

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



**ANALISA ALIRAN DAYA PADA SISTEM DISTRIBUSI
MENGUNAKAN METODE *THREE CURRENT INJECTION
METHOD (TCIM)* DI G.I. BLIMBING MALANG**

SKRIPSI

**Disusun Oleh :
PAULUS HARRY D. L
NIM : 95 12 008**

MARET 2007

LEMBAR PERSETUJUAN

**ANALISA ALIRAN DAYA PADA SISTEM DISTRIBUSI
MENGUNAKAN METODE *THREE CURRENT INJECTION
METHOD (TCIM)* DI G.I. BLIMBING MALANG**

SKRIPSI

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

Disusun Oleh :

**PAULUS HARRY D. L
95. 12. 008**



**Mengetahui,
Ketua Jurusan,**

**Menyetujui,
Dosen Pembimbing,**

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

**Ir. H. Taufik Hidayat, MT
NIP. Y. 1018700151**

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

ABSTRAKSI

ANALISA ALIRAN DAYA PADA SISTEM DISTRIBUSI MENGUNAKAN METODE *THREE CURRENT INJECTION METHOD (TCIM)* DI G.I. BLIMBING MALANG

**(Paulus Harry D.L, 95.12.008, Teknik Elektro/Teknik Energi Listrik S-1)
(Dosen Pembimbing : Ir. H. Taufik Hidayat, MT)**

Jaringan distribusi tipe radial merupakan tipe jaringan yang banyak digunakan untuk mendistribusikan daya listrik pada tegangan menengah. Pertumbuhan beban dan perluasan jaringan adalah hal yang biasa terjadi sebagai akibat dari permintaan konsumen yang cenderung meningkat seiring bertambahnya jumlah penduduk dan taraf hidup masyarakat. Namun dengan bertambahnya panjang saluran pada sistem akan menyebabkan rugi-rugi daya yang cukup besar dan penurunan tegangan yang tidak standart yang ditetapkan yaitu antara -10% sampai 5%. Pada kenyataannya beban tidak seimbang sehingga perlu studi aliran daya untuk menghitung rugi-rugi daya dan penurunan tegangan pada saluran tiga fasa dengan beban tidak seimbang secara cepat. Metode alternatif yang bisa digunakan adalah metode *Current Injection (CIM)* Tiga Fasa karena metode ini dapat menganalisa aliran daya dan rugi-rugi daya pada saluran.

Kata Kunci : *Aliran Daya Tidak Seimbang, Aliran Daya Metode Current Injection, Aliran Daya Jaringan Distribusi.*

ABSTRAKSI

ENERGY STREAM ANALYSIS AT DISTRIBUTION SYSTEM USE THREE CURRENT INJECTION METHOD (TCIM) IN MAINS WATCHMAN STATION BLIMBING MALANG

**Paulus Harry D.L, 95.12.008,
Technique Electro/Technique Of Energy Electrics Of S-1
(Lecturer Counsellor : Ir. H. Taufik Hidayat,MT)**

Distribution type network of radial represent network type which used many to distribute electricity at average tension. Growth of burden and extension of network is commonplace happened in consequence of request of consumer which tend to increase along increasing it is amount of level and resident live society. But by increasing channel length at system will cause big enough energy loss and degradation of tension which do not applied standart that is between - 10% until 5%. Practically uneven burden so that need energy stream study to calculate energy loss and degradation of tension at channel three phasa with burden of uneven quickly. Alternative method which can be used by method of Current Injection (CIM) Three Phase because this method can analyse energy stream and energy loss at channel.

Key Word : *Uneven Stream Energy, Stream Energy Method of Current Injection, Stream Energy Network Distribution.*

KATA PENGANTAR

Puji dan Syukur kehadiran Tuhan Yang Maha Kuasa, karena dengan segala Rahmat-Nya. Penulis dapat menyelesaikan Skripsi yang berjudul :

"ANALISA ALIRAN DAYA PADA SISTEM DISTRIBUSI MENGGUNAKAN METODE THREE CURRENT INJECTION METHOD (TCIM)"

Skripsi ini disusun sebagai salah satu persyaratan untuk memperoleh derajat sarjana (S1) pada Jurusan Teknik Elektro Energi Elektrik Di Institut Teknologi Nasional - Malang

Skripsi ini dapat terselesaikan juga berkat bantuan dari berbagai pihak untuk itu penulis ingin mengucapkan terima kasih kepada, yang terhormat :

1. Bapak **Prof. Dr. Ir. Abraham Lomi, MSEE**, selaku Rektor Intitut Teknologi Nasional (ITN) - Malang
2. Bapak **Ir. Mochtar Asroni, MSME**, selaku Dekan Fakultas Teknologi Industri (FTI) Institut Teknologi Nasional Malang
3. Bapak **Ir. F. Yudi Limpraptono, MT**, selaku Ketua Jurusan Teknik Elektro ITN-Malang.
4. Bapak **Ir. H. Taufik Hidayat, MT**, selaku Dosen Pembimbing.
5. Seluruh dosen dan staf akademik Institut Teknologi Nasional Malang yang telah memberikan saran dan masukan untuk perbaikan skripsi ini.
6. Alm. Bapak dan Ibu beserta keluarga yang telah memberikan dukungan moral dan spiritual.
7. Sahabat dan teman-teman seperjuangan yang selama ini telah memberikan bantuan dan motivasinya kepada penulis sehingga skripsi ini terselesaikan dengan cepat.

Penulis menyadari bahwa skripsi ini masih belum sempurna, untuk itu penulis sangat mengharapkan kritik dan saran yang bersifat membangun dari semua pihak buat kesempurnaan skripsi ini

Malang , April 2007

Penulis

DAFTAR ISI

	HAL
KATA PENGANTAR.....	i
LEMBAR PERSETUJUAN.....	ii
DAFTAR ISI.....	iii
DAFTAR GAMBAR.....	iv
DAFTAR TABEL.....	vi
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	2
1.3. Tujuan.....	2
1.4. Batasan Masalah.....	2
1.5. Metode Penelitian.....	3
1.6. Sistematika Penulisan.....	3
BAB II SISTEM DISTRIBUSI TENAGA LISTRIK.....	5
2.1. Sistem Distribusi Tenaga Listrik.....	5
2.1.1. Sistem Distribusi Primer.....	6
2.1.2. Sistem Distribusi Sekunder.....	6
2.2. Struktur Jaringan Distribusi Tenaga Listrik.....	7
2.3. Sistem Jaringan Distribusi Radial.....	7
2.3.1. Sistem Radial Pohon.....	8
2.3.2. Sistem Radial Dengan <i>Tie</i> dan <i>Swit</i> h Pemisah.....	9

