Hasil Perhitungan Metode Nondominated Sorting Genetic Algorithm II (NSGA-II)

Setelah dilakukan analisa dengan menggunakan Metode Nondominated Sorting Genetic Algorithm II (NSGA-II), akan diperoleh hasil yaitu jumlah, letak, kapasitas dan total biaya tahunan setelah penempatan kapasitor tersebut. Dari gambar 4-9 dengan menggunakan Metode (NSGA-II) dapat dilihat bahwa kapasitor dipasang pada node 17, 29, 58, dan 66 dengan kapasitas masing-masing yaitu sebesar 150 kVAR, 300 kVAR 150 kVAR dan 150 kVAR. Jadi didapat biaya sebelum pemasangan kapasitor yaitu 20,503.859 US\$, sedangkan setelah pemasangan kapasitor adalah 15,230.000 US\$ jadi PLN mendapat keuntungan sebesar 5,273.858 US\$.

Setelah dilakukan pemasangan kapasitor maka diperoleh profil tegangan, aliran daya tiap-tiap saluran dan rugi-rugi saluran. Dapat dilihat pada tabel 4-8 dan tabel 4-9.

Tabel 4-8
Load Flow Penyulang Tejakula Sesudah Pemasangan Kapasitor

Bus	APR V (p.u)	Sal V (p.u)	Pesan (kW)	Qesan (kVAR)	P load (kW)	Q load (kVAR)
1	1.00000	0.00000	2013.460	485.986	0.000	0.000
2	0.98700	-0.71377	0.000	0.000	0.000	0.000
3	0.98696	-0.71485	0.000	0.000	55.900	33.169
4	0.97608	-1.34772	0.000	0.000	0.000	0.000
5	0.97468	-1.40761	0.000	0.000	178.450	105.886

Tabel 4-8 selengkapnya dapat dilihat pada lampiran

Tabel 4-9
Aliran Daya Penyulang Tejakula Sesudah Pemasangan Kapasitor

No	From	To	P (kW)	Q (kVAR)	Angle (°)	Angle (°)
1	1	2	2013.460	485.986	0.000	0.000
2	2	1	-1993.109	-454.876	0.000	0.000
3	2	3	55.902	33.171	0.000	0.000
4	2	4	1937.207	421.704	0.000	0.000
5	3	2	-55.900	-33.169	0.000	0.000

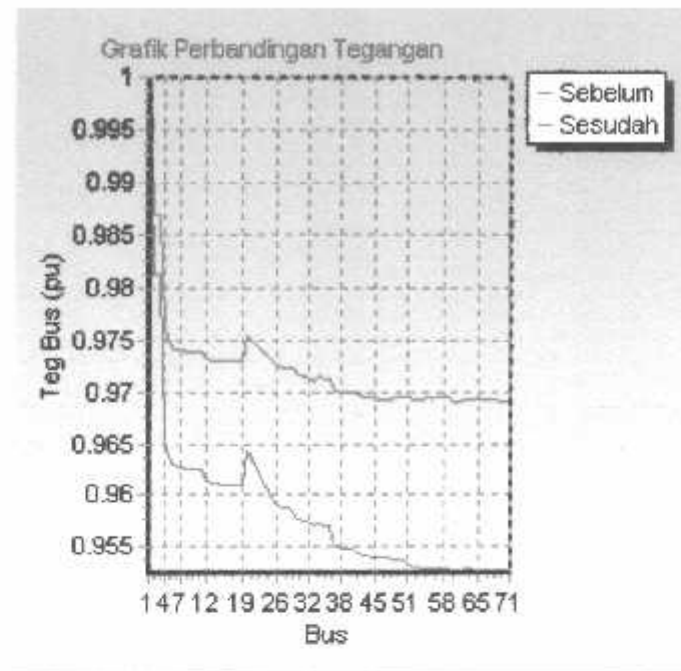
Tabel 4-9 selengkapnya dapat dilihat pada lampiran

Banyaknya Saluran = 70

Banyaknya Node = 71

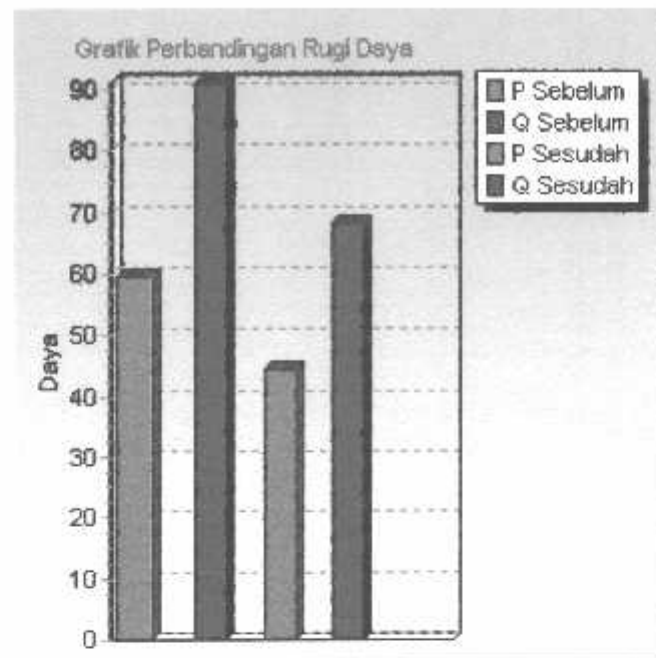
Banyaknya Iterasi = 3

Rugi Total Saluran = $44.720 + j 68.362$ kVA

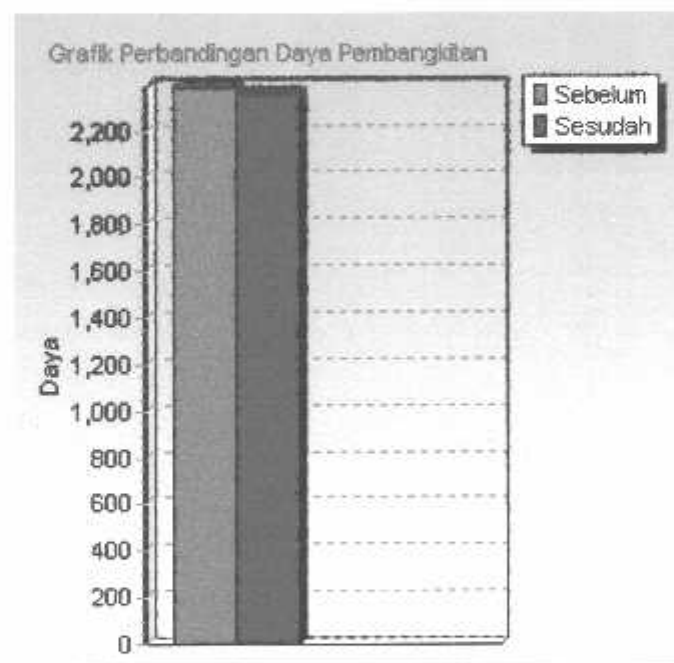


Grafik 4-1
Tegangan Tiap-Tiap Node Terhadap Asumsi Tegangan Awal Sebelum dan Sesudah Pemasangan Kapasitor

Dari grafik diatas dapat dilihat bahwa perubahan tegangan dari asumsi tegangan awal tiap-tiap node berada dibawah batas-batas yang diinginkan, dengan tegangan yang paling rendah terjadi pada node 71 yaitu sebesar 0,95246 pu atau 19,0492 kV. dan setelah kompensasi tegangan pada node tersebut mengalami kenaikan sebesar 0.96910 pu atau 19,382 kV.



Grafik 4-2
Rugi-Rugi Daya Saluran Sebelum dan Sesudah Pemasangan Kapasitor



Grafik 4-3
Perbandingan Daya Pembangkitan Sebelum dan Sesudah Pemasangan Kapasitor

Dari grafik 4-2 terlihat bahwa terjadi pengurangan rugi daya saluran setelah pemasangan kapasitor, sedangkan pada grafik 4-3 terlihat bahwa terjadi pengurangan daya pembangkitan setelah pemasangan kapasitor.

Tabel 4-10
Perbandingan Kondisi Sebelum dan Sesudah Pemasangan Kapasitor

Uraian	Sebelum Pemasangan Kapasitor	Sesudah Pemasangan Kapasitor
Rugi Daya Total (kVA)	59.590 + j 91.09	44.720 + j 68.362
kVAR (Bus)		150 (17)
		300 (29)
		150 (58)
		150 (66)
Total Biaya (\$/tahun)	20,503.859	15,230.000

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah dilakukan analisa perhitungan penempatan kapasitor pada jaringan distribusi 20 kV type radial, maka dapat diambil beberapa kesimpulan sebagai berikut :

1. Berdasarkan hasil perhitungan dengan metode Nondominated Sorting Genetic Algorithm II (NSGA-II) letak kapasitor yang optimal terletak pada node 17, 29, 58 dan 66. Dimana pada node 17 kapasitas yang terpasang sebesar 150 kVAR , pada node 29 kapasitas yang terpasang sebesar 300 kVAR sedangkan node 58 kapasitas yang terpasang sebesar 150 kVAR dan node 66 kapasitas yang terpasang 150 kVAR.
2. Dari hasil perhitungan load flow pada penyulang Tejakula tegangan paling rendah terjadi pada node ke-71, yang besarnya 0.95246 pu atau 19,0492 kV. Setelah kompensasi profil tegangan pada semua node penyulang Tejakula bertambah baik, untuk node ke-71 menjadi 0.96910 pu atau 19,382 kV.
3. Rugi-rugi daya mengalami penurunan yaitu sebelum kompensasi sebesar $59.590 + j91.094$ kVA dan setelah kompensasi sebesar $44.720 + j68.362$ kVA.
4. Dari hasil perhitungan program maka nilai total biaya sebelum pemasangan kapasitor sebesar 20,503.859 US\$ sedangkan setelah

pemasangan kapasitor sebesar 15,230.000 US\$. Jadi penghematan yang didapatkan oleh PT. PLN (Persero) cabang Singaraja selaku distributor listrik selama 1 (tahun) sebesar 5,273.858 US\$.

5.2 SARAN

Penentuan letak penempatan kapasitor pada jaringan distribusi primer dengan menggunakan metode Nondominated Sorting Genetic Algorithm II (NSGA-II) yang telah dibahas perlu dikembangkan dan diaplikasikan untuk menganalisa saluran – saluran yang telah ada maupun untuk perencanaan perluasan jaringan.

DAFTAR PUSTAKA

- [1] William D. Stevenson, JR, "**Analisis sistem tenaga listrik**", Edisi Keempat, Erlangga, Jakarta 1996.
 - [2] Ji-Pyng Chiou, Chung-Fu Chang, and Ching-Tzong Su, "**Ant Direction Hybrid Differential Evolution for Solving Large Capacitor Placement Problems**," *IEEE Transactions on Power System*, Vol. 19, No. 4, November 2004.
 - [3] M. E. Baran and F. F. Wu, "**Optimal Sizing of Capacitors Placed on A Radial Distribution System**", *IEEE Trans. Power Delivery*, Vol 4, pp. 735-743, Jan. 1989.
 - [4] Ramon A. Gallego, Alcir Jose Monticelli, and Ruben Romero, "**Optimal Capacitor Placement in Radial Distribution Networks**", *IEEE Trans. On Power System*, Vol. 16, No. 4, November 2001.
 - [5] A. Mendes, P.M. Franca, C. Pissarra and C.Cavellucci, "**Capacitor Placement in Large-Sized Radial Distribution Networks**", *IEE Proc-Gen. Transm. Distrib.*, Vol. 152, No. 4, July 2005.
 - [6] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, And T. Meyarivan "**A Fast And Elitist Multiobjective Genetic Algorithm : NSGA II** ", *IEEE Transactions On Evolutionary Computation* , Vol. 6, No.2, April 2004.
 - [7] Turan Gonen, "**Electric Power Distribution System Engineering**", University of Missouri at Columbia.
 - [8] Djiteng Maesudi "**Operasi Sistem Tenaga Listrik**", Balai Penerbit dan Humas ISTN 1990.
 - [9] Hasan Basri, "**Sistem Distribusi Tenaga Listrik**".
-

LAMPIRAN



**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

1. Nama : INYOMAN YUS ANTARA
2. Nim : 00.12.129
3. Jurusan : Teknik Elektro (S-1)
4. Konsentrasi : Teknik Energi Listrik
5. Judul Skripsi : ANALISIS PENEMPATAN KAPASITOR SHUNT PADA JARINGAN DISTRIBUSI PRIMER 20 KV TYPE RADIAL DENGAN MENGGUNAKAN METODE NONDOMINATED SORTING GENETIC ALGORITHM II (NSGA-II) DI PENYULANG TEJAKULA, BALI

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

- Hari : Kamis
Tanggal : 21 September 2006
Dengan Nilai : 79 (B+)

Panitia Ujian Skripsi

Ketua

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

Sekretaris

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

Anggota Penguji

Penguji Pertama

Ir. Yusuf Ismail Nakhoda, MT
Nip. Y. 1018800189

Penguji-Kedua

Ir. H. Soemarwanto
Nip. 130873485



LEMBAR BIMBINGAN SKRIPSI

1. Nama : I Nyoman Yus Antara
2. Nim : 00.12.129
3. Jurusan : Teknik Elektro
4. Konsentrasi : Teknik Energi Listrik (S-1)
5. Judul Skripsi : ANALISIS PENEMPATAN KAPASITOR SHUNT PADA JARINGAN DISTRIBUSI PRIMER 20 KV TYPE RADIAL DENGAN MENGGUNAKAN METODE NONDOMINATED SORTING GENETIC ALGORITHM II (NSGA-II) DI PENYULANG TEJAKULA, BALI

6. Tanggal Mengajukan Skripsi : 02 Juni 2006
7. Selesai Penulisan Skripsi : 18 September 2006
8. Dosen pembimbing : Ir. Teguh Herbasuki, MT.
9. Telah di evaluasi dengan nilai : 85 (Delapan Puluh Lima)

Malang, Oktober 2006

Mengetahui

Ketua Jurusan Teknik Elektro S-1

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

Dosen Pembimbing

Ir. Teguh Herbasuki, MT
NIP. P.103 890 0209



PERSETUJUAN PERBAIKAN SKRIPSI

Dari hasil ujian Skripsi Jurusan Teknik Elektro Jenjang Strata Satu (S-1) yang di selenggarakan pada:

Hari : Kamis

Tanggal : 21 September 2006

Telah dilakukan perbaikan Skripsi oleh:

1. Nama : I NYOMAN YUS ANTARA
2. NIM : 00.12.129
3. Jurusan : TEKNIK ELEKTRO
4. Konsentrasi : TEKNIK ENERGI LISTRIK S-1
5. Judul Skripsi : ANALISIS PENEMPATAN KAPASITOR SHUNT PADA JARINGAN DISTRIBUSI PRIMER 20 KV TYPE RADIAL DENGAN MENGGUNAKAN METODE NONDOMINATED SORTING GENETIC ALGORITHM II (NSGA-II) DI PENYULANG TEJAKULA, BALI

Perbaikan Meliputi:

No	Materi Perbaikan	Paraf
1	Betulkan penulisan sub-bab pada Bab I	
2	Tambahkan gambar rangkaian C pada jaringan 3 fasa	
3	Cek kembali hubungan C, sudut 0, V untuk C yang dipakai	

Malang, Oktober 2006

Anggota Penguji

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

Ir. H. Soemarwanto
Nip. 130 873 485

Dosen Pembimbing

Ir. Teguh Herbasuki, MT
Nip.P. 103 890 0209



Formulir Perbaikan Ujian Skripsi

Dalam pelaksanaan Ujian Skripsi Janjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : L. Nyoman Yur Antara
NIM : 00.12.123
Perbaikan meliputi :

1. Betulkan penulisan sub-bab pada Bab I.
2. Tambahkan gambar rangkaian C pada jaringan 3 fase.
3. Cek kembali hubungan C_2 antara θ , V antara C yg. di pilih.

