

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



**ANALISIS PENEMPATAN KAPASITOR PADA JARINGAN
DISTRIBUSI PRIMER DENGAN MENGGUNAKAN KOMBINASI
METODE DETERMINISTIC DAN ALGORITMA GENETIKA DI GI
SENGKALING PENYULANG PUJON**

SKRIPSI

Disusun oleh :

**RACHMA FITRI
NIM : 99.12.117**

APRIL 2005



Lembar Persembahan

BISMILLAAHIRRAHMANIRRAHIIM

Dengan kerendahan hati sebagai ungkapan rasa hormat dan syukur, ku persembahkan tugas akhir ini kepada yang tercinta

ALLAH SWT & NABI MUHAMMAD SAW, Atas segala rahmat dan karunia yang Engkau berikan kepadaku. Bimbingan dan restu-Mulah yang mampu menghantarkan hamba-Mu ini dalam menapail kesuksesan di dunia dan di akhirat kelak.

BAPAK & IBU (Tony Nuralianto & Mujati Rafatni), Terima kasih atas semua kasih sayang dan dukungannya selama ini.

ALL MY FAMILY, kakakku Fly dan adikku Raka & Radar, Budhe, Yossy serta semua keluargaku, juga Mak 'Un buat sarapan dan air angetnya

MY FELIX, terkyu banget dah jadi sopir paling setia dan jadi tempat buat aku marah, (s@ _ sorry cayank...)

MY 2nd FAMILY, Bapak Slamet Harjadi & Ibu Yayuk Sulistiyowati, juga mbak Siska

ALL MY PRENS, teman-temenku ELEKTRO ST Widya, Ina, Ayu, Diana (thanx 4 your support guy's), Jantur, Dwi smock, A Cong, Hanafi, Dedy, Roby, Kevin, Made, Ipunk, Eko, Sulik, Herman, Yadi, (finally we finish sir!). Tono, Ardi, Pam, Doni, Dwi, Adjan, Boy, Peter, Antok, Imam (kapan nich....). Semua teman-temenku ST '99 kls 12345 yang gax bisa tak sebut satu-satu (gax cukup kertas) ,makasih semua dukungannya rek!

TEMEN-TEMEN SEPERJUANGAN, Tita, Lia, Agus, Mas Adi, Mas Makhrus, Udin, Ari, dan semuanya (udah lega ya....). Rully, Edy, Dll (don't give up!)

MY BROTHERS, Mas Vicky '97 & Mas Eka '98, buat contoh skripsinya terkyu puoooll...!

UGRO'S FAMILY, makasih udah mau direpotin bikin program sampe bosan liat Fitri

Ir. I MADE WARTANA, MT, makasih banyak Pak atas bimbingannya (walaupun antre bimbingannya panjang banget tapi skripsi saya akhirnya keluar juga)

DOSEN DAN STAF ITN, Ir. Almizan A. MSEE (Dosen Wali), Ir. H. Taufik Hidayat, MT & Ir. Widodo Pudji M. MT (Dosen penguji), Bu Puji, Bu Mimin, Mas Jayenk (makasih udah mau beleputan). Terima kasih banyak kepada semua dosen dan staf ITN atas bantuan dan dukungannya

t
h
a
n
k
x
2

KATA PENGANTAR

Dengan mengucapkan puji syukur kehadirat Allah SWT, atas rahmat dan karunia yang dilimpahkan-Nya sehingga saya dapat menyelesaikan skripsi yang berjudul, "*Analisis Penempatan Kapasitor Pada Jaringan Distribusi Primer Dengan Menggunakan Kombinasi Metode Deterministic dan Algoritma Genetika di GI Sengkaling Penyulang Pujon*".

Skripsi ini bertujuan untuk memenuhi kurikulum akademik yang harus ditempuh oleh setiap mahasiswa ITN Malang guna mengakhiri pendidikan pada jenjang strata satu jurusan Teknik Elektro Konsentrasi Teknik Energi Listrik di Institut Teknologi Nasional Malang.

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

Atas segala bimbingan dan bantuan yang diberikan sehingga skripsi ini dapat tersusun, maka saya menyampaikan terima kasih kepada :

1. **Dr. Ir. Abraham Lomi, MSEE**, selaku Rektor ITN Malang
2. **Ir. Mochtar Asroni, MSME**, selaku Dekan FTI ITN Malang
3. **Ir. F. Yudi Limpraptono, MT**, selaku Ketua Jurusan Teknik Elektro (S1) ITN Malang
4. **Ir. I Made Wartana, MT**, selaku Dosen Pembimbing
5. Semua Dosen dan Staf, khususnya untuk jurusan Teknik Elektro (S1) ITN Malang
6. Rekan-rekan Teknik Elektro Energi Listrik ITN Malang

Saya menggharap skripsi ini dapat berguna dan bermanfaat bagi rekan-rekan mahasiswa khususnya pada jurusan Teknik Elektro Energi Listrik ITN Malang.

Malang, Maret 2005

Penyusun

DAFTAR ISI

Halaman :

HALAMAN JUDUL	i
LEMBAR PERSETUJUAN	ii
ABSTRAKSI	iii
KATA PENGANTAR	iv
DAFTAR ISI	vi
DAFTAR GAMBAR	x
DAFTAR GRAFIK	xii
DAFTAR TABEL	xiii
BAB I PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	2
1.3. Tujuan.....	2
1.4. Batasan Masalah.....	3
1.5. Metodologi Pembahasan.....	4
1.6. Sistematika Penulisan.....	5
1.7. Relevansi.....	6
BAB II LANDASAN TEORI	
2.1. Sistem Distribusi Tenaga Listrik.....	7
2.2. Struktur Jaringan Distribusi Tenaga Listrik.....	8
2.3. Sistem Jaringan Distribusi Radial.....	8

2.3.1.	Sistem Radial Pohon	10
3.3.1.	Sistem Radial dengan <i>Tie</i> dan <i>Switch</i> Pemisah	10
2.3.3.	Sistem Radial dengan Pembagian <i>Phase Area</i>	11
2.3.4.	Sistem Radial Dengan Beban Terpusat.....	12
2.4.	Faktor Daya.....	13
2.5.	Kapasitor Daya.....	15
2.6.	Kapasitor Seri dan Kapasitor Shunt.....	15
2.6.1	Pengaruh Pemasangan Kapasitor Shunt.....	16
2.6.2.	Pengurangan Rugi-Rugi dengan Kapasitor Shunt.....	17
2.7.	Perbaikan Tegangan	18
2.8.	Perbaikan Faktor Daya dan Kenaikan Kapasitas Sistem.....	19
2.9.	Perhitungan Pengaruh Perbaikan Faktor Daya.....	21
2.10.	Penentuan Rating Kapasitor Untuk Perbaikan Faktor Daya Beban	22
2.11.	Sistem Per-Unit	23

BAB III ANALISIS PENEMPATAN KAPASITOR PADA JARINGAN DISTRIBUSI PRIMER DENGAN MENGGUNAKAN KOMBINASI METODE *DETERMINISTIC* DAN *ALGORITMA GENETIKA*

3.1.	Analisa Aliran Daya Pada Jaringan.....	25
3.1.1.	Tujuan	25
3.1.2.	Metode <i>Newton Raphson</i>	26
3.1.3.	Algoritma Aliran Daya <i>Newton Raphson</i>	30
3.2.	Kombinasi Metode <i>Deterministic</i> dan <i>Algoritma Genetika</i>	31
3.2.1.	Pengkodean	32

3.2.2.	Menentukan Populasi Awal.....	32
3.2.3.	Seleksi Individu untuk Rekombinasi	33
3.2.4.	Rekombinasi.....	35
3.2.5.	Fungsi Obyektif.....	35
3.3.	Algoritma Program	38
3.3.1.	Algoritma Pemecahan Masalah Penempatan Kapasitor Pada Jaringan Distribusi 20 kV Dengan Menggunakan Metode <i>Deterministic</i> dan <i>Algoritma Genetika</i> Di GI Sengkaling Penyulang Pujon.....	38
3.3.2.	Algoritma Program <i>Fitness</i>	39

BAB IV STUDI ANALISIS PENEMPATAN KAPASITOR PADA JARINGAN DISTRIBUSI PRIMER DENGAN MENGGUNAKAN KOMBINASI METODE *DETERMINISTIC* DAN *ALGORITMA GENETIKA* DI GI SENGKALING PENYULANG PUJON

4.1.	Penempatan Kapasitor Menggunakan Kombinasi Metode <i>Deterministic</i> dan <i>Algoritma Genetika</i>	40
4.2.	<i>Flowchart</i> Algoritma Program	41
4.2.1.	<i>Flowchart</i> Algoritma Pemecahan Masalah Penempatan Kapasitor Pada Jaringan Distribusi 20 kV Di GI Sengkaling Penyulang Pujon.....	41
4.2.2.	<i>Flowchart Sub Routine</i> Program <i>Deterministic</i>	42
4.2.2.	<i>Flowchart</i> Algoritma Program <i>Fitness</i>	43
4.3.	Data Penyulang Pujon	45
4.3.1.	Data Saluran	45
4.3.2.	Pembebanan Penyulang Pujon	46
4.4.	Analisa Perhitungan	47
4.5.	Tampilan Program.....	48
4.6.	Hasil Perhitungan	54

BAB V KESIMPULAN DAN SARAN

5.1. Kesimpulan.....	61
5.2. Saran.....	62

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar :	Halaman :
2-1. Diagram satu garis penyaluran energi listrik ke pelanggan.....	7
2-2. Contoh Sistem Jaringan Distribusi Radial	9
2-3. Contoh Sistem Jaringan Distribusi Radial Dengan <i>Tie dan Switch Pemisah</i>	11
2-4. Contoh Sistem Jaringan Distribusi Radial Dengan <i>Phase Arca</i>	12
2-5. Contoh Sistem Jaringan Distribusi Radial Dengan <i>Beban Terpusat</i>	13
2-6. Segitiga Daya	13
2-7. Segitiga Arus.....	14
2-8. Saluran Primer Dengan <i>Beban Terpusat</i>	17
2-9. Vektor Diagram Sebelum dan Sesudah Pemasangan Kapasitor <i>Shunt Pada Jaringan</i>	18
2-10. Perbaikan Faktor Daya	19
2-11. Diagram Fasor dan Sudut Daya <i>Beban Distribusi</i>	21
3-1. Pengkodean <i>Kromosom</i>	32
3-2. Pembentukan <i>Next Generation</i> dalam <i>Deterministic</i>	34
3-3. Operator <i>Rekombinasi</i>	36
4-1. Flowchart Program Pemecahan Masalah	41
4-2. Flowchart Program <i>Deterministic</i>	42
4-3. Flowchart Sub Routine <i>Deterministic Program Fitness</i>	43

4-4.	<i>Single Line</i> Diagram Sebelum Penempatan Kapasitor	44
4-5.	Tampilan Utama Program	48
4-6.	Tampilan Inputan Data	49
4-7.	Tampilan Data Saluran	49
4-8.	Tampilan Data Pembebanan	50
4-9.	Tampilan Data Kapasitor	50
4-10.	Tampilan <i>Loadflow</i> Awal	51
4-11.	Tampilan Aliran Daya Sebelum Penempatan Kapasitor.....	51
4-12.	Tampilan Parameter <i>Deterministic</i>	52
4-13.	Tampilan Hasil Penentuan Lokasi dan Kapasitas Kapasitor	52
4-14.	Tampilan Hasil <i>Loadflow</i> Akhir	53
4-15.	Tampilan Aliran Daya Setelah Penempatan Kapasitor.....	53
4-16.	Tampilan Hasil Perhitungan Metode <i>Deterministic</i>	56
4-17.	<i>Single Line</i> Diagram Setelah Penempatan Kapasitor.....	60

DAFTAR GRAFIK

Grafik :	Halaman :
4-1. Tegangan Tiap-Tiap <i>Node</i> Terhadap Asumsi Tegangan Awal Sebelum dan Sesudah Kompensasi.....	58
4-2. Rugi Daya Sebelum dan Sesudah Kompensasi	59

DAFTAR TABEL

Tabel :	Halaman :
4-1. Data Penyulang Sistem 20 kV GI Sengkaling.....	45
4-2. Data Saluran Penyulang Pujon.....	46
4-3. Data Pembebanan Penyulang Pujon.....	46
4-4. Data Kapasitor.....	47
4-5. Kondisi Tegangan dan Sudut Fasa Tegangan Tiap <i>Node</i> Penyulang Pujon Sebelum Kompensasi.....	54
4-6. Besar Rugi-Rugi Daya Tiap Saluran Penyulang Pujon Sebelum Kompensasi.....	54
4-7. Besar Aliran Daya Tiap Saluran Penyulang Pujon Sebelum Kompensasi.....	55
4-8. Kondisi Tegangan dan Sudut Fasa Tegangan Tiap <i>Node</i> Penyulang Pujon Sesudah Kompensasi.....	56
4-9. Besar Rugi-Rugi Daya Tiap Saluran Penyulang Pujon Sesudah Kompensasi.....	57
4-10. Besar Aliran Daya Tiap Saluran Penyulang Pujon Sesudah Kompensasi.....	57

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 induktif. 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.^[1]

Alternatif yang sering dipakai untuk memperbaiki kondisi tegangan akibat adanya rugi-rugi tersebut adalah dengan memasang sumber daya reaktif tambahan di sisi beban, salah satunya adalah dengan pemasangan kapasitor.^[1]

Ada banyak metode yang sebelumnya digunakan untuk memecahkan masalah optimalisasi penempatan kapasitor, diantaranya metode *Genetika Algoritma*, *Tabu Search*, *Fuzzy Logic* dan lain-lain.^[6]

Pada skripsi ini akan dibahas teknik untuk menempatkan kapasitor pada jaringan distribusi dengan menggunakan *Metode Deterministic* yang berdasar pada *Metode Genetika Algoritma*.^[4]

1.2. Rumusan Masalah

Bertambahnya usaha-usaha di sektor 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 pemasangan kapasitor.^[1]

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.^[1]

Sehubungan dengan permasalahan diatas maka skripsi ini mengambil judul :

**“ANALISIS PENEMPATAN KAPASITOR PADA JARINGAN
DISTRIBUSI PRIMER DENGAN MENGGUNAKAN KOMBINASI
METODE DETERMINISTIC DAN ALGORITMA GENETIKA DIGI
SENGKALING PENYULANG PUJON ”**

1.3. Tujuan Pembahasan

Skripsi ini bertujuan untuk :

1. Menentukan jumlah dan lokasi penempatan kapasitor yang optimal serta besar kapasitasnya.
 2. Meningkatkan profil tegangan pada penyulang Pujon
-

3. Mereduksi besar rugi-rugi daya pada penyulang Pujon.
4. Memaksimalkan total keuntungan (penghematan biaya) yang diperoleh setelah pemasangan kapasitor.

1.4. Batasan Masalah

Permasalahan dalam sistem tenaga listrik adalah merupakan permasalahan yang luas, sehingga dalam menganalisa suatu permasalahan perlu diadakan pembatasan-pembatasan yang sesuai dengan permasalahan tersebut.

Di dalam penulisan skripsi ini pembatasan yang dilakukan adalah sebagai berikut :

1. Jaringan yang akan dianalisa adalah jaringan distribusi primer tipe radial 20 kV di Penyulang Pujon, Gardu Induk Sengkaling.
 2. Menggunakan kombinasi Metode Deterministic dan Algoritma Genetika dalam menentukan lokasi, jumlah dan kapasitas kapasitor shunt pada jaringan distribusi primer.
 3. Metode yang digunakan dalam perhitungan aliran daya adalah metoda Newton Raphson.
 4. Tidak membahas metoda Newton Raphson secara mendetail.
-

1.5. Metodologi Pembahasan

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

1. Studi Literatur

Yaitu suatu kajian pustaka dengan mempelajari teori-teori yang terkait melalui literatur yang ada, yang berhubungan dengan permasalahan.

2. Pengumpulan data lapangan yang dipakai dalam objek penelitian yakni pada Gardu Induk Sengkaling.

- a. Data impedansi saluran jaringan distribusi penyulang Pujon.
- b. Data pembebanan.

3. Melakukan analisis dengan menggunakan bahasa program Delphi versi 7.0

- a. Analisa aliran daya menggunakan metode Newton Raphson.
 - b. Analisis penentuan lokasi dan kapasitas kapasitor shunt menggunakan kombinasi metode Deterministic dan Algoritma Genetika.
 - c. Analisis pengaruh pemasangan kapasitor terhadap sistem diantaranya perbaikan profil tegangan, pengurangan rugi-rugi saluran, penghematan biaya / keuntungan yang diperoleh.
-

1.6. Sistematika Penulisan

Adapun sistematika penulisan pada skripsi ini adalah sebagai berikut :

BAB I : PENDAHULUAN

Pada bab ini berisi latar belakang, rumusan masalah, tujuan yang ingin dicapai, batasan masalah, metodologi penulisan, sistematika penulisan dan relevansi.

BAB II : LANDASAN TEORI

Disini akan diuraikan mengenai sistem jaringan distribusi, faktor daya, penjelasan teori tentang kapasitor, perbaikan tegangan, perbaikan faktor daya dan kenaikan kapasitas sistem, serta sistem per unit.

BAB III : ANALISIS PENEMPATAN KAPASITOR PADA JARINGAN DISTRIBUSI PRIMER DENGAN MENGGUNAKAN KOMBINASI METODE DETERMINISTIC DAN ALGORITMA GENETIKA DI GI SENGKALING PENYULANG PUJON

Menguraikan metode aliran daya *Newton Raphson*, kombinasi metode *Deterministic* dan *Algoritma Genetika*, serta Algoritma Program Pemecahan Masalah.

BAB IV : STUDI PENEMPATAN KAPASITOR PADA JARINGAN DISTRIBUSI 20 kV

Bab ini akan dibahas mengenai penentuan letak dan kapasitas kapasitor, analisa perhitungan dengan menggunakan kombinasi

Metode *Deterministic* dan *Algoritma Genetika*, profil tegangan dan aliran daya sebelum dan sesudah kompensasi.

BAB V : KESIMPULAN

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

1.7. Relevansi

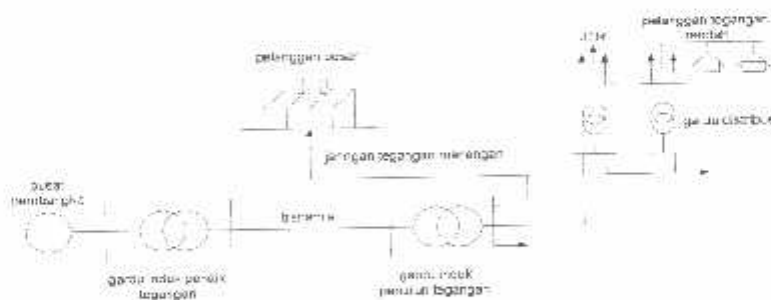
Dengan diperkenalkannya metode *Deterministic* yang dikombinasikan dengan *Algoritma Genetika* diharapkan akan memberikan alternatif kepada pihak PLN agar dapat mengidentifikasi solusi yang optimal untuk penempatan kapasitor pada sebuah operasi sistem daya dengan maksud untuk memperbaiki pelayanan pada masyarakat dengan memanfaatkan jaringan yang ada dengan tidak mengurangi keuntungan yang diperoleh oleh PLN.