2.3.3. Sistem Radial Dengan Pembagian Phase Area.....	10
2.3.4. Sistem Radial Dengan Beban Terpusat.....	11
2.4. Rugi-Rugi Daya Saluran Distribusi.....	11
2.5. Persamaan Rugi-Rugi Daya.....	12
2.6. Persamaan Aliran Daya.....	12
2.7. Faktor Daya.....	12
2.8. Sistem Per-Unit.....	13
2.9. Mengubah Dasar Kuantitas Per-Unit.....	14
BAB III METODE PENYELESAIAN ALIRAN DAYA.....	16
3.1. Tujuan Studi Aliran Daya.....	16
3.2. Pendekatan Studi Aliran daya.....	16
3.3. Klasifikasi Bus.....	17
3.3.1. Bus <i>Slack/</i> Swing Bus.....	17
3.3.2. Bus Generator / Bus PV.....	17
3.3.3. Bus Beban / Bus PQ.....	18
3.4. Saluran Tiga fasa.....	18
3.5. Aliran Daya Arus Injeksi Tiga fase.....	19
3.6. Struktur Matrik Jacobian.....	23
3.6.1. Sistem Sederhana Yang Memiliki Bus PQ.....	24
3.6.2. Sistem Sederhana Yang Memiliki Bus PV.....	25
3.7. Diagram Alir Perhitungan.....	27

BAB IV ANALISA ALIRAN DAYA PADA JARINGAN DISTRIBUSI PRIMER TIPE RADIAL TIDAK SEIMBANG PADA PENYULANG GLINTUNG DENGAN METODE <i>CURRENT INJECTION</i> DI G.I.BLIMBING MALANG.....	28
4.1. Analisa Dengan Metode <i>Current Injection</i>	28
4.2. Algoritma Pemecahan Masalah	29
4.3. Diagram Alir Penyelesaian.....	30
4.4. Data Perhitungan.....	31
4.4.1. Data Saluran.....	31
4.4.2. Data Pembebanan.....	33
4.5. Analisa Perhitungan	34
4.5.1. Perhitungan Aliran Daya Penyulang Glintung	35
4.6. Tampilan Program	35
4.7. Hasil Perhitungan.....	41
BAB.V.KESIMPULAN	48
5.1. Kesimpulan.....	48

LAMPIRAN

LISTENING PROGRAM

DAFTAR PUSTAKA

DAFTAR GAMBAR

GAMBAR	HAL
2-1. Diagram Satu Garis Sistem Penyaluran Energi Listrik Ke Pelanggan ...	6
2-2. Sistem Jaringan Distribusi Radial.....	8
2-3. Sistem Jaringan Distribusi Radial Dengan <i>Tie</i> dan <i>Switch</i> Penisah	9
2-4. Sistem Jaringan Distribusi Radial Dengan Phase Area	10
2-5. Sistem Jaringan Distribusi Radial Dengan Beban Terpusat.....	11
2-6. Segi Tiga Daya	12
3-1. Rangkaian Saluran Tiga Fasa	18
3-2. Sistem Radial Dengan Sebuah Bus PQ.....	23
3-3. Struktur Matrik Jacobian Untuk Sistem Dari Gambar 3-2.....	24
3-4. Sistem Radial Dengan sebuah Bus PV	25
3-5. Struktur Matrik Jacobian Untuk Sistem Dari Gambar 3-4.....	26
3-6. Diagram Alir Perhitungan Metode <i>Current Injection</i>	27
4-1. Diagram Alir Penyelesaian Metode <i>Current Injection</i>	30
4-2. Struktur Dan Jarak Antar Konduktor	33
4-3. Tampilan Utama Program	35
4-4. Tampilan Inputan Data.....	36
4-5. Tampilan Data Saluran.....	37
4-6. Tampilan Pembebanan	37
4-7. Tampilan Hasil Profil Teg Tiap Bus.....	38
4-8. Tampilan Hasil Aliran Daya Tiap Saluran	39

4-9. Tampilan Total Aliran Daya Awal Pada Jaringan	39
4-10. Grafik Tegangan Tiap-Tiap Node Terhadap Asumsi Tegangan Awal...	40
4-11. Profil Tegangan Tiap Bus Pada Kondisi Akhir.....	41
4-12. Tampilan Hasil Aliran daya Tiap Saluran	42
4-13. Tampilan Total Aliran Daya Pada Jaringan	42
4-14. Grafik Perbandingan Teg Tiap-Tiap Node Terhadap Asumsi Tegangan Awal.....	43
4-15. Perbandingan Total Hasil Perhitungan Program Besar Rugi-Rugi Daya Dan Biaya.....	43

DAFTAR TABEL

TABEL	HAL
4-1. Data Penyulang Sistem 20 KV G.I.BLIMBING	32
4-2. Data Saluran Penyulang Glintung.....	32
4-3. Data Pembebanan Penyulang Glintung.....	34
4-4. Kondisi Teg Sudut Fasa Teg Tiap Node Penyulang Glintung Pada Load Flow Awal	44
4-5. Kondisi Teg dan Sudut Fasa Teg Tiap Node Penyulang Glintung Pada Load Flow Akhir	44
4-6. Besar Rugi-Rugi Daya Tiap Saluran Penyulang Glintung Dengan Metode <i>Current Injection</i>	45
4-7. Arus Tiap Saluran Penyulang Glintung Dengan Metode <i>Current Injection</i>	45
4-8. Perubahan Rugi Daya Pada Saluran	46

BAB I

PENDAHULUAN

1.1. Latar Belakang

Sistem distribusi merupakan bagian dari sistem tenaga listrik yang berfungsi untuk menyalurkan dan mendistribusikan tenaga listrik ke pusat-pusat beban atau konsumen. Permasalahan yang dihadapi jaringan distribusi adalah bagaimana menyalurkan daya dengan baik pada saat tertentu maupun dimasa yang akan datang. Disamping tersedianya energi listrik yang mencukupi, konsumen juga menuntut peningkatan kualitas energi listrik. Untuk meningkatkan energi listrik maka perlu dilakukan analisis yang tepat, yaitu analisa aliran daya. Hasil analisa aliran daya memberikan informasi mengenai besar tegangan, arus, aliran daya, dan rugi-rugi daya pada pengoperasian normal.