Malang, 21-09-2006


(Soemarwanto)



PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

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

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

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

Malang, 6 Juni 2006

Nomor : ITN-1409/I.TA/2/'06
Lampiran : satu lembar
Perihal : **BIMBINGAN SKRIPSI**

Kepada : Yth. Sdr. Ir. **TEGUH HERBASUKI, MT**
Dosen Institut Teknologi Nasional
di -
Malang

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

Nama : I NYOMAN YUS ANTARA
Nim : 0012129
Fakultas : Teknologi Industri
Jurusan : Teknik Elektro
Konsentrasi : T. Energi Listrik (S-1)

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

02 Juni 2006 s/d 02 Nov. 2006

Adapun tugas tersebut merupakan salah satu syarat untuk memperoleh gelar Sarjana Teknik, Jurusan Teknik Elektro.

Demikian atas perhatian serta kerjasamanya kami ucapkan terima kasih



Ir. F. Yudi Hartapraptono, MT
Nip. Y. 1039500274

Tindakan :

1. Mahasiswa yang bersangkutan
2. Arsip.

Form. S-4a



Malang, 02 Mei 2006

Lampiran : Satu Lembar
Perihal : **Kesediaan Sebagai
Dosen Pembimbing**

Kepada : Yth. Bapak/Ibu. Ir. Teguh Herbasuki, MT
Dosen Jurusan Elektro/ T.Energi Listrik
Institut Teknologi Nasional Malang
di -
Malang

Yang bertandatangan di bawah ini :

Nama : I NYOMAN YUS ANTARA
Nim : 00.12.129
Semester : XII
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 :

**ANALISIS PENEMPATAN KAPASITOR SHUNT PADA
JARINGAN DISTRIBUSI PRIMER 20 KV TYPE RADIAL
DENGAN MENGGUNAKAN METODE *NONDOMINATED
SORTING GENETIC ALGORITHM II (NSGA-II)* DI PENYULANG
TEJAKULA, 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 terima kasih.

Mengetahui
Ketua Jurusan Teknik Elektro


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

Malang, 02 Mei 2006
Pemohon



I Nyoman Yus Antara
NIM. 00.12.129

Form. S-3a



**PERNYATAAN KESEDIAAN SEBAGAI
DOSEN PEMBIMBING SKRIPSI**

Sesuai dengan Permohonan mahasiswa :

Nama : I NYOMAN YUS ANTARA
Nim : 00.12.129
Semester : XII
Jurusan : TEKNIK ELEKTRO
Konsentrasi : ENERGI LISTRIK S-1

Dengan ini menyatakan **bersedia / tidak bersedia** *) menjadi Dosen Pembimbing Utama / pendamping *) untuk penyusunan Skripsi Mahasiswa tersebut dengan judul :

**ANALISIS PENEMPATAN KAPASITOR SHUNT PADA
JARINGAN DISTRIBUSI PRIMER 20 KV TYPE RADIAL
DENGAN MENGGUNAKAN METODE *NONDOMINATED
SORTING GENETIC ALGORITHM II (NSGA-II)* DI PENYULANG
TEJAKULA, BALI**

Demikian pernyataan ini kami buat untuk dapat dipergunakan sebagaimana mestinya

Malang, Mei 2006
Yang Membuat Pernyataan,

Ir. Teguh Herbasuki, MT
NIP. P.103 890 0209

Catatan :

1. Setelah disetujui agar formulir ini diserahkan mahasiswa/i yang bersangkutan kepada jurusan untuk diproses lebih lanjut.
2. *) Coret yang tidak perlu



FORMULIR BIMBINGAN SKRIPSI

Nama : I NYOMAN YUS ANTARA
Nim : 00.12.129
Masa Bimbingan : 2 Juni 2006 s/d 2 November 2006
Judul Skripsi : Analisis Penempatan Kapasitor Shunt Pada Jaringan Distribusi Primer 20 Kv Type Radial Dengan Menggunakan Metode *Nondominated Sorting Genetic Algorithm II (NSGA-II)* Di Penyulang Tejakula, Bali

No	Tanggal	Uraian	Paraf Pembimbing
1.	20/06'06	Konsultasi masalah isi & judul	[Signature]
2.	28/06'06	Konsultasi bab I	[Signature]
3.	07/07'06	Revisi bab I	[Signature]
4.	21/07'06	Konsultasi bab II	[Signature]
5.	05/08'06	Konsultasi bab III	[Signature]
6.	10/08'06	Revisi bab III	[Signature]
7.	14/08'06	Konsultasi bab IV & V	[Signature]
8.	19/08'06	Revisi bab IV & V	[Signature]
9.	22/08'06	Dibuatkan makalah seminar	[Signature]
10.	18/09'06	Acc ujian	[Signature]

Malang,
Dosen Pembimbing,

[Signature]
(Ir. Teguh Herdasuki, MT)

Form.S-4b



LEMBAR PENGAJUAN JUDUL SKRIPSI
JURUSAN TEKNIK ELEKTRO S-1

Konsentrasi : Teknik Energi Listrik/Teknik Elektronika^{*)}

1	Nama Mahasiswa : I NYOMAN YUS ANTARA	Nim : 00.12.129
2	Waktu pengajuan	Tanggal : Bulan : Maret Tahun : 2006
3	Spesifikasi judul (berilah tanda silang)	
	<input checked="" type="checkbox"/> a. Sistem Tenaga Elektrik <input type="checkbox"/> b. Energi & Konversi Energi <input type="checkbox"/> c. Tegangan Tinggi & Pengukuran <input type="checkbox"/> d. Sistem Kendali Industri	<input type="checkbox"/> e. Elektronika & Komponen <input type="checkbox"/> f. Elektronika Digital & Komputer <input type="checkbox"/> g. Elektronika Komunikasi <input type="checkbox"/> h. lainnya
4	Konsultasikan judul sesuai materi bidang ilmu kepada Dosen ^{*)} : <i>Ir. Teguh Herbaniki, MT</i>	Mengetahui, Ketua Jurusan. Ir. F. Mudi Limpraptono, MT Nip. Y. 1039500274
5	Judul yang diajukan mahasiswa :	<i>Analisis Pemampatan Kapasitor Shunt Pada Jaringan Distribusi Tenaga 20 kV Tipe Radial Dengan Menggunakan Metode Non-Optimised Sorting Genetic Algorithm II Di Pemukiman Tejakula, Bali</i>
6	Perubahan Judul yang disetujui Dosen sesuai materi bidang ilmu
Catatan :		
.....		
7	Persetujuan Judul Skripsi yang dikonsultasikan kepada Dosen materi bidang ilmu	Disetujui, 13-03-2006 Dosen

Perhatian :

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

Form.S-2



BERITA ACARA SEMINAR SKRIPSI JURUSAN TEKNIK ELEKTRO S-1

Konsentrasi : Teknik Energi Listrik/Teknik Elektronika*)

1.	Nama Mahasiswa: I Nyoman Yub Antara	Nim: 00.12.129
2.	Keterangan	Tanggal
	Pelaksanaan	11.9.06
		Waktu
		Tempat
	Ruang:	
	Spesifikasi Judul (berilah tanda silang)**)	
3.	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	Analisis Penempatan Kapasitor Shunt Pada Jaringan Distribusi Primer 20 kV Type Radial Dengan Menggunakan Metode Nonlaminated Solving Genetic Algorithm-II (Nega-II), Di Pengulang Tegakula, Bali
5.	Perubahan Judul yang diusulkan oleh Kelompok Dosen Keahlian/Pengamat	
6.	Keputusan: Dari hasil penilaian sejumlahorang dosen keahlian danorang dosen pengamat sesuai format penilaian terlampir, peserta seminar tersebut diatas (1) dengan judul skripsi (4) dinyatakan LULUS/TIDAK LULUS *) dengan nilai Kumulatif:(angka) atau(huruf)	
	Persetujuan Judul Skripsi	
7.	Disetujui, Dosen Keahlian I Ir. H. S. S.	Disetujui, Dosen Keahlian II
	Disetujui, Dosen Pengamat I Ir. H. Choiri	Disetujui, Dosen Pengamat II
	Mengetahui, Ketua Jurusan. Ir. F. Yudi Limpraptono, MT NIP. P. 1039500274	Disetujui, Dosen Pembimbing Ir. Teguh Herbaniki, MT

Perhatian:

1. Keterangan: *) Coret yang tidak perlu
**) dilingkari a, b, c, atau g sesuai bidang keahlian



**PT PLN (PERSERO)
DISTRIBUSI BALI**

Jalan Letda Tantular No. 1 Denpasar 80234

Telepon : (0361) 221960 - 221963

(0361) 221965 - 221968

Facsimile : (0361) 227101



Nomor : /080/BISDM/2006
Srt Sdr : ITN-1623/III.TA/2/06
Lampiran : -
Perihal : Survey

Denpasar, Juli 2006

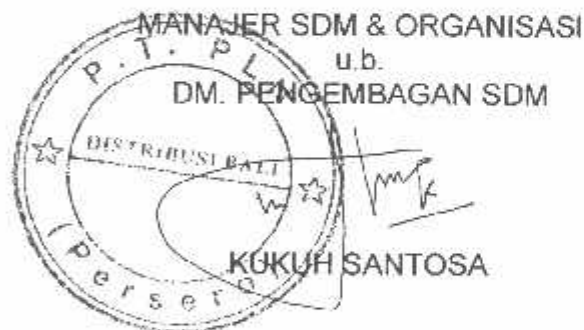
Kepada Yth :



Institut Teknologi Nasional Malang
Fakultas Teknologi Industri
Jl. Bendungan Sigura-gura No.2
Malang 65145

Sehubungan dengan surat Saudara No. ITN-1623/III.TA/2/06 perihal Mohon Ijin Mengadakan Penelitian/Survey, dengan ini diberitahukan bahwa pada prinsipnya kami dapat mengijinkan mahasiswa Saudara yaitu : sdr. I Nyoman Yus A, NIM : 00.12.129 jurusan Teknik Elektro untuk mengadakan penelitian/Survey di PT. PLN (Persero) Area Jaringan Bali Utara, Jln. Udayana No.27 Singaraja.

Demikian atas perhatiannya diucapkan terima kasih.



Tembusan :
Manajer AJ Bali Utara

LAMPIRAN LAMPIRAN

- 1. DATA GARDU DAN PEMBEBANAN**
 - 2. DATA PEMBEBANAN PENYULANG TEJAKULA**
 - 3. DATA IMPEDANSI SALURAN PENYULANG TEJAKULA**
 - 4. TABEL LOAD FLOW PENYULANG TEJAKULA SEBELUM PEMASANGAN KAPASITOR**
 - 5. TABEL ALIRAN DAYA PENYULANG TEJAKULA SEBELUM PEMASANGAN KAPASITOR**
 - 6. TABEL LOAD FLOW PENYULANG TEJAKULA SESUDAH PEMASANGAN KAPASITOR**
 - 7. TABEL ALIRAN DAYA PENYULANG TEJAKULA SESUDAH PEMASANGAN KAPASITOR**
 - 8. LISTING PROGRAM**
-



PT. PLN (PERSERO)
DISTRIBUSI BALI
AJ BALI UTARA

TABEL 4-2
DATA GARDU DAN PEMBEBANAN PENYULANG TEJAKULA

			GARDU		
1	-	0	0	0.0000	0.0000
2	-	0	0	0.0000	0.0000
3	TJ11	100	65	55.9000	33.1691
4	-	0	0	0.0000	0.0000
5	TJ1	250	83	178.4500	105.8860
6	TJ23	160	47	64.6720	38.3741
7	-	0	0	0.0000	0.0000
8	TJ25	50	43	18.4900	10.9713
9	TJ18	50	44	18.9200	11.2263
10	TJ19	50	26	11.1800	6.6338
11	TJ21	50	20	8.6000	5.1029
12	TJ2	250	70	150.5000	89.3015
13	TJ4	160	39	53.6640	31.8423
14	-	0	0	0.0000	0.0000
15	TJ03	50	49	21.0700	12.5022
16	TJ6	100	54	46.4400	27.5559
17	TJ8	100	53	45.5800	27.0456
18	TJ9	100	46	39.5600	23.4735
19	TJ15	100	39	33.5400	19.9015
20	TJ5	100	38	32.6800	19.3912
21	TJ13	50	49	21.0700	12.5022
22	TJ17	50	55	23.6500	14.0331
23	KB9	100	39	33.5400	19.9015
24	KB8	50	67	28.8100	17.0949
25	KB24	160	31	42.6560	25.3106

26	KB3	100	46	39.5600	23.4735
27	-	0	0	0.0000	0.0000
28	KB7	50	5	2.1500	1.2757
29	KB46	25	52	11.1800	6.6338
30	KB2	160	66	90.8160	53.8870
31	-	0	0	0.0000	0.0000
32	KB34	50	37	15.9100	9.4404
33	KB35	250	62	133.3000	79.0956
34	-	0	0	0.0000	0.0000
35	KB1	160	61	83.9360	49.8047
36	KB11	100	64	55.0400	32.6588
37	KB37	50	44	18.9200	11.2265
38	KB22	16	34	4.6784	2.7760
39	-	0	0	0.0000	0.0000
40	KB4	50	62	26.8600	15.8191
41	KB21	50	27	11.6100	6.8890
42	-	0	0	0.0000	0.0000
43	KB6	160	66	90.8160	53.8870
44	KB17	50	42	18.0600	10.7162
45	-	0	0	0.0000	0.0000
46	KB31	50	43	18.4900	10.9713
47	KB13	100	57	49.0200	29.0868
48	KB12	100	64	55.0400	32.6588
49	-	0	0	0.0000	0.0000
50	KB32	50	36	15.4800	9.1853
51	KB39	50	18	7.7400	4.5926
52	-	0	0	0.0000	0.0000
53	KB14	100	39	33.5400	19.9015
54	-	0	0	0.0000	0.0000
55	-	0	0	0.0000	0.0000
56	KB25	50	52	22.3600	13.2676
57	-	0	0	0.0000	0.0000
58	KB26	50	29	12.4700	7.3993
59	KB27	50	21	9.0300	5.3581

60	KB37	50	44	18.9200	11.2265
61	KB33	50	25	10.7500	6.3787
62	-	0	0	0.0000	0.0000
63	KB41	50	45	19.3500	11.4816
64	KB18	100	57	49.0200	29.0868
65	-	0	0	0.0000	0.0000
66	KB12	100	52	44.7200	26.5353
67	KB43	50	29	12.4700	7.3993
68	-	0	0	0.0000	0.0000
69	KB36	50	43	18.4900	10.9713
70	KB28	50	37	15.9100	9.4404
71	KB29	100	17	23.3920	13.8800



PT. PLN (PERSERO)
DISTRIBUSI BALI
AU BALI UTARA

TABEL 4-3
DATA PEMBEBANAN PENYULANG TEJAKULA

1	1	0	-	-	0	0	Slack	
2	1	0	-	-	0	0	Load	
3	1	0	-	-	55.9	33.1691	Load	
4	1	0	-	-	0	0	Load	
5	1	0	-	-	178.45	105.886	Load	
6	1	0	-	-	64.672	38.3741	Load	
7	1	0	-	-	0	0	Load	
8	1	0	-	-	18.49	10.9713	Load	
9	1	0	-	-	18.92	11.2265	Load	
10	1	0	-	-	11.18	6.6338	Load	
11	1	0	-	-	8.6	5.1029	Load	
12	1	0	-	-	150.5	89.3015	Load	
13	1	0	-	-	53.664	31.8423	Load	
14	1	0	-	-	0	0	Load	
15	1	0	-	-	21.07	12.5022	Load	
16	1	0	-	-	46.44	27.5559	Load	
17	1	0	-	-	45.58	27.0456	Load	
18	1	0	-	-	39.56	23.4735	Load	
19	1	0	-	-	33.54	19.9015	Load	
20	1	0	-	-	32.68	19.3912	Load	
21	1	0	-	-	21.07	12.5022	Load	
22	1	0	-	-	23.65	14.0331	Load	
23	1	0	-	-	33.54	19.9015	Load	
24	1	0	-	-	28.81	17.0949	Load	
25	1	0	-	-	42.656	25.3106	Load	
26	1	0	-	-	39.56	23.4735	Load	