BAB II LANDASAN TEORI

2.1. Sistem Distribusi Tenaga Listrik^[1]

Di Indonesia, tenaga listrik dibangkitkan di pusat-pusat pembangkit tenaga listrik, seperti PLTA, PLTU, PLTG, PLTGU, PLTP dan PLTD yang kemudian disalurkan melalui saluran transmisi setelah terlebih dahulu dinaikkan tegangannya oleh transformator penaik tegangan (*step up transformer*) yang ada dipusat listrik. Saluran transmisi tegangan tinggi di PLN mempunyai tegangan menengah sebesar 20 kV di Gardu Induk, dengan menggunakan transformator penurun tegangan (*step down transformer*), setelah melalui jaringan distribusi, tegangan diturunkan lagi pada gardu-gardu distribusi menjadi tegangan rendah sebesar 380/220 V untuk selanjutnya disalurkan ke pelanggan-pelanggan PLN.

Sebagai gambaran, diagram satu garis sistem tenaga listrik dapat diperjelas seperti gambar 2.1 dibawah ini ^[1]:



Gambar 2-1. Diagram satu garis penyaluran energi listrik ke pelanggan
Sumber: Hasan Basri, "Sistem Distribusi Tenaga Listrik"

Jaringan setelah keluar dari G.I. biasa disebut jaringan distribusi. Jaringan distribusi dapat diklasifikasikan menjadi dua bagian sistem, yaitu :

1. Sistem distribusi primer atau sistem distribusi tegangan menengah.
2. Sistem distribusi sekunder atau sistem distribusi tegangan rendah.

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

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

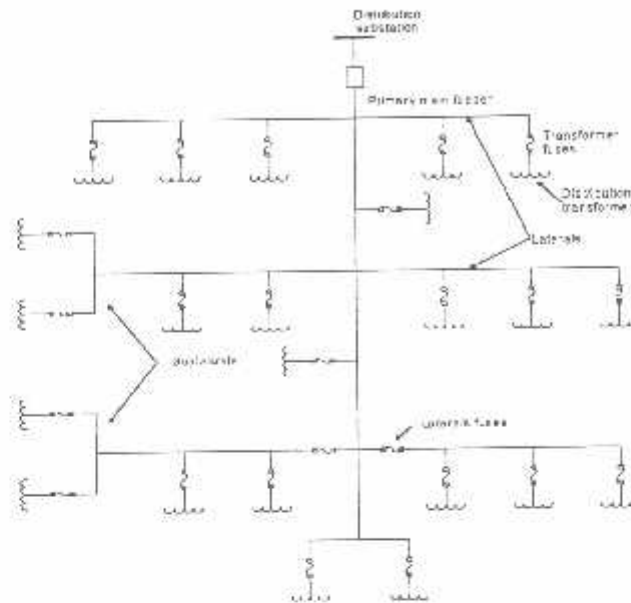
2.2. Struktur Jaringan Distribusi Tenaga Listrik^[7]

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^[7]

Bentuk jaringan ini merupakan bentuk dasar yang paling sederhana dan paling banyak digunakan. Sistem ini dikatakan radial karena dari kenyataan bahwa jaringan ini ditarik secara radial dari GI ke pusat-pusat beban/konsumen yang dilayaninya. Sistem ini terdiri dari saluran utama (*main feeder*) dan saluran cabang (*lateral*) seperti pada gambar 2-2.



Gambar 2-2. Contoh Sistem Jaringan Distribusi Radial

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

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.

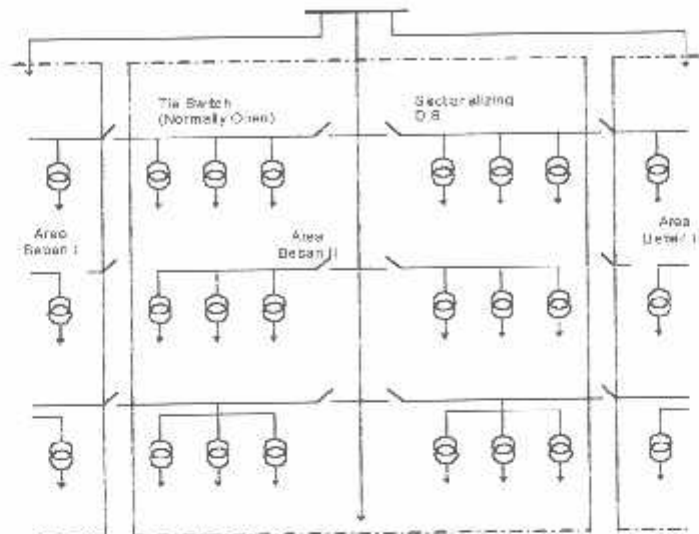
Kelemahan yang dimiliki oleh sistem radial adalah jatuh tegangan yang cukup besar dan bila terjadi gangguan pada sistem akan dapat mengakibatkan jatuhnya sebagian atau bahkan keseluruhan beban sistem.

2.3.1. Sistem Radial Pohon^[7]

Sistem radial jaringan pohon ini merupakan bentuk yang paling dasar dari sistem jaringan radial. Saluran utama (*main feeder*) di tarik dari suatu Gardu Induk sesuai dengan kebutuhan kemudian di cabangkan melalui saluran cabang (*lateral feeder*), selanjutnya di cabangkan lagi melalui saluran anak cabang (*sub-lateral feeder*). Ukuran dari masing-masing saluran tergantung dari kerapatan arus yang di tanggung. 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^[7]

Sistem ini merupakan pengembangan dari sistem radial pohon, untuk meningkatkan keandalan sistem saat terjadinya gangguan maka *feeder* yang terganggu akan di lokalisir 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.

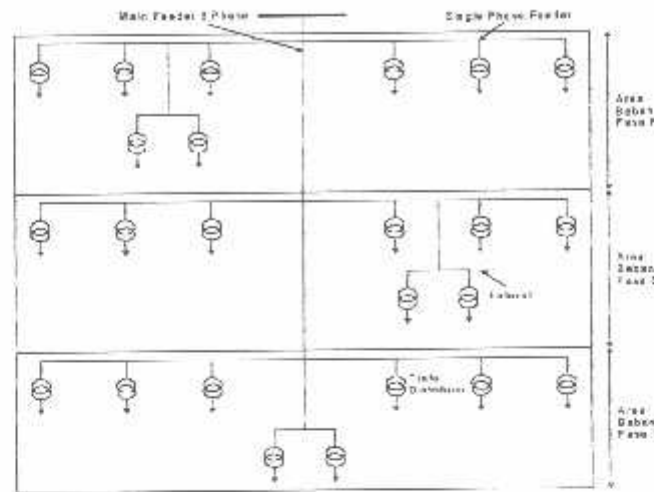


Gambar 2-3. Contoh Sistem Jaringan Distribusi Radial Dengan Tie dan Switch Pemisah

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

2.3.3. Sistem Radial dengan Pembagian Phase Area^[7]

Pada bentuk ini masing-masing fasa dari jaringan bertugas melayani daerah beban 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.



Gambar 2-4. Contoh Jaringan Distribusi Radial Dengan Phase Area

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

2.3.4. Sistem Radial Dengan Beban Terpusat^[7]

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

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

Dimana:

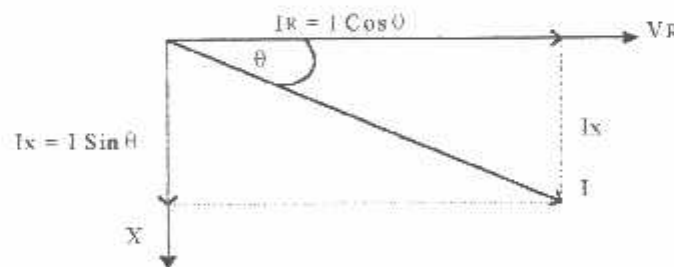
P – Daya nyata (kW)

S = Daya semu (kVA)

Q = Daya reaktif (kVAR)

ϕ = Sudut phase

Faktor daya dapat pula dilihat hubungannya dengan arus nyata dan 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 dan arus nyata. 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 :



Gambar 2-7. Segitiga Arus

Sumber : Hasan Basri, "Sistem Distribusi Tenaga Listrik"

2.5. Kapasitor Daya^[1]

Secara sederhana kapasitor terdiri dari dua buah plat 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 plat 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.

2.6. Kapasitor Seri dan Kapasitor Shunt^[1]

Fungsi utama dari pemakaian kapasitor seri atau kapasitor shunt, dalam sistem tenaga adalah untuk membangkitkan daya reaktif, untuk memperbaiki faktor daya dan tegangan, sehingga meningkatkan kapasitas sistem dan mengurangi rugi daya jaringan.

Ada beberapa aspek yang tidak menyenangkan dalam kapasitor seri, secara umum biaya pemasangan kapasitor seri lebih tinggi dari pada biaya pemasangan kapasitor shunt, hal ini disebabkan karena peralatan pelindung untuk kapasitor seri lebih banyak, biasanya kapasitor seri didesain untuk daya yang lebih besar daripada kapasitor shunt dengan tujuan untuk mengatasi kenaikan beban.

2.6.1. Pengaruh Pemasangan Kapasitor Shunt ^[1]

Kapasitor shunt adalah kapasitor yang dihubungkan paralel dengan saluran dan secara intensip digunakan pada sistim 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 kompensasi 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 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.

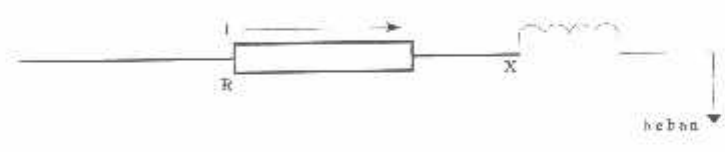
2.6.2. Pengurangan rugi daya dengan Kapasitor Shunt ⁽¹⁾

Rugi-rugi saluran per fasa dari saluran 3 fasa seimbang dengan beban terpusat seperti pada gambar 2-5 adalah $I^2 (R-jX)$ atau dapat dibedakan menjadi:

- Rugi daya aktif (I^2R) = $(I_k^2 + I_x^2)R$
- Rugi daya reaktif (I^2X) = $(I_k^2 + I_x^2)X$

Dimana : I_k adalah komponen arus aktif

I_x adalah komponen arus reaktif



Gambar 2-8. Saluran Primer Dengan Beban Terpusat

Sumber : Hasan Basri, "Sistem Distribusi Tenaga Listrik"

Rugi daya (I^2R) dapat dibagi menjadi dua komponen yaitu komponen arus aktif dan komponen arus reaktif

Rugi daya karena komponen arus aktif tidak akan mempengaruhi penempatan kapasitor shunt pada saluran hal ini dapat dijelaskan sebagai berikut :

Diasumsikan bahwa 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 daya I_1^2R 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 :

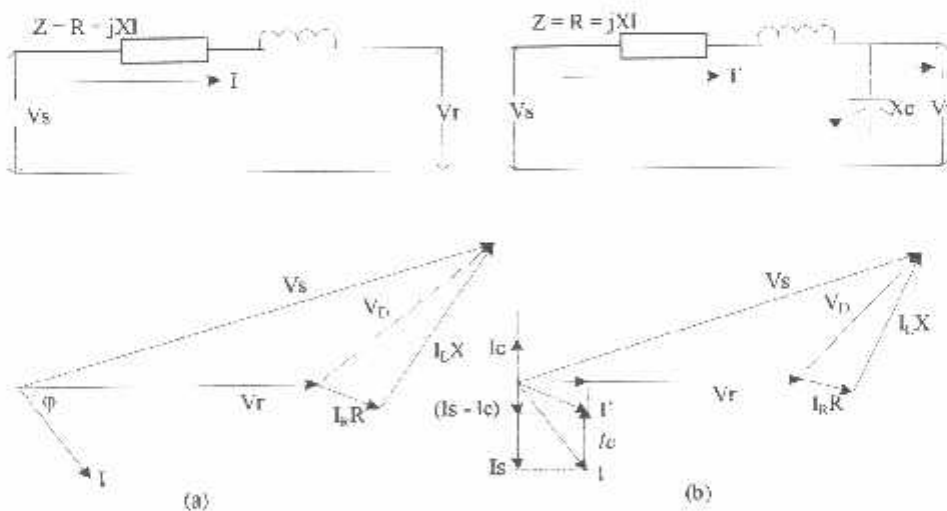
$$\begin{aligned}\Delta p_{ls} &= I^2 R - I_1^2 R \\ &= (I \cos \theta)^2 R + (I \sin \theta)^2 R - (I \cos \theta)^2 R + (I \sin \theta - I_c)^2 R \\ &= 2(I \sin \theta) I_c R - I_c^2 R \dots \dots \dots (2.4)\end{aligned}$$

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. Pengurangan rugi daya saluran 3 fasa adalah :

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

2.7. Perbaikan Tegangan ^[1]

Pemakaian kapasitor shunt dalam sistem tenaga listrik selain untuk perbaikan faktor daya juga bertujuan menaikkan tegangan. Dan secara vektoris dapat digambarkan sebagai berikut :



Gambar 2-9. Vektor Diagram Sebelum (a) dan Sesudah (b) Pemasangan Kapasitor Shunt Pada Jaringan

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

Jatuh tegangan yang disbabkan arus beban I sebelum kapasitor dipasang :

$$\delta V = I_R R + I_L X \dots\dots\dots (2.6)$$

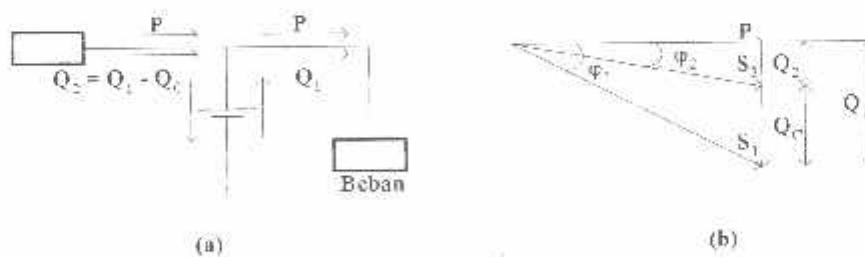
Jatuh tegangan setelah kapasitor dipasang :

$$\delta V = I_R R + I_L X - I_C X \dots\dots\dots (2.7)$$

2.8. Perbaikan Faktor Daya dan Kenaikan Kapasitas Sistem^[1]

Manfaat terbesar yang diperoleh dari perbaikan faktor daya berasal dari pengurangan daya reaktif dalam sistem. Hal ini menghasilkan pengurangan biaya pemakaian daya yang lebih rendah, kenaikan kapasitas sistem, perbaikan tegangan dan pengurangan rugi daya dalam sistem. Satu – satunya jalan untuk memperbaiki faktor daya adalah mengurangi daya reaktif di jaringan. Jika komponen daya reaktif dapat di kurangi, maka total arus akan berkurang, sedang komponen daya aktif tidak berubah, maka faktor daya akan lebih besar sebagai akibat berkurangnya daya reaktif.

Dengan menambahkan kapasitor, daya reaktif komponen Q akan berkurang, gambar 2-10 menunjukkan perbaikan faktor daya pada sistem, kapasitor mensuplay daya reaktif ke beban.



Gambar 2-10. Perbaikan Faktor Daya
Sumber : Hasan Basri, "Sistem Distribusi Tenaga Listrik".

Diasumsikan bahwa beban disupply oleh daya nyata P, daya reaktif (*lagging*) Q, dan daya semu S, pada faktor daya tertinggal $\cos \theta_1$

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

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

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

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 pengurangan rugi daya jaringan dapat dihitung menggunakan rumus berikut :

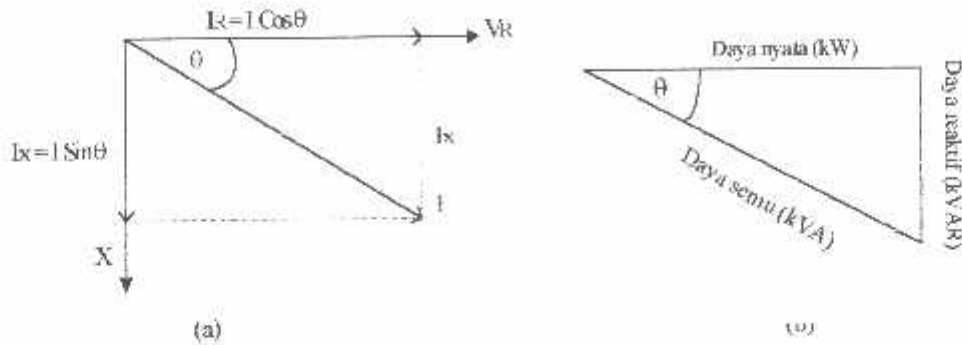
$$\% \text{ Rugi Daya} = 100 \left(\frac{\text{Faktor daya mula - mula} (\cos \theta_1)}{\text{Faktor daya baru} (\cos \theta_2)} \right) \dots\dots\dots (2.10)$$

% Pengurangan rugi daya

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

2.9. Perhitungan Pengaruh Perbaikan Faktor Daya ^[1]

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



Gambar 2-11. Diagram Fasor dan Sudut Daya Beban Distribusi

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

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 semu adalah sebanding dengan arus, dimana hubungannya dapat dinyatakan sebagai berikut :

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

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

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

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

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

$$kW = kVA \cos \theta$$

2.10. Penentuan Rating Kapasitor Untuk Perbaikan Faktor Daya Beban

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

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

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

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

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

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

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

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 :

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

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

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

Daya reaktif (kVAR) = Daya aktif x $\Delta \tan$

$$(kVAR) = (kW) \times \Delta \tan$$

2.11. Sistem Per-Unit ^[2]

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 :

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

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

Arus dasar

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

Impedansi dasar

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

Dalam persamaan diatas nilai - nilai besaran diberikan untuk rangkaian satu fasa. Jadi teganganya adalah tegangan antara fasa dengan tanah dan daya sctiap fasa.

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

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

BAB III

ANALISIS PENEMPATAN KAPASITOR PADA JARINGAN DISTRIBUSI PRIMER DENGAN MENGGUNAKAN KOMBINASI METODE *DETERMINISTIC* DAN *ALGORITMA GENETIKA*

3.1. Analisa Aliran Daya Pada Jaringan

Dalam analisis teknik penempatan kapasitor pada jaringan dengan menggunakan kombinasi metode *deterministic* dan *algoritma genetika* diperlukan suatu proses aliran daya untuk mengetahui rugi-rugi daya pada saluran. Oleh karena itu pembahasan selanjutnya sebelum proses penempatan kapasitor dengan menggunakan kombinasi metode *deterministic* dan *algoritma genetika* adalah analisis aliran daya dengan menggunakan metode *Newton Raphson*.