Maka dari itu didalam skripsi ini akan dijelaskan perhitungan pada saluran distribusi primer tipe radial yang mempunyai beban tidak seimbang, sehingga perbedaan tegangan dan arus antar phase dapat diketahui dan hasilnya dapat kita terapkan untuk perhitungan-perhitungan lainnya seperti rugi daya.

1.2. Rumusan Masalah

Berdasarkan latar belakang tersebut, permasalahan yang timbul adalah bagaimana mengetahui kondisi jaringan dalam menyalurkan daya listrik terutama perubahan parameter-parameternya pada kondisi operasi.

Maka judul skripsi yang diambil adalah:

**Analisa Aliran Daya Pada Sistem Distribusi Menggunakan Metode
*Three Current Injection Method (TCIM)***

1.3. Tujuan

Menganalisa besarnya aliran daya pada jaringan distribusi tegangan menengah 20 kV pada kondisi operasi untuk mendapatkan perubahan parameter-parameter saluran yang berpengaruh terhadap analisa aliran daya memperkenalkan salah satu metode studi aliran daya jaringan yaitu metode TCIM (*Three Current Injection Method*).

1.4. Batasan Masalah

1. Jaringan listrik yang akan dianalisa adalah jaringan distribusi 20 kV.
2. Metode analisa pembanding adalah metode aliran daya *Newton Raphson*.
3. Analisa dilakukan dengan bantuan perangkat lunak komputer dengan bahasa pemrograman *Borland Delphi versi 0.6*.
4. Tidak membahas masalah proteksi jaringan distribusi.
5. Tidak membahas masalah ekonomi.

1.5. Metode Penelitian

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

1. Studi Literatur.

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

2. Pengumpulan Data.

Bentuk data yang digunakan adalah:

a. **Data Kuantitatif**, yaitu data yang dapat dihitung atau data yang berbentuk angka-angka. Data ini dapat dilihat pada lampiran.

b. **Data Kualitatif**, yaitu data yang berbentuk diagram. Dalam hal ini adalah single line diagram penyulang.

3. Simulasi dan Pembahasan Masalah.

Analisa perhitungan tegangan, sudut phase tegangan, rugi-rugi daya, dan aliran daya pada tiap saluran menggunakan metode injeksi arus yang disimulasikan dengan program komputer.

1.6. Sistematika Penulisan

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

BAB I: PENDAHULUAN

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

BAB II: SISTEM DISTRIBUSI TENAGA LISTRIK

Pada bab ini akan dibahas mengenai sistem jaringan distribusi tegangan menengah, klasifikasi sistem distribusi tenaga listrik dan struktur jaringan distribusi tenaga listrik.

BAB III: METODE PENYELESAIAN STUDI ALIRAN DAYA

Pada bab ini akan dibahas metode penyelesaian studi aliran daya, analisa aliran daya dan rugi-rugi daya dengan metode *current injection*.

BAB IV: ANALISA ALIRAN DAYA PADA JARINGAN DISTRIBUSI PRIMER TIPE RADIAL TIDAK SEIMBANG PADA PENYULANG GLINTUNG DENGAN *METODE CURRENT INJECTION* DI GARDU INDUK BLIMBING MALANG

Bab ini membahas perhitungan aliran daya jaringan 20 kV sistem distribusi radial tidak seimbang dengan *metode current injection* yang meliputi analisa jaringan dengan menentukan tegangan dan sudut fasa tegangan pada tiap *node*, arus pada tiap saluran, aliran daya, dan rugi-rugi daya.

BAB V: KESIMPULAN DAN SARAN

Merupakan intisari dari hasil pembahasan berisi kesimpulan dan saran yang dapat digunakan sebagai pertimbangan untuk pengembangan penulisan selanjutnya.

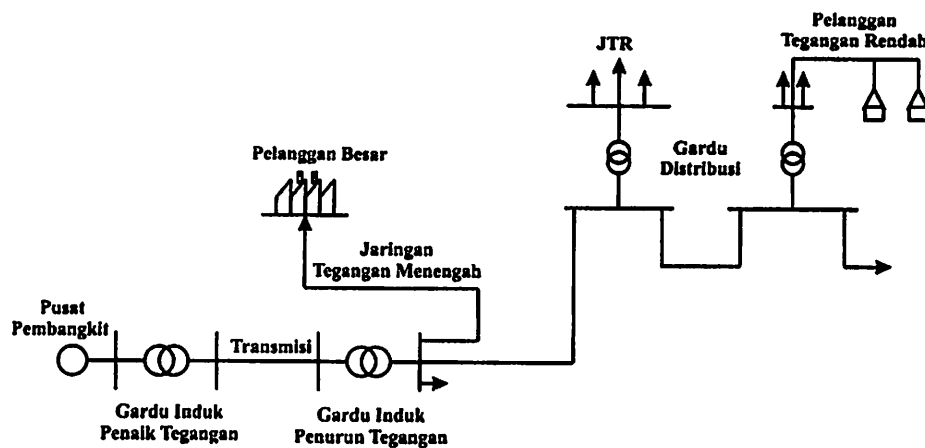
BAB II

SISTEM DISTRIBUSI TENAGA LISTRIK

2.1. Sistem Distribusi Tenaga Listrik

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 pada pusat listrik. Saluran transmisi tegangan tinggi di PLN mempunyai tegangan 150 kV dan 500 kV, selanjutnya tegangan tinggi tersebut diturunkan menjadi 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 pada gambar 2.1 di bawah ini:



Gambar 2-1

Diagram Satu Garis Sistem Penyaluran Energi Listrik Ke Pelanggan

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

Jaringan setelah keluar dari gardu induk biasa disebut jaringan distribusi.

Jaringan distribusi dapat diklasifikasikan menjadi dua bagian sistem, yaitu :

- ❑ Sistem distribusi primer atau sistem distribusi tegangan menengah.
- ❑ Sistem distribusi sekunder atau sistem distribusi tegangan rendah.

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

2.1.1. Sistem Distribusi Primer

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

2.1.2. Sistem Distribusi Sekunder

Tingkat tegangan yang digunakan pada sistem distribusi sekunder meliputi tegangan rendah 127/220 Volt atau 220/380 Volt karena itu sistem distribusi ini

sering disebut dengan sistem distribusi tegangan rendah. 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.