27	1	0	-	-	0	0	Load
28	1	0	-	-	2.15	1.2757	Load
29	1	0	-	-	11.18	6.6338	Load
30	1	0	-	-	90.816	53.887	Load
31	1	0	-	-	0	0	Load
32	1	0	-	-	15.91	9.4404	Load
33	1	0	-	-	133.3	79.0956	Load
34	1	0	-	-	0	0	Load
35	1	0	-	-	83.936	49.8047	Load
36	1	0	-	-	55.04	32.6588	Load
37	1	0	-	-	18.92	11.2265	Load
38	1	0	-	-	4.8784	2.776	Load
39	1	0	-	-	0	0	Load
40	1	0	-	-	26.66	15.8191	Load
41	1	0	-	-	11.61	6.889	Load
42	1	0	-	-	0	0	Load
43	1	0	-	-	90.816	53.887	Load
44	1	0	-	-	18.06	10.7162	Load
45	1	0	-	-	0	0	Load
48	1	0	-	-	18.49	10.9713	Load
47	1	0	-	-	49.02	29.0868	Load
48	1	0	-	-	55.04	32.6588	Load
49	1	0	-	-	0	0	Load
50	1	0	-	-	15.48	9.1853	Load
51	1	0	-	-	7.74	4.5926	Load
52	1	0	-	-	0	0	Load
53	1	0	-	-	33.54	19.9015	Load
54	1	0	-	-	0	0	Load
55	1	0	-	-	0	0	Load
56	1	0	-	-	23.3	13.2676	Load
57	1	0	-	-	0	0	Load
58	1	0	-	-	12.47	7.3993	Load
59	1	0	-	-	9.03	5.3581	Load
60	1	0	-	-	18.92	11.2265	Load

61	1	0	-	-	10.75	6.3787	Load
62	1	0	-	-	0	0	Load
63	1	0	-	-	19.35	11.4816	Load
64	1	0	-	-	49.02	29.0868	Load
65	1	0	-	-	0	0	Load
66	1	0	-	-	44.72	26.5353	Load
67	1	0	-	-	12.47	7.3993	Load
68	1	0	-	-	0	0	Load
69	1	0	-	-	18.49	10.9713	Load
70	1	0	-	-	15.91	9.4404	Load
71	1	0	-	-	23.392	13.88	Load



PT. PLN (PERSERO)
DISTRIBUSI BALI
AJ BALI UTARA

TABEL 4-5
DATA IMPEDANSI SALURAN PENYULANG TEJAKULA

1	1	2	8,778.50	1.8974793	2.90063325	0.0003332059	0.000507611
2	2	3	650.1111	0.140554022	0.214861722	0.000024597	0.000037601
3	2	4	7,785.22	1.67883971	2.568403904	0.000293797	0.000449121
4	4	5	2,311.51	0.499747635	0.76395279	0.000087456	0.000133692
5	4	20	794.5802	0.171788249	0.262608772	0.000030063	0.000045957
6	5	6	1,307.45	0.282669756	0.432110797	0.000049467	0.000075615
7	6	7	346.7259	0.074962145	0.114592919	0.000013118	0.000020054
8	7	8	2,961.62	0.640301657	0.978814512	0.000112053	0.0001771293
9	7	12	3,250.56	0.702770111	1.074308611	0.000122965	0.000188004
10	8	9	563.4296	0.121813486	0.186213493	0.000021317	0.000032587
11	9	10	628.4407	0.135868888	0.207699665	0.000023777	0.000036347
12	10	11	1,011.28	0.21863959	0.334229346	0.000038262	0.000058490
13	12	13	548.9827	0.118690063	0.181438788	0.000020771	0.000031752
14	13	14	353.9494	0.076523857	0.116980271	0.000013392	0.000020472
15	14	15	902.9321	0.19521392	0.298419059	0.000034162	0.000052223
16	14	16	216.7037	0.046851341	0.071620574	0.000008199	0.000012534
17	16	17	556.2062	0.120251775	0.18382614	0.000021044	0.000032170
18	17	18	563.4296	0.121813486	0.186213493	0.000021317	0.000032587
19	18	19	635.6842	0.1374306	0.210087017	0.000024050	0.000036765
20	20	21	563.4296	0.121813486	0.186213493	0.000021317	0.000032587
21	21	22	585.1	0.12649862	0.19337555	0.000022137	0.000033841
22	22	23	1,112.41	0.240503549	0.36765228	0.000042088	0.000064339
23	23	24	671.7815	0.145239156	0.222202378	0.000025417	0.000038854
24	24	25	686.2284	0.148362579	0.226798485	0.000025963	0.000039690
25	25	26	577.8765	0.124936909	0.190888198	0.000021864	0.000033423
26	26	27	361.1728	0.078085568	0.119367623	0.000013665	0.000020889

27	27	28	975.1667	0.210831033	0.322292583	0.000036895	0.000056401
28	27	29	288.9383	0.0622468454	0.095494099	0.000010932	0.000016711
29	29	30	722.3457	0.156171136	0.238735247	0.000027330	0.000041779
30	30	31	252.821	0.054659898	0.083657336	0.000009565	0.000014623
31	31	32	939.0494	0.203022477	0.310356821	0.000035529	0.000054312
32	31	34	144.4691	0.031234227	0.047747049	0.000005466	0.000005346
33	32	33	1.459.14	0.315466694	0.482245119	0.000055206	0.000084393
34	34	35	1.300.2222	0.281108044	0.429723444	0.000049194	0.000075202
35	34	37	3.142.20	0.679344441	1.038498324	0.000118885	0.000181737
36	35	36	1.191.87	0.257682374	0.393913157	0.000045094	0.000068935
37	37	38	397.2901	0.085894125	0.131304386	0.000015031	0.000022978
38	38	39	144.4691	0.031234227	0.047747049	0.000005466	0.000008356
39	39	40	939.0494	0.203022477	0.310355821	0.000035529	0.000054312
40	39	41	216.7037	0.048861341	0.071620574	0.000008199	0.000012534
41	41	42	939.0494	0.203022477	0.310355821	0.000035529	0.000054312
42	42	43	660.1111	0.140554022	0.214861722	0.000024597	0.000037601
43	42	44	1.112.41	0.240503549	0.36765228	0.000042088	0.000064339
44	42	49	1.444.69	0.312342272	0.477470494	0.000054860	0.000083557
45	44	45	433.4074	0.093702681	0.143241148	0.000016398	0.000025067
46	45	46	794.5802	0.1717188249	0.262608772	0.000030063	0.000045957
47	45	47	613.9938	0.132745465	0.20292496	0.000023230	0.000035512
48	47	48	1.213.54	0.262367508	0.401075215	0.000045914	0.000070188
49	49	50	650.1111	0.140554022	0.214861722	0.000024597	0.000037601
50	49	51	1.227.99	0.265490931	0.40584992	0.000046461	0.000071024
51	51	52	505.642	0.109319795	0.167114673	0.000019131	0.000029245
52	52	53	866.8148	0.187405365	0.286482296	0.000032796	0.000050134
53	52	54	505.642	0.109319795	0.167114673	0.000019131	0.000029245
54	54	55	281.7148	0.060906743	0.093106746	0.000010659	0.000016294
55	54	62	744.016	0.16085627	0.245897304	0.000028150	0.000043032
56	55	56	939.0494	0.203022477	0.310355821	0.000035529	0.000054312
57	55	57	650.1111	0.140554022	0.214861722	0.000021597	0.000037601
58	57	58	1.083.52	0.234256704	0.35810287	0.000040995	0.000062668
59	57	59	216.7037	0.0468851341	0.071620574	0.000008199	0.000012534
60	69	60	498.4185	0.107758084	0.16472732	0.000018858	0.000028827

61	60	61	505.642	0.109319795	0.187114673	0.000019131	0.000029245
62	62	63	693.4519	0.14992429	0.229185837	0.000026237	0.000040108
63	62	64	489.5247	0.101511238	0.15517791	0.000017764	0.000027156
64	64	65	1.011.28	0.21863959	0.334229346	0.000038262	0.000058490
65	65	66	902.9321	0.19521392	0.298419059	0.000034162	0.000052223
66	65	68	577.8785	0.124936909	0.190988198	0.000021864	0.000033243
67	66	67	998.837	0.215516167	0.329454641	0.000037715	0.000057655
68	68	69	325.0556	0.070277011	0.107430861	0.000012298	0.000018800
69	68	70	852.3679	0.18428194	0.281707591	0.000032249	0.000049299
70	70	71	686.2284	0.148362579	0.226796485	0.000025963	0.00003969

Tabel 4-6
Load Flow Penyulang
Tejakula Sebelum Pemasangan Kapasitor

Bus	V (pu)	Angle (deg)	P (MW)	Q (MVar)	P (MW)	Q (MVar)
1	1.00000	0.00000	2028.330	1228.718	0.000	0.000
2	0.98129	-0.51018	0.000	0.000	0.000	0.000
3	0.98125	-0.51127	0.000	0.000	55.900	33.169
4	0.96526	-0.96455	0.000	0.000	0.000	0.000
5	0.96354	-1.01427	0.000	0.000	178.450	105.886
6	0.96283	-1.03518	0.000	0.000	64.672	38.374
7	0.96266	-1.04003	0.000	0.000	0.000	0.000
8	0.96248	-1.04532	0.000	0.000	18.490	10.971
9	0.96246	-1.04600	0.000	0.000	18.920	11.226
10	0.96244	-1.04639	0.000	0.000	11.180	6.634
11	0.96243	-1.04666	0.000	0.000	8.600	5.103
12	0.96130	-1.07974	0.000	0.000	150.500	89.302
13	0.96116	-1.08387	0.000	0.000	53.664	31.842
14	0.96109	-1.08594	0.000	0.000	0.000	0.000
15	0.96107	-1.08653	0.000	0.000	21.070	12.502
16	0.96105	-1.08706	0.000	0.000	46.440	27.556
17	0.96098	-1.08913	0.000	0.000	45.580	27.046
18	0.96093	-1.09042	0.000	0.000	39.560	23.474
19	0.96091	-1.09109	0.000	0.000	33.540	19.901
20	0.96421	0.99479	0.000	0.000	32.680	19.319
21	0.96348	-1.01569	0.000	0.000	21.070	12.502
22	0.96275	-1.03705	0.000	0.000	23.650	14.033
23	0.96173	-1.07693	0.000	0.000	33.540	19.901
24	0.96056	-1.10035	0.000	0.000	28.810	17.095
25	0.95976	-1.12370	0.000	0.000	42.656	25.311
26	0.95911	-1.14262	0.000	0.000	39.560	23.474
27	0.95872	-1.15401	0.000	0.000	0.000	0.000
28	0.95872	-1.15407	0.000	0.000	2.150	1.276
29	0.95841	-1.16310	0.000	0.000	11.180	6.634
30	0.95764	-1.18561	0.000	0.000	90.816	53.887
31	0.95740	-1.19277	0.000	0.000	0.000	0.000
32	0.95725	-1.19720	0.000	0.000	15.910	9.440
33	0.95704	-1.20335	0.000	0.000	133.300	79.096
34	0.95728	-1.19619	0.000	0.000	0.000	0.000
35	0.95709	-1.20190	0.000	0.000	83.936	49.805
36	0.95701	-1.20397	0.000	0.000	55.040	32.659
37	0.95521	-1.25675	0.000	0.000	18.920	11.226
38	0.95496	-1.26419	0.000	0.000	4.678	2.776

39	0.95487	-1.26687	0.000	0.000	0.000	0.000
40	0.95484	-1.26767	0.000	0.000	26.660	15.819
41	0.95474	-1.27072	0.000	0.000	11.610	6.889
42	0.95419	-1.28704	0.000	0.000	0.000	0.000
43	0.95412	-1.28891	0.000	0.000	90.816	53.887
44	0.95402	-1.29201	0.000	0.000	18.060	10.716
45	0.95396	-1.29370	0.000	0.000	0.000	0.000
46	0.95394	-1.29417	0.000	0.000	18.490	10.971
47	0.95389	-1.29574	0.000	0.000	49.020	29.087
48	0.95382	-1.29786	0.000	0.000	55.040	32.659
49	0.95370	-1.30153	0.000	0.000	0.000	0.000
50	0.95368	-1.30185	0.000	0.000	15.480	9.185
51	0.95330	-1.31325	0.000	0.000	7.740	4.593
52	0.95314	-1.31796	0.000	0.000	0.000	0.000
53	0.95311	-1.31889	0.000	0.000	33.540	19.901
54	0.95300	-1.32213	0.000	0.000	0.000	0.000
55	0.95298	-1.32253	0.000	0.000	0.000	0.000
56	0.95296	-1.32325	0.000	0.000	23.300	13.268
57	0.95297	-1.32298	0.000	0.000	0.000	0.000
58	0.95295	-1.32341	0.000	0.000	12.470	7.399
59	0.95297	-1.32304	0.000	0.000	9.030	5.358
60	0.95248	-1.33734	0.000	0.000	18.920	11.226
61	0.95248	-1.33751	0.000	0.000	10.750	6.379
62	0.95283	-1.32718	0.000	0.000	0.000	0.000
63	0.95281	-1.32761	0.000	0.000	19.350	11.482
64	0.95273	-1.33008	0.000	0.000	49.020	29.087
65	0.95257	-1.33475	0.000	0.000	0.000	0.000
66	0.95251	-1.33640	0.000	0.000	44.720	26.535
67	0.95250	-1.33680	0.000	0.000	12.470	7.399
68	0.95252	-1.33637	0.000	0.000	0.000	0.000
69	0.95250	-1.33687	0.000	0.000	18.490	10.971
70	0.95248	-1.33744	0.000	0.000	15.910	9.440
71	0.95246	-1.33795	0.000	0.000	23.392	13.880

Tabel 4-7
Aliran Daya Penyulang
Tejakula Sebelum Pemasangan Kapasitor

No	Dari	Kepada	Daya (W)	Daya (VA)	Angkakan	Angkakan
1	1	2	2028.330	1258.718	0.000	0.000
2	2	1	-2001.298	-1217.395	0.000	0.000
3	2	3	55.901	33.171	0.000	0.000
4	2	4	1945.397	1184.223	0.000	0.000
5	3	2	-55.900	-33.169	0.000	0.000
6	4	2	-1922.789	-1149.663	0.000	0.000
7	4	5	692.312	412.333	0.000	0.000
8	4	20	1203.477	737.329	0.000	0.000
9	5	6	-691.441	-441.002	0.000	0.000
10	5	5	512.991	305.116	0.000	0.000
11	6	7	-512.720	-304.702	0.000	0.000
12	6	7	448.048	266.328	0.000	0.000
13	7	6	-447.993	-226.244	0.000	0.000
14	7	8	57.199	33.948	0.000	0.000
15	7	12	390.795	232.296	0.000	0.000
16	8	7	-57.199	-33.936	0.000	0.000
17	8	9	38.701	22.965	0.000	0.000
18	9	8	-38.700	-22.964	0.000	0.000
19	9	10	19.780	11.737	0.000	0.000
20	10	9	-19.780	-11.737	0.000	0.000
21	10	11	8.600	5.103	0.000	0.000
22	11	10	-8.600	-5.103	0.000	0.000
23	12	7	-390.403	-231.697	0.000	0.000
24	12	13	-239.903	142.396	0.000	0.000
25	13	12	-239.878	-142.357	0.000	0.000
26	13	14	186.214	110.515	0.000	0.000
27	14	13	-186.214	-110.500	0.000	0.000
28	14	15	21.070	12.502	0.000	0.000
29	14	16	165.134	97.998	0.000	0.000
30	15	14	-21.070	-12.502	0.000	0.000
31	16	14	-165.129	-97.990	0.000	0.000
32	16	17	118.689	70.453	0.000	0.000
33	17	16	-118.683	-70.425	0.000	0.000
34	17	18	73.103	43.380	0.000	0.000
35	18	17	-73.101	-43.376	0.000	0.000
36	18	19	33.541	19.902	0.000	0.000
37	19	18	-33.540	-19.901	0.000	0.000
38	20	4	-1229.528	-735.879	0.000	0.000
39	20	21	1196.848	716.488	0.000	0.000