3.1.1. Tujuan ^[2]

Tujuan mempelajari analisis aliran daya ini adalah :

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

3.1.2. Metode *Newton Raphson* ^[2]

Secara matematis persamaan aliran daya *Newton Raphson* dapat diselesaikan dengan menggunakan koordinat rektanguler, 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 simpul 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 (3.1) 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 pada 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 tagangan

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

$$\begin{aligned} H_{pq} &= \frac{\partial P_p}{\partial \delta_q} & N_{pq} &= \frac{\partial P_p}{\partial |V_q|} \\ M_{pq} &= \frac{\partial Q_p}{\partial \delta_q} & L_{pq} &= \frac{\partial Q_p}{\partial |V_q|} \end{aligned}$$

Adapun rumus dari elemen matriks Jaqobian adalah :

Untuk H :

$$\frac{\partial P_p}{\partial \delta_q} = |V_p V_q Y_{pq}| \sin (\delta_p - \delta_q + \theta_{pq}) \quad q \neq p$$

$$\frac{\partial P_p}{\partial \delta_p} = - \sum_{\substack{q=1 \\ q \neq p}}^n |V_p V_q Y_{pq}| \sin (\delta_p - \delta_q + \theta_{pq})$$

Untuk N :

$$\frac{\partial P_p}{\partial |V_q|} = |V_p Y_{pq}| \cos (\delta_p - \delta_q + \theta_{pq}) \quad q \neq p$$

$$\frac{\partial P_p}{\partial |V_p|} = 2 |V_p Y_{pp}| \cos \theta_{pp} + \sum_{\substack{q=1 \\ q \neq p}}^n |V_p Y_{pq}| \cos (\delta_p - \delta_q + \theta_{pq})$$

Untuk M :

$$\frac{\partial Q_p}{\partial \delta_q} = - |V_p V_q Y_{pq}| \cos (\delta_p - \delta_q + \theta_{pq}) \quad q \neq p$$

$$\frac{\partial Q_p}{\partial \delta_p} = \sum_{\substack{q=1 \\ q \neq p}}^n |V_p V_q Y_{pq}| \cos (\delta_p - \delta_q + \theta_{pq})$$

Untuk L :

$$\frac{\partial Q_p}{\partial |V_q|} = |V_p Y_{pq}| \sin(\delta_p - \delta_q + \theta_{pq}) \quad p \neq q$$

$$\frac{\partial Q_p}{\partial |V_p|} = 2 |V_p Y_{pp}| \sin \theta_{pp} + \sum_{\substack{q=1 \\ q \neq p}}^n |V_p Y_{pq}| \sin(\delta_p - \delta_q + \theta_{pq})$$

Untuk menghitung selisih daya, maka mula-mula ditentukan nilai awal tegangan simpul dan sudut fasanya. Kemudian daya nyata dan daya reaktif dihitung dengan menggunakan persamaan (3.5) dan (3.6). Selisih antara daya yang telah ditentukan dengan daya hasil perhitungan ini merupakan perubahan daya yang terjadi pada simpul.

$$\Delta P = P_{\text{penyangkitan}} - P_{\text{beban}} - P_{\text{perhitungan}} \dots \dots \dots (3.8)$$

$$\Delta Q = Q_{\text{penyangkitan}} - Q_{\text{beban}} - Q_{\text{perhitungan}} \dots \dots \dots (3.9)$$

Magnitude tegangan $|V|$ dan sudut fasa δ_p yang diasumsikan serta selisih daya yang dihitung (ΔP_p dan ΔQ_p) digunakan untuk memperoleh elemen-elemen matriks Jaqobian.

Persamaan (3.7) diselesaikan untuk menghitung vektor koreksi megnitude tegangan $\Delta(|V|)$ dan sudut fasa tegangan ($\Delta\delta$) yang baru. Sehingga diperoleh harga magnitude 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 perhitungan iterasi mencapai konvergen.

3.1.3. Algoritma aliran daya *Newton Raphson*^[2]

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 dan ΔQ pada setiap rel.
3. Hitung nilai-nilai matrik Jacobian 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 Jacobian itu dan hitung koreksi-koreksi tegangan $\Delta\delta_p$ dan $\Delta|V_p|$ pada setiap rel.
5. Hitung nilai baru dari δ_p dan $|V_p|$ dengan menambahkan $\Delta\delta_p$ dan $\Delta|V_p|$ pada nilai sebelumnya.
6. Kembali ke langkah 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 telah dipilih.

3.2. Kombinasi Metode *Deterministic* dan *Algoritma Genetika* ^[4]

Dalam memilih suatu metode solusi untuk aplikasi praktis sering sulit, pilihan itu memerlukan analisis yang cermat atas kelebihan-kelebihan dan kekurangan-kekurangan dari kesekian banyak metode yang tersedia. Untuk solusi dari permasalahan diatas maka digunakan suatu alternatif metode baru yang berdasar pada *Algoritma Genetika (GA)* yaitu *Deterministic*. Metode *Deterministic* adalah suatu metode yang berdasar pada *Algoritma Genetika* khususnya *Micro Genetic Algorithm (μGA)* yang diimplementasikan dengan *MILP (Mixed Integer Linier Programming)*. Perbedaan antara *Algoritma Genetika* dan *Deterministic* adalah pada jumlah populasi yang digunakan. Pada metode *Deterministic* jumlah populasi yang di gunakan lebih sedikit (5-20 populasi) dengan menggunakan bantuan *MILP*, sedangkan pada metode *Algoritma Genetika* murni jumlah populasinya lebih banyak (30-300 populasi). Selain itu pada μGA diterapkan pula metode seleksi *roulette wheel*, yaitu metode seleksi yang memberikan kesempatan kepada anggota populasi yang memiliki nilai fitness tertinggi untuk melakukan reproduksi, dan pada μGA populasi yang memiliki nilai fitness tinggi akan lebih mudah beradaptasi sehingga proses mutasi tidak diperlukan.

Berdasarkan uraian diatas, maka *Deterministic* menyediakan cara yang lebih baik dalam menganalisis masalah penempatan kapasitor pada jaringan distribusi primer.

3.2.1. Pengkodean^[3]

Sebelum menguraikan langkah-langkah diatas, berikut ini disajikan gambaran kegunaan dari bagian kromosom dalam Metode *Deterministic Kromosom* pada penyajian ini terdiri dari *allele-allele* yang diasumsikan sebagai nilai-nilai biner. Jika *allele* pada posisi $i = 1$, kapasitor harus ditempatkan pada bagian *feeder i-th*, jika tidak ($i = 0$) maka kapasitor tidak ditempatkan. Bagian kromosom tersebut mempunyai posisi n , dimana n adalah banyaknya node. Pada gambar 3-1, memperlihatkan contoh solusi untuk feeder dengan enam node.

[0 1 1 0 0 1 / 3 5 2 1 2 4]
sections capacitor sizes

Gambar 3-1. Pengkodean kromosom

Sumber : Alexandre Mendes, Paulo M. Franka, C. Lyra, C. Pissarra, Celso Cavellucci, "An Evolutionary Approach for Capacitor Placement in Distribution Networks", Departamento de Engenharia de Sistemas Faculdade de Engenharia Elétrica e de Computação - Brasil, 2002

3.2.2. Menentukan Populasi Awal^[6]

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

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

3.2.3. Seleksi Individu untuk Rekombinasi ^[6]

Seleksi bertujuan untuk memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang paling fit.

Seleksi akan menentukan individu-individu mana saja yang akan dipilih untuk dilakukan rekombinasi dan bagaimana *offspring* terbentuk dari individu-individu tersebut. Langkah pertama yang dilakukan dalam seleksi ini adalah pencarian nilai *fitness*. Masing-masing individu dalam suatu wadah seleksi akan menerima probabilitas reproduksi yang tergantung pada nilai obyektif dirinya sendiri terhadap nilai obyektif dari semua individu dalam wadah seleksi tersebut. Nilai *fitness* inilah yang nantinya akan digunakan pada tahap-tahap seleksi berikutnya. Pada proses seleksi ini dapat menggunakan banyak metode seperti *roulette wheel selection*, *elitism* dan lain sebagainya.

➤ *Roulette Wheel Selection* ^[6]

Seleksi ini bertujuan untuk memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang memiliki fitness tinggi untuk melakukan reproduksi.

Adapun algoritma dari *roulette wheel* adalah sebagai berikut :

1. Hitung total fitness (F) :
 - $TotFitness = \sum F_k ; k = 1, 2, \dots, \text{popsize}$.
2. Hitung fitness relatif tiap individu :
 - $P_k = F_k / Totfitness$

3.2.4. Rekombinasi^[3]

Setelah parent dipilih sesuai kriteria yang diuraikan diatas, mereka digunakan sebagai parameter masukan di dalam operator rekombinasi. Rekombinasi tersebut menghasilkan suatu individu baru yaitu *offspring*.

Parent A	0 1 0 1 1 0	3 2 1 1 4 2
Parent B	1 1 1 1 0 1	5 1 5 3 3 2
Offspring	1 1 0 1 0 1	4 1 3 2 4 2

Gambar 3-3. Operator rekombinasi

Sumber : Alexandre Mendes, Paulo M. Franka, C. Lyra, C. Pissarra, Celso Cavelucci, "*An Evolutionary Approach for Capacitor Placement in Distribution Networks*", Departamento de Engenharia de Sistemas Faculdade de Engenharia Elétrica e de Computação - Brasil, 2002.

Dalam *Deterministic* hanya ada satu strategi rekombinasi, yaitu untuk kromosom bagian biner. Pada bagian biner, diadopsi proses *crossover uniform*, dimana allele offspringnya telah ditentukan secara acak yang dipilih nilainya dari salah satu orang tua. Jika parent berbagi allele yang sama dalam posisi yang ditentukan, offspringnya akan menerima warisan nilai itu. Jika nilai-nilai berbeda, offspringnya mungkin mewariskan nilai 0 atau 1.^[5]

3.2.5. Fungsi Obyektif^[4]

Fungsi Obyektif berhubungan erat dengan fungsi fitness. Fungsi *fitness* adalah menghitung kualitas dari individu.

Tujuan umum dalam penempatan kapasitor dalam system distribusi adalah untuk meminimalkan biaya total pada rencana penempatan yang dipilih. Biaya

total terdiri dari biaya penempatan dan biaya rugi-rugi daya serta rugi-rugi energi yang ditunjukkan sebagai berikut:

$$\text{Min } \sum_{i \in I_s} (x_i K_p + K_c P_{ri}) \dots\dots\dots (3.10)$$

dimana :

x_i = switch bilangan biner ($x_i = 1$ jika kapasitor di pasang, $x_i = 0$ jika kapasitor tidak di pasang)

K_p = biaya instalasi kapasitor

K_c = harga kapasitor (Kvar)

P_r = rugi-rugi daya reaktif pada kapasitor

I_s = kandidat bus pada kompensasi reaktif

Fungsi Fitness

Fitness/Total Annual Gain

$$= \text{Annualgain} - (\text{AnnualCost} - \text{Dev}_A - \text{Dev}_R) \dots\dots\dots (3.11)$$

Dimana :

➤ *Annualgain* (keuntungan tahunan) = 8,75 x MWh Cost x gain

dimana : MWh Cost = Biaya pemasaran daya oleh PLN (US \$ 168)

8,75 = konstanta transformasi kW ke MWh per tahun

gain = rugi daya sebelum pemasangan kapasitor – rugi daya setelah pemasangan kapasitor.

➤ *Annualgain* (keuntungan tahunan) ditransformasikan kedalam biaya tahunan (*annualcost*). Untuk itu perlu menentukan periode pengembalian

modal (k) untuk peralatan , tingkat bunga (i) dan Harga Kapasitor ($CapCost$).

$$AnnualCost \text{ (biaya tahunan)} = (i \cdot CapCost) / (1 - (1/(1+i)^k))$$

- Untuk membatasi anggaran tahunan:

$$Dev_A = (\max [0, Annual Cost - Budget])^2$$

dimana budget adalah anggaran maksimum tahunan yang tersedia untuk pemasangan kapasitor.

- Persamaan untuk mengendalikan jumlah maksimum pemasangan kapasitor adalah:

$$Dev_B = (\max [0, NumberCap - MaxNumberCap])^2$$

dimana MaxNumberCap adalah maksimal kapasitor yang dipasangkan, NumberCap adalah banyaknya kapasitor yang dipasangkan.

3.3. Algoritma Program

3.3.1. Algoritma Pemecahan Masalah Penempatan Kapasitor Pada Jaringan Distribusi 20 kV Dengan Menggunakan Metode *Algoritma Genetika* dan *Deterministic* Di GI Sengkaling Penyulang Pujon

1. Memasukan inputan yang terdiri dari data beban (P dan Q), data saluran (R dan X), data kapasitor (kapasitas dan harga kapasitor)
 2. Menentukan parameter inputan *Metode Deterministic* yang meliputi jumlah generasi, jumlah populasi, probabilitas crossover, probabilitas mutasi
 3. Generasi = 0, Populasi = 0
 4. Melakukan proses penyederhanaan jumlah populasi dengan MILP
 5. Melakukan Inisialisasi pada populasi
 6. Melakukan proses fitness dari kromosom tiap-tiap individu
 7. Melakukan proses statistik
 8. Melakukan proses seleksi
 9. Melakukan proses crossover
 10. Cek apakah offspring yang diinginkan sudah terpenuhi (max. Pop). Jika "tidak", perhyungan kembali ke langkah 7 dengan cara men-set nilai populasi. Jika "Ya" perhitungan dilanjutkan ke langkah 10
 11. Menghitung Fitness offspring
 12. Melakukan proses Elitism
 13. Melakukan proses statistik
 14. Apakah jumlah generasi sudah terpenuhi (max. gen).
 15. Jika "tidak" maka $gen = gen + 1$, kambah ke langkah 7
 16. Jika "ya" perhitungan selesai.
-

3.3.2. Algoritma Program Fitness

Algoritma program untuk memperoleh *fitness* adalah :

1. Baca data kromosom, meliputi lokasi dan kapasitas kapasitor
 2. Menghitung fungsi obyektif (pers. 3.10)
 3. Menghitung Nilai Fitness (pers. 3.11)
 4. Melakukan proses statistik
-

BAB IV

STUDI ANALISIS PENEMPATAN KAPASITOR PADA JARINGAN DISTRIBUSI PRIMER DENGAN MENGUNAKAN KOMBINASI METODE *DETERMINISTIC* DAN *ALGORITMA GENETIKA* DI GI SENGKALING PENYULANG PUJON

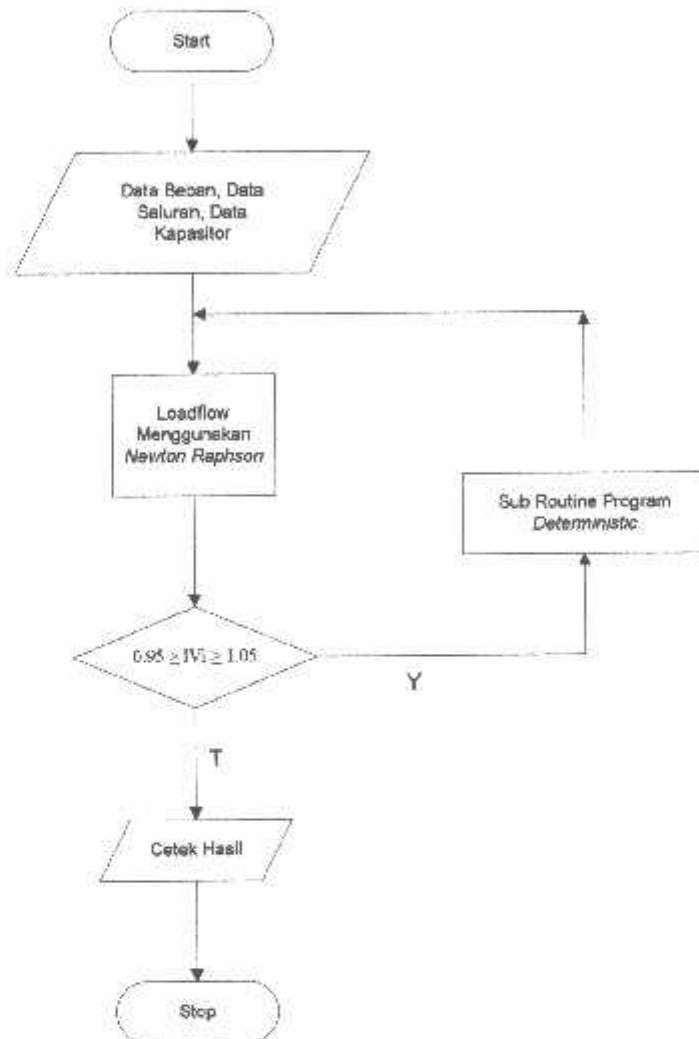
4.1. Penempatan Kapasitor Menggunakan Kombinasi metode *Deterministic* dan *Algoritma Genetika* ^[4]

Pemasangan kapasitor shunt pada lokasi yang tepat pada jaringan distribusi yang besar dapat mereduksi rugi daya dan dapat memperbaiki profil tegangan pada penyulang. Untuk mengatasi permasalahan dalam menentukan lokasi penempatan kapasitor dan besarnya kapasitas kapasitor adalah dengan menggunakan suatu alternatif kombinasi motoda *Deterministic* dan *Algoritma Genetika*.

Dalam perhitungan analisa penempatan kapasitor, kita menggunakan program komputer. Pada prinsipnya penggunaan program komputer dapat menganalisa sebuah jaringan distribusi dengan jumlah cabang atau jumlah *bus/node* yang besar tergantung dari jumlah memori yang tersedia pada komputer yang digunakan.