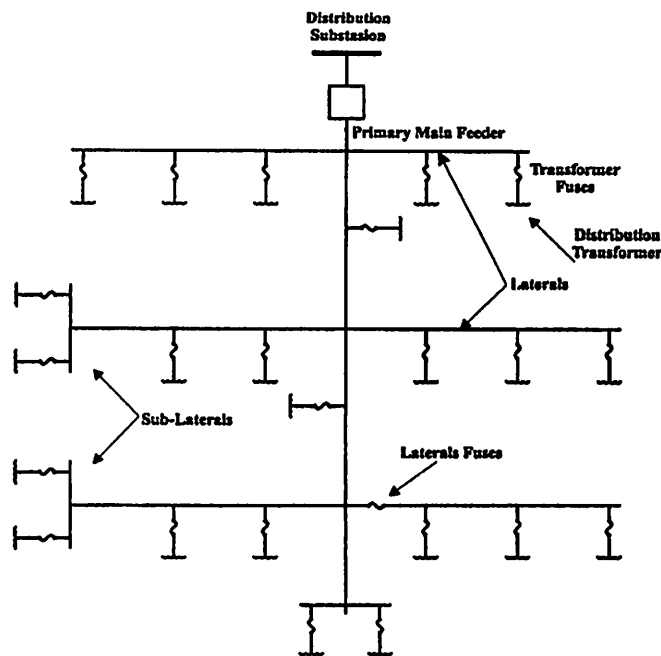
2.2. Struktur Jaringan Distribusi Tenaga Listrik

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

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

2.3. Sistem Jaringan Distribusi Radial

Bentuk jaringan ini merupakan bentuk dasar yang paling banyak digunakan dan yang paling sederhana. Sistem ini dikatakan radial karena dari kenyataan bahwa jaringan ini ditarik secara radial dari gardu induk ke pusat-pusat beban atau 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 suatu daerah beban tertentu dilaksanakan dengan memasang transformator pada sembarang titik pada jaringan yang terdekat dengan daerah beban yang dilalui. 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 jaringan distribusi radial adalah jatuh tegangan yang cukup besar dan bila terjadi gangguan pada sistem mengakibatkan jatuhnya sebagian atau bahkan keseluruhan beban sistem.

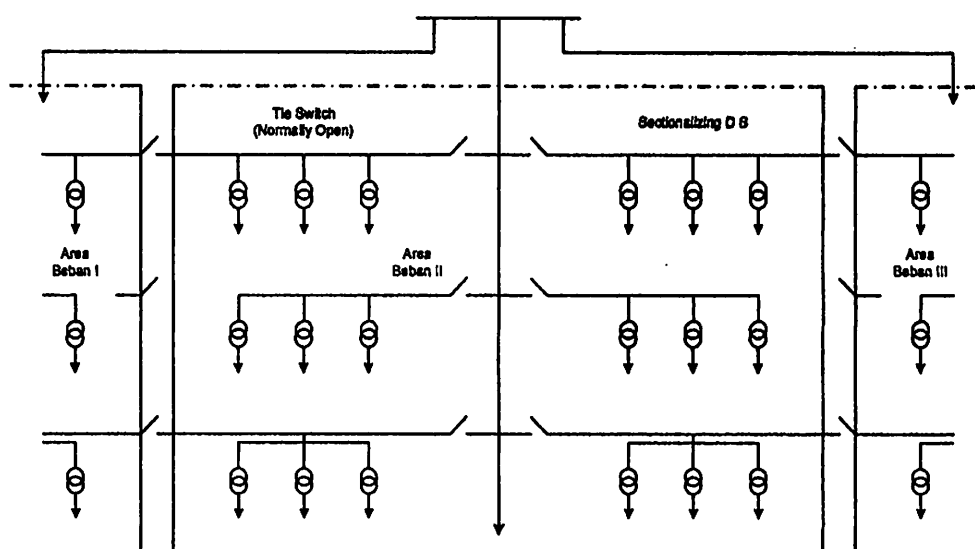
2.3.1. Sistem Radial Pohon

Sistem jaringan radial pohon merupakan bentuk yang paling dasar dari sistem distribusi jaringan radial. Saluran utama (*Main Feeder*) ditak dari suatu

G.I. sesuai dengan kebutuhan kemudian dicabangkan melalui saluran cabang (*Lateral*), selanjutnya dicabangkan lagi melalui saluran anak cabang (*Sub-Lateral Feeder*). Ukuran dari masing-masing saluran tergantung dari kerapatan arus yang mengalir. Dari gambar 2-2 (*Main Feeder*) merupakan saluran yang dialiri arus terbesar, selanjutnya arus berkurang pada tiap cabang tergantung dari besarnya beban.

2.3.2. Sistem Radial Dengan Tie dan Switch Pemisah

Sistem ini merupakan pengembangan dari sistem radial pohon, untuk meningkatkan keandalan sistem saat terjadi gangguan maka *Feeder* yang terganggu akan dilokalisir sedangkan area yang semula dilayani *Feeder* tersebut pelayanannya dialihkan pada *Feeder* yang tidak terganggu. Sistem radial dengan Tie dan Switch Pemisah dapat dilihat pada gambar 2-3

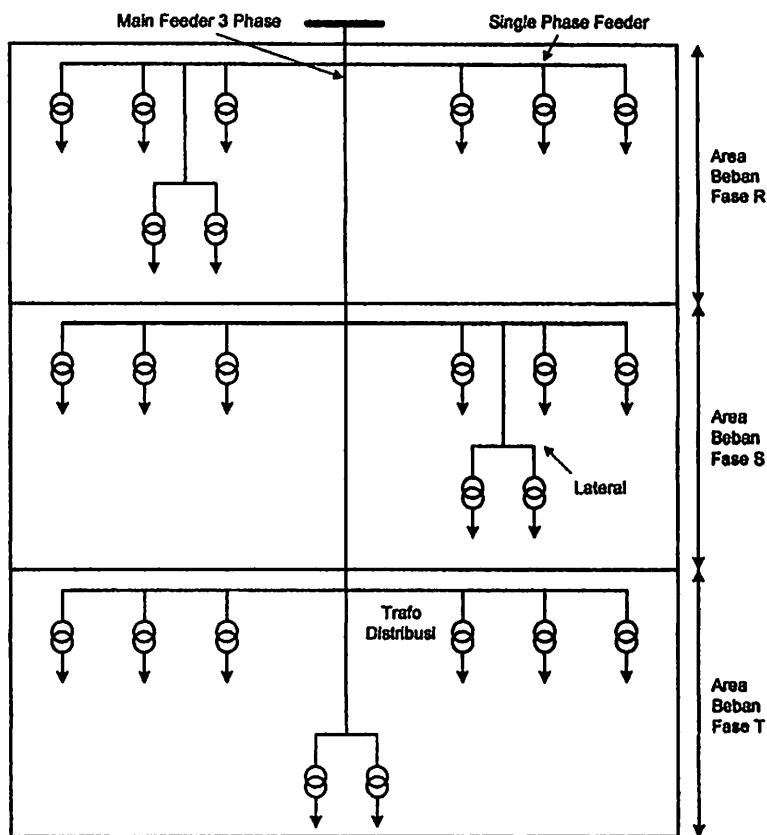


Gambar 2-3
Contoh Sistem Jaringan Distribusi Radial
Dengan Tie dan Switch Pemisah