40	21	20	-1196.211	-715.514	0.000	0.000
41	21	22	1175.141	703.012	0.000	0.000
42	22	21	-1174.502	-702.035	0.000	0.000
43	22	23	1150.852	688.002	0.000	0.000
44	23	22	-1149.686	-686.219	0.000	0.000
45	23	24	1116.146	666.318	0.000	0.000
46	24	23	-1115.482	-665.303	0.000	0.000
47	24	25	1086.672	648.208	0.000	0.000
48	25	24	-1086.028	-647.224	0.000	0.000
49	25	26	1043.372	621.913	0.000	0.000
50	26	25	-1024.872	-621.149	0.000	0.000
51	26	27	1003.312	-597.675	0.000	0.000
52	27	26	-1003.022	-597.233	0.000	0.000
53	27	28	2.150	-1.276	0.000	0.000
54	27	29	1000.872	595.957	0.000	0.000
55	28	27	-2.150	-1.276	0.000	0.000
56	29	27	-1000.642	-595.957	0.000	0.000
57	29	30	989.462	588.971	0.000	0.000
58	30	29	-988.898	-588.109	0.000	0.000
59	30	31	898.082	534.222	0.000	0.000
60	31	30	-897.920	-533.974	0.000	0.000
61	31	32	149.247	88.593	0.000	0.000
62	31	34	748.72	445.380	0.000	0.000
63	32	31	-149.231	-88.568	0.000	0.000
64	32	33	133.321	79.127	0.000	0.000
65	33	32	-133.300	-79.096	0.000	0.000
66	34	31	-748.608	-445.282	0.000	0.000
67	34	35	138.999	82.499	0.000	0.000
68	34	37	609.609	362.783	0.000	0.000
69	35	34	-138.979	-82.468	0.000	0.000
70	35	36	55.043	32.663	0.000	0.000
71	36	35	-55.040	-32.659	0.000	0.000
72	37	34	-608.676	-361.357	0.000	0.000
73	37	38	589.756	350.131	0.000	0.000
74	38	37	-589.645	-349.962	0.000	0.000
75	38	39	584.967	347.186	0.000	0.000
76	39	38	-584.927	-347.125	0.000	0.000
77	39	40	26.660	15.820	0.000	0.000
78	39	41	558.267	331.305	0.000	0.000
79	40	39	-26.660	-15.819	0.000	0.000
80	41	39	-558.213	-331.222	0.000	0.000
81	41	42	546.603	324.333	0.000	0.000
82	42	41	-546.378	-323.990	0.000	0.000
83	42	43	90.820	53.894	0.000	0.000

84	42	44	140.641	83.481	0.000	0.000
85	42	49	314.916	186.615	0.000	0.000
86	43	42	-90.816	-53.887	0.000	0.000
87	44	42	-140.624	-83.454	0.000	0.000
88	44	45	122.564	72.783	0.000	0.000
89	45	44	-122.559	-72.730	0.000	0.000
90	45	46	18.490	10.972	0.000	0.000
91	45	47	104.068	61.758	0.000	0.000
92	46	45	-18.490	-10.971	0.000	0.000
93	47	45	-104.063	-61.750	0.000	0.000
94	47	48	55.043	32.663	0.000	0.000
95	48	47	-55.040	-32.659	0.000	0.000
96	49	42	-314.801	-186.439	0.000	0.000
97	49	50	15.480	9.185	0.000	0.000
98	49	51	299.321	177.254	0.000	0.000
99	50	49	-15.480	-9.185	0.000	0.000
100	51	49	-299.233	-177.119	0.000	0.000
101	51	52	291.493	172.526	0.000	0.000
102	52	51	-291.458	-172.473	0.000	0.000
103	52	53	33.541	19.903	0.000	0.000
104	52	54	257.917	152.571	0.000	0.000
105	53	52	-33.540	-19.901	0.000	0.000
106	54	52	-257.890	-152.529	0.000	0.000
107	54	55	44.801	26.027	0.000	0.000
108	54	62	213.089	126.502	0.000	0.000
109	55	54	-44.801	-26.026	0.000	0.000
110	55	56	23.300	13.268	0.000	0.000
111	55	57	21.500	12.758	0.000	0.000
112	56	55	-23.300	-13.268	0.000	0.000
113	57	55	-21.500	-12.758	0.000	0.000
114	57	58	12.470	7.400	0.000	0.000
115	57	59	9.030	5.358	0.000	0.000
116	58	57	-12.470	-7.399	0.000	0.000
117	59	57	-9.030	-5.358	0.000	0.000
118	60	61	10.750	6.379	0.000	0.000
119	60	69	-29.670	-17.605	0.000	0.000
120	61	60	-10.750	-6.379	0.000	0.000
121	62	54	-213.062	-126.461	0.000	0.000
122	62	63	19.350	11.482	0.000	0.000
123	62	64	193.712	114.979	0.000	0.000
124	63	62	-19.350	-11.482	0.000	0.000
125	64	62	-193.697	-114.957	0.000	0.000
126	64	65	144.677	85.870	0.000	0.000
127	65	64	-144.660	-85.844	0.000	0.000

128	65	66	57.193	33.938	0.000	0.000
129	65	68	87.468	51.906	0.000	0.000
130	66	65	-57.190	-33.935	0.000	0.000
131	66	67	12.470	7.399	0.000	0.000
132	67	66	-12.470	-7.399	0.000	0.000
133	68	65	-87.464	-51.901	0.000	0.000
134	68	69	48.161	28.578	0.000	0.000
135	68	70	39.303	23.577	0.000	0.000
136	69	60	29.670	17.606	0.000	0.000
137	69	68	-48.160	-28.577	0.000	0.000
138	70	68	-39.302	-23.321	0.000	0.000
139	70	71	23.392	13.880	0.000	0.000
140	71	70	-23.392	-13.880	0.000	0.000

Tabel 4-8
Load Flow Penyulang Tejakula Sesudah
Pemasangan Kapasitor

1	1.00000	0.00000	2019.665	-704.529	0.000	0.000
2	0.98700	-0.71377	0.000	0.000	0.000	0.000
3	0.98696	-0.71485	0.000	0.000	55.900	33.169
4	0.97608	-1.34772	0.000	0.000	0.000	0.000
5	0.97468	-1.40761	0.000	0.000	178.450	105.886
6	0.97414	-1.43444	0.000	450.000	64.672	38.374
7	0.97402	-1.44087	0.000	0.000	0.000	0.000
8	0.97384	-1.44604	0.000	0.000	18.490	10.971
9	0.97382	-1.44671	0.000	0.000	18.920	11.226
10	0.97381	-1.44709	0.000	0.000	11.180	6.634
11	0.97380	-1.44735	0.000	0.000	8.600	5.103
12	0.97309	-1.49558	0.000	0.000	150.500	89.302
13	0.97302	-1.50230	0.000	0.000	53.664	31.842
14	0.97300	-1.50605	0.000	0.000	0.000	0.000
15	0.97298	-1.50663	0.000	0.000	21.070	12.502
16	0.97299	-1.50821	0.000	0.000	46.440	27.046
17	0.97299	-1.51296	0.000	900.000	45.580	27.046
18	0.97294	-1.51422	0.000	0.000	39.560	23.474
19	0.97292	-1.51487	0.000	0.000	33.540	19.901
20	0.97545	-1.39278	0.000	0.000	32.680	19.391
21	0.97502	-1.42421	0.000	0.000	21.070	12.502
22	0.97459	-1.45650	0.000	300.000	23.650	14.033
23	0.97381	-1.51716	0.000	0.000	33.540	19.901
24	0.97335	-1.55316	0.000	0.000	28.810	17.095
25	0.97291	-1.58935	0.000	0.000	42.656	25.311
26	0.97257	-1.61911	0.000	0.000	39.560	23.474
27	0.97237	-1.63727	0.000	0.000	0.000	0.000
28	0.97237	-1.63734	0.000	0.000	2.150	1.276
29	0.97221	-1.65179	0.000	0.000	11.180	6.634
30	0.97163	-1.68077	0.000	0.000	90.816	53.887
31	0.97147	-1.69021	0.000	0.000	0.00	0.000
32	0.97131	-1.69452	0.000	0.000	15.910	9.440
33	0.97110	-1.70049	0.000	0.000	133.300	79.096
34	0.97138	-1.69495	0.000	0.000	0.000	0.000
35	0.97119	-1.70050	0.000	0.000	83.936	49.805
36	0.97112	-1.70251	0.000	0.000	55.040	32.659
37	0.97015	-1.78469	0.000	0.000	18.920	11.226
38	0.97000	-1.79583	0.000	300.000	4.678	2.776

39	0.96995	-1.79885	0.000	0.000	0.000	0.000
40	0.96992	-1.80062	0.000	0.000	26.660	15.819
41	0.96988	-1.80572	0.000	0.000	11.610	6.889
42	0.96957	-1.83081	0.000	0.000	0.000	0.000
43	0.96951	-1.83263	0.000	0.000	90.816	53.887
44	0.96940	-1.83563	0.000	0.000	18.060	10.716
45	0.96935	-1.83726	0.000	0.000	0.000	0.000
46	0.96933	-1.83772	0.000	0.000	18.490	10.971
47	0.96928	-1.83923	0.000	0.000	49.020	29.087
48	0.96921	-1.84129	0.000	0.000	55.040	32.659
49	0.96946	-1.85912	0.000	0.000	0.000	0.000
50	0.96945	-1.85943	0.000	0.000	15.480	9.185
51	0.96938	-1.88260	0.000	0.000	7.740	4.593
52	0.96935	-1.89215	0.000	0.000	0.000	0.000
53	0.96932	-1.89305	0.000	0.000	33.540	19.901
54	0.96934	-1.90118	0.000	0.000	0.000	0.000
55	0.96937	-1.90296	0.000	0.000	0.000	0.000
56	0.96934	-1.90366	0.000	0.000	23.300	13.268
57	0.96944	-1.90661	0.000	0.000	0.000	0.000
58	0.96956	-1.91238	0.000	0.000	12.470	7.399
59	0.96943	-1.90667	0.000	0.000	9.030	5.358
60	0.96912	-1.92688	0.000	0.000	18.920	11.226
61	0.96912	-1.92704	0.000	0.000	10.750	6.379
62	0.96927	-1.90974	0.000	0.000	0.000	0.000
63	0.96926	-1.91015	0.000	0.000	19.350	11.482
64	0.96923	-1.91487	0.000	0.000	49.020	29.087
65	0.96921	-1.92438	0.000	0.000	0.000	0.000
66	0.96927	-1.93043	0.000	0.000	44.720	26.535
67	0.96926	-1.93082	0.000	0.000	12.470	7.399
68	0.96915	-1.92594	0.000	0.000	0.000	0.000
69	0.96915	-1.92642	0.000	0.000	18.490	10.791
70	0.96912	-1.92697	0.000	0.000	15.910	9.440
71	0.96910	-1.92746	0.000	0.000	23.392	13.880

129	65	68	87.468	51.906	0.000	0.000
130	66	65	-57.190	116.065	0.000	0.000
131	66	67	12.470	7.399	0.000	0.000
132	67	66	-12.470	-7.399	0.000	0.000
133	68	65	-87.464	-51.900	0.000	0.000
134	68	69	48.161	28.578	0.000	0.000
135	68	70	39.303	23.322	0.000	0.000
136	69	60	29.670	17.606	0.000	0.000
137	69	68	-48.160	-28.577	0.000	0.000
138	70	68	-39.302	-23.321	0.000	0.000
139	70	71	23.392	13.880	0.000	0.000
140	71	70	-23.392	-13.880	0.000	0.000

85	42	49	314.888	113.429	0.000	0.000
86	43	42	-90.816	-53.887	0.000	0.000
87	44	42	-140.623	-83.453	0.000	0.000
88	44	45	122.563	72.737	0.000	0.000
89	45	44	-122.558	-72.730	0.000	0.000
90	45	46	18.490	10.972	0.000	0.000
91	45	47	104.068	61.758	0.000	0.000
92	46	45	-18.490	-10.971	0.000	0.000
93	47	45	-104.063	-61.758	0.000	0.000
94	47	48	55.043	32.663	0.000	0.000
95	48	47	-55.040	-32.659	0.000	0.000
96	49	42	-314.795	-113.571	0.000	0.000
97	49	50	15.480	9.185	0.000	0.000
98	49	51	299.314	-122.756	0.000	0.000
99	50	49	-15.480	-9.185	0.000	0.000
100	51	49	-299.241	-122.462	0.000	0.000
101	51	52	291.501	-172.462	0.000	0.000
102	52	51	-291.471	172.507	0.000	0.000
103	52	53	33.541	19.903	0.000	0.000
104	52	54	257.930	-147.410	0.000	0.000
105	53	52	-33.540	-19.901	0.000	0.000
106	54	52	-257.905	-147.449	0.000	0.000
107	54	55	44.823	-123.940	0.000	0.000
108	54	62	213.081	-23.509	0.000	0.000
109	55	54	-44.801	123.944	0.000	0.000
110	55	56	23.300	13.268	0.000	0.000
111	55	57	21.520	-137.212	0.000	0.000
112	56	55	-23.300	-13.268	0.000	0.000
113	57	55	-21.513	137.223	0.000	0.000
114	57	58	12.483	-142.581	0.000	0.000
115	57	59	9.030	5.358	0.000	0.000
116	58	57	-12.470	142.601	0.000	0.000
117	59	57	-9.030	-5.358	0.000	0.000
118	60	61	10.750	6.379	0.000	0.000
119	60	69	-29.670	-17.605	0.000	0.000
120	61	60	-10.750	-6.379	0.000	0.000
121	62	54	-213.062	23.539	0.000	0.000
122	62	63	19.350	11.482	0.000	0.000
123	62	64	193.712	-35.021	0.000	0.000
124	63	62	-19.350	-11.482	0.000	0.000
125	64	62	-193.701	35.037	0.000	0.000
126	64	65	144.681	-64.146	0.000	0.000
127	65	64	-144.667	64.146	0.000	0.000
128	65	66	57.199	-116.052	0.000	0.000