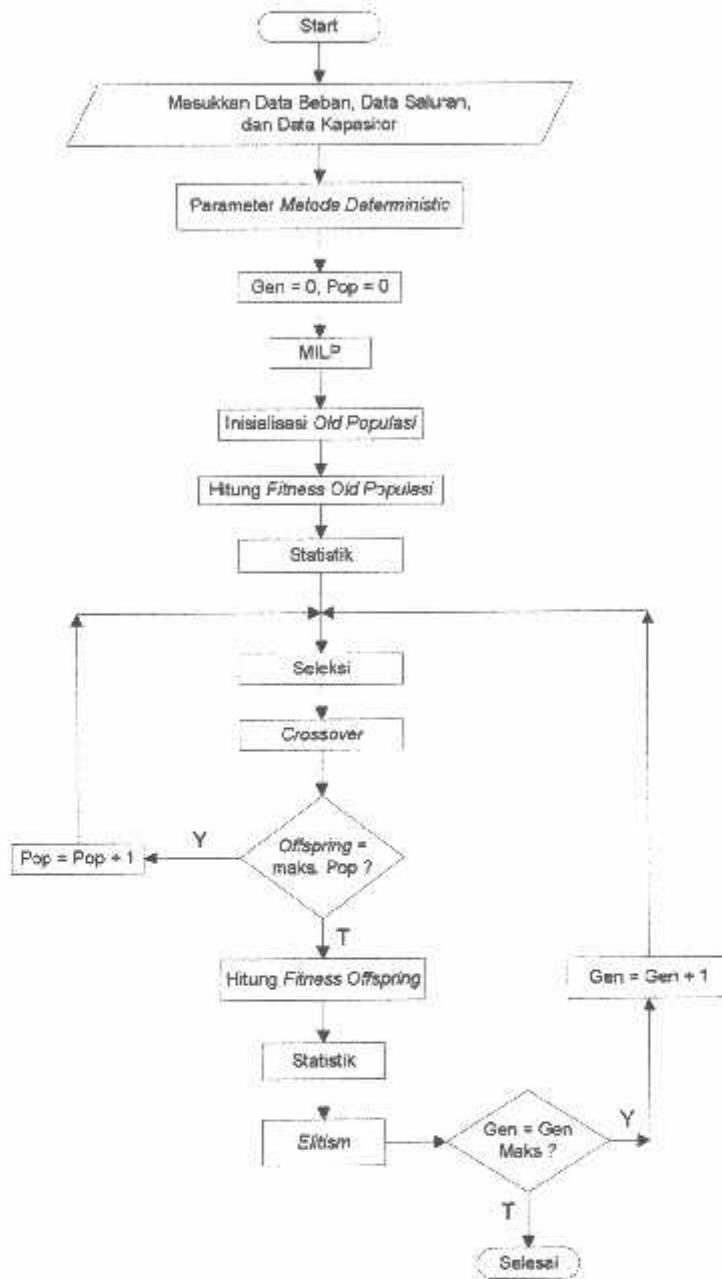
4.2. Flowchart Algoritma Program

4.2.1. Flowchart Algoritma Pemecahan Masalah Penempatan Kapasitor Pada Jaringan Distribusi 20 kV DI GI Sengkaling Penyulang Pujon



Gambar 4-1. Flowchart Program Pemecahan Masalah

4.2.2 Flowchart Sub Routine Program *Deterministic*



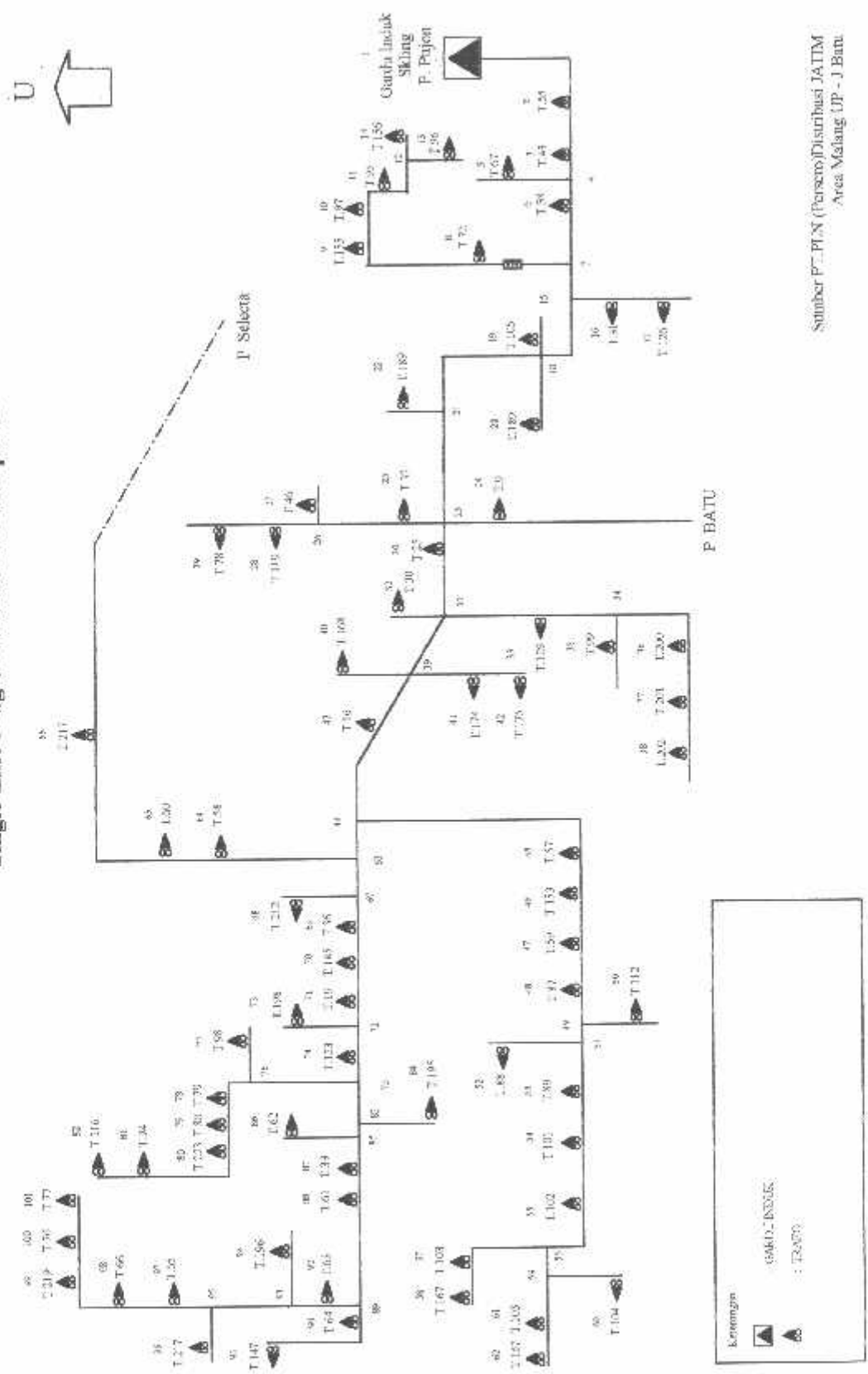
Gambar 4-2. Flowchart Program *Deterministic*

4.2.3. Flowchart Sub Routine *Deterministic Program Fitness*



Gambar 4-3. Flowchart Sub Routine *Deterministic Program Fitness*

Gambar 4-4
Single Line Diagram Sebelum Penempatan



Sumber PT. PTN (Persero) Distribusi JATIM
Area Malang UP - J. Batu

4.3. Data Penyulang Pujon

Penyulang Pujon yang merupakan keluaran dari Gardu Induk Sengkaling Malang memakai tegangan 20 kV. Untuk menyelesaikan perhitungan terlebih dahulu ditetapkan *single line diagram* dari penyulang Pujon yang akan di analisa. Agar perhitungan lebih mudah maka digunakan sistem per-unit (pu), dimana dasar yang di gunakan:

- ▲ Tegangan dasar : 20 kV
- ▲ Daya dasar : 100 kVA

Selanjutnya *node-node* yang ada diklasifikasikan, yaitu Busbar GI Sengkaling diasumsikan sebagai *slack bus*, sedangkan *node-node* yang lain sepanjang saluran radial di pandang sebagai *load bus*. Dalam hal ini tidak ada *bus generator* karena sepanjang saluran tidak terdapat pembangkitan.

4.3.1. Data Saluran

Untuk menyalurkan tenaga listrik dari GI Sengkaling ke pelanggan, pada penyulang Pujon digunakan saluran jenis hantaran udara, dengan spesifikasi jenis konduktor yang digunakan adalah seperti pada tabel 4-1.

Tabel 4-1
Data Penyulang Sistem 20 kV GI Sengkaling

Jenis Konduktor	Penampang Nominal (mm ²)	Impedansi Saluran (Ω/km)
AAAC	35	0.9217 + j 0.3790
AAAC	50	0.6452 + j 0.3678
AAAC	70	0.4608 + j 0.3572
AAAC	120	0.2688 + j 0.3376
AAAC	150	0.2162 + j 0.3305

Adapun data saluran penyulang Pujon seperti pada tabel 4-2.

Tabel 4-2
Data Saluran Penyulang Pujon

Saluran	Dari Node	Ke Node	Panjang Saluran (m)	Impedansi Saluran	
				R (Ohm)	X (Ohm)
1	1	2	580.3	0.1255	0.1918
2	2	3	1432.6	0.3097	0.4735
3	3	4	863.9	0.1868	0.2855
4	4	5	85.4	0.0185	0.0282
5	5	6	683.3	0.1477	0.2258

Tabel 4-2 selengkapnya dapat dilihat pada lampiran

4.3.2. Pembebanan Penyulang Pujon

Pembebanan diperoleh dengan mengambil data dari kVA 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 *node*. Dengan mengasumsikan $\cos \varphi$ 0.86 diperoleh pembebanan aktif dan pembebanan reaktif seperti pada tabel 4-3.

Tabel 4-3
Data Pembebanan Penyulang Pujon

Nomor Node	Kode Trafo	Lokasi	Pembebanan		Type Bus
			P (kW)	Q (kVAR)	
1			0	0	Slack
2	T55	Jl Raya Ngundat	38.7	22.96	Load
3	T43	Jl Raya Mojorejo	99.072	58.775	Load
4			0	0	Load
5	T63	Dk. Mantung	115.584	68.571	Load

Tabel 4-3 selengkapnya dapat dilihat pada lampiran

Tabel 4-4
Data Kapasitor

Index	Size in kVAR	Cost in dollars (US\$)
1	150	1,498
2	300	1,604
3	450	1,620
4	600	1,833
5	900	2,550
6	1200	2,955

Sumber: PT PLN (Persero) UBD JATIM Area Pelayanan Malang GA Batu Sub UP-J Batu

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

Untuk menentukan letak penempatan kapasitor, besar kapasitas yang terpasang dan total *AnnualGain*/keuntungan tahunan akan ditentukan oleh program komputer.

Untuk memudahkan perhitungan dan analisa pada sistem tenaga, biasanya dipakai harga-harga dalam per-satuan/ per-unit. Harga per-satuan/ per-unit adalah harga yang sebenarnya dibagi dengan harga dasar. Harga dasar 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 rata-rata diasumsikan sebesar 0,86.

Perhitungan diawali dengan menampilkan *single line* diagram dari penyulang yang mewakili keadaan sistem yang sesungguhnya. Dari gambar 4-4

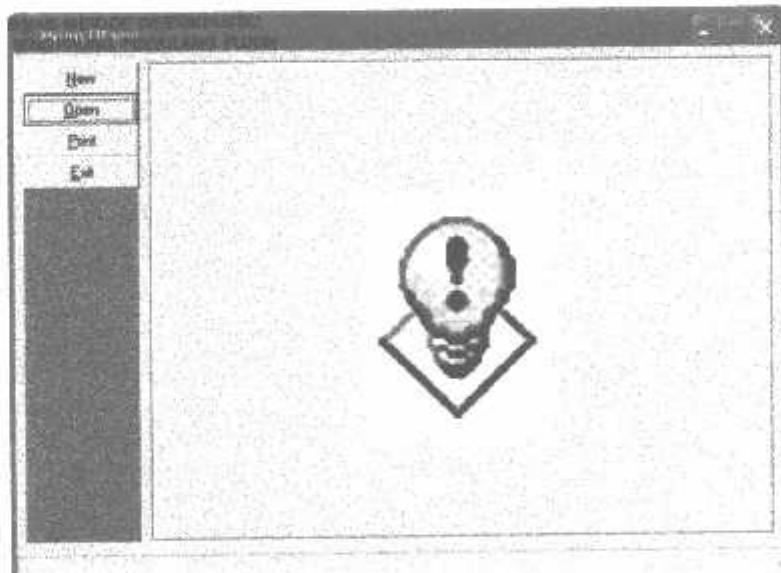
diperlihatkan *single line* diagram dari penyulang Pujon. Pada penyulang ini jumlah bus dan jumlah saluran masing-masing adalah :

- ▲ *Slack bus* = 1
- ▲ *Load bus* = 100
- ▲ Jumlah saluran = 100

4.5 Tampilan program

Program dalam skripsi ini dijalankan dengan menggunakan bahasa pemrograman Borland Delphi versi 7.0 dan diaplikasikan pada komputer processor AMD Athlon XP 2100 MHz, dengan memori 256 MB. Mengenai jalannya program, ikuti prosedur program sebagai berikut:

1. Tampilan utama dari program



Gambar 4-5. Tampilan Utama Program

2. Tekan tombol open file untuk membuka file yang sudah tersimpan

The screenshot shows a window titled "Tampilan Data" with four tabs: "Data Umum", "Data Saluran", "Data Pembebanan", and "Data Kapasitor". The "Data Umum" tab is active. It contains the following input fields:

- Jumlah Bus: 101
- Jumlah Saluran: 100
- Tegangan Dasar: 20 kV
- Daya Dasar: 100 kVA
- Parameter: ohm

Buttons for "Next" and "Close" are located at the bottom right.

Gambar 4-6. Tampilan Inputan data

3. Kemudian tekan tombol data saluran

The screenshot shows the same "Tampilan Data" window, but with the "Data Saluran" tab selected. It displays a table with the following data:

No	Daftar	Ka	R (ohm)	X (ohm)	Lc (ohm)	Ti
1	2		0.1255	0.1918	0	0
2	3		0.3097	0.4735	0	0
3	4		0.1868	0.2855	0	0
4	4	5	0.0185	0.0282	0	0
5	4	5	0.1477	0.2258	0	0
6	6	7	0.4503	0.6884	0	0
7	7	8	0.0334	0.051	0	0
8	8	9	0.0896	0.137	0	0
9	9	10	0.1906	0.2923	0	0
10	10	11	0.1002	0.1512	0	0
11	11	12	0.0977	0.1494	0	0
12	12	13	0.308	0.4736	0	0
13	12	14	0.2565	0.4075	0	0
14	7	15	0.2691	0.4114	0	0
15	15	15	0.0195	-0.0299	0	0

Buttons for "Next" and "Data" are located at the bottom right.

Gambar 4-7. Tampilan Data Saluran

4. Kemudian tekan tombol data pembebanan

No	Abs V (psi)	Sud V (deg)	Pgen (kw)	Qgen (kVAR)	Flood (kw)	Gload (kVAR)	Cap	Tjo Bus
1	1	0	0	0	0	0	0	1
2	1	0	0	0	38.7	22.96	0	3
3	1	0	0	0	39.072	58.775	0	3
4	1	0	0	0	0	0	0	3
5	1	0	0	0	115.584	68.571	0	3
6	1	0	0	0	22.575	13.393	0	3
7	1	0	0	0	0	0	0	3
8	1	0	0	0	57.792	34.286	0	3
9	1	0	0	0	65.79	39.03	0	3
10	1	0	0	0	82.56	49.979	0	3
11	1	0	0	0	74.384	44.882	0	3
12	1	0	0	0	0	0	0	3
13	1	0	0	0	19.78	11.725	0	3
14	1	0	0	0	69.048	39.194	0	3
15	1	0	0	0	0	0	0	3

Gambar 4-8. Tampilan Data Pembebanan

5. Tekan tombol data kapasitor

No	Cap (kVAR)	Harga (US\$)
1	100	1498.00000
2	300	1694.00000
3	450	1620.00000
4	600	1933.00000
5	900	2560.00000
6	1200	2955.00000

Gambar 4-9. Tampilan Data Kapasitor

6. Tekan tombol next kemudian tekan tombol hitung *Newton Raphson* untuk melihat hasil perhitungan aliran daya pada kondisi awal

Bus	Abs V (pu)	Sud V (deg)	Pgen (kW)	Qgen (kVAr)	Pload (kVA)	Qload (kVA)
1	1.00000	0.00000	4083.802	2528.556	0.800	0.000
2	0.99751	-0.06691	0.000	0.300	38.700	22.960
3	0.99142	-0.23194	0.000	0.300	99.022	58.775
4	0.98784	-0.32986	0.000	0.300	0.800	0.000
5	0.99703	-0.33015	0.000	0.000	115.584	68.571
6	0.99508	-0.40547	0.000	0.300	22.575	13.330
7	0.97678	-0.63217	0.000	0.000	0.000	0.000
8	0.97672	-0.63888	0.000	0.000	57.752	34.266
9	0.97658	-0.64277	0.000	0.000	65.780	39.030
10	0.97635	-0.64928	0.000	0.000	82.560	48.979
11	0.97627	-0.65163	0.000	0.000	74.304	44.882
12	0.97623	-0.65281	0.000	0.000	0.000	0.000
13	0.97620	-0.65367	0.000	0.000	19.788	11.735

Gambar 4-10. Tampilan Loadflow Awal

7. Kemudian tekan tombol aliran daya

No	Dari	Ke	P (kWatt)	Q (kVA)	Ansa
1	1	2	4083.802	2528.556	0
2	2	1	-4076.564	-2517.493	0
3	2	3	4027.864	2494.531	0
4	3	2	-4020.335	-2467.734	0
5	3	4	3521.263	2408.955	0
6	4	3	-3911.200	-2393.579	0
7	4	5	115.585	68.572	0
8	4	9	3795.615	2305.007	0
9	5	4	115.584	68.571	0
10	6	4	-3788.118	-2312.545	0
11	6	7	3785.543	2300.152	0
12	7	5	-3742.957	-2285.623	0

Summary Result

Total Pembangkitan
4083.802 + 2528.556 kVA

Total Pembebanan
3570.267 + 2355.630 kVA

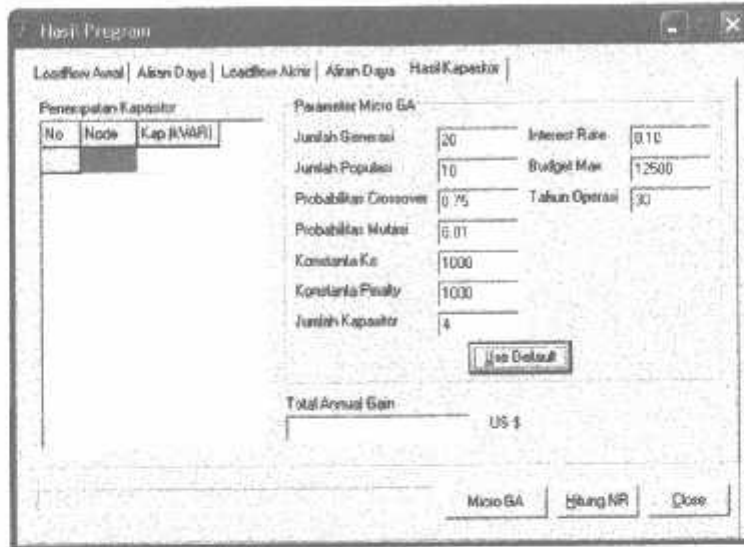
Total Rugi-Rugi Saluran
113.535 + 117.258 kVA