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

2.3.3. Sistem Radial Dengan Pembagian Phase Area

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



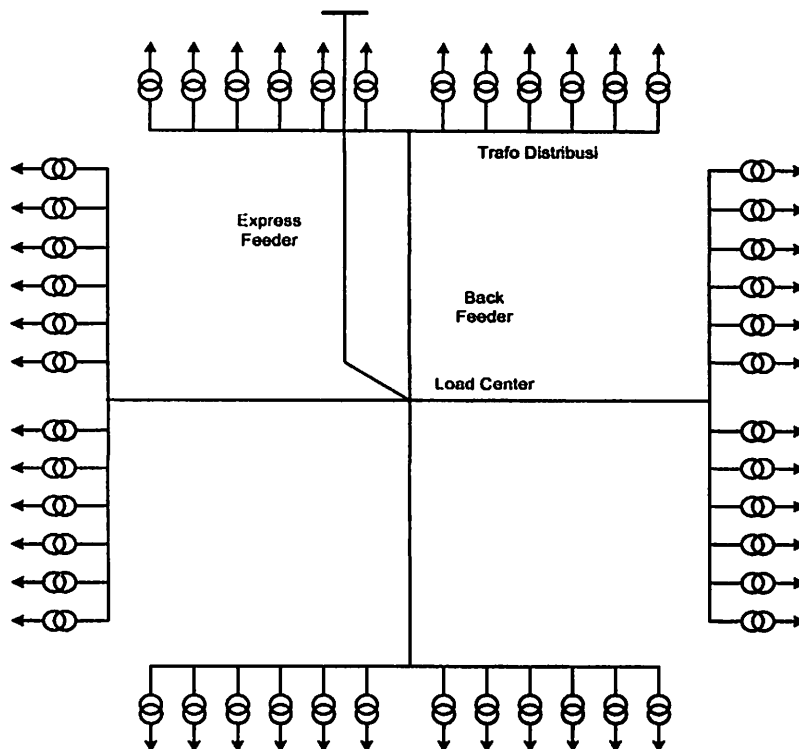
Gambar 2-4

Contoh Jaringan Distribusi Radial Dengan Phase Area

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

2.3.4. Sistem Radial Dengan Beban Terpusat

Bentuk dari sistem ini mensuplai daya dengan menggunakan *Main Feeder* yang disebut *Express Feeder* langsung ke pusat beban, dari titik pusat beban ini disebut dengan menggunakan *Back Feeder* secara radial seperti terlihat pada gambar 2-5.



Gambar 2-5

Contoh Jaringan Distribusi Radial Dengan Beban Terpusat

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

2.4. Rugi-Rugi Daya Saluran Distribusi

Rugi daya adalah besar daya yang hilang, dalam penyaluran daya elektrik. Rugi daya ini terdiri dari rugi daya aktif dan rugi daya reaktif. Rugi-rugi daya ini dapat terjadi pada komponen umum pada sistem tenaga listrik seperti:

- ❑ Rugi daya pada penyulang utama dan peralatan saluran.
- ❑ Rugi daya pada trafo distribusi.

2.5. Persamaan Rugi-Rugi Daya

Rugi-rugi daya aktif dan reaktif dapat dihitung melalui persamaan:

$$P + jQ = [Z] \cdot |I|^2 \dots\dots\dots(2.1)$$

2.6. Persamaan Aliran Daya

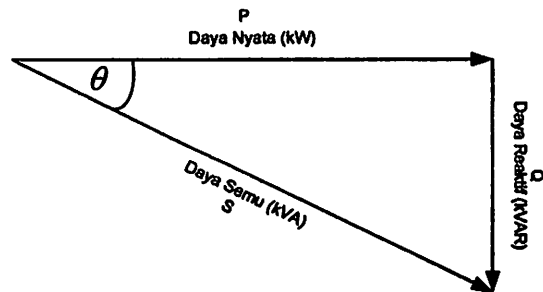
Aliran daya aktif dan reaktif dapat dihitung melalui persamaan:

$$P_\alpha = \text{Real}(V_i \cdot I_\alpha^*) \dots\dots\dots(2.2)$$

$$Q_\alpha = \text{Im}(V_i \cdot I_\alpha^*) \dots\dots\dots(2.3)$$

2.7. Faktor Daya

Faktor daya adalah perbandingan antara komponen daya kW terhadap kVA, seperti terlihat pada gambar 2-6:



Gambar 2-6
Segitiga Daya

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

Faktor daya:

$$\cos \theta = \frac{kW}{kVA} \dots\dots\dots(2.4)$$

$$(kVA)^2 = (kW)^2 + (kVAR)^2 \dots\dots\dots(2.5)$$

$$kW = kVA \cos \theta \dots\dots\dots(2.6)$$

$$kVAR = kVA \sin \theta = kW \cdot \tan \theta \dots\dots\dots(2.7)$$

$$\tan \theta = \frac{kVAR}{kW} \dots\dots\dots(2.8)$$

Rugi daya:

$$P = I^2 \cdot R \dots\dots\dots(2.9)$$

Faktor daya akan mendahului (*Leading*) apabila arus mendahului tegangan dan akan tertinggal (*Lagging*) bila arus tertinggal terhadap tegangan.