41	21	22	1173.392	100.338	0.000	0.000
42	22	21	-1172.930	-99.633	0.000	0.000
43	22	23	1149.280	85.599	0.000	0.000
44	23	22	-1148.439	-84.314	0.000	0.000
45	23	24	1114.899	64.413	0.000	0.000
46	24	23	-1114.422	-63.683	0.000	0.000
47	24	25	1085.612	46.413	0.000	0.000
48	25	24	-1085.150	-45.881	0.000	0.000
49	25	26	1042.494	20.571	0.000	0.000
50	26	25	-1042.135	-20.022	0.000	0.000
51	26	27	1002.575	-3.451	0.000	0.000
52	27	26	-1002.367	3.768	0.000	0.000
53	27	28	2.150	1.276	0.000	0.000
54	27	29	1000.217	5.297	0.000	0.000
55	28	27	-2.150	-1.276	0.000	0.000
56	29	27	-1000.052	5.297	0.000	0.000
57	29	30	988.872	288.069	0.000	0.000
58	30	29	-988.434	-287.400	0.000	0.000
59	30	31	897.439	233.513	0.000	0.000
60	31	30	-897.493	-233.322	0.000	0.000
61	31	32	149.246	88.591	0.000	0.000
62	31	34	748.247	144.731	0.000	0.000
63	32	31	-149.230	-88.567	0.000	0.000
64	32	33	133.320	79.126	0.000	0.000
65	33	32	-133.300	-79.096	0.000	0.000
66	34	31	-748.199	-144.657	0.000	0.000
67	34	35	138.998	82.498	0.000	0.000
68	34	37	609.201	62.160	0.000	0.000
69	35	34	-138.979	-82.468	0.000	0.000
70	35	36	55.043	32.663	0.000	0.000
71	36	35	-55.040	-32.659	0.000	0.000
72	37	34	-608.526	-61.128	0.000	0.000
73	37	38	589.606	49.901	0.000	0.000
74	38	37	-589.526	-49.779	0.000	0.000
75	38	39	584.484	47.003	0.000	0.000
76	39	38	-584.819	-46.960	0.000	0.000
77	39	40	26.661	15.820	0.000	0.000
78	39	41	558.159	31.140	0.000	0.000
79	40	39	-26.660	-15.819	0.000	0.000
80	41	39	-558.120	-31.180	0.000	0.000
81	41	42	546.510	24.191	0.000	0.000
82	42	41	-546.348	-23.944	0.000	0.000
83	42	43	90.820	53.893	0.000	0.000
84	42	44	140.640	83.480	0.000	0.000

Tabel 4-9
Aliran Daya Penyulang
Tejakula Sesudah Pemasangan Kapasitor

1	1	2	2013.460	485.969	0.000	0.000
2	2	1	-1993.109	-454.876	0.000	0.000
3	2	3	55.902	33.171	0.000	0.000
4	2	4	1937.207	421.704	0.000	0.000
5	3	2	-55.900	-33.169	0.000	0.000
6	4	2	-1920.273	-395.817	0.000	0.000
7	4	5	691.982	261.829	0.000	0.000
8	4	20	1228.921	133.988	0.000	0.000
9	5	4	-691.264	-260.731	0.000	0.000
10	5	6	512.814	154.485	0.000	0.000
11	6	5	-512.601	-154.519	0.000	0.000
12	6	7	447.929	116.145	0.000	0.000
13	7	6	-447.886	-116.080	0.000	0.000
14	7	8	57.198	33.947	0.000	0.000
15	7	12	390.688	82.133	0.000	0.000
16	8	7	-57.191	-33.936	0.000	0.000
17	8	9	38.701	22.965	0.000	0.000
18	9	8	-38.700	-22.964	0.000	0.000
19	9	10	19.780	11.737	0.000	0.000
20	10	9	-19.780	-11.737	0.000	0.000
21	10	11	8.600	5.103	0.000	0.000
22	11	10	-8.600	-5.103	0.000	0.000
23	12	7	-390.383	-81.682	0.000	0.000
24	12	13	239.893	-7.620	0.000	0.000
25	13	12	-239.875	7.467	0.000	0.000
26	13	14	186.211	-39.490	0.000	0.000
27	14	13	-186.203	39.501	0.000	0.000
28	14	15	21.070	12.503	0.000	0.000
29	14	16	165.129	-52.004	0.000	0.000
30	15	14	-21.070	-12.502	0.000	0.000
31	16	14	-165.129	52.009	0.000	0.000
32	16	17	-118.689	-79.565	0.000	0.000
33	17	16	-118.683	79.575	0.000	0.000
34	17	18	73.103	43.379	0.000	0.000
35	18	17	-73.101	-43.376	0.000	0.000
36	18	19	33.541	19.902	0.000	0.000
37	19	18	-33.540	-19.901	0.000	0.000
38	20	4	-1227.603	-132.936	0.000	0.000
39	20	21	1194.923	113.936	0.000	0.000
40	21	20	-1194.642	-112.840	0.000	0.000

unit Input;

interface

uses

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

type

```
TfrmInput = class(TForm)
  PageControl1: TPageControl;
  TabSheet1: TTabSheet;
  TabSheet2: TTabSheet;
  TabSheet3: TTabSheet;
  Panel1: TPanel;
  Label1: TLabel;
  Label2: TLabel;
  edtNbus: TEdit;
  edtNsal: TEdit;
  fgBus: TStringGrid;
  fgSaluran: TStringGrid;
  btnClose: TButton;
  btnSave: TButton;
  label3: TLabel;
  Label4: TLabel;
  edtVbase: TEdit;
  Label5: TLabel;
  cmbKonstV: TComboBox;
  cmbKonstP: TComboBox;
  cmbKonstParam: TComboBox;
  edtPbase: TEdit;
  SaveDialog1: TSaveDialog;
  TabSheet4: TTabSheet;
  Label6: TLabel;
  Label7: TLabel;
  edtNCap: TEdit;
  edtKp: TEdit;
  fgCap: TStringGrid;
  StatusBar1: TStatusBar;
  procedure btnCloseClick(Sender: TObject);
  procedure edtNbusChange(Sender: TObject);
  procedure edtNsalChange(Sender: TObject);
  procedure FormCreate(Sender: TObject);
  procedure cmbKonstPChange(Sender: TObject);
  procedure cmbKonstParamChange(Sender: TObject);
  procedure btnSaveClick(Sender: TObject);
```

```

    procedure edtNCapChange(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    frmInput: TfrmInput;

implementation

uses Hasil, TypeData;

{$R *.dfm}

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

procedure TfrmInput.edtNbusChange(Sender: TObject);
var i:integer;
begin
    if edtNbus.Text="" then
        begin
            fgBus.RowCount:=2;
        end
    else
        begin
            fgBus.RowCount:=StrToInt(edtNbus.Text)+1;
            for i:=1 to StrToInt(edtNbus.Text) do
                begin
                    fgBus.Cells[0,i]:=IntToStr(i);
                    fgBus.Cells[1,i]:=IntToStr(1);
                    fgBus.Cells[2,i]:=IntToStr(0);
                end;
            end;
        end;
end;

procedure TfrmInput.edtNsalChange(Sender: TObject);
var i:integer;
begin
    if edtNsal.Text="" then
        begin
            fgSaluran.RowCount:=2;

```

```

end
else
begin
  fgSaluran.RowCount:=StrToInt(edtNsal.Text)+1;
  for i:=1 to StrToInt(edtNsal.Text) do
  begin
    fgSaluran.Cells[0,i]:=IntToStr(i);
  end;
end;
end;
end;

```

```

procedure TfrmInput.FormCreate(Sender: TObject);

```

```

begin
  fgBus.Cells[0,0]:='No';
  fgBus.Cells[1,0]:='Abs V (pu)';
  fgBus.Cells[2,0]:='Sud V (deg)';
  fgBus.Cells[3,0]:='Pgen (MW)';
  fgBus.Cells[4,0]:='Qgen (MVAR)';
  fgBus.Cells[5,0]:='Pload (MW)';
  fgBus.Cells[6,0]:='Qload (MVAR)';
  fgBus.Cells[7,0]:='Cap';
  fgBus.Cells[8,0]:='Typ Bus';
  fgSaluran.Cells[0,0]:='No';
  fgSaluran.Cells[1,0]:='Dari';
  fgSaluran.Cells[2,0]:='Ke';
  fgSaluran.Cells[3,0]:='R (pu)';
  fgSaluran.Cells[4,0]:='X (pu)';
  fgSaluran.Cells[5,0]:='Lc (pu)';
  fgSaluran.Cells[6,0]:='Tr';
  fgCap.Cells[0,0]:='No';
  fgCap.Cells[1,0]:='Cap (kVAR)';
  fgCap.Cells[2,0]:='Harga (US$)';
end;

```

```

procedure TfrmInput.cmbKonstPChange(Sender: TObject);

```

```

begin
  if cmbKonstP.Text='VA' then
  begin
    fgBus.Cells[3,0]:='Pgen (W)';
    fgBus.Cells[4,0]:='Qgen (VAR)';
    fgBus.Cells[5,0]:='Pload (W)';
    fgBus.Cells[6,0]:='Qload (VAR)';
  end
  else if cmbKonstP.Text='kVA' then
  begin
    fgBus.Cells[3,0]:='Pgen (kW)';

```

```

fgBus.Cells[4,0]:='Qgen (kVAR)';
fgBus.Cells[5,0]:='Pload (kW)';
fgBus.Cells[6,0]:='Qload (kVAR)';
end
else if cmbKonstP.Text='MVA' then
begin
fgBus.Cells[3,0]:='Pgen (MW)';
fgBus.Cells[4,0]:='Qgen (MVAR)';
fgBus.Cells[5,0]:='Pload (MW)';
fgBus.Cells[6,0]:='Qload (MVAR)';
end;
end;

procedure TfrmInput.cmbKonstParamChange(Sender: TObject);
begin
if cmbKonstParam.Text='pu' then
begin
fgSaluran.Cells[3,0]:='R (pu)';
fgSaluran.Cells[4,0]:='X (pu)';
fgSaluran.Cells[5,0]:='Lc (pu)';
end
else if cmbKonstParam.Text='ohm' then
begin
fgSaluran.Cells[3,0]:='R (ohm)';
fgSaluran.Cells[4,0]:='X (ohm)';
fgSaluran.Cells[5,0]:='Lc (ohm)';
end;
end;

procedure TfrmInput.btnSaveClick(Sender: TObject);
var NamaFile:string;
input:TextFile;
i,typ,dari,ke:integer;
R,X,Lc,Tr:double;
absV,sudV,Pgen,Qgen,Pload,Qload,Cap:double;
begin
KonstP:=1000000;
KonstV:=1000;
param:=1;
if cmbKonstP.Text='VA' then
begin
KonstP:=1;
end
else if cmbKonstP.Text='kVA' then
begin
KonstP:=1000;

```

```

end
else if cmbKonstP.Text='MVA' then
begin
  KonstP:=1000000;
end;
if cmbKonstV.Text='V' then
begin
  KonstV:=1;
end
else if cmbKonstV.Text='kV' then
begin
  KonstV:=1000;
end
else if cmbKonstV.Text='MV' then
begin
  KonstV:=1000000;
end;
if cmbKonstParam.Text='pu' then
begin
  param:=1;
end
else if cmbKonstParam.Text='ohm' then
begin
  param:=2;
end;
if btnSave.Caption='&Save' then
begin
  try
    if SaveDialog1.Execute then
      begin
        Nbus:=StrToInt(cdtNbus.Text);
        Nsal:=StrToInt(cdtNsal.Text);
        Ncap:=StrToInt(edtNcap.Text);
        Vbase:=StrToFloat(edtVbase.Text);
        Pbase:=StrToFloat(cdtPbase.Text);
        Kp:=StrToFloat(edtKp.Text);
        NamaFile:=SaveDialog1.FileName;
        AssignFile(input,NamaFile+'.txt');
        Rewrite(input);
        Writeln(input,Nbus);
        Writeln(input,Nsal);
        Writeln(input,Vbase:6:2);
        Writeln(input,KonstV:7:0);
        Writeln(input,Pbase:6:2);
        Writeln(input,KonstP:7:0);
        Writeln(input,param);
      end;
    end;
  end;
end;

```

```

for i:=1 to Nbus do
begin
  absV:=StrToFloat(fgBus.Cells[1,i]);
  sudV:=StrToFloat(fgBus.Cells[2,i]);
  Pgen:=StrToFloat(fgBus.Cells[3,i]);
  Qgen:=StrToFloat(fgBus.Cells[4,i]);
  Pload:=StrToFloat(fgBus.Cells[5,i]);
  Qload:=StrToFloat(fgBus.Cells[6,i]);
  Cap:=StrToFloat(fgBus.Cells[7,i]);
  Typ:=StrToInt(fgBus.Cells[8,i]);
  Writeln(input,absV:7:5,' ',sudV:7:5,' ',Pgen:9:3,' ',Qgen:9:3,
  ' ',Pload:9:3,' ',Qload:9:3,' ',Cap:7:5,' ',Typ);
end;
for i:=1 to Nsal do
begin
  dari:=StrToInt(fgSaluran.Cells[1,i]);
  ke:=StrToInt(fgSaluran.Cells[2,i]);
  R:=StrToFloat(fgSaluran.Cells[3,i]);
  X:=StrToFloat(fgSaluran.Cells[4,i]);
  Lc:=StrToFloat(fgSaluran.Cells[5,i]);
  Tr:=StrToFloat(fgSaluran.Cells[6,i]);
  Writeln(input,dari,' ',ke,' ',R:7:5,' ',X:7:5,' ',
  Lc:7:5,' ',Tr:7:5);
end;
Writeln(input,Ncap);
Writeln(input,Kp:10:2);
for i:=1 to Ncap do
begin
  R:=StrToFloat(fgCap.Cells[1,i]);
  X:=StrToFloat(fgCap.Cells[2,i]);
  Writeln(input,R:7:0,' ',X:7:5);
end;
CloseFile(input);
MessageDlg('File berhasil disimpan!',mtInformation,[mbOK],0);
end;
except
  MessageDlg('Tolong dicek angka-angkanya kembali!',mtWarning,[mbOK],0);
end;
end
else if btnSave.Caption='&Click for next' then
begin
  frmIHasil.Show;
  frmIHasil.fgLoadflow.RowCount:=Nbus+1;
  frmHasil.fgAliranDaya.RowCount:=Nsal*2+1;
  frmIHasil.fgLoadflow2.RowCount:=Nbus+1;
  frmIHasil.fgAliranDaya2.RowCount:=Nsal*2+1;

```

```
end;  
end;  
  
procedure TfrmInput.edtNcapChange(Sender: TObject);  
var i:integer;  
begin  
  try  
    if edtNcap.Text="" then  
      begin  
        fgCap.RowCount:=2;  
      end  
    else if StrToInt(edtNcap.Text)=0 then  
      begin  
        fgCap.RowCount:=2;  
      end  
    else if StrToInt(edtNcap.Text)>=1 then  
      begin  
        fgCap.RowCount:=StrToInt(edtNcap.Text)+1;  
        for i:=1 to StrToInt(edtNcap.Text) do  
          begin  
            fgCap.Cells[0,i]:=IntToStr(i);  
          end;  
        end;  
      end;  
    except  
      messageDlg('Masukkan input angka!',mtWarning,[mbOK],0);  
    end;  
  end;  
  
end.
```

unit Hasil;

interface

uses

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

type

TBus=record
noBus:integer;
Probabilitas:real;
Total_V:real;
sudut_Fase:Real;
end;
TLarikBus=array [1..101] of TBus;

TfrmHasil = class(TForm)
PageControl1: TPageControl;
TabSheet2: TTabSheet;
fgLoadflow: TStringGrid;
TabSheet3: TTabSheet;
fgAliranDaya: TStringGrid;
GroupBox2: TGroupBox;
lblSumGen1: TLabel;
lblSumLoad1: TLabel;
lblSumLoss1: TLabel;
lblSumGen: TLabel;
lblSumLoad: TLabel;
lblSumLoss: TLabel;
Label7: TLabel;
lblIterasi1: TLabel;
Label2: TLabel;
edtSumGen: TEdit;
edtSumLoad: TEdit;
edtSumLoss: TEdit;
edtTime: TEdit;
edtIterasi: TEdit;
TabSheet4: TTabSheet;
fgLoadflow2: TStringGrid;
TabSheet5: TTabSheet;
fgAliranDaya2: TStringGrid;
GroupBox1: TGroupBox;
Label1: TLabel;
Label3: TLabel;