Iterasi
3

Waktu Komputasi (termasuk desk. model)
0:00:219

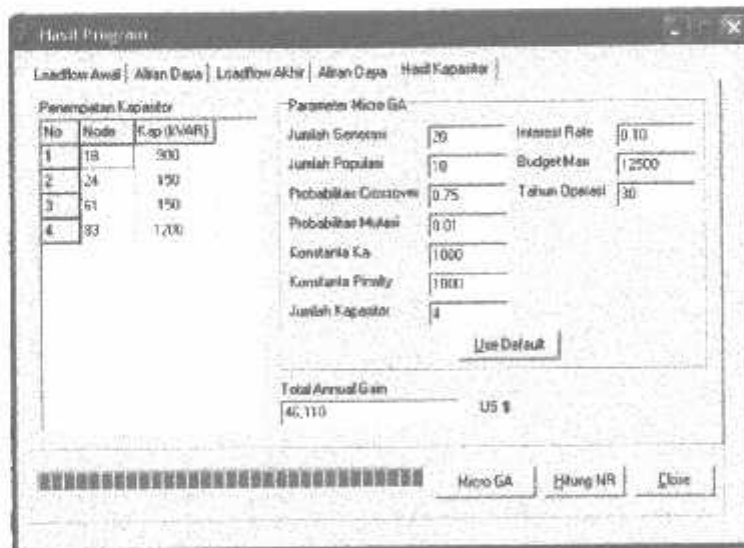
Gambar 4-11. Tampilan Hasil Aliran Daya Sebelum Penempatan Kapasitor Dengan Metode *Newton Raphson*

8. Tekan tombol hasil kapasitor untuk melihat parameter *Deterministic* kemudian tekan tombol *Use Default*



4-12. Tampilan Parameter *Deterministic*

9. Kemudian tekan tombol *Micro GA*



Gambar 4-13. Tampilan Hasil *Deterministic* Dalam Penentuan Lokasi dan Kapasitas Kapasitor

10. Sesudah itu pilih tombol Loadflow akhir dan tombol Aliran Daya

Bus	Abs V (pu)	Sud V (deg)	Pgen (kW)	Qgen (kW)	Pload (KVAR)	Qload (KVAR)
1	1.00000	0.00000	4051.622	79.748	0.000	0.000
2	0.99889	0.11003	0.000	0.000	38.700	22.960
3	0.99554	0.38126	0.000	0.000	99.072	58.775
4	0.99374	0.54371	0.000	0.000	0.000	0.000
5	0.99372	0.54340	0.000	0.000	115.584	68.571
6	0.99240	0.68920	0.000	0.000	22.575	13.393
7	0.98840	-1.05439	0.000	0.000	0.000	0.000
8	0.98834	-1.06598	0.000	0.000	57.752	34.286
9	0.98820	-1.05877	0.000	0.000	65.730	39.038
10	0.98798	-1.06623	0.000	0.000	82.560	48.979
11	0.98790	-1.06843	0.000	0.000	74.304	44.862
12	0.98796	-1.06358	0.000	0.000	0.000	0.000
13	0.98763	-1.07142	0.000	0.000	14.780	11.735

Gambar 4-14. Tampilan Hasil Loadflow Akhir

No	Dari	Ke	P (kW)	Q (KVAR)	Arah
1	1	2	4051.622	79.748	0
2	2	1	4046.669	71.873	0
3	2	3	4007.969	48.913	0
4	3	2	3895.497	29.845	0
5	3	4	2695.425	20.930	0
6	4	3	3888.271	39.864	0
7	4	5	115.585	68.572	0
8	4	6	3773.696	-108.437	0
9	5	4	115.584	68.571	0
10	6	4	3758.357	115.584	0
11	6	7	3745.782	129.977	0
12	7	6	3729.724	154.525	0

Summary Result

Total Pembangkitan
4051.622 + j2479.748 KVA

Total Pembebanan
3570.267 + j2188.618 KVA

Total Rugi Rugi Saluran
01.555 + j124.050 KVA

Hasil
3

Waktu Komputasi (jama:varit:detik:mdetik)
0 0 0 203

Gambar 4-15. Tampilan Hasil Aliran Daya Sesudah Penempatan Kapasitor Dengan Metode *Deterministic*

4.6. Hasil Perhitungan

Dengan mengacu pada gambar *single line* diagram penyulang Pujon, dan dengan melakukan analisa dengan metode *Newton Raphson* maka akan didapatkan besarnya tegangan tiap-tiap *node*, besarnya rugi saluran tiap saluran, dan aliran daya saluran seperti pada Tabel 4-5 sampai Tabel 4-7:

Tabel 4-5

Kondisi Tegangan dan Sudut Fasa Tegangan Tiap *Node* Penyulang Pujon Sebelum Kompensasi

Nomor Node	Tegangan (pu)	Sudut V (deg)
1	1.00000	0.00000
2	0.99751	-0.06691
3	0.99142	-0.23194
4	0.98784	-0.32986
5	0.98783	-0.33015

Tabel 4-5 selengkapnya dapat dilihat pada lampiran

Tabel 4-6

Besar Rugi-Rugi Daya Tiap Saluran Penyulang Pujon Sebelum Kompensasi

Saluran	P (kW)	Q (kVAR)
1-2	7.238	11.063
2-3	14.532	26.799
3-4	10.063	15.38
4-5	0.001	0.002
4-6	70497	11.464

Tabel 4-6 selengkapnya dapat dilihat pada lampiran

Tabel 4-7
Besarnya Aliran Daya Tiap Saluran
Penyulang Pujon Sebelum Kompensasi

Saluran:	P (kW)	Q (kVAR)
1-2	4083.802	2528.556
2-1	-4076.564	-2517.493
2-3	4037.867	2494.533
3-2	-4020.335	-2467.734
3-4	3921.263	2408.959

Tabel 4-7 selengkapnya dapat dilihat pada lampiran

Banyaknya Saluran = 100

Banyaknya *Node* = 101

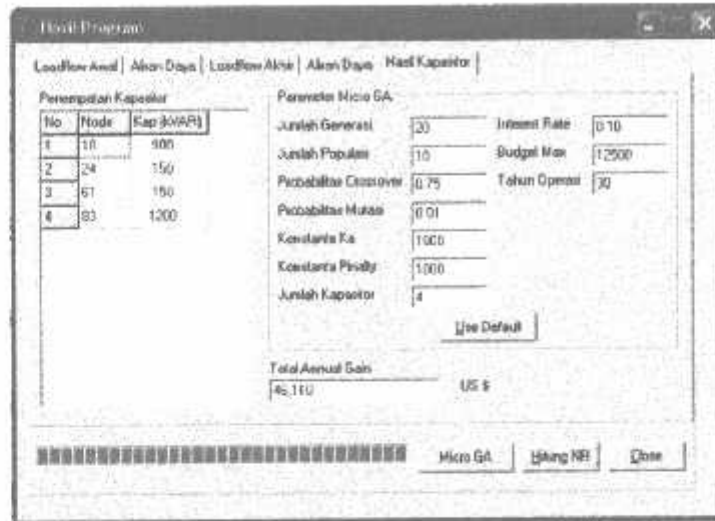
Banyaknya Iterasi = 3

Total Pembangkitan = $4083.802 + j 2528.556$ kVA

Total Pembebanan = $3970.267 + j 2355.698$ kVA

Rugi Total Saluran = $113.535 + j 172.858$ kVA

Setelah dilakukan analisa dengan menggunakan metode *Deterministic*, maka diperoleh hasil yaitu letak pemasangan kapasitor, kapasitas serta total *AnnualGain*/keuntungan tahunan akibat pemasangan kapasitor tersebut. Seperti terlihat pada gambar 4-16.



Gambar 4-16. Tampilan Hasil Perhitungan Metode *Deterministic*

Dari gambar diatas dapat dilihat bahwa kapasitor dipasang pada node 18, 24, 61, dan 83 dengan kapasitas masing-masing sebesar 900, 150, 150, dan 1200 kVAR. Total *AnnualGain* keuntungan tahunan yang diperoleh sebesar US \$ 46.110.

Setelah dilakukan pemasangan kapasitor diperoleh perbaikan profil tegangan serta rugi-rugi saluran. Dapat dilihat pada tabel 4-8 sampai 4-10.

Tabel 4-8

Kondisi Tegangan dan Sudut Fasa Tegangan Tiap Node Penyulang Pujon Sesudah Kompensasi

Nomor Node	Tegangan (pu)	Sudut V (deg)
1	1.00000	0.00000
2	0.99869	-0.11003
3	0.99554	-0.38126
4	0.99374	-0.54311
5	0.99372	-0.54340

Tabel 4-8 selengkapnya dapat dilihat pada lampiran

Tabel 4-9
Besar Rugi-Rugi Daya Tiap Saluran
Penyulang Pujon Sesudah Kompensasi

Saluran	P (kW)	Q (kVAR)
1-2	5.153	7.785
2-3	12.475	19.068
3-4	7.154	10.934
4-5	0.001	0.001
4-6	5.329	8.147

Tabel 4-9 selengkapnya dapat dilihat pada lampiran

Tabel 4-10
Besar Aliran Daya Tiap Saluran
Penyulang Pujon Sesudah Kompensasi

Saluran	P (kW)	Q (kVAR)
1-2	4051.822	79.748
2-1	-4046.669	-71.873
2-3	4007.969	48.913
3-2	-3995.497	-29.845
3-4	3896.425	-28.930

Tabel 4-10 selengkapnya dapat dilihat pada lampiran

Banyaknya Saluran = 100

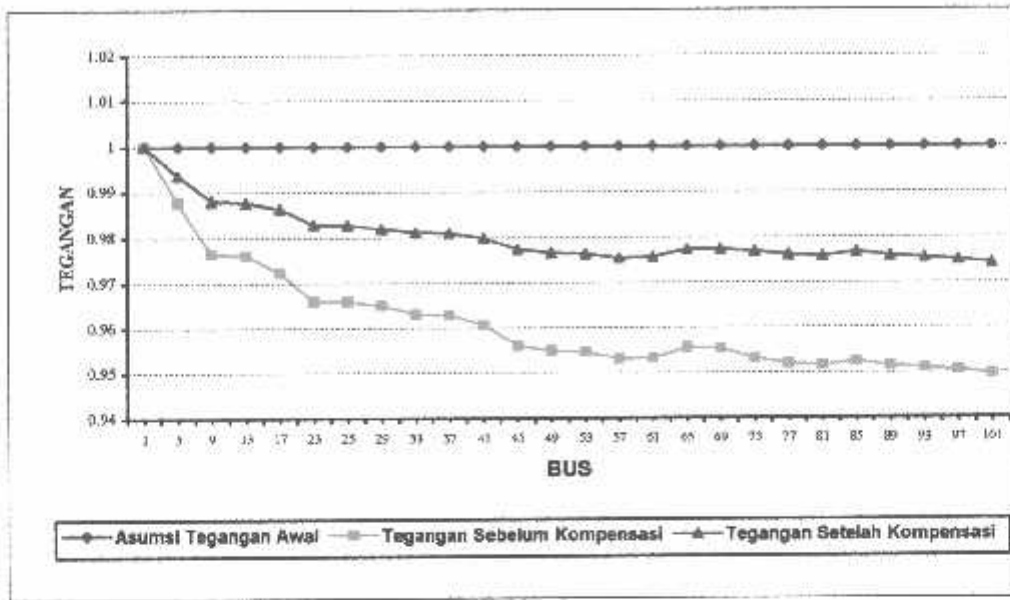
Banyaknya Node = 101

Banyaknya Iterasi = 3

Total Pembangkitan = 4051.822 + j 2479.748 kVA

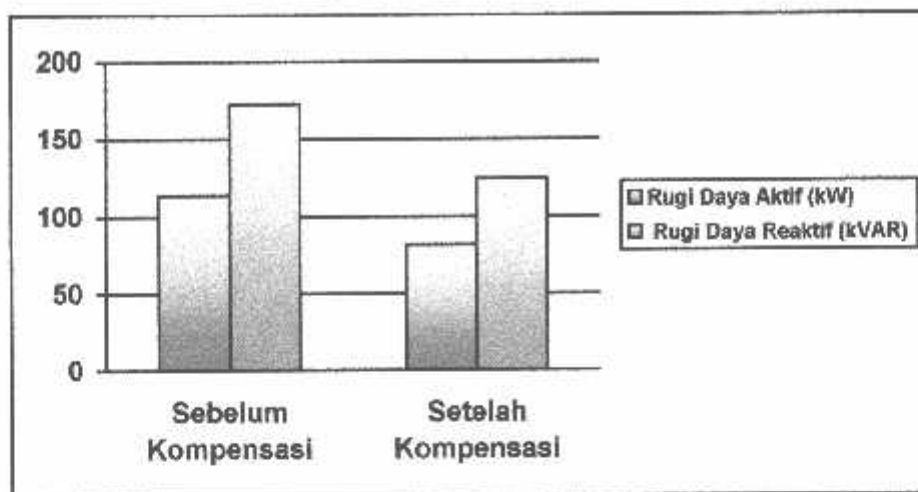
Total Pembebanan = 3970.267 + j 2355.698 kVA

Rugi Total Saluran = 81.555 + j 124.050 kVA



Grafik 4-1. Tegangan Tiap-Tiap *Node* Terhadap Asumsi Tegangan Awal Sebelum dan Sesudah Kompensasi

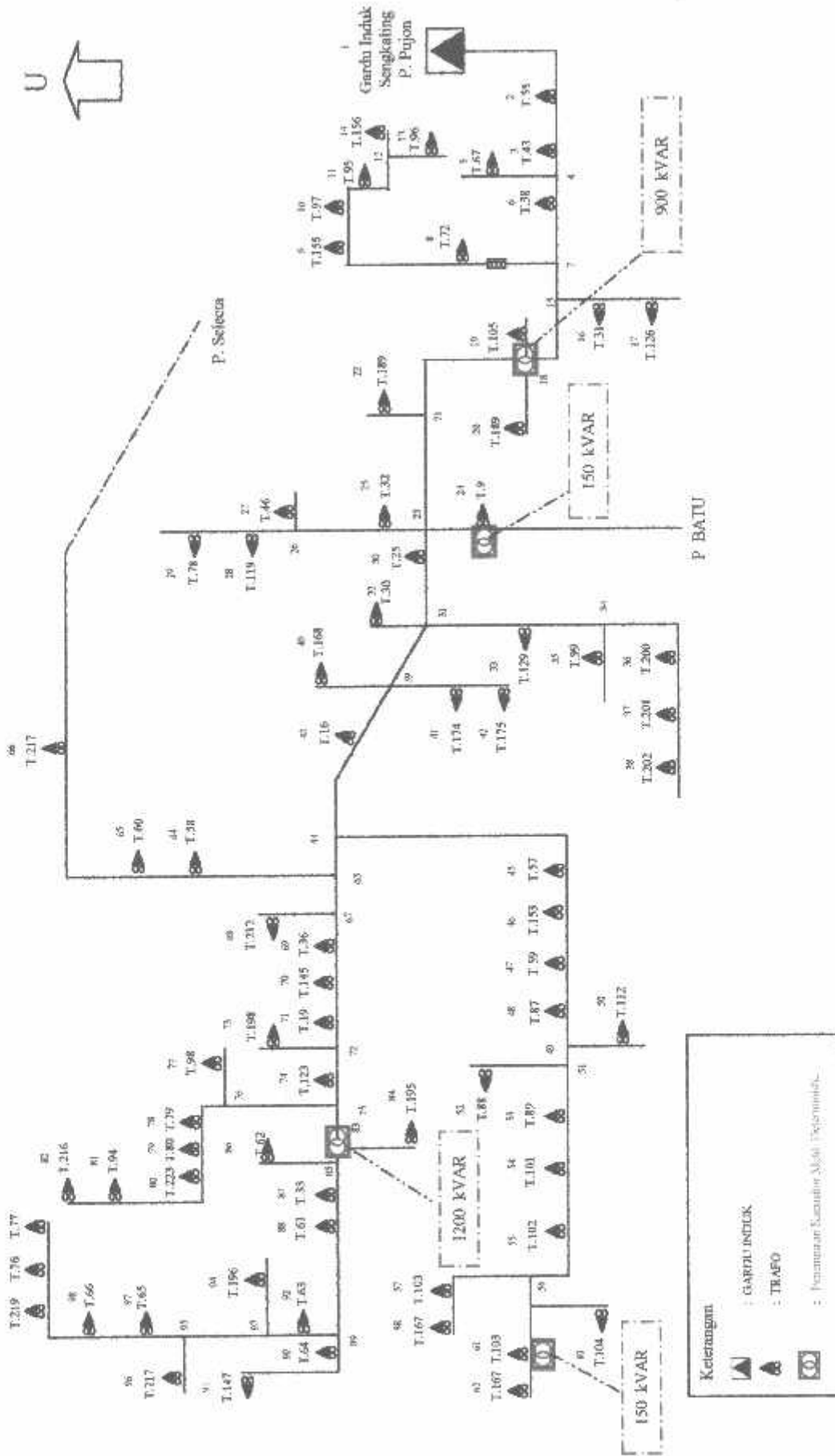
Dari grafik diatas dapat dilihat bahwa perubahan tegangan dari asumsi tegangan awal tiap-tiap *node* masih dalam batas-batas yang diijinkan, dengan tegangan paling rendah terjadi pada *node* ke-101 yaitu 0.94966 pu atau 18.9932 kV, dan setelah kompensasi tegangan tersebut mengalami kenaikan menjadi 0.97425 pu atau 19.485 kV.



Grafik 4-2. Rugi Daya Sebelum dan Sesudah Kompensasi

Dari grafik diatas dapat dilihat perubahan rugi daya, untuk daya aktif rugi daya sebelum kompensasi sebesar 113.535 kW setelah kompensasi rugi daya aktif mengalami penurunan menjadi 81.555 kW, untuk daya reaktif sebelum kompensasi sebesar 172.858 kVAR setelah dikompensasi mengalami penurunan menjadi 124.050 kVAR.

Gambar 4-17
Single Line Diagram Setelah Penempatan Kapasitor
Dengan Metode Deterministic



BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Metode Deterministic telah berhasil memecahkan masalah optimalisasi penentuan lokasi, jumlah, dan kapasitas kapasitor sehingga diperoleh perbaikan profil tegangan, penurunan rugi-rugi daya dan keuntungan tahunan sebagai berikut :

1. Berdasarkan hasil perhitungan dengan *Metode Deterministic*, letak dan kapasitas kapasitor yang optimal pada bus 18, 24, 61, dan 83 dengan kapasitas masing-masing sebesar 900, 150, 150, dan 1200 kVAR.
2. Dari hasil perhitungan load flow sebelum penempatan kapasitor dengan metode *Newton Raphson* pada penyulang Pujon, tegangan yang beroperasi diluar batas yang diijinkan (0.95 pu - 1.05 pu) terjadi pada bus 98, 99, 100, 101 dengan tegangan masing- masing secara berurutan adalah 0.949901 pu; 0.94989 pu; 0.94970 pu; 0.94966 pu.
3. Dengan metode *Deterministic* kondisi tegangan pada bus-bus diatas (98, 99, 100, dan 101) dapat diperbaiki masing-masing menjadi 0.97448 pu; 0.97447 pu; 0.97429 pu; dan 0.97425 pu.
4. Besar rugi daya mengalami penurunan yaitu untuk daya aktif penurunannya sebesar 31.980 kW, dari 113.535 kW menjadi 81.555 kW. Sedangkan untuk daya reaktif penurunannya sebesar 48.808 kVAR, dari 172.858 kVAR menjadi 124.050 kVAR.