2.8. Sistem Per-Unit

Untuk memudahkan perhitungan-perhitungan dalam sistem tenaga listrik digunakan satuan per-unit (p.u) yang didefinisikan sebagai perbandingan besar nilai tegangan yang sebenarnya dengan besar tegangan dasar (*Base Value*), sehingga dapat dirumuskan sebagai berikut:

$$\text{besaran per - unit} = \frac{\text{besaran sesungguhnya}}{\text{besaran dasar dengan dimensi yang sama}} \dots\dots(2.10)$$

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

❑ Untuk data 1 fasa

Arus dasar

$$I_d = \frac{kVA \text{ dasar 1 fasa}}{kV \text{ dasar } L-N} \dots\dots\dots(2.11)$$

Impedansi dasar

$$Z_d = \frac{(kV \text{ dasar } L-N)^2 \times 1000}{kVA \text{ dasar 1 fasa}} \dots\dots\dots(2.12)$$

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

Daya Dasar, kW_{1Ø} = Dasar kVA_{1Ø}

Daya Dasar, MW_{1Ø} = Dasar MVA_{1Ø}

$$\text{Im pedansi per - unit dari suatu elemen rangkaian} = \frac{\text{Im pedansi sebenarnya, } \Omega}{\text{Im pedansi dasar, } \Omega} \dots\dots\dots(2.13)$$

▣ Untuk data 3 fasa

$$I_d = \frac{kVA \text{ dasar 3 fasa}}{\sqrt{3} kV \text{ dasar } L-L} \dots\dots\dots(2.14)$$

$$Z_d = \frac{(kV \text{ dasar } L-L)^2 \times 1000}{kVA \text{ dasar 3 fasa}} \dots\dots\dots(2.15)$$

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

2.9. Mengubah Dasar Kuantitas Per-Unit

Impedansi per-unit untuk suatu komponen dari suatu sistem dinyatakan menurut dasar yang berbeda menurut dasar yang berbeda dengan dasar yang dipilih untuk bagian dari sistem dimana komponen tersebut berada. Karena semua impedansi dalam bagian manapun dari suatu sistem harus dinyatakan dengan dasar impedansi yang sama, maka dalam perhitungannya kita perlu mempunyai cara untuk dapat mengubah impedansi per-unit dari suatu dasar ke dasar yang lain.

Dengan mensubstitusikan impedansi dasar yang diberikan dalam persamaan (2.11) atau (2.13) ke dalam persamaan (2.12). Sehingga diperoleh:

Im pedansi per – unit dari suatu elemen rangkaian

$$= \frac{(\text{Im pedansi sebenarnya, } \Omega) \times (\text{kVA dasar})}{(\text{Tegangan dasar, kV})^2 \times 1.000} \dots\dots\dots(2.16)$$

Rumus diatas memperlihatkan bahwa impedansi per-unit berbanding lurus dengan kVA dasar dan berbanding terbalik dengan kuadrat tegangan dasar. Karena itu untuk mengubah dari impedansi per-unit menurut suatu dasar yang diberikan menjadi impedansi per-unit menurut suatu dasar yang baru dapat dipakai persamaan berikut:

$$Z_{\text{baru per-unit}} = Z_{\text{diberikan per – unit}} \left(\frac{kV_{\text{diberikan dasar}}}{kV_{\text{baru dasar}}} \right)^2 \times \left(\frac{kVA_{\text{baru dasar}}}{kVA_{\text{diberikan dasar}}} \right) \dots(2.17)$$

Persamaan ini sangat berguna untuk mengubah suatu impedansi per-unit yang diberikan menurut suatu dasar tertentu ke suatu dasar yang baru. Selain dengan menggunakan persamaan (2.16) perubahan dasar dapat juga diperoleh dengan mengubah nilai per-unit menurut suatu dasar menjadi nilai ohm dan membaginya dengan impedansi dasar yang baru.

Sumber: William D. Stevenson, Jr. "Anallisis Sistem Tenaga Listrik", Edisi keempat

BAB III

METODE PENYELESAIAN ALIRAN DAYA

3.1. Tujuan Studi Aliran Daya

Dengan semakin kompleksnya problem didalam sistem tenaga listrik, sebagai akibat dari meningkatnya permintaan konsumen, bertambahnya jumlah saluran transmisi dan distribusi, perlu adanya studi aliran daya dalam analisa sistem tenaga listrik. Studi aliran daya ini dilakukan untuk menentukan:

1. Aliran daya aktif dan reaktif pada cabang-cabang rangkaian.
2. Tidak ada rangkaian yang mempunyai beban lebih dari tegangan busbar dalam batas-batas yang dapat diterima.
3. Pengaruh penambahan atau perubahan suatu sistem.
4. Pengaruh hilangnya hubungan dalam keadaan darurat.
5. Kondisi optimum pembebanan sistem.
6. Hilangnya daya optimum sistem.

3.2. Pendekatan Studi Aliran Daya

Didalam pengoperasian sistem tenaga listrik parameter-parameter yang perlu diperhatikan sehubungan dengan analisa aliran daya adalah besarnya magnitude tegangan $|V|$, sudut fasa tegangan (θ), daya nyata (P) dan, daya reaktif (Q). Daya nyata P mempunyai ketergantungan yang kuat dengan sudut fasa

tegangan θ dan daya reaktif Q mempunyai ketergantungan yang kuat dengan besarnya $|V|$. Bila P dan Q berubah, maka θ dan $|V|$ berubah pula dan sebaliknya.

3.3. Klasifikasi Bus

Terdapat empat (4) besaran pada setiap Bus/Rel dalam sistem tenaga listrik yaitu besarnya magnitude tegangan $|V|$, sudut fasa tegangan (θ), daya nyata (P) dan, daya reaktif (Q). Pada setiap Bus ditentukan dua besaran dari empat besaran tersebut. Dalam studi aliran daya konvensional, seluruh Bus dalam jaringan digolongkan menjadi Bus *Slack/Swing*, Bus PV/Bus Generator dan Bus *PQ/Load Bus* (Bus Beban).

3.3.1. Bus *Slack/Swing Bus*

Pada Bus ini nilai $|V|$ dan θ ditentukan besarnya sementara P dan Q dihitung. Nilai $|V|$ adalah 1 pu dan nilai θ adalah 0° karena pada Bus ini fasor tegangan dipakai sebagai referensi.

3.3.2. Bus Generator/Bus PV

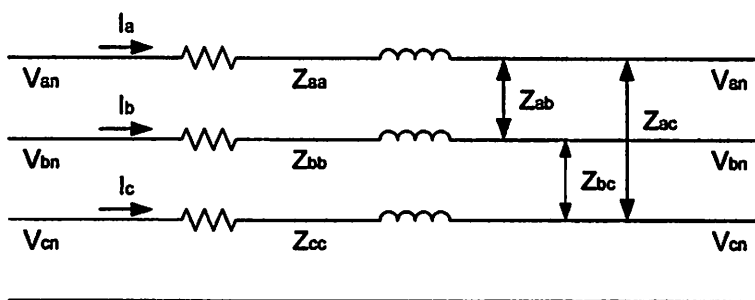
Pada Bus Generator ini hanya terdapat daya pembangkit dimana parameter P dan $|V|$ diketahui, sementara Q dan θ dihitung.