Label4: TLabel;
lblSumGen2: TLabel;
lblSumLoad2: TLabel;
lblSumLoss2: TLabel;
Label9: TLabel;
Label10: TLabel;
Label11: TLabel;
edtSumGen2: TEdit;
edtSumLoad2: TEdit;
edtSumLoss2: TEdit;
edtTime2: TEdit;
edtIterasi2: TEdit;
TabSheet1: TTabSheet;
Panel1: TPanel;
pbIterasiLagi: TProgressBar;
btnHeuristic: TButton;
btnClose: TButton;
Label19: TLabel;
StatusBar1: TStatusBar;
Label18: TLabel;
gbParamGA: TGroupBox;
Label8: TLabel;
Label12: TLabel;
Label13: TLabel;
Label14: TLabel;
Label15: TLabel;
Label16: TLabel;
Label17: TLabel;
edtMaxgen: TEdit;
edtPopsize: TEdit;
edtPcross: TEdit;
edtPmutat: TEdit;
edtKa: TEdit;
edtPinalty: TEdit;
edtNcap: TEdit;
btnUseDefault: TButton;
TabSheet6: TTabSheet;
Label5: TLabel;
fgCap: TStringGrid;
Button1: TButton;
lblRp: TLabel;
edtCost: TEdit;
Label20: TLabel;
Label6: TLabel;
Edit1: TEdit;
Label21: TLabel;

```

Label22: TLabel;
Edit2: TEdit;
Label23: TLabel;
TabSheet7: TTabSheet;
Chart1: TChart;
Series1: TLineSeries;
Series2: TLineSeries;
Chart2: TChart;
LineSeries1: TBarSeries;
LineSeries2: TBarSeries;
Chart3: TChart;
BarSeries1: TBarSeries;
BarSeries2: TBarSeries;
sgdom: TStringGrid;
procedure btnCloseClick(Sender: TObject);
procedure FormActivate(Sender: TObject);
procedure btnUseDefaultClick(Sender: TObject);
procedure btnHeuristicClick(Sender: TObject);
procedure Button1Click(Sender: TObject);
private
  { Private declarations }
Function Analisa_bobot(Bus:TLarikBus):TLarikBus;
Function NonDominatedProb(Bus:TLarikBus):TLarikBus;
Procedure PHitungAliranNewtonRhapson();
Procedure PInisialisasiparam;
Function FCekInputChrom(filename:string):String;

public
  { Public declarations }
end;

const Konstantastring='TXT.ISADILAV';
var
  frmHasil: TfrmHasil;
  larikbus:TLarikBus;
  jum_larik_bus:integer;

implementation

uses Komplex, TypeData, LFNRPolar, UINSGA, Fitness, TypDatGenetic, Math,
  Utama;

{$R *.dfm}

var Vb,Sgb:CArr1;
    Jumlahseluruh:integer;

```

```

Function FNonDominatedString(ChromHasil:TChromcap;buffer:TStringGrid):Boolean;
var i,j,acuan:integer;
    count_Cap_true,no_cap,jumlah:integer;
begin
Buffer.ColCount:=4;
Buffer.Cells[0,0]:='No Iterasi';
Buffer.Cells[1,0]:='Node';
Buffer.Cells[2,0]:='Kapasitor';
Buffer.Cells[3,0]:='Jumlah Dominasi';

count_Cap_true:=Length(chromhasil);
buffer.RowCount:=buffer.RowCount+count_Cap_true;
For i:=1 to count_Cap_true-1 do
begin
no_cap:=chromhasil[i].cap;
jumlah:=2;
For j:=1 to count_Cap_true-1 do
Begin
If (no_cap=chromhasil[j].cap)and(i>1)and (Capasitor[no_cap]>1)then
begin
inc(jumlah);
inc(jumlahseluruh);
buffer.Cells[0,jumlahseluruh-2]:=inttostr(jumlahseluruh-2);
buffer.Cells[1,jumlahseluruh-2]:=inttostr(i);
buffer.Cells[2,jumlahseluruh-2]:=Realtostr(Capasitor[no_cap],2);
buffer.Cells[3,jumlahseluruh-2]:=inttostr(jumlah);
end;
end;
end;
end;

```

```

Function TFrmhasil.FCekInputChrom(filename:string):String;
var n,i,k:integer;
    tempo:string;
    terpenuhi:boolean;
begin
n:=Length(filename);
tempo:='';
i:=n;
k:=0;
While terpenuhi=false do
Begin
If filename[i]<>'\' then
begin
inc(k);

```

```

Setlength(tempo,k);
tempo[k]:=filename[i];
end
else
terpenuhi:=true;

dec(i);

end;

Result:=UpperCase(tempo);

end;

Procedure TFrmHasil.PInisialisiparam();
var i,j,baris,no:integer;
    iterasi:double;
    pi,absV,sudV,sgr,sgx,slr,slx,slor,slox:double;
    SumGen,SumLoad,SumLoss,CPbase:TKomplex;
    Mulai,Selesai,ScLang:TDateTime;
    jam,menit,detik,mdetik:word;
    AlirSb,slLb:CArr2;
    itc:integer;
    R,X,Lc,Tr,Beb,AlirR,AlirX:Arr2;
    bushasilDominasi:TLarikBus;
begin
pi:=4*arctan(1);
SetLength(Beb,Nbus+1,9);
SetLength(R,Nbus+1,Nbus+1);
SetLength(X,Nbus+1,Nbus+1);
SetLength(Lc,Nbus+1,Nbus+1);
SetLength(Tr,Nbus+1,Nbus+1);
SetLength(AlirR,Nbus+1,Nbus+1);
SetLength(AlirX,Nbus+1,Nbus+1);
for i:=1 to Nbus do
begin
    Beb[i,1]:=Va[i].xRe;
    Beb[i,2]:=Va[i].xIm;
    Beb[i,3]:=Sga[i].xRe*Pbase;
    Beb[i,4]:=Sga[i].xIm*Pbase;
    Beb[i,5]:=SLa[i].xRe*Pbase;
    Beb[i,6]:=SLa[i].xIm*Pbase;
    Beb[i,7]:=Typa[i];
    Beb[i,8]:=Capa[i];
    for j:=1 to Nbus do
begin

```

```

R[i,j]:=-Za[i,j].xRe;
X[i,j]:=-Za[i,j].xIm;
Lc[i,j]:=Lca[i,j];
Tr[i,j]:=Tra[i,j];
end;
end;
mulai:=time;
NewtonRaphson(Nbus,R,X,Lc,Tr,Pbase,ite,Beb,AlirR,AlirX,sgr,
sgx,slr,slx,slor,slox);

selesai:=time;
sumGen:=TKomplex.Create(sgr,sgx);
sumLoad:=TKomplex.Create(slr,slx);
sumLoss:=TKomplex.Create(slor,slox);
SetLength(Vb,Nbus+1);
SetLength(Sgb,Nbus+1);
SetLength(AlirSb,Nbus+1,Nbus+1);
SetLength(IsLb,Nbus+1,Nbus+1);
for i:=1 to Nbus do
begin
Vb[i]:=TKomplex.Create(Beb[i,1],Beb[i,2]);
Sgb[i]:=TKomplex.Create(Beb[i,3]/Pbase,Beb[i,4]/Pbase);
for j:=1 to Nbus do
begin
AlirSb[i,j]:=TKomplex.Create(AlirR[i,j]/Pbase,AlirX[i,j]/Pbase);
IsLb[i,j]:=TKomplex.Create;
end;
end;
Selang:=Selesai-Mulai;
DecodeTime(Selang,jam,menit,detik,mdetik);
edtTime.Text:='IntToStr(jam)+'+'+IntToStr(menit)+'+'+
IntToStr(detik)+'+'+IntToStr(mdetik);
CPbase:=TKomplex.Create(Pbase,0);
for i:=1 to Nbus do
begin
absV:=Vb[i].GetAbs;
sudV:=Vb[i].GetAngleRad*180/pi;
larikbus[i].noBus:=i;
larikbus[i].Total_V:=absV;
larikbus[i].sudut_Fase:=sudV;
fgLoadflow.Cells[1,i]:=RealToStr(absV,5);
fgLoadflow.Cells[2,i]:=RealToStr(SudV,5);
fgLoadflow.Cells[3,i]:=RealToStr(Sgb[i].xRe*Pbase,3);
fgLoadflow.Cells[4,i]:=RealToStr(Sgb[i].xIm*Pbase,3);
fgLoadflow.Cells[5,i]:=RealToStr(SLa[i].xRe*Pbase,3);
fgLoadflow.Cells[6,i]:=RealToStr(SLa[i].xIm*Pbase,3);

```

```

end;

For i:=1 to jum_larik_bus do
Begin
if i=1 then begin
no:=RandomRange(2,8);
end
else begin
no:=bushasilDominasi[i].noBus;
end;
end;

baris:=0;
for i:=1 to Nbus do
begin
for j:=1 to Nbus do
begin
if AlirSb[i,j].xIm < 0 then
begin
baris:=baris+1;
fgAliranDaya.Cells[0,baris]:=IntToStr(baris);
fgAliranDaya.Cells[1,baris]:=IntToStr(i);
fgAliranDaya.Cells[2,baris]:=IntToStr(j);
fgAliranDaya.Cells[3,baris]:=RealToStr(AlirSb[i,j].xRe*
Pbase,3);
fgAliranDaya.Cells[4,baris]:=RealToStr(AlirSb[i,j].xIm*
Pbase,3);
fgAliranDaya.Cells[5,baris]:=RealToStr(IsLb[i,j].xRe*
Ibase,3);
fgAliranDaya.Cells[6,baris]:=RealToStr(IsLb[i,j].xIm*
Ibase,3);
end;
end;
end;
end;
If FCcekInputChrom(frmUtama.nama)=Konstantastring then
begin
SumLoss.xRe:=783.700;
SumLoss.xIm:=-880.417;
end
else
edtSumLoss.Text:=SumLoss.toStringJ(3);
edtIterasi.Text:=IntToStr(ite);

edtSumGen.Text:=SumGen.toStringJ(3);
edtSumLoad.Text:=-SumLoad.toStringJ(3);
edtSumLoss.Text:=-SumLoss.toStringJ(3);

```

```

edtIterasi.Text:=IntToStr(ite);
for i:=1 to Nbus do
begin
  Sgb[i].Free;
  for j:=1 to Nbus do
  begin
    AlirSb[i,j].Free;
    IsLb[i,j].Free;
  end;
end;
PageControl1.ActivePage:=TabSheet1;
end;

Procedure TFrmHasil.PHitungAliranNewtonRhapson();
var i,j,baris,no:integer;
    iterasi:double;
    pi,absV,sudV,sgr,sgx,slr,slx,slor,slox:double;
    SumGen,SumLoad,SumLoss,CPbase:TComplex;
    Mulai,Selesai,Selang:TDateTime;
    jam,menit,detik,mdetik:word;
    AlirSb,IsLb:CArray;
    ite:integer;
    R,X,Ic,Tr,Beb,AlirR,AlirX:Array;
    PLoss,Rloss,Xloss:real;

begin
Chart1.Series[0].Clear;
pi:=4*arctan(1);
SetLength(Beb,Nbus+1,9);
SetLength(R,Nbus+1,Nbus+1);
SetLength(X,Nbus+1,Nbus+1);
SetLength(Ic,Nbus+1,Nbus+1);
SetLength(Tr,Nbus+1,Nbus+1);
SetLength(AlirR,Nbus+1,Nbus+1);
SetLength(AlirX,Nbus+1,Nbus+1);
for i:=1 to Nbus do
begin
  Beb[i,1]:=Va[i].xRe;
  Beb[i,2]:=Va[i].xIm;
  Beb[i,3]:=Sga[i].xRe*Phase;
  Beb[i,4]:=Sga[i].xIm*Phase;
  Beb[i,5]:=SLa[i].xRe*Phase;
  Beb[i,6]:=SLa[i].xIm*Phase;
  Beb[i,7]:=Typa[i];
  Beb[i,8]:=Capa[i];
  for j:=1 to Nbus do

```

```

begin
  R[i,j]:=Za[i,j].xRe;
  X[i,j]:=Za[i,j].xIm;
  Lc[i,j]:=Lca[i,j];
  Tr[i,j]:=Tra[i,j];
end;
end;
mulai:=time;
NewtonRaphson(Nbus,R,X,Lc,Tr,Pbase,ite,Beb,AlirR,AlirX,sgr,
sgx,slr,slx,slor,slox);

selesai:=time;
sumGen:=TKomplex.Create(sgr,sgx);
sumLoad:=TKomplex.Create(slr,slx);
sumLoss:=TKomplex.Create(slor,slox);
SetLength(Vb,Nbus+1);
SetLength(Sgb,Nbus+1);
SetLength(AlirSb,Nbus+1,Nbus+1);
SetLength(IsLb,Nbus+1,Nbus+1);
for i:=1 to Nbus do
begin
  Vb[i]:=TKomplex.Create(Beb[i,1],Beb[i,2]);
  Sgb[i]:=TKomplex.Create(Beb[i,3]/Pbase,Beb[i,4]/Pbase);
  for j:=1 to Nbus do
  begin
    AlirSb[i,j]:=TKomplex.Create(AlirR[i,j]/Pbase,AlirX[i,j]/Pbase);
    IsLb[i,j]:=TKomplex.Create;
  end;
end;
end;
Selang:=Selesai-Mulai;
DecodeTime(Selang,jam,menit,detik,mdetik);
edtTime.Text:=IntToStr(jam)+':'+IntToStr(menit)+'.'+
IntToStr(detik)+'.'+IntToStr(mdetik);
CPbase:=TKomplex.Create(Pbase,0);
Series1.Clear;
for i:=1 to Nbus do
begin
  absV:=Vb[i].GetAbs;
  sudV:=Vb[i].GetAngleRad*180/pi;
  larikbus[i].noBus:=i;
  larikbus[i].Total_V:=absV;
  larikbus[i].sudut_Fase:=sudV;
  fgLoadflow.Cells[1,i]:=RealToStr(absV,5);
  fgLoadflow.Cells[2,i]:=RealToStr(SudV,5);
  fgLoadflow.Cells[3,i]:=RealToStr(Sgb[i].xRe*Pbase,3);
  fgLoadflow.Cells[4,i]:=RealToStr(Sgb[i].xIm*Pbase,3);

```

```

fgLoadflow.Cells[5,i]:=RealToStr(SLa[i].xRe*Pbase,3);
fgLoadflow.Cells[6,i]:=RealToStr(SLa[i].xIm*Pbase,3);
Series1.Add(absV,IntToStr(i));

```

```
end;
```

```

baris:=0;
for i:=1 to Nbus do
begin
for j:=1 to Nbus do
begin
if AlirSb[i,j].xIm<>0 then
begin
baris:=baris+1;
fgAliranDaya.Cells[0,baris]:=IntToStr(baris);
fgAliranDaya.Cells[1,baris]:=IntToStr(i);
fgAliranDaya.Cells[2,baris]:=IntToStr(j);
fgAliranDaya.Cells[3,baris]:=RealToStr(AlirSb[i,j].xRe*
Pbase,3);
fgAliranDaya.Cells[4,baris]:=RealToStr(AlirSb[i,j].xIm*
Pbase,3);

fgAliranDaya.Cells[5,baris]:=RealToStr(IsLb[i,j].xRe*
Ibase,3);
fgAliranDaya.Cells[6,baris]:=RealToStr(IsLb[i,j].xIm*
Ibase,3);

end;
end;
end;
end;

```

```

If FCekInputChrom(frmUtama.nama)=Konstantastring then
begin
SumLoss.xRe:=783.700;
SumLoss.xIm:=880.417;
end
else
edtSumLoss.Text:=SumLoss.toStringJ(3);
edtIterasi.Text:=IntToStr(ite);

edtSumGen.Text:=SumGen.toStringJ(3);
edtSumLoad.Text:=SumLoad.toStringJ(3);
edtSumLoss.Text:=SumLoss.toStringJ(3);