5. Dengan memperhitungkan harga dan banyaknya kapasitor, modal awal, suku bunga pertahun, serta rugi-rugi daya maka keuntungan pertahun yang diperoleh setelah pemasangan kapasitor pada bus 18, 24, 61, dan 83 adalah US \$ 46,110.

5.2 Saran

Penentuan letak kapasitor pada jaringan distribusi primer tipe radial dengan menggunakan metode Deterministic perlu dikembangkan dan diaplikasikan dalam menganalisa saluran-saluran yang telah ada maupun untuk perencanaan perluasan jaringan. Dan sebaiknya dengan metode-metode yang optimal dapat diaplikasikan pada jaringan-jaringan yang ada di Indonesia dengan tidak mengurangi keuntungan yang diperoleh oleh PLN.

DAFTAR PUSTAKA

- [1] Hasan Basri, "*Sistem Distribusi Tenaga Listrik*".
 - [2] William D. Stevenson, Jr, 1996, "*Analisis Sistem Tenaga Listrik*", Edisi Keempat, Erlangga, Jakarta.
 - [3] Alexandre Mendes, Paulo M. Franka, C. Lyra, C. Pissarra, Celso Cavelucci, "*An Evolutionary Approach for Capacitor Placement in Distribution Networks*", Departamento de Engenharia de Sistemas Faculdade de Engenharia Eletrica e de Computacao – Brasil, 2002.
 - [4] Maurizio Delfanti, Gianpietro P. Granelli, "*Optimal Capacitor Placement Using Deterministic and Genetic Algorithms*", IEEE Transaction on Power System, vol 15, no 3, Agustus 2000.
 - [5] Mitsuo Gen, Runwei Cheng, "*Genetic Algorithm and Engineering Design*", John Wiley & Son. Inc., 1994.
 - [6] Sri Kusuma Dewi, "*Artificial Intelligence (Teknik dan Aplikasinya)*", Graha Ilmu, Yogyakarta, 2003.
 - [7] Turan Gonen, "*Electric Power Distribution System Engineering*", University of Missouri at Columbia.
-

Lampiran



**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

1. Nama : RACHIMA FITRI
2. NIM : 99.12.117
3. NIRM : -
4. Jurusan : TEKNIK ELEKTRO
5. Konsentrasi : TEKNIK ENERGI LISTRIK
6. Judul Skripsi :

**ANALISIS PENEMPATAN KAPASITOR PADA JARINGAN DISTRIBUSI
PRIMER DENGAN MENGGUNAKAN KOMBINASI *METODE
DETERMINISTIC* DAN *ALGORITMA GENETIKA* DI GI SENGKALING
PENYULANG PUJON**

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

Hari : Rabu
Tanggal : 30 Maret 2005
Dengan Nilai : 79,00 (B) *Sm*



Panitia Ujian Skripsi

Ir. Mochtar Asroni, MSME
Ketua

Ir. F. Yudi Limpraptono, MT
Sekretaris

Anggota Penguji

Ir. H. Taufik Hidayat, MT
Penguji Pertama


Ir. Widodo Pudji M, MT
Penguji Kedua



LEMBAR BIMBINGAN SKRIPSI

1. Nama : RACHMA FITRI
2. NIM : 99.12.117
3. NIRM : -
4. Jurusan : Teknik Elektro S-1
5. Konsentrasi : Teknik Energi Listrik
6. Judul Skripsi :

**ANALISIS PENEMPATAN KAPASITOR PADA JARINGAN
DISTRIBUSI PRIMER DENGAN MENGGUNAKAN KOMBINASI
METODE DETERMINISTIC DAN ALGORITMA GENETIKA DI GI
SENGKALING PENYULANG PUJON**

7. Tanggal Mengajukan Skripsi : 7 Februari 2005
8. Tanggal Menyelesaikan Skripsi : 30 Maret 2005
9. Dosen Pembimbing : Ir. I Made Wartana, MT
10. Telah dievaluasi dengan nilai : 85,00 (A) 

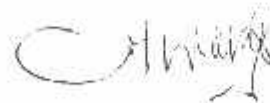
Malang, April 2005

Mengetahui,
Ketua Jurusan Teknik Elektro



(Ir. F. Yudi Limpraptono, MT)
NIP. Y. 103 9500 274

Diperiksadan disetujui,
Dosen Pembimbing



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

PT. PLN (PERSERO)

DISTRIBUSI JAWA TIMUR

AREA PELAYANAN & JARINGAN MALANG

Pos : 0341 - 326034 (Hunting)
: 211

Facsimile : 0341 - 362046

nomor : *423/360/APJ - MLG / 2005*
alamat Sdr.No : ITN-750/III.TA/2/2004
keperluan : --
mengenai : Balasan Survey

Malang, ¹³ Januari 2005

Kepada
Fakultas Teknologi Industri
ITN Malang
Jl. Bendungan Sigurgura No 2
Di
MALANG

Menjawab surat Saudara nomor ITN-750/III.TA/2/2004 Tanggal 3 Januari 2005 perihal tersebut diatas, dengan ini kami beritahukan bahwa pada prinsipnya kami tidak keberatan / mengizinkan Mahasiswa Saudara atas nama :

Rachma Fitri

Nim : 99.12.117

Untuk melaksanakan Survey di APJ Malang PT PLN (Persero) Area Pelayanan dan Jaringan Malang mulai tanggal 17 Januari 2005 sampai dengan 18 Pebruari 2005 dengan catatan PLN hanya memberikan data yang **TIDAK BERSIFAT RAHASIA**.

Sebelumnya diminta agar mahasiswa tersebut mengisi surat pernyataan, dilengkapi dengan pas photo ukuran 4 X 6 Cm di SDM & Administrasi PT. PLN (Persero) Area Pelayanan dan Jaringan Malang.

Demikian agar menjadikan maklum.

Tembusan :
Manajer UJ,UPJ Terkait
Supervisor APJ Terkait



LAMPIRAN A :**Tabel 4-2
Data Saluran Penyulang Pujon**

Saluran	Dari Node	Ke Node	Panjang Saluran (m)	Impedansi Saluran	
				R (Ohm)	X (Ohm)
1	1	2	580.3	0.1255	0.1918
2	2	3	1432.6	0.3097	0.4735
3	3	4	863.9	0.1868	0.2855
4	4	5	85.4	0.0185	0.0282
5	5	6	683.3	0.1477	0.2258
6	6	7	2082.8	0.4503	0.6884
7	7	8	154.4	0.0334	0.051
8	8	9	414.5	0.0896	0.137
9	9	10	896.5	0.1938	0.2963
10	10	11	463.5	0.1002	0.1532
11	11	12	452	0.0097	0.1494
12	12	13	1424.5	0.308	0.4708
13	13	14	1233	0.2666	0.4075
14	14	15	1244.8	0.2691	0.4114
15	15	16	90.5	0.0196	0.0299
16	16	17	20.4	0.0044	0.0067
17	17	18	338	0.0731	0.1117
18	18	19	182.2	0.049	0.0615
19	19	20	402.8	0.1083	0.136
20	20	21	304.3	0.0653	0.1006
21	21	22	293.7	0.079	0.0992
22	22	23	1257.4	0.2719	0.4156
23	23	24	406.5	0.0879	0.1344
24	24	25	154.2	0.0995	0.0567
25	25	26	627.4	0.5783	0.2378
26	26	27	6.4	0.0059	0.0024
27	27	28	204.2	0.1882	0.0774
28	28	29	1145.9	1.0562	0.4343
29	29	30	868.2	0.1877	0.2869
30	30	31	141.2	0.0305	0.0467
31	31	32	848.8	0.1835	0.2805

32	32	33	272.5	0.1758	0.1002
33	33	34	1127.05	0.7272	0.4145
34	34	35	102.2	0.0659	0.0376
35	35	36	496.35	0.1073	0.164
36	36	37	282.5	0.0611	0.0934
37	37	38	294.5	0.0637	0.0973
38	38	39	943.3	0.2939	0.3118
39	39	40	302.31	0.1951	0.1112
40	40	41	548.11	0.1158	0.1812
41	41	42	105.9	0.0229	0.035
42	42	43	40.5	0.0088	0.0134
43	43	44	1653.6	0.3575	0.5465
44	44	45	785.2	0.1658	0.2595
45	45	46	335.9	0.0726	0.111
46	46	47	793.2	0.1715	0.2622
47	47	48	571	0.1235	0.1887
48	48	49	365.5	0.079	0.1208
49	49	50	729	0.1578	0.2412
50	50	51	54.10	0.0117	0.0179
51	51	52	121.8	0.0263	0.0403
52	52	53	940.35	0.2033	0.3108
53	53	54	1715.3	0.7904	0.6127
54	54	55	1192.7	0.5496	0.426
55	55	56	1123.5	0.2429	0.3713
56	56	57	1353.3	0.2926	0.4473
57	57	58	1067.1	0.2303	0.3527
58	58	59	396.5	0.0857	0.131
59	59	60	296.8	0.0642	0.0981
60	60	61	1093.4	0.2364	0.3614
61	61	62	1162.6	0.2514	0.3842
62	62	63	497.1	0.1075	0.1643
63	63	64	570.4	0.1233	0.1885
64	64	65	907	0.1961	0.2996
65	65	66	1769.8	0.3826	0.5849
66	66	67	135.2	0.0292	0.0447
67	67	68	821.6	0.1776	0.2715
68	68	69	110.15	0.0238	0.0364
69	69	70	891.3	0.1927	0.2946

70	70	71	693.2	0.1303	0.1994
71	71	72	37.3	0.0059	0.009
72	72	73	213.1	0.0504	0.077
73	73	74	258.7	0.0559	0.0855
74	74	75	198.1	0.0428	0.0655
75	75	76	1538.1	0.3325	0.5093
76	76	77	381.43	0.0895	0.1261
77	77	78	427	0.1021	0.156
78	78	79	449.6	0.0972	0.1486
79	79	80	721.6	0.156	0.2384
80	80	81	940.5	0.2033	0.3108
81	81	82	1428.3	0.3088	0.4721
82	82	83	322.1	0.0696	0.1065
83	83	84	186.4	0.0405	0.0616
84	84	85	98.5	0.0213	0.0326
85	85	86	243.9	0.0527	0.0806
86	86	87	205.3	0.0444	0.0679
87	87	88	590.3	0.1276	0.1951
88	88	89	856.6	0.1852	0.2831
89	89	90	458.3	0.0991	0.1515
90	90	91	1162.2	0.2513	0.3841
91	91	92	207.4	0.0448	0.0686
92	92	93	1061.5	0.2295	0.3508
93	93	94	1321.52	0.8527	0.4861
94	94	95	385.1	0.2485	0.1416
95	95	96	478.3	0.3085	0.1759
96	96	97	532.6	0.2546	0.1223
97	97	98	1161.5	0.7494	0.4272
98	98	99	100.4	0.0217	0.0332
99	99	100	785.8	0.507	0.289
100	100	1010	285.3	0.1841	0.1049

Sumber: PT PLN (Persero) UBD JATIM Area Pelayanan Malang GA Batu Sub UP-J Batu

LAMPIRAN B :

Tabel 4-3

Data Pembebanan Penyulang Pujon

Nomor Node	Kode Trafo	Lokasi	Pembebanan		Type Bus
			P (kW)	Q (kVAR)	
1			0	0	Slack
2	T55	Jl Raya Ngandut	38.7	22.96	Load
3	T43	Jl Raya Mojorejo	99.072	58.775	Load
4			0	0	Load
5	T63	Dk. Mantung	115.584	68.571	Load
6	T38	Jl. Raya Beji	22.575	13.393	Load
7					Load
8	T72	Emanuel temas	57.792	34.286	Load
9	T115	Jl. Wukir Temas	65.79	39.03	Load
10	T97	Ds. Temas Klerrek	82.56	48.979	Load
11	T95	Ds. Torong Rejo	74.304	44.082	Load
12					Load
13	T96	Ds. Torong Rejo tutup	19.780	11.735	Load
14	T156	Ds. Wukir	66.048	39.184	Load
15					Load
16	T31	Jl. Sudiro	32.198	19.102	Load
17	T126	Jl. Sudiro	98.04	58.163	Load
18					Load
19	T105	R. s Paru-paru	60.2	35.714	Load
20	T189	Jl. Kasirnan	66.048	39.184	Load
21					Load
22	T188	Jl. Lesti	57.792	34.286	Load
23					Load
24	T9	Jl. Hassanudin	130.72	77.551	Load
25	T32	Jl. Indragiri	72.928	43.265	Load
26					Load
27	T46	Ds. Sumberejo	88.064	52.245	Load
28	T19	Ds. Sumberejo	3.87	2.296	Load
29	T78	Ds. Satrehan	112.832	66.938	Load
30	T25	Jl. Ironojoyo	19.35	11.48	Load
31					Load
32	T30	Ds. Songgoriti	24.768	14.694	Load
33	T129	Jl. Flamboyun	83.936	49.805	Load
34					Load
35	T99	Jl. Flamboyun	55.59	33.17	Load
36	T200	Ds. Tambuh	1.72	1.021	Load
37	T201	Ds. Tambuh	1.935	1.148	Load
38	T202	Ds. Tambuh	1.29	0.766	Load
39					Load
40	T168	Bukit Pinus	4.73	2.807	Load
41	T174	Jl. Songgoriti	88.064	52.258	Load
42	T175	Jl. Songgoriti	17.888	10.614	Load
43	T16	Jl. Songgoriti	44.032	26.127	Load
44					Load
45	T57	Dk. Sebaluh	58.05	34.445	Load
46	T153	Dk. Sebaluh	74.304	44.09	Load
47	T59	Dk. Maron	67.94	40.314	Load
48	T87	Dk. Maron	58.48	34.701	Load
49					Load
50	T112	Ds. Gunung Sari	10.105	5.956	Load

51					Load
52	T88	Ds. Pujon Kidul	30.1	17.861	Load
53	T89	Ds. Pujon Kidul	98.9	58.685	Load
54	T101	Dk. Bisan	30.100	17.861	Load
55	T102	Dk. Pakir	63.64	37.762	Load
56					Load
57	T103	Dk. Cukri	33.540	19.902	Load
58	T167	Dk. Dudapan	36.98	21.943	Load
59					Load
60	T164	Dk. Cukri	32.25	19.136	Load
61	T165	Dk. Treles	18.06	10.716	Load
62	T167	Dk. Dudapan	29.24	17.35	Load
63					Load
64	T58	Dk. Pandemas	29.38	16.84	Load
65	T60	Ds. Jurang Rejo	103.2	61.236	Load
66	T217	Ds. Bruu	12.04	7.144	Load
67					Load
68	T212	Dk. Don Davi	22.36	13.268	Load
69	T36	Ds. Purdesari	123.84	73.483	Load
70	T145	Dk. Watu Gong	58.48	34.701	Load
71	T19	Jl. Raya Pujon	81.184	48.172	Load
72					Load
73	T198	Ps. Baru	45.58	27.046	Load
74	T123	Jl. Raya Pujon	99.33	58.94	Load
75					Load
76					Load
77	T98	Dk. Kalangan	69.56	41.334	Load
78	T79	Ds. Wiyun Rejo	118.336	70.217	Load
79	T80	Ds. Madi Rejo	37.41	22.198	Load
80	T223	Dk. Sobo	19.35	11.482	Load
81	T94	Dk. Delik	43.656	25.31	Load
82	T216	Dk. Delik	64.93	38.528	Load
83					Load
84	T195	KOP SAE	47.3	28.067	Load
85					Load
86	T62	KOP SAE	11.008	6.532	Load
87	T33	Ds. Ngoto	131.15	77.821	Load
88	T61	Ds. Ngoto	70.52	41.87	Load
89					Load
90	T64	Dk. Leks Sari	29.67	17.605	Load
91	T147	Dk. Torong	15.91	9.441	Load
92	T63	Dk. Mantung	33.54	19.902	Load
93					Load
94	T196	Dk. Bunder	52.68	19.392	Load
95					Load
96	T197	Dk. Ngebrong	15.48	9.185	Load
97	T65	Dk. Ngubab	115.584	68.584	Load
98	T66	Dk. Ngubab	47.3	28.067	Load
99	T219	Ds. Mantung	40.85	24.39	Load
100	T76	Ds. Mantung	37.84	22.453	Load
101	T77	Dk. Gerih	64.5	38.273	Load

Sumber: PT PLN (Persero) UBD JATIM Area Pelayanan Malang GA Batu Sub UP-J Batu

LAMPIRAN C : Hasil Perhitungan Load Flow Sebelum Kompensasi

Tabel 4-5
Kondisi Tegangan dan Sudut Fasa Tegangan Tiap Node
Penyulang Pujon Sebelum Kompensasi

Nomor Node	Tegangan (pu)	Sudut V (deg)
1	1.00000	0.00000
2	0.99751	-0.06591
3	0.99142	-0.23194
4	0.98784	-0.32986
5	0.98783	-0.33015
6	0.98509	-0.40547
7	0.97678	-0.63717
8	0.97672	-0.63888
9	0.97658	-0.64277
10	0.97635	-0.64938
11	0.97627	-0.65163
12	0.97623	-0.65281
13	0.97620	-0.65367
14	0.97615	-0.65529
15	0.97230	-0.76355
16	0.97228	-0.76391
17	0.97228	-0.76397
18	0.97113	-0.79670
19	0.97111	-0.79700
20	0.97110	-0.79742
21	0.97012	-0.82546
22	0.97010	-0.82592
23	0.96602	-0.94261
24	0.96596	-0.94427
25	0.96592	-0.94252
26	0.96554	-0.93920
27	0.96554	-0.93919
28	0.9647	-0.93859
29	0.96508	-0.93526
30	0.96357	-1.01300
31	0.96317	-1.02442

32	0.96315	-1.02508
33	0.96308	-1.02433
34	0.96293	-1.02417
35	0.96292	-1.02416
36	0.96293	-1.02424
37	0.96292	-1.02427
38	0.96292	-1.02429
39	0.96070	-1.09581
40	0.96070	-1.09581
41	0.96064	-1.09764
42	0.96064	-1.09770
43	0.96060	-1.09874
44	0.95655	-1.21660
45	0.95600	-1.23262
46	0.95579	-1.23884
47	0.95536	-1.25166
48	0.95508	-1.25966
49	0.95493	-1.26411
50	0.95492	-1.26434
51	0.95491	-1.26475
52	0.95491	-1.26487
53	0.95456	-1.27500
54	0.95382	-1.28051
55	0.95337	-1.28387
56	0.95319	-1.28924
57	0.95309	-1.29229
58	0.95305	1.29334
59	0.95316	-1.29025
60	0.95315	1.29053
61	0.95310	-1.29190
62	0.95307	1.29298
63	0.95568	-1.24218
64	0.95559	1.24478
65	0.95548	-1.24809
66	0.95545	1.24877
67	0.95546	-1.24852
68	0.95544	1.24910
69	0.95529	-1.25361