3.3.3. Bus Beban/Bus PQ

Bus beban adalah suatu Bus yang ditentukan besar dayanya. Pada simpul ini parameter-parameter P dan Q diketahui, sedangkan parameter $|V|$ dan θ berubah-ubah menurut kebutuhan. Karena itu parameter $|V|$ dan θ harus diketahui.

3.4. Saluran Tiga Fasa

Rangkaian ekuivalen untuk saluran tiga fasa adalah:



Gambar 3-1
Rangkaian Saluran Tiga Fasa

Sumber: Turan Gonen. "Electric Power Distribution System Engineering", University of Missouri at Columbia [1]

Impedansi Z_{abc} akan ditunjukkan sebagai matrik impedansi fasa. Untuk saluran tiga fasa dengan netral akan menghasilkan matrik impedansi *primitive* 4 x 4 seperti persamaan dibawah ini:

$$[Z_{abc}] = \begin{bmatrix} Z_{aa} & Z_{ab} & Z_{ac} & Z_{an} \\ Z_{ba} & Z_{bb} & Z_{bc} & Z_{bn} \\ Z_{ca} & Z_{cb} & Z_{cc} & Z_{cn} \\ Z_{na} & Z_{nb} & Z_{nc} & Z_{nn} \end{bmatrix} \dots\dots\dots(3.1)$$

Maka matrik tersebut menjadi matrik 3 x 3 dengan persamaan:

$$[Z_{abc}] = [Z_{abc}] - [Z_{an}] \cdot [Z_{nn}]^{-1} [Z_{na}] \dots\dots\dots(3.2)$$

Impedansi saluran ditunjukkan oleh matrik impedansi 3 x 3 sebagai

berikut:

$$Z = \begin{bmatrix} (R_{aa} + jX_{aa}) & (R_{ab} + jX_{ab}) & (R_{ac} + jX_{ac}) \\ (R_{ba} + jX_{ba}) & (R_{bb} + jX_{bb}) & (R_{bc} + jX_{bc}) \\ (R_{ca} + jX_{ca}) & (R_{cb} + jX_{cb}) & (R_{cc} + jX_{cc}) \end{bmatrix} \dots\dots\dots(3.3)$$

3.5. Aliran Daya Arus Injeksi Tiga Fasa

Metode *TCIM* dapat dioperasikan pada kondisi beban seimbang dan tak seimbang, metode *TCIM* dipergunakan pada aliran daya pada setiap cabang arus. Dalam perhitungan aliran daya menggunakan metode *TCIM* secara berangsur-angsur memberikan hasil yang konvergen (nilai solusinya tidak berubah lagi).

Persamaan dasar arus *Mismatches* tiga fasa untuk bus k adalah:

$$\Delta I_k^2 = \frac{(P_k^{sp})^s - j(Q_k^{sp})^s}{(E_k^s)^*} - \sum_{i \in \Omega_k} \sum_{t \in \alpha_p} Y_{ki}^{st} E_i^t \dots\dots\dots(3.4)$$

dimana:

- s, t ∈ α_p.
- α_p = { a, b, c }.
- k = { 1, ..., n }, n adalah jumlah nomor bus.

$$E_k = V_{rk} + jV_{mk} \dots\dots\dots(3.5)$$

$$(P_k^{sp})^s = P_{gk}^s - P_{lk}^s \dots\dots\dots(3.6)$$

$$(Q_k^{sp})^s = Q_{gk}^s - Q_{lk}^s \dots\dots\dots(3.7)$$

$(P_k^{sp})^s, (Q_k^{sp})^s$: Daya specified aktif dan reaktif pada bus k untuk fase s.

P_{gk}^s, Q_{gk}^s : Daya aktif dan reaktif generator untuk fase s.

P_{lk}^s, Q_{lk}^s : Daya aktif dan reaktif beban untuk fase s.

$Y_{ki}^{st} = G_{ki}^{st} + jB_{ki}^{st}$: Elemen matrik admitansi bus.

Akibat dari tingkat tegangan pada sistem beban yang diberikan oleh persamaan *second order polynomial* adalah:

$$P_k^s = P_{0k}^s + P_{1k}^s V_k + P_{2k}^s V_k^2 \dots \dots \dots (3.8)$$

$$Q_k^s = Q_{0k}^s + Q_{1k}^s V_k + Q_{2k}^s V_k^2 \dots \dots \dots (3.9)$$

dimana:

- P_{0k}^s, Q_{0k}^s : komponen daya konstan dari beban fase s pada bus k.
- P_{1k}^s, Q_{1k}^s : komponen arus konstan dari beban fase s pada bus k.
- P_{2k}^s, Q_{2k}^s : komponen impedansi konstan dari beban fase s pada bus k.
- $V_k = |E_k|$: nilai mutlak dari tegangan pada bus k.

Persamaan (3.6) dapat dinyatakan dalam bentuk kompleks yang terdiri dari bagian *real* dan *imaginer* seperti persamaan berikut:

$$\Delta I_{n_k}^s = \frac{(P_k^{sp})^s V_{n_k}^s + (Q_k^{sp})^s V_{m_k}^s}{(V_{n_k}^s)^2 + (V_{m_k}^s)^2} - \sum_{i=1}^n \sum_{t \in \alpha_p} (G_{ki}^{st} V_{n_k}^t - B_{ki}^{st} V_{m_k}^t) \dots \dots \dots (3.10)$$

$$\Delta I_{m_k}^s = \frac{(P_k^{sp})^s V_{m_k}^s + (Q_k^{sp})^s V_{n_k}^s}{(V_{n_k}^s)^2 + (V_{m_k}^s)^2} - \sum_{i=1}^n \sum_{t \in \alpha_p} (G_{ki}^{st} V_{m_k}^t - B_{ki}^{st} V_{n_k}^t) \dots \dots \dots (3.11)$$