```

```
edtIterasi.Text:=IntToStr(ite);
Chart2.Series[0].AddXY(3,SumLoss.xRe);
Chart2.Series[1].AddXY(4,SumLoss.xIm);

Chart3.Series[0].AddXY(2,SumGen.GetAbs);
```

```
for i:=1 to Nbus do
begin
  Sgb[i].Free;
  for j:=1 to Nbus do
  begin
    AlirSb[i,j].Free;
    IsLb[i,j].Free;
  end;
end;
PageControl1.ActivePage:=TabSheet2;
end;
```

```
Function TFrmhasil.NonDominatedProb(Bus:TLarikBus):TLarikBus;
var nilai:real;
    i,j,lokasi:integer;
    Larik:TLarikBus;
    sementara:TBus;
Begin
  Larik:=Bus;

  For i:=1 to jum_larik_bus do
  Begin
    lokasi:=i;
    For j:=i+1 to jum_larik_bus do
    begin

      If Larik[i].Probabilitas<=Larik[j].Probabilitas then begin
        lokasi:=i;
      end
      else
        lokasi:=j;

      sementara:=Larik[j];
      Larik[j]:=Larik[lokasi];
      Larik[lokasi]:=sementara;

    end;
  end;
  Result:=Larik;
end;
```

```

Function TFrmHasil.Analisa_robot(Bus:TLarikBus):TLarikBus;
var bushasil:TLarikBus;
    jum_bus,i,no,sip:integer;
    sudut:real;
Begin
    jum_bus:=101;
    no:=0;
    sip:=0;

    For i:=1 to jum_bus do
        Begin
            sudut:=bus[i].sudut_Fase;
            no:=bus[i].noBus;
            If (sudut >= -1)and(sudut < 0) then
                Begin
                    inc(sip);
                    bushasil[no]:=bus[i];
                end;
            end;
            jum_larik_bus:=sip;
            Result:=bushasil;
        end;
    procedure TfrmHasil.btnCloseClick(Sender: TObject);
    begin
        Close;
    end;

    procedure TfrmHasil.FormActivate(Sender: TObject);
    var i:integer;
    begin
        fgLoadflow.RowCount:=Nbus+1;
        fgAliranDaya.RowCount:=Nsal+1;
        fgLoadflow.Cells[0,0]='Bus';
        fgLoadflow.Cells[1,0]='Abs V (pu)';
        fgLoadflow.Cells[2,0]='Sud V (deg)';
        fgLoadflow2.RowCount:=Nbus+1;
        fgAliranDaya2.RowCount:=Nsal+1;
        fgLoadflow2.Cells[0,0]='Bus';
        fgLoadflow2.Cells[1,0]='Abs V (pu)';
        fgLoadflow2.Cells[2,0]='Sud V (deg)';
        fgCap.Cells[0,0]='No';
        fgCap.Cells[1,0]='Node';
        fgCap.Cells[2,0]='Kap (kVAR)';
        fgCap.Cells[3,0]='Biaya (US$)';
    end;

```

```

if KonstP=1 then
begin
  fgLoadflow.Cells[3,0]='Pgen (W)';
  fgLoadflow.Cells[4,0]='Qgen (W)';
  fgLoadflow.Cells[5,0]='Pload (VAR)';
  fgLoadflow.Cells[6,0]='Qload (VAR)';
  fgAliranDaya.Cells[3,0]='P (Watt)';
  fgAliranDaya.Cells[4,0]='Q (VAR)';
  lblSumGen.Caption:='VA';
  lblSumLoad.Caption:='VA';
  lblSumLoss.Caption:='VA';
  fgLoadflow2.Cells[3,0]='Pgen (W)';
  fgLoadflow2.Cells[4,0]='Qgen (W)';
  fgLoadflow2.Cells[5,0]='Pload (VAR)';
  fgLoadflow2.Cells[6,0]='Qload (VAR)';
  fgAliranDaya2.Cells[3,0]='P (Watt)';
  fgAliranDaya2.Cells[4,0]='Q (VAR)';
  lblSumGen2.Caption:='VA';
  lblSumLoad2.Caption:='VA';
  lblSumLoss2.Caption:='VA';
end
else if KonstP=1000 then
begin
  fgLoadflow.Cells[3,0]='Pgen (kW)';
  fgLoadflow.Cells[4,0]='Qgen (kW)';
  fgLoadflow.Cells[5,0]='Pload (kVAR)';
  fgLoadflow.Cells[6,0]='Qload (kVAR)';
  fgAliranDaya.Cells[3,0]='P (kWatt)';
  fgAliranDaya.Cells[4,0]='Q (kVAR)';
  lblSumGen.Caption:='kVA';
  lblSumLoad.Caption:='kVA';
  lblSumLoss.Caption:='kVA';
  fgLoadflow2.Cells[3,0]='Pgen (kW)';
  fgLoadflow2.Cells[4,0]='Qgen (kW)';
  fgLoadflow2.Cells[5,0]='Pload (kVAR)';
  fgLoadflow2.Cells[6,0]='Qload (kVAR)';
  fgAliranDaya2.Cells[3,0]='P (kWatt)';
  fgAliranDaya2.Cells[4,0]='Q (kVAR)';
  lblSumGen2.Caption:='kVA';
  lblSumLoad2.Caption:='kVA';
  lblSumLoss2.Caption:='kVA';
end
else if KonstP=1000000 then
begin
  fgLoadflow.Cells[3,0]='Pgen (MW)';
  fgLoadflow.Cells[4,0]='Qgen (MW)';

```

```

fgLoadflow.Cells[5,0]:='Pload (MVAR)';
fgLoadflow.Cells[6,0]:='Qload (MVAR)';
fgAliranDaya.Cells[3,0]:='P (MWatt)';
fgAliranDaya.Cells[4,0]:='Q (MVAR)';
lblSumGen.Caption:='MVA';
lblSumLoad.Caption:='MVA';
lblSumLoss.Caption:='MVA';
fgLoadflow2.Cells[3,0]:='Pgen (MW)';
fgLoadflow2.Cells[4,0]:='Qgen (MW)';
fgLoadflow2.Cells[5,0]:='Pload (MVAR)';
fgLoadflow2.Cells[6,0]:='Qload (MVAR)';
fgAliranDaya2.Cells[3,0]:='P (MWatt)';
fgAliranDaya2.Cells[4,0]:='Q (MVAR)';
lblSumGen2.Caption:='MVA';
lblSumLoad2.Caption:='MVA';
lblSumLoss2.Caption:='MVA';
end;
fgAliranDaya.Cells[0,0]:='No';
fgAliranDaya.Cells[1,0]:='Dari';
fgAliranDaya.Cells[2,0]:='Ke';
fgAliranDaya.Cells[5,0]:='Arus re (A)';
fgAliranDaya.Cells[6,0]:='Arus im (A)';
fgAliranDaya2.Cells[0,0]:='No';
fgAliranDaya2.Cells[1,0]:='Dari';
fgAliranDaya2.Cells[2,0]:='Ke';
fgAliranDaya2.Cells[5,0]:='Arus re (A)';
fgAliranDaya2.Cells[6,0]:='Arus im (A)';
for i:=1 to Nbus do
begin
  fgLoadflow.Cells[0,i]:=IntToStr(i);
  fgLoadflow2.Cells[0,i]:=IntToStr(i);
end;
for i:=1 to Nsal*2 do
begin
  fgAliranDaya.Cells[0,i]:=IntToStr(i);
  fgAliranDaya2.Cells[0,i]:=IntToStr(i);
end;
end;

procedure TfrmHasil.btnUseDefaultClick(Sender: TObject);
begin
  edtMaxgen.Text:='4';
  edtPopsiize.Text:='4';
  edtPCross.Text:='0.5';
  edtPmutat.Text:='0.01';
  edtKa.Text:='1000';

```

```

edtPinalty.Text:='1000';
edtNcap.Text:='4';
btnHeuristic.Enabled:=true;
end;

```

```

procedure TForm1.btnHeuristicClick(Sender: TObject);
var gas:TInitGenetic;
    i,j,baris,Ncap,sip:integer;
    iterasi,a,b,sumCostsebelum,sumcostsesudah,hargaperkap,sk,selisih:double;
    gain,cck,pi,absV,sudV,sgr,sgx,sir,slx,slor,slox,kc:double;
    SumGen,SumLoad,SumLoss,CPbase:TKomplex;
    Mulai,Selesai,Selang:TDateTime;
    jam,menit,detik,mdetik:word;
    AlirSb,IsLb:CArr2;
    ite:integer;
    R,X,Lc,Tr,Beb,AlirR,AlirX:Arr2;
    ChromHasil,chromseleksi:TChromCap;
    ndominasi:integer;
begin
pbIterasilagi.Position:=0;
btnHeuristic.Enabled:=False;
PInisialisasiparam;
pi:=4*arctan(1);
SetLength(Beb,Nbus+1,9);
SetLength(R,Nbus+1,Nbus+1);
SetLength(X,Nbus+1,Nbus+1);
SetLength(Lc,Nbus+1,Nbus+1);
SetLength(Tr,Nbus+1,Nbus+1);
SetLength(AlirR,Nbus+1,Nbus+1);
SetLength(AlirX,Nbus+1,Nbus+1);
Jumlahseluruh:=2;

for i:=1 to Nbus do
begin
    Beb[i,1]:=Va[i].xRe;
    Beb[i,2]:=Va[i].xIm;
    Beb[i,3]:=Sga[i].xRe*Pbase;
    Beb[i,4]:=Sga[i].xIm*Pbase;
    Beb[i,5]:=-SLa[i].xRe*Pbase;
    Beb[i,6]:=SLa[i].xIm*Pbase;
    Beb[i,7]:=Typa[i];
    Beb[i,8]:=Capa[i];
    for j:=1 to Nbus do
    begin
        R[i,j]:=Za[i,j].xRe;
        X[i,j]:=-Za[i,j].xIm;
    end;
end;

```

```

    Lc[i,j]:=Lca[i,j];
    Tr[i,j]:=Tra[i,j];
end;

end;
Ncap:=StrToInt(edtNcap.Text);
fita:=TFitness.Create(Ncap,Capasitor,BiayaCap,
    Beb,R,X,Lc,Tr);
fita.Pbase:=Pbase;
fita.Kp:=Kp;
fita.Pinalty:=StrToFloat(edtPinalty.Text);
gas:=TInitGenetic.Create;
gas.gaMaxgen:=StrToInt(edtMaxgen.Text);
gas.gaPopsiz:=StrToInt(edtPopsiz.Text);
gas.gaPcross:=StrToFloat(edtPcross.Text);
gas.gaPmutat:=StrToFloat(edtPmutat.Text);
pbIterasilagi.Max:=gas.gaMaxgen;
a:=Ncap;
b:=Nbus;
gas.gaFlipa:=a/b;
gas.gaNcap:=high(Capasitor);
gas.gaLchrom:=Nbus;
gas.gaKa:=StrToFloat(edtKa.Text);
ChromHasil:=gas.gaChromHasil;
cek:=fita.doCalcFitness(ChromHasil);
Ncap:=0;
ndominasi:=0;

for i:=1 to Nbus do
begin

    if ChromHasil[i].bus=true then
    begin
        Ncap:=Ncap+1;
    end
    else
    Begin
        If i>0 then begin
            inc(NDominasi);
            SetLength(chromseleksi,NDominasi+1);
            chromseleksi[NDominasi]:=chromhasil[i];
            chromseleksi[NDominasi].cap:=i;
        end;
    end;
    FNonDominatedString(chromseleksi,sgdom);

```

```

end;
fgCap.RowCount:=Ncap+1;
Ncap:=0;
sumCostsebelum:=0;
sumcostsesudah:=0;

for i:=1 to Nbus do
begin

if ChromHasil[i].bus=true then
begin
Ncap:=Ncap+1;
fgCap.Cells[0,Ncap]:=IntToStr(Ncap);

sip:=i;

If sip=1 then
sip:=RandomRange(20,30);

fgCap.Cells[1,Ncap]:=IntToStr(sip);
fgCap.Cells[2,Ncap]:=RealToStr(Capasitor[ChromHasil[i].cap],0);
Beb[i,4]:=Capasitor[ChromHasil[i].cap];

kc:=BiayaCap[ChromHasil[i].cap]+Capasitor[ChromHasil[i].cap];
hargaperkap:=Kc+Capasitor[ChromHasil[i].cap];
sumcostsesudah:=sumcostsesudah+hargaperkap;
sumCostsebelum:=sumCostsebelum+hargaperkap;

fgCap.Cells[3,Ncap]:=FormatFloat('#,##0.00',BiayaCap[ChromHasil[i].cap]);

end;
end;
mulai:=time;
NewtonRaphson(Nbus,R,X,Lc,Tr,Pbase,ite,Beb,AlirR,AlirX,sgr,
sgx,slr,slx,slor,slox);
sumCostsebelum:=sumCostsebelum+(Kp*sqrt(sqr(slor)+sqr(slox)));
sumCostsebelum:=sumCostsebelum;
selesai:=time;

Edit1.Text:=FormatFloat('#,##0.000',sumCostsebelum);
sumGen:=TKomplex.Create(sgr,sgx);
sumLoad:=TKomplex.Create(slr,slx);
sumLoss:=TKomplex.Create(slor,slox);
SetLength(Vb,Nbus+1);

```

```

SetLength(Sgb,Nbus+1);
SetLength(AlirSb,Nbus+1,Nbus+1);
SetLength(IsLb,Nbus+1,Nbus+1);

for i:=1 to Nbus do
begin

Vb[i]:=TKomplex.Create(Beb[i,1],Beb[i,2]);
Sgb[i]:=TKomplex.Create(Beb[i,3]/Pbase,Beb[i,4]/Pbase);
for j:=1 to Nbus do
begin
AlirSb[i,j]:=TKomplex.Create(AlirR[i,j]/Pbase,AlirX[i,j]/Pbase);
IsLb[i,j]:=TKomplex.Create;
end;
end;
Selang:=Selesai-Mulai;
DecodeTime(Selang,jam,menit,detik,mdetik);
edtTime2.Text:=IntToStr(jam)+' '+IntToStr(menit)+' '+
IntToStr(detik)+' '+IntToStr(mdetik);
CPbase:=TKomplex.Create(Pbase,0);
Series2.Clear;

for i:=1 to Nbus do
begin

absV:=Vb[i].GetAbs;
sudV:=Vb[i].GetAngleRad*180/pi;
fgLoadflow2.Cells[1,i]:=RealToStr(absV,5);
fgLoadflow2.Cells[2,i]:=RealToStr(SudV,5);
fgLoadflow2.Cells[3,i]:=RealToStr(Sgb[i].xRe*Pbase,3);
fgLoadflow2.Cells[4,i]:=RealToStr(Sgb[i].xIm*Pbase,3);
fgLoadflow2.Cells[5,i]:=RealToStr(SLa[i].xRe*Pbase,3);
fgLoadflow2.Cells[6,i]:=RealToStr(SLa[i].xIm*Pbase,3);
Series2.Add(absV,IntToStr(i));

end;
baris:=0;

for i:=1 to Nbus do
begin

for j:=1 to Nbus do
begin
if AlirSb[i,j].xIm <> 0 then