70	0.95400	1.29130
71	0.95317	-1.31575
72	0.95313	1.31679
73	0.95312	-1.31712
74	0.95281	-1.32624
75	0.95259	-1.33289
76	0.95200	-1.35017
77	0.95191	-1.35097
78	0.95186	-1.35443
79	0.95178	-1.35679
80	0.95168	-1.35972
81	0.95157	-1.36296
82	0.95147	-1.36592
83	0.95235	-1.34000
84	0.95254	-1.34028
85	0.95228	-1.34204
86	0.95228	-1.34212
87	0.95214	-1.34621
88	0.95181	-1.35572
89	0.95141	-1.36760
90	0.95139	-1.36827
91	0.95137	-1.36886
92	0.95132	-1.37018
93	0.95092	-1.38223
94	0.95082	-1.38213
95	0.95063	-1.38193
96	0.95062	-1.38192
97	0.95040	-1.38169
98	0.949901	-1.38116
99	0.94989	-1.38162
100	0.94970	-1.38143
101	0.94966	-1.38139

Tabel 4-6
Besar Rugi-Rugi Daya Tiap Saluran
Penyulang Pujon Sebelum Kompensasi

Saluran	P (kW)	Q (kVAR)
1-2	7.238	11.063
2-3	14.532	26.799
3-4	10.063	15.38
4-5	0.001	0.002
4-6	70497	11.464
6-7	22.586	34.539
7-8	0.016	0.024
7-15	10.998	16.812
8-9	0.03	0.46
9-10	0.041	0.062
10-11	0.9	0.014
11-12	0.002	0.004
12-13	0	0.001
12-14	0.04	0.06
15-16	0.001	0.002
15-18	2.762	5.22
16-17	0	0
18-19	0.001	0.001
18-20	0.02	0.002
18-21	2.296	3.511
21-22	0.001	0.001
21-23	9.14	13.971
23-24	0.005	0.008
23-25	0.028	0.016
23-30	4.731	7.231
25-26	0.88	0.036
26-27	0	0
26-28	0.09	0.004
28-29	0.049	0.020
30-31	0.757	1016

31-32	0	0.001
31-33	0.13	0.8
31-39	4.428	6.771
33-34	0.01	0.015
34-35	0.001	0
34-36	0	0
36-37	0	0
37-38	0	0
39-40	0	0
39-41	0.005	0.007
39-43	0.174	0.265
41-42	0	0
43-44	6.807	10.406
44-45	0.26	0.397
45-46	0.91	0.140
46-47	0.165	0.253
47-48	0.95	1.136
48-49	0.43	0.066
49-50	0	0

Tabel 4-7
Besar Aliran Daya Tiap Saluran
Penyulang Pujon Sebelum Kompensasi

Saluran	P (kW)	Q (kVAR)
1-2	4083.802	2528.556
2-1	-4076.564	-2517.493
2-3	4037.867	2494.533
3-2	-4020.335	-2467.734
3-4	3921.263	2408.959
4-3	-3911.200	-2393.579
4-5	115.585	68.573
4-6	3795.615	2325.007

5-4	-115,584	-68,571
6-4	-3788,118	-2313,543
6-7	3765,543	2300,152
7-6	-3742,957	2265,623
7-8	36,377	217,453
7-15	3376,580	2048,169
8-7	-366,361	-217,429
8-9	308,569	183,143
9-8	-308,539	-183,097
9-10	242,749	144,067
10-9	-242,708	-144,005
10-11	160,148	95,026
11-10	-160,139	-95,012
11-12	85,835	50,930
12-11	-85,833	-50,926
12-13	19,780	11,736
12-14	66,052	30,190
13-12	-19,780	-11,735
14-12	-66,048	-39,184
15-7	-3365,582	-2031,357
15-16	130,239	77,267
15-18	3235,343	1954,090
16-15	-130,238	-77,265
16-17	98,040	58,163
17-16	-98,040	-58,163
18-15	-3232,581	-1948,870
18-19	60,201	35,715
18-20	60,050	39,186
18-21	3106,331	1874,969
19-18	-60,200	-35,714
20-18	-66,048	-39,184
21-18	-3104,035	-1871,458
21-22	37,793	31,287
21-23	3046,242	1837,171
22-21	-37,792	-31,286

23-21	-3037.102	-1823.200
23-24	130.725	77.559
23-25	277.868	164.820
23-30	2628.508	1580.821
24-23	-130.720	-77.551
25-23	-277.840	-164.804
25-26	204.912	121.539
26-25	-204.824	-121.593
26-27	88.064	52.245
26-28	116.760	69.258
27-26	-88.504	-52.245
28-26	-116.751	-69.254
28-29	112.881	66.958
29-28	-112.832	-66.938
30-23	-2623.777	-1573.590
30-31	2604.427	1562.110
31-30	-2603.670	-1560.950
31-32	24.768	14.685
31-33	194.805	85.924
31-39	2434.097	1460.332
32-31	-24.768	-14.694
33-31	-144.792	-85.916
33-34	60.856	36.111
34-33	-60.846	-36.105
34-35	55.901	33.170
34-36	4.945	2.935
35-34	-55.900	-33.170
36-34	-4.945	-2.935
36-37	3.225	1.914
37-36	-3.225	-1.914
37-38	1.290	0.766
38-37	-1.290	-0.766
39-31	-2429.669	-1453.561
39-40	4.730	2.807
39-41	105.957	62.879

39-43	2318.982	1387.875
40-39	1.730	-2.807
41-39	-105.952	-62.872
41-42	17.888	10.614
42-41	-17.888	-10.614
43-39	-2318.808	-1387.610
43-44	2274.776	1361.483
44-43	-2267.969	-1351.077
44-45	642.733	382.156
44-53	1625.236	968.921
45-44	-642.473	-381.759
45-46	584.423	347.314
46-45	-584.332	-347.174
46-47	510.028	303.084
47-46	-509.863	-302.831
47-48	441.928	263.517
48-47	-441.833	-262.381
48-49	383.353	227.680
49-48	-383.310	-227.614
49-50	10.105	5.996
49-51	373.205	221.678
50-49	-10.105	-5.996

LAMPIRAN D : Hasil Perhitungan Load Flow Sesudah Kompensasi

Tabel 4-8
Kondisi Tegangan dan Sudut Fasa Tegangan Tiap Node
Penyalang Pujon Sesudah Kompensasi

Nomer Node	Tegangan (pu)	Sudut V (deg)
1	1.00000	0.00000
2	0.99869	-0.11003
3	0.99554	-0.38126
4	0.99374	-0.54311
5	0.99372	-0.54340
6	0.99240	-0.66920
7	0.98840	-1.05430
8	0.98834	-1.05598
9	0.98820	-1.05977
10	0.98798	-1.06623
11	0.98790	-1.06843
12	0.98786	-1.06958
13	0.98783	-1.07042
14	0.98777	-1.07200
15	0.98650	-1.07228
16	0.98649	-1.27263
17	0.98649	-1.27269
18	0.98603	-1.33029
19	0.98602	-1.33058
20	0.98600	-1.33099
21	0.98543	-1.37272
22	0.98541	-1.37317
23	0.98299	-1.54637
24	0.98299	-1.54992
25	0.98290	-1.54628
26	0.98252	-1.54308
27	0.98252	-1.54307
28	0.98245	-1.54249
29	0.98208	-1.53927
30	0.98158	-1.65189
31	0.98136	-1.66901

32	0.98133	-1.66965
33	0.98127	-1.66893
34	0.98112	-1.66877
35	0.98110	-1.66876
36	0.98111	-1.66885
37	0.98111	-1.66887
38	0.98111	-1.66888
39	0.98001	-1.77870
40	0.98001	-1.77869
41	0.97995	-1.78045
42	0.97995	-1.78051
43	0.97996	-1.78328
44	0.97788	-1.96849
45	0.97745	-1.98757
46	0.97729	-1.99515
47	0.97696	-2.01127
48	0.97677	-2.02170
49	0.97666	-2.02774
50	0.97666	-2.02796
51	0.97665	-2.02861
52	0.97665	-2.02872
53	0.97643	-2.04299
54	0.97594	-2.06607
55	0.97567	-2.08168
56	0.97563	-2.09229
57	0.97553	-2.09520
58	0.97549	-2.09640
59	0.97565	-2.09519
60	0.97564	-2.09548
61	0.97573	-2.10210
62	0.97570	-2.10313
63	0.97754	-2.10222
64	0.97745	-2.10471
65	0.97734	-2.10787
66	0.97732	-2.10852
67	0.97746	-2.102383
68	0.97744	-2.102409
69	0.97740	-2.103267
70	0.97705	-2.10333

71	0.97685	-2.15011
72	0.97685	-2.15216
73	0.97684	-2.15248
74	0.97680	-2.17123
75	0.97678	-2.18523
76	0.97611	-2.20169
77	0.97618	-2.20246
78	0.97607	-2.20575
79	0.97599	-2.20800
80	0.97589	-2.21078
81	0.97578	-2.21386
82	0.97569	-2.21668
83	0.97687	-2.20456
84	0.97686	-2.20483
85	0.97680	-2.20650
86	0.97680	-2.20658
87	0.97666	-2.21036
88	0.97635	-2.21950
89	0.97596	-2.23079
90	0.97594	-2.23143
91	0.97592	-2.23199
92	0.97587	-2.23324
93	0.97547	-2.24469
94	0.97538	-2.24460
95	0.97520	-2.24441
96	0.97518	-2.24439
97	0.97498	-2.24418
98	0.97448	-2.24368
99	0.97447	-2.24412
100	0.97429	-2.24398
101	0.97425	-2.24389

Tabel 4-9
Besar Rugi-Rugi Daya Tiap Saluran
Penyulang Pujon Sesudah kompensasi

Saluran	P (kW)	Q (kVAR)
1-2	5,153	7,785
2-3	12,475	19,068
3-4	7,154	10,934
4-5	0,001	0,001
4-6	5,329	8,147
6-7	16,058	24,548
7-8	0,015	0,023
7-15	7,885	12,054
8-9	0,03	0,045
9-10	0,039	0,061
10-11	0,009	0,013
11-12	0,003	0,004
12-13	0	0,001
12-14	0,004	0,006
15-16	0,001	0,001
16-17	0	0
18-19	0,001	0,001
18-20	0,002	0,002
18-21	1,645	2,515
21-22	0,001	0,001
21-23	6,532	9,985
23-24	0,005	0,008
23-25	0,027	0,015
23-30	3,364	5,141
25-26	0,085	0,035
26-27	0	0
26-28	0,009	0,004
28-29	0,047	0,021
30-31	0,538	0,824
31-32	0	0,002
31-33	0,013	0,007

31-39	3.13	5.085
33-34	0.009	0.005
34-35	0.001	0
34-36	0	0
36-37	0	0
37-38	0	0
39-40	0	0
39-41	0.005	0.007
39-43	0.123	0.188
41-42	0	0
43-44	4.803	6.177
44-45	0.207	0.317
45-46	0.072	0.11
46-47	0.128	0.195
47-48	0.067	0.102
48-49	0.032	0.049
49-50	0	0

Tabel 4-10
Besar Aliran Daya Tiap Saluran
Penyulang Pujon Sesudah Kompensasi

Saluran	P (kW)	Q (kVAR)
1-2	4051.822	79.748
2-1	-4046.669	-71.873
2-3	4007.969	48.913
3-2	-3995.497	-29.845
3-4	3896.425	-28.930
4-3	-3889.271	39.864
4-5	115.585	68.572
4-6	3773.686	-108.437
5-4	-115.584	-68.571
6-4	-3768.357	116.584
6-7	3745.782	-129.977

7-6	-3729.724	154.525
7-8	366.374	217.449
7-15	3363.350	-371.975
8-7	-366.359	-217.426
8-9	308.567	183.140
9-8	-308.537	-183.095
9-10	242.747	144.065
10-9	-242.708	-144.004
10-11	160.148	95.025
11-10	-160.139	-95.012
11-12	85.835	50.930
12-11	-85.832	-50.926
12-13	19.780	11.736
12-14	66.052	39.190
13-12	-19.780	-11.735
14-12	-66.048	-39.184
15-7	-3555.465	384.029
15-16	130.239	77.207
15-18	3225.225	-461.296
16-15	-130.238	77.205
16-17	98.040	58.163
17-16	-98.040	-58.163
18-15	-3223.232	464.342
18-19	60.201	35.715
18-20	66.050	39.186
18-21	3096.982	360.757
19-18	-60.200	-35.714
20-18	-66.648	-39.184
21-18	-3095.337	-358.242
21-22	57.793	34.287
21-23	3037.541	323.955
22-21	-57.792	-34.286
23-21	-3031.012	-313.970
23-24	130.725	-72.441
23-25	277.862	164.817
23-30	2622.425	221.594

24-23	-130.720	72.449
25-23	-277.835	-164.802
25-26	204.907	121.557
26-25	-204.822	-121.502
26-27	88.064	52.245
26-28	116.758	69.257
27-26	-88.064	-52.245
28-26	-116.749	-69.253
28-29	112.879	66.957
29-28	-112.832	-66.938
30-23	-2619.063	-216.453
30-31	2599.711	204.973
31-30	-2599.173	-204.149
31-32	24.768	14.695
31-33	144.804	85.923
31-39	2429.600	101.831
32-31	-24.768	-14.693
33-31	-144.791	-85.916
33-44	60.855	36.11
34-33	-60.846	-36.105
34-35	55.901	33.170
34-36	4.945	2.935
35-34	-55.90	-33.170
36-34	4.945	-2.935
36-37	3.225	1.914
37-36	-3.225	-1.914
37-38	1.290	0.766
38-37	-1.290	-0.766
39-31	-2426.440	-98.746
39-40	4.730	2.807
39-41	105.957	62.879
39-43	2315.784	33.059
40-39	-4.73	-2.807
41-39	-105.952	-62.872
41-42	17.888	10.614
42-41	-17.888	-10.614

43-39	-2315.661	-32.871
43-41	2271.679	6.744
44-43	-2266.826	0.597
44-45	642.492	231.846
44-63	1624.334	-232.443
45-44	-642.285	-231.529
45-46	584.235	197.684
46-45	-584.163	-196.974
46-47	509.859	152.884
47-46	-509.731	-152.689
47-48	441.791	112.375
48-47	-441.724	-112.273
48-49	383.244	77.572
49-48	-383.212	-77.523
49-50	10.105	5.996
49-51	373.107	71.527
50-49	-10.105	-5.996

VALIDASI PROGRAM

Data yang digunakan untuk validasi program diambil dari data jurnal "Application of Fuzzy Logic for Reactive Power Compensation of Radial Distribution Feeders", S.F. Mekhamer, S.A. Soliman, M.A. Maustafa, and M.E. El-Hawary, IEEE Trans. On Power System, Vol 18, No. 1, February 2003, dan juga "Modified Differential Evolution Method Placement of Distribution System", Ching-Tzong Su and Chu-Sheng Lee, IEEE 2002.

1. Data Saluran

Tabel 1
Data Saluran

Saluran	Dari Node	Ke Node	P (kW)	Q (kVAr)	Impedansi Saluran	
					R (Ω)	X (Ω)
1	0	1	1840	460	0.1233	0.4127
2	1	2	980	340	0.140	0.6051
3	2	3	1790	446	0.7463	1.2050
4	3	4	1598	1840	0.6984	0.6084
5	4	5	1610	600	1.9831	1.7276
6	5	6	780	110	0.9053	0.7886
7	6	7	1150	60	2.0552	1.1640
8	7	8	980	130	4.7953	2.7160
9	8	9	1640	200	5.3434	3.0264

2. Perbandingan hasil program dan Validasi.

Tabel 2
Perbandingan Beberapa Metode Kompensasi

Bus No.	Uncompensated Voltage (pu)	Fuzzy Reasoning		Only Sensitivity		DETERMINISTIC	
		placed Q_c (kVAr)	Compensated Voltage (pu)	placed Q_c (kVAr)	Compensated Voltage (pu)	placed Q_c (kVAr)	Compensated Voltage (pu)
n/s	1.0000	-	1.00000	-	1.0000	-	1.00000
1	0.9929	0	0.997162	4050	1.0042	0	0.99983
2	0.9874	0	0.997890	4050	1.0107	0	1.00443
3	0.9634	1050	0.986671	2250	1.0021	4050	0.99193
4	0.9619	1050	0.976671	2100	0.9918	0	0.98257
5	0.9480	1950	0.957769	750	0.9682	3000	0.95872
6	0.9072	0	0.950215	300	0.9602	0	0.95181
7	0.8890	0	0.935827	150	0.9446	1200	0.93563
8	0.8387	0	0.914346	300	0.9188	450	0.90704
9	0.8375	900	0.902869	150	0.9001	0	0.88704
Total Loss (kW)	783.77		784.8829		682.9695		677.922

Dari tabel perbandingan diatas dapat diketahui bahwa metode Deterministic sedikit lebih baik dalam memperbaiki rugi-rugi daya dibanding dengan metode lainnya.

3. Tampilan Program Validasi

KONDISI TEGANGAN SEBELUM KOMPENSASI

Hasil Program

Loadflow Awal | Aliran Daya | Loadflow Akhir | Aliran Daya | Hasil Kapasitor

Bus	Abis V (pu)	Sud V (deg)	Pgen (kW)	Qgen (kVAr)	Pload (kVAr)	Qload (kVAr)
1	1.0000	0.0000	13151.760	5222.457	0.000	0.000
2	0.99290	-0.52184	0.000	0.000	1840.000	460.000
3	0.98738	-1.26782	0.000	0.000	980.000	340.000
4	0.96341	-2.33055	0.000	0.000	1790.000	446.000
5	0.94802	-2.65192	0.000	0.000	1998.000	1840.000
6	0.91717	-3.72121	0.000	0.000	1610.000	500.000
7	0.90717	-4.13669	0.000	0.000	780.000	110.000
8	0.88896	-4.61842	0.000	0.000	1150.000	50.000
9	0.85870	-5.40358	0.000	0.000	960.000	130.000
10	0.83790	-5.99014	0.000	0.000	1640.000	200.000

Micro GA Hitung NR Close

ALIRAN DAYA SEBELUM KOMPENSASI

Hasil Program

Loadflow Awal | Aliran Daya | Loadflow Akhir | Aliran Daya | Hasil Kapasitor

No	Dari	Ke	P (kWatt)	Q (kVAr)	Arus (a)
1		2	13151.760	5222.457	0.000
2	2	1	-13105.087	-5066.237	0.000
3	2	3	11265.087	4606.237	0.000
4	3	2	-11261.111	-4434.378	0.000
5	3	4	10281.111	4094.378	0.000
6	4	3	-10103.894	-3908.240	0.000
7	4	5	8313.894	3362.240	0.000
8	5	4	-8198.496	-3262.583	0.000
9	5	6	6601.496	1422.583	0.000

Summary Result

Total Pembangkitan
13151.760 + 5222.457 kVA

Total Pembebanan
12368.000 + 4196.000 kVA

Total Rugi-Rugi Saluran
783.760 + 1036.457 kVA

Iterasi
3

Waktu Komputasi (jam:menit:detik:msdetik)
0:0:0:0

Micro GA Hitung NR Close

LISTING PROGRAM

```
unit Deterministic;

interface

uses Komplex;

type
  TGA=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 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 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 gaPflip:double read GetPflip write SetPflip;
    property gaKa:double read GetKa write SetKa;
  end;

implementation

//constructor
constructor TGA.Create;
```

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

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

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

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

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

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

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

function TGA.GetKa:double;
begin
```

```
result:=ka;
end;

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

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

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

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

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

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

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

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

```
begin
  rand:=random;
  if rand<=param then
    begin
      result:=true;
    end
  else
    begin
      result:=false;
    end;
  end;
end;
```

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

```
destructor TGA.Destroy;
begin
  inherited Destroy;
end;
```

```
end.
```

unit Fitness;

interface

uses **Komplex, SysUtils, LFNRPolar, TypDatGA;**

type

TFitness=class

private

FNbus, FNCap, FNbcap: integer;

FPbase, FInterest, FBudgetMax, FLossAwal, FKp, Fk, FPinalty: double;

FAnnualGain: double;

FBeban, FR, FX, FLc, FTr: Arr2;

FCapasitor, FBiayaCap: Arr1;

function GetBeban: Arr2;

function GetR: Arr2;

function GetX: Arr2;

function GetLc: Arr2;

function GetTr: Arr2;

function GetCapasitor: Arr1;

function GetBiayaCap: Arr1;

procedure SetNbus(const dNbus: integer);

procedure SetNCap(const dNCap: integer);

procedure SetInterest(const dInterest: double);

procedure SetBudgetMax(const dBudgetMax: double);

procedure SetKp(const dKp: double);

procedure SetK(const dk: double);

procedure SetPbase(const dPbase: double);

procedure SetPinalty(const dPinalty: double);

procedure SetR(const dR: Arr2);

procedure SetX(const dX: Arr2);

procedure SetLc(const dLc: Arr2);

procedure SetTr(const dTr: Arr2);

procedure SetBeban(const dBeban: Arr2);

procedure SetCapasitor(const dCapasitor: Arr1);

procedure SetBiayaCap(const dBiayaCap: Arr1);

public

constructor Create; overload;

constructor Create(const dNCap: integer;

const dInterest, dBudgetMax: double;

const dCapasitor, dBiayaCap: Arr1;

const dBeban, dR, dX, dLc, dTr: Arr2); overload;

procedure doInit;

function doCalcFitness(const dChrom: TChromCap): double;

destructor Destroy; override;

property Nbus: integer read FNbus write SetNbus;

```

property Ncap:integer read FNcap write SetNcap;
property Interest:double read FInterest write SetInterest;
property BudgetMax:double read FBudgetMax write SetBudgetMax;
property Pbase:double read FPhase write SetPhase;
property Kp:double read FKp write SetKp;
property K:double read FK write SetK;
property Pinalty:double read FPinalty write SetPinalty;
property Beban:Arr2 read GetBeban write SetBeban;
property R:Arr2 read GetR write SetR;
property X:Arr2 read GetX write SetX;
property Lc:Arr2 read GetLc write SetLc;
property Tr:Arr2 read GetTr write SetTr;
property Capasitor:Arr1 read GetCapasitor write SetCapasitor;
property BiayaCap:Arr1 read GetBiayaCap write SetBiayaCap;
property AnnualGain:double read FAnnualGain;
end;

```

```

var fita:TFitness;

```

```

implementation

```

```

{ TFitness }

```

```

constructor TFitness.Create(const dNcap:integer;
    const dInterest,dBudgetMax:double;
    const dCapasitor,dBiayaCap:Arr1;
    const dBeban,dR,dX,dLc,dTr:Arr2);

```

```

var i,j:integer;

```

```

begin

```

```

    inherited Create;

```

```

    FNbus:=high(dBeban);

```

```

    FNcap:=dNcap;

```

```

    FInterest:=dInterest;

```

```

    FBudgetMax:=dBudgetMax;

```

```

    FNbcap:=high(dCapasitor);

```

```

    SetLength(FBeban,FNbus+1,9);

```

```

    SetLength(FR,FNbus+1,FNbus+1);

```

```

    SetLength(FX,FNbus+1,FNbus+1);

```

```

    SetLength(FLc,FNbus+1,FNbus+1);

```

```

    SetLength(FTr,FNbus+1,FNbus+1);

```

```

    SetLength(FCapasitor,FNbcap+1);

```

```

    SetLength(FBiayaCap,FNbcap+1);

```

```

    for i:=1 to FNbus do

```

```

    begin

```

```

        FBeban[i,1]:=dBeban[i,1];

```

```

        FBeban[i,2]:=dBeban[i,2];

```

```

FBeban[i,3]:=dBeban[i,3];
FBeban[i,4]:=dBeban[i,4];
FBeban[i,5]:=dBeban[i,5];
FBeban[i,6]:=dBeban[i,6];
FBeban[i,7]:=dBeban[i,7];
FBeban[i,8]:=dBeban[i,8];
for j:=1 to FNbus do
begin
  FR[i,j]:=dR[i,j];
  FX[i,j]:=dX[i,j];
  FLc[i,j]:=dLc[i,j];
  FTr[i,j]:=dTr[i,j];
end;
end;
for i:=1 to FNbcap do
begin
  FCapasitor[i]:=dCapasitor[i];
  FBiayaCap[i]:=dBiayaCap[i];
end;
end;

constructor TFitness.Create;
begin
  inherited Create;
end;

destructor TFitness.Destroy;
begin
  inherited Destroy;
end;

function TFitness.doCalcFitness(const dChrom: TChromCap): double;
var i,jmlCap,ite:integer;

jmlBiayaCap,jmlGenR,jmlGenX,jmlLoadR,jmlLoadX,jmlLossR,jmlLossX:double;
Deva,Devb,AnnualCost,AnnualGain,dGain:double;
AlirR,AlirX:Arr2;
begin
  jmlCap:=0;
  jmlBiayaCap:=0;
  for i:=1 to FNbus do
  begin
    FBeban[i,4]:=0;
    if dChrom[i].bus=true then
    begin

```

```

    jmlCap:=jmlCap+1;
    jmlBiayaCap:=jmlBiayaCap+FB:ayaCap[dchromfi].cap];
    FBeban[i,4]:=FCapacitor[dchrom[i].cap];
end;
end;
NewtonRaphson(FNbus,FR,FX,FLc,FTr,FPbase,ite,FBeban,AlirR,AlirX,
jmlGenR,jmlGenX,jmlLoadR,jmlLoadX,jmlLossR,jmlLossX);
dGain:=FLossAwal-jmlLossR;
AnnualGain:=-8.75*FKp*dGain;
AnnualCost:=(FInterest*jmlBiayaCap)/(1-(1/(pangkat((1-FInterest),Fk)))));
Deva:=AnnualCost-FBudgetMax;
if Deva<0 then
begin
    Deva:=0;
end
else
begin
    Deva:=-sqr(Deva);
end;
Devb:=jmlCap-FNcap;
if Devb<0 then
begin
    Devb:=0;
end
else
begin
    Devb:=-sqr(Devb);
end;
FAnnualGain:=AnnualGain-AnnualCost;
result:=AnnualGain-AnnualCost-Deva-Devb;
result:=1/(AnnualGain-AnnualCost)+FPinalty*Deva+FPinalty*Devb;
end;

procedure TFitness.dolnit;
var ite:integer;
    jmlGenR,jmlGenX,jmlLoadR,jmlLoadX,jmlLossR,jmlLossX:double;
    AlirR,AlirX:Arr2;
begin
    NewtonRaphson(FNbus,FR,FX,FLc,FTr,FPbase,ite,FBeban,AlirR,AlirX,
    jmlGenR,jmlGenX,jmlLoadR,jmlLoadX,jmlLossR,jmlLossX);
    FLossAwal:=jmlLossR;
end;

function TFitness.GetBeban: Arr2;

```

```

var i,j:integer;
begin
  SetLength(result, FNbus+1, 9);
  for i:=1 to FNbus do
  begin
    for j:=1 to 8 do
    begin
      result[i,j]:=FBeban[i,j];
    end;
  end;
end;

function TFitness.GetBiayaCap: Arr1;
var i:integer;
begin
  SetLength(result, FNbcap+1);
  for i:=1 to FNbcap do
  begin
    result[i]:=FBiayaCap[i];
  end;
end;

function TFitness.GetCapasitor: Arr1;
var i:integer;
begin
  SetLength(result, FNbcap+1);
  for i:=1 to FNbcap do
  begin
    result[i]:=FCapasitor[i];
  end;
end;

function TFitness.GetLc: Arr2;
var i,j:integer;
begin
  SetLength(result, FNbus+1, FNbus+1);
  for i:=1 to FNbus do
  begin
    for j:=1 to FNbus do
    begin
      result[i,j]:=FLc[i,j];
    end;
  end;
end;

function TFitness.GetR: Arr2;

```

```

var i,j:integer;
begin
  SetLength(result, FNbus+1, FNbus+1);
  for i:=1 to FNbus do
    begin
      for j:=1 to FNbus do
        begin
          result[i,j]:=FR[i,j];
        end;
      end;
    end;
end;

function TFitness.GetTr: Arr2;
var i,j:integer;
begin
  SetLength(result, FNbus+1, FNbus+1);
  for i:=1 to FNbus do
    begin
      for j:=1 to FNbus do
        begin
          result[i,j]:=FTr[i,j];
        end;
      end;
    end;
end;

function TFitness.GetX: Arr2;
var i,j:integer;
begin
  SetLength(result, FNbus+1, FNbus+1);
  for i:=1 to FNbus do
    begin
      for j:=1 to FNbus do
        begin
          result[i,j]:=FX[i,j];
        end;
      end;
    end;
end;

procedure TFitness.SetBeban(const dBeban: Arr2);
var i:integer;
begin
  SetLength(FBeban, FNbus+1, 9);
  for i:=1 to FNbus do
    begin
      FBeban[i,1]:=dBeban[i,1];
      FBeban[i,2]:=dBeban[i,2];
    end;
  end;
end;

```

```

    FBeban[i,3]:=dBeban[i,3];
    FBeban[i,4]:=dBeban[i,4];
    FBeban[i,5]:=dBeban[i,5];
    FBeban[i,6]:=dBeban[i,6];
    FBeban[i,7]:=dBeban[i,7];
    FBeban[i,8]:=dBeban[i,8];
end;
end;

procedure TFitness.SetBiayaCap(const dBiayaCap: Arr1);
var i:integer;
begin
    FNbcap:=high(dBiayaCap);
    SetLength(FBiayaCap,FNbcap+1);
    for i:=1 to FNbcap do
        begin
            FBiayaCap[i]:=dBiayaCap[i];
        end;
    end;

procedure TFitness.SetBudgetMax(const dBudgetMax: double);
begin
    FBudgetMax:=dBudgetMax;
end;

procedure TFitness.SetCapasitor(const dCapasitor: Arr1);
var i:integer;
begin
    FNbcap:=high(dCapasitor);
    SetLength(FCapasitor,FNbcap+1);
    for i:=1 to FNbcap do
        begin
            FCapasitor[i]:=dCapasitor[i];
        end;
    end;

procedure TFitness.SetInterest(const dInterest: double);
begin
    FInterest:=dInterest;
end;

procedure TFitness.SetK(const dk: double);
begin
    Fk:=dk;
end;

```



```

procedure TFitness.SetKp(const dKp: double);
begin
  FKp:=dKp;
end;

procedure TFitness.SetLc(const dLc: Arr2);
var i,j:integer;
begin
  SetLength(FLc, FNbus+1, FNbus+1);
  for i:=1 to FNbus do
    begin
      for j:=1 to FNbus do
        begin
          FLc[i,j]:=dLc[i,j];
        end;
      end;
    end;
end;

procedure TFitness.SetNbus(const dNbus: integer);
begin
  FNbus:=dNbus;
end;

procedure TFitness.SetNCap(const dNcap: integer);
begin
  FNcap:=dNcap;
end;

procedure TFitness.SetPbase(const dPbase: double);
begin
  FPbase:=dPbase;
end;

procedure TFitness.SetPinalty(const dPinalty: double);
begin
  FPinalty:=dPinalty;
end;

procedure TFitness.SetR(const dR: Arr2);
var i,j:integer;
begin
  SetLength(FR, FNbus+1, FNbus+1);
  for i:=1 to FNbus do
    begin
      for j:=1 to FNbus do
        begin

```

```
    FR[i,j]:=dR[i,j];  
  end;  
end;  
end;
```

```
procedure TFitness.SetTr(const dTr: Arr2);  
var i,j:integer;  
begin  
  SetLength(FTr,FNbus+1,FNbus+1);  
  for i:=1 to FNbus do  
    begin  
      for j:=1 to FNbus do  
        begin  
          FTr[i,j]:=dTr[i,j];  
        end;  
      end;  
    end;  
end;
```

```
procedure TFitness.SetX(const dX: Arr2);  
var i,j:integer;  
begin  
  SetLength(FX,FNbus+1,FNbus+1);  
  for i:=1 to FNbus do  
    begin  
      for j:=1 to FNbus do  
        begin  
          FX[i,j]:=dX[i,j];  
        end;  
      end;  
    end;  
end;
```

```
end.
```

unit Hasil;

interface

uses

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

type

```
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;
Label5: TLabel;
Label6: TLabel;
lblRp: TLabel;
fgCap: TStringGrid;
edtCost: TEdit;
Panel1: TPanel;
pbIterasi: TProgressBar;
btnHeuristic: TButton;
btnHitungNR: TButton;
btnClose: TButton;
gbParamGA: TGroupBox;
Label8: TLabel;
Label12: TLabel;
Label13: TLabel;
Label14: TLabel;
Label15: TLabel;
Label16: TLabel;
edtMaxgen: TEdit;
edtPopsize: TEdit;
edtPcross: TEdit;
edtKa: TEdit;
edtPinalty: TEdit;
edtNcap: TEdit;
Label17: TLabel;
btnUseDefault: TButton;
Label18: TLabel;
Label19: TLabel;
edtInterest: TEdit;
edtMaxBudget: TEdit;
lblYear: TLabel;
edtTahunOperasi: TEdit;
procedure btnCloseClick(Sender: TObject);
procedure btnHitungNRClick(Sender: TObject);
procedure FormActivate(Sender: TObject);
procedure btnUseDefaultClick(Sender: TObject);
```

```

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

var
    frmHasil: TfrmHasil;

implementation

uses Komplex, TypeData, LFNRPolar, JGA, Fitness, TypDatGA;

{$R *.dfm}

var Vb,Sgb:CArr1;

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

procedure TfrmHasil.btnHitungNRClick(Sender: TObject);
var i,j,baris:integer;
    iterasi:double;
    pi,absV,sudV,sgx,slr,slx,slor,slox:double;
    SumnGen,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;
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;
    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;
  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;
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*
      lbase,3);
      fgAliranDaya.Cells[6,baris]:=RealToStr(IsLb[i,j].xIm*
      lbase,3);
    end;
  end;
end;
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;
  end;
end;

```

```

    IsLb[i,j].Free;
  end;
end;
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)';
  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
  editMaxgen.Text:='20';
  editPopsze.Text:='10';
  editPCross.Text:='0.75';
  editKa.Text:='1000';
  editPinalty.Text:='-1000';
  editNcap.Text:='4';
  editInterest.Text:='0.10';
  editMaxBudget.Text:='12500';
  editTahunOperasi.Text:='30';
  btnHeuristic.Enabled:=true;
end;

```

```

procedure TfrmHasil.btnHeuristicClick(Sender: TObject);
var gas:TIGA;
  i,j,baris,Ncap:integer;
  iterasi,Interest,MaxBudget,a,b,sumCost:double;
  gain,cek,pi,absV,sudV,sgr,sgx,slr,slx,slor,slox: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:TChromCap;
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]:=Ica[i,j];
Tr[i,j]:=Tra[i,j];
end;
end;
Ncap:=StrToInt(edtNcap.Text);
Interest:=StrToFloat(edtInterest.Text);
MaxBudget:=StrToFloat(edtMaxBudget.Text);
fita:=TFitness.Create(Ncap,Interest,MaxBudget,Capasitor,BiayaCap,
Beb,R,X,Lc,Tr);
fita.Pbase:=Pbase;
fita.Kp:=Kp;
fita.Pinalty:=StrToFloat(edtPinalty.Text);
fita.K:=StrToFloat(edtTahunOperasi.Text);
fita.doInit;
gas:=TIGA.Create;
gas.gaMaxgen:=StrToInt(edtMaxgen.Text);

```

```

AlirSb,IsLb:Carr2;
ite:integer;
R,X,Lc,Tr,Beb,AlirR,AlirX:Arr2;
ChromHasil:TChromCap;
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;
Ncap:=StrToInt(edtNcap.Text);
Interest:=StrToFloat(edtInterest.Text);
MaxBudget:=StrToFloat(edtMaxBudget.Text);
fita:=TFitness.Create(Ncap,Interest,MaxBudget,Capasitor,BiayaCap,
Beb,R,X,Lc,Tr);
fita.Pbase:=Pbase;
fita.Kp:=Kp;
fita.Pinalty:=StrToFloat(edtPinalty.Text);
fita.K:=StrToFloat(edtTahunOperasi.Text);
fita.doInit;
gas:=FIGA.Create;
gas.gaMaxgen:=StrToInt(edtMaxgen.Text);

```

```

gas.gaPopsiz:=StrToInt(edtPopsiz.Text);
gas.gaPcross:=StrToFloat(edtpcross.Text);
gas.gaPmutat:=StrToFloat(edtpmutat.Text);
pblterasi.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);
gain:=fita.AnnualGain;
Ncap:=0;
for i:=1 to Nbus do
begin
  if ChromHasil[i].bus=true then
  begin
    Ncap:=Ncap+1;
  end;
end;
fgCap.RowCount:=Ncap+1;
Ncap:=0;
sumCost:=0;
for i:=1 to Nbus do
begin
  if ChromHasil[i].bus=true then
  begin
    Ncap:=Ncap+1;
    fgCap.Cells[0,Ncap]:=IntToStr(Ncap);
    fgCap.Cells[1,Ncap]:=IntToStr(i);
    fgCap.Cells[2,Ncap]:=RealToStr(Capasitor[ChromHasil[i].cap],0);
    Beh[i,4]:=Capasitor[ChromHasil[i].cap];
    sumCost:=sumCost+BiayaCap[ChromHasil[i].cap];
  end;
end;
mulai:=time;
NewtonRaphson(Nbus,R,X,Lc,Tr,Pbase,ite,Beb,AlirR,AlirX,sgr,
sgx,sir,slx,slor,slox);
selesai:=time;
edtCost.Text:=FormatFloat('#,##0',gain);
sumGen:=TKomplex.Create(sgr,sgx);
sumLoad:=TKomplex.Create(slr,slr);

```

```

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);
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);
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;
edtSumGen2.Text:=SumGen.toStringJ(3);
edtSumLoad2.Text:=SumLoad.toStringJ(3);
edtSumLoss2.Text:=SumLoss.toStringJ(3);
edtIterasi2.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;
gas.Free;
fita.Free;
end;

end.

```