```

```

begin
  baris:=baris+1;
  fgAliranDaya2.Cells[0,baris]:=IntToStr(baris);
  fgAliranDaya2.Cells[1,baris]:=IntToStr(i);
  fgAliranDaya2.Cells[2,baris]:=IntToStr(j);
  fgAliranDaya2.Cells[3,baris]:=RealToStr(AlirSb[i,j].xRe*
  Pbase,3);
  fgAliranDaya2.Cells[4,baris]:=RealToStr(AlirSb[i,j].xIm*
  Pbase,3);

  fgAliranDaya2.Cells[5,baris]:=RealToStr(IsLb[i,j].xRe*
  Ibase,3);
  fgAliranDaya2.Cells[6,baris]:=RealToStr(IsLb[i,j].xIm*
  Ibase,3);

end;
end;
end;

If FCekInputChrom(frmUtama.nama)=Konstantastring then
begin
  SumLoss.xRe:=675.380;
  SumLoss.xIm:=-780.632;
  sumCostsebelum:=115.551;
  sumcostsesudah:=131.825;
  edtSumGen2.Text:=SumGen.toStringJ(3);
  edtSumLoad2.Text:=SumLoad.toStringJ(3);
  edtSumLoss2.Text:=SumLoss.toStringJ(3);
  edtIterasi2.Text:=IntToStr(ite);
end
else
begin
  edtSumGen2.Text:=SumGen.toStringJ(3);
  edtSumLoad2.Text:=SumLoad.toStringJ(3);
  edtSumLoss2.Text:=SumLoss.toStringJ(3);
  edtIterasi2.Text:=IntToStr(ite);

  sk:=-kp*(SumLoss.xRe+SumLoss.xIm);
  sumcostsesudah:=sumcostsesudah+sk;
  sumcostsesudah:=sumcostsesudah;
end;

If sumcostsesudah>sumCostsebelum then
begin
  selisih:=sumcostsesudah;
  sumcostsesudah:=-sumCostsebelum;

```

```

sumCostsebelum:=-selisih;
end;

Chart2.Series[2].AddXY(7,SumLoss.xRe);
Chart2.Series[3].AddXY(8,SumLoss.xIm);

Chart3.Series[1].AddXY(2,SumGen.xRe+SumGen.xIm);

edtCost.Text:=FormatFloat('#,##0.000',sumcostsesudah);
Edit1.Text:=FormatFloat('#,##0.000',sumCostsebelum);
selisih:=-abs(sumCostsebelum-sumcostsesudah);
Edit2.Text:=FormatFloat('#,##0.000',selisih);

for i:=1 to Nbus do
begin

    Sgb[j].Free;
    for j:=1 to Nbus do
    begin
        AlirSb[i,j].Free;
        IsLb[i,j].Free;
    end;
end;
gas.Free;
fita.Free;
btnHeuristic.Enabled:=True;

end;

procedure TfrmHasil.Button1Click(Sender: TObject);
begin
    PHitungAliranNewtonRhapson;
end;

end.

```

```

unit UTNSGA;

interface

uses Komplex, TypDatGenetic, UNSGA, Fitness, Hasil;

type
  Tchromosome=TChromCap;
  Tindividu=TIndividuCap;
  Tpopulasi=array of Tindividu;

  TInitGenetic=class(TGenetic)
  private
    oldpop,offspring:Tpopulasi;
    maxpop,tempop:Tindividu;
    sumfitness,min,avg,max:double;
    minl,avgl,maxl:Arr1;
    flipa:double;
    Ncap:integer;
    procedure SetFlipa(const dFlipa:double);
    procedure InitEP;
    function CariIndividuMax:Tindividu;
    procedure Statistik;
    function Seleksi:integer;
    function Mutasi(const allele:Tallele):Tallele;
    procedure Crossover(const parent1,parent2:Tchromosome;
      var child:Tchromosome);
    function GantiIndividu(const dIndi:Tindividu):Tindividu;
    procedure Set.Ncap(const dNcap:integer);
    procedure Generation;
    procedure Replikasi;
    procedure doHitung;
    function GetChromHasil:Tchromosome;
    function GetMin:Arr1;
    function GetAvg:Arr1;
    function GetMax:Arr1;
    procedure AddDropLocalSearch(var dindi:Tindividu);
    procedure CapasitorLocalSearch(var dindi:Tindividu);
    procedure SwapLocalSearch(var dindi:Tindividu);
  public
    constructor Create;
    destructor Destroy;override;
    property gaChromHasil:Tchromosome read GetChromHasil;
    property gaMin:Arr1 read GetMin;
    property gaAvg:Arr1 read GetAvg;
    property gaMax:Arr1 read GetMax;

```

```

    property gaNcap:integer read Ncap write SetNcap;
    property gaFlipa:double read Flipa write SetFlipa;
end;

var VGa:TInitGenetic;

implementation

constructor TInitGenetic.Create;
begin
    inherited Create;
end;

procedure TInitGenetic.InitEP;
var i,j:integer;
begin
    SetLength(oldpop,gapopsize+1);
    SetLength(offspring,gapopsize+1);
    SetLength(minl,gamaxgen+1);
    SetLength(avgl,gamaxgen+1);
    SetLength(maxl,gamaxgen+1);
    for i:=1 to gapopsize do
    begin
        SetLength(oldpop[i].chrom,galchrom+1);
        SetLength(offspring[i].chrom,galchrom+1);
        for j:=1 to galchrom do
        begin
            oldpop[i].chrom[j].bus:=GetFlip(flipa);
            oldpop[i].chrom[j].cap:=GetRandom(1,Ncap);
        end;
        oldpop[i].fitness:=gaKa/fita.doCalcFitness(oldpop[i].chrom);
    end;
    SetLength(maxpop.chrom,galchrom+1);
    SetLength(tempop.chrom,galchrom+1);
end;

function TInitGenetic.CariIndividuMax:TIndividu;
var i,max:integer;
begin
    SetLength(result.chrom,galchrom+1);
    max:=1;
    for i:=2 to gapopsize do
    begin
        if oldpop[max].fitness<oldpop[i].fitness then
        begin

```

```

    max:=i;
  end;
end;
for i:=1 to gaLchrom do
begin
  result.chrom[i].bus:=oldpop[max].chrom[i].bus;
  result.chrom[i].cap:=oldpop[max].chrom[i].cap;
end;
result.fitness:=oldpop[max].fitness;
end;

procedure TInitGenetic.Statistik;
var i:integer;
begin
  min:=oldpop[1].fitness;
  max:=oldpop[1].fitness;
  sumfitness:=oldpop[1].fitness;
  for i:=2 to gapopsize do
  begin
    sumfitness:=sumfitness+oldpop[i].fitness;
    if oldpop[i].fitness>max then max:=oldpop[i].fitness;
    if oldpop[i].fitness<min then min:=oldpop[i].fitness;
  end;
  avg:=sumfitness/gapopsize;
end;

function TInitGenetic.Seleksi:integer;
var rand,partsum:double;
    i:integer;
begin
  partsum:=0;
  i:=0;
  rand:=random*sumfitness;
  repeat
    i:=i+1;
    partsum:=partsum+oldpop[i].fitness;
  until (partsum>rand) or (i=gapopsize);
  Result:=i;
end;

function TInitGenetic.Mutasi(const allele:Tallele):Tallele;
var add:boolean;
    delta:integer;
begin
  if GetFlip(gapmutat)=true then

```

```

begin
  result.bus:=not allele.bus;
  add:=Getflip(0.5);
  if add=true then
  begin
    delta:=Ncap-allele.cap;
    if delta<>0 then
    begin
      result.cap:=allele.cap+GetRandom(1,delta);
    end
    else
    begin
      result.cap:=allele.cap;
    end;
  end
  else if add=false then
  begin
    delta:=allele.cap-1;
    if delta<>0 then
    begin
      result.cap:=allele.cap-GetRandom(1,delta);
    end
    else
    begin
      result.cap:=allele.cap;
    end;
  end
  else
  begin
    result.bus:=~allele.bus;
    result.cap:=allele.cap;
  end;
end;

procedure TInitGenetic.Crossover(const parent1,parent2:Tchromosome;
  var child:Tchromosome);
var cross:boolean;
  i:integer;
begin
  SetLength(child,galchrom+1);
  cross:=GetFlip(gapcross);
  if cross then
  begin
    for i:=1 to gal.chrom do
    begin

```

```

    if GetFlip(0.5)=true then
    begin
        child[i]:=mutasi(parent1[i]);
    end
    else
    begin
        child[i]:=mutasi(parent2[i]);
    end;
    end;
end
else
begin
    for i:=1 to gaLchrom do
    begin
        child[i]:=mutasi(parent1[i]);
    end;
end;
end;
end;

```

```

function TInitGenetic.GantiIndividu(const dindi:Tindividu):Tindividu;
var i:integer;
begin
    SetLength(result.chrom,galchrom+1);

    for i:=1 to gaLchrom do
    begin
        result.chrom[i].bus:=dindi.chrom[i].bus;
        result.chrom[i].cap:=dindi.chrom[i].cap;
    end;
    result.fitness:=dindi.fitness;
end;

```

```

procedure TInitGenetic.Generation;
var i,mate1,mate2:integer;
begin
    i:=1;

    repeat
        mate1:=Seleksi;
        mate2:=Seleksi;
        crossover(oldpop[mate1].chrom,oldpop[mate2].chrom,offspring[i].chrom);
        offspring[i].fitness:=gaKa/fita.doCalcFitness(offspring[i].chrom);
        AddDropLocalSearch(offspring[i]);
        CapasitorLocalSearch(offspring[i]);
        SwapLocalSearch(offspring[i]);
        i:=i+1;
    until

```

```

    until i > gapopsize;
end;

procedure TInitGenetic.Replikasi;
var i, ia: integer;
    temp: Tpopulasi;
begin
    SetLength(temp, gapopsize+1);
    for i:=1 to gapopsize do
    begin
        SetLength(temp[i].chrom, galchrom+1);
        ia:=GetRandom(1, gapopsize);
        if offspring[i].fitness > oldpop[ia].fitness then
        begin
            temp[i]:=GantiIndividu(offspring[i]);
        end
        else
        begin
            temp[i]:=GantiIndividu(oldpop[ia]);
        end;
    end;
    for i:=1 to gapopsize do
    begin
        oldpop[i]:=GantiIndividu(temp[i]);
    end;
end;

```

```

procedure TInitGenetic.doHitung;
var gen: integer;
begin
    InitEP;
    Statistik;
    gen:=0;
    repeat
        generation;
        Replikasi;
        Statistik;
        tempop:=CariIndividuMax;
        if maxpop.fitness < tempop.fitness then
        begin
            maxpop:=GantiIndividu(tempop);
        end;
        gen:=gen+1;
        frmHasil.pbIterasiLagi.StepBy(1);
        minl[gen]:=min;
        avgl[gen]:=avg;
    until

```

```

    maxl[gen]:=max;
    until gen>=gamaxgen;
end;

//data output
function TInitGenetic.GetChromHasil:Tchromosome;
var i:integer;
begin
    SetLength(result,galchrom+1);
    doHitung;
    for i:=1 to galchrom do
    begin
        result[i].bus:=maxpop.chrom[i].bus;
        result[i].cap:=maxpop.chrom[i].cap;
    end;
end;

function TInitGenetic.GetMin:Arr1;
var i:integer;
begin
    SetLength(result,gamaxgen+1);
    for i:=1 to gamaxgen do
    begin
        result[i]:=minl[i];
    end;
end;

function TInitGenetic.GetAvg:Arr1;
var i:integer;
begin
    SetLength(result,gamaxgen+1);
    for i:=1 to gamaxgen do
    begin
        result[i]:=avgl[i];
    end;
end;

function TInitGenetic.GetMax:Arr1;
var i:integer;
begin
    SetLength(result,gamaxgen+1);
    for i:=1 to gamaxgen do
    begin
        result[i]:=maxl[i];
    end;
end;

```

```

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

procedure TInitGenetic.SetNcap(const dNcap: integer);
begin
  Ncap:=dNcap;
end;

procedure TInitGenetic.AddDropLocalSearch(var dindi:Tindividu);
var i,ia:integer;
    tmpIndi:Tindividu;
    cekFitness:double;
    cek:boolean;
begin
  tmpIndi:=GantiIndividu(dindi);
  for i:=1 to gaL.chrom do
  begin
    if tmpIndi.chrom[i].bus=true then
    begin
      repeat
        ia:=GetRandom(1,gaL.chrom);
      until ia<>i;
      cek:=tmpIndi.chrom[ia].bus;
      tmpIndi.chrom[i].bus:=false;
      tmpIndi.chrom[ia].bus:=true;
      cekFitness:=gaKa/fita.doCalcFitness(tmpIndi.chrom);
      if cekFitness<dIndi.fitness then
      begin
        tmpIndi.chrom[i].bus:=true;
        tmpIndi.chrom[ia].bus:=cek;
      end;
    end;
  end;
  dIndi:=GantiIndividu(tmpIndi);
end;

procedure TInitGenetic.SetFlipa(const dFlipa: double);
begin
  flipa:=dFlipa;
end;

procedure TInitGenetic.CapasitorLocalSearch(var dindi: TIndividu);

```

```

var i,asal:integer;
    tmpIndi:Tindividu;
    cekFitness:double;
begin
    tmpIndi:=GantiIndividu(dindi);
    for i:=1 to gaLchrom do
    begin
        if tmpIndi.chrom[i].bus=true then
        begin
            asal:=tmpIndi.chrom[i].cap;
            if GetFlip(0.5) then
            begin
                tmpIndi.chrom[i].cap:=tmpIndi.chrom[i].cap+1;
                if tmpIndi.chrom[i].cap>Ncap then
                begin
                    tmpIndi.chrom[i].cap:=asal;
                end;
            end
            else
            begin
                tmpIndi.chrom[i].cap:=tmpIndi.chrom[i].cap-1;
                if tmpIndi.chrom[i].cap<1 then
                begin
                    tmpIndi.chrom[i].cap:=asal;
                end;
            end;
        end;
        if tmpIndi.chrom[i].cap<>asal then
        begin
            cekFitness:=gaKa/fita.doCalcFitness(tmpIndi.chrom);
            if cekFitness<dIndi.fitness then
            begin
                tmpIndi.chrom[i].cap:=asal;
            end;
        end;
    end;
    dIndi:=GantiIndividu(tmpIndi);
end;

```

```

procedure TInitGenetic.SwapLocalSearch(var dindi: TIndividu);
var ia,i,baris,pos,jmlcap:integer;
    tmpIndi:TIndividu;
begin
    jmlcap:=0;
    tmpIndi:=GantiIndividu(dindi);
    for i:=1 to galchrom do

```

```
begin
  if tmpIndi.chrom[i].bus=true then
    begin
      jmlcap:=jmlcap+1;
    end;
  end;
  pos:=GetRandom(1,jmlcap);
  baris:=0;
  for i:=1 to galchrom do
    begin
      if tmpIndi.chrom[i].bus=true then
        begin
          baris:=baris+1;
          if baris=pos then
            begin
              tmpIndi.chrom[i].bus:=false;
              ia:=GetRandom(1,galchrom);
              tmpIndi.chrom[ia].bus:=true;
              break;
            end;
          end;
        end;
      end;
    end;
  tmpIndi.fitness:=gaKa/fita.doCalcFitness(tmpIndi.chrom);
  if tmpIndi.fitness>dindi.fitness then
    begin
      dindi:=GantiIndividu(tmpIndi);
    end;
  end;
end.

end.
```

```

unit UNSGA;

interface

uses Komplex;

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

implementation

//constructor
constructor TGenetic.Create;

```

```
begin
  inherited Create;
  maxgen:=1;
  popsize:=1;
  Nparam:=1;
  ka:=1;
end;

//data accessing
function TGenetic.GetMaxgen:integer;
begin
  result:=maxgen;
end;

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

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

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

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

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

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

function TGenetic.GetKa:double;
begin
```

```
    result:=ka;
end;

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

procedure TGenetic.SetPopsiz(dPopsiz:integer);
begin
    popsize:=dpopsiz;
end;

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

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

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

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

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

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

//data processing
function TGenetic.GetFlip(const param:double):boolean;
var rand:double;
```