Persamaan (3.12) dan (3.13) dapat ditulis dalam bentuk nilai *specified* dan *calculated* seperti persamaan berikut:

$$\Delta I_{r_k}^s = (I_{r_k}^{sp})^s - (I_{r_k}^{calc})^s \dots\dots\dots(3.12)$$

$$\Delta I_{m_k}^s = (I_{m_k}^{sp})^s - (I_{m_k}^{calc})^s \dots\dots\dots(3.13)$$

penggunaan metode *Newton-Rhapson* ke persamaan (3.10) dan (3.11), sehingga menghasilkan persamaan linier sebagai berikut:

$$\begin{bmatrix} \Delta I_{m_1}^{abc} \\ \Delta I_{r_1}^{abc} \\ \Delta I_{m_2}^{abc} \\ \Delta I_{r_2}^{abc} \\ \vdots \\ \Delta I_{m_n}^{abc} \\ \Delta I_{r_n}^{abc} \end{bmatrix} = \begin{bmatrix} (Y_{11}^*)^{abc} & Y_{12}^{abc} & \dots & Y_{1n}^{abc} \\ Y_{21}^{abc} & (Y_{22}^*)^{abc} & \dots & Y_{2n}^{abc} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{n1}^{abc} & Y_{n2}^{abc} & \dots & (Y_{nn}^*)^{abc} \end{bmatrix} \begin{bmatrix} \Delta V_{r_1}^{abc} \\ \Delta V_{m_1}^{abc} \\ \Delta V_{r_2}^{abc} \\ \Delta V_{m_2}^{abc} \\ \vdots \\ \Delta V_{r_n}^{abc} \\ \Delta V_{m_n}^{abc} \end{bmatrix} \dots\dots\dots(3.14)$$

elemen diluar diagonal identik dengan elemen dari matrik admitansi bus.

Ketika koordinat retangular tiga fasa digunakan, setiap elemen terdiri dari ordo 6 x 6 dengan struktur seperti berikut:

$$Y_{im}^{abc} = \begin{bmatrix} B_{im}^{abc} & G_{im}^{abc} \\ G_{im}^{abc} & -B_{im}^{abc} \end{bmatrix}; i, m = 1, \dots, n \dots\dots\dots(3.15)$$

elemen diagonalnya sebagai berikut:

$$(Y_{kk}^*)^{abc} = \begin{bmatrix} (B_{kk}')^{abc} & (G_{kk}')^{abc} \\ (G_{kk}'')^{abc} & (B_{kk}'')^{abc} \end{bmatrix} \dots\dots\dots(3.16)$$

dimana:

$$(B_{kk}')^{abc} = B_{kk}^{abc} - \begin{bmatrix} a_k^a & & \\ & a_k^b & \\ & & a_k^c \end{bmatrix} \dots\dots\dots(3.17)$$

$$(G'_{kk})^{abc} = G_{kk}^{abc} - \begin{bmatrix} b_k^a & & \\ & b_k^b & \\ & & b_k^c \end{bmatrix} \dots\dots\dots(3.18)$$

$$(G''_{kk})^{abc} = G_{kk}^{abc} - \begin{bmatrix} c_k^a & & \\ & c_k^b & \\ & & c_k^c \end{bmatrix} \dots\dots\dots(3.19)$$

$$(B''_{kk})^{abc} = B_{kk}^{abc} - \begin{bmatrix} d_k^a & & \\ & d_k^b & \\ & & d_k^c \end{bmatrix} \dots\dots\dots(3.20)$$

Elemen a, b, c, dan d, pada persamaan (3.17), (3.18), (3.19), dan (3.20) tergantung pada pemakaian model beban. Elemen tersebut digunakan untuk mengupdate matrik jacobian dan persamaan elemen tersebut adalah:

$$a_k^s = \frac{Q_{0_k}^s [(V_{r_k}^s)^2 - (V_{m_k}^s)^2] - 2V_{r_k}^s V_{m_k}^s P_{0_k}^s}{(V_k^s)^4} + \frac{V_{r_k}^s V_{m_k}^s P_{1_k}^s + Q_{1_k}^s (V_{m_k}^s)^2}{(V_k^s)^3} + Q_{2_k}^s \dots\dots\dots(3.21)$$

$$b_k^s = \frac{P_{0_k}^s [(V_{r_k}^s)^2 - (V_{m_k}^s)^2] + 2V_{r_k}^s V_{m_k}^s Q_{0_k}^s}{(V_k^s)^4} - \frac{(V_{r_k}^s V_{m_k}^s) Q_{1_k}^s + P_{1_k}^s (V_{r_k}^s)^2}{(V_k^s)^3} - P_{2_k}^s \dots\dots\dots(3.22)$$

$$c_k^s = \frac{P_{0_k}^s [(V_{m_k}^s)^2 - (V_{r_k}^s)^2] - 2V_{r_k}^s V_{m_k}^s Q_{0_k}^s}{(V_k^s)^4} + \frac{(V_{r_k}^s V_{m_k}^s) Q_{1_k}^s - P_{1_k}^s (V_{m_k}^s)^2}{(V_k^s)^3} - P_{2_k}^s \dots\dots\dots(3.23)$$

$$d_k^s = \frac{Q_{0_k}^s [(V_{r_k}^s)^2 - (V_{m_k}^s)^2] - 2V_{r_k}^s V_{m_k}^s P_{0_k}^s}{(V_k^s)^4} + \frac{(V_{r_k}^s V_{m_k}^s) P_{1_k}^s - Q_{1_k}^s (V_{m_k}^s)^2}{(V_k^s)^3} - Q_{2_k}^s \dots\dots\dots(3.24)$$

Besar arus *mismatch* dari persamaan (3.14) untuk bus k dan phasa s dapat dinyatakan seperti berikut:

$$\Delta I_{r_k}^s = \frac{V_{r_k}^s \Delta P_k^s + V_{m_k}^s \Delta Q_k^s}{(V_{r_k}^s)^2 + (V_{m_k}^s)^2} \dots\dots\dots(3.25)$$

$$\Delta I_{m_k}^s = \frac{V_{m_k}^s \Delta P_k^{s^*} + V_{m_k}^s \Delta Q_k^s}{(V_{r_k}^s)^2 + (V_{m_k}^s)^2} \dots\dots\dots(3.26)$$

Daya mismatches aktif dan reaktif ΔP_k^s dan ΔQ_k^s adalah:

$$\Delta P_k^s = (P_k^{sp})^s - (P_k^{calc})^s \dots\dots\dots(3.27)$$

$$\Delta Q_k^s = (Q_k^{sp})^s - (Q_k^{calc})^s \dots\dots\dots(3.28)$$

dimana:

$$(P_k^{calc})^s = V_{r_k}^s (I_{r_k}^{calc})^s + V_{m_k}^s (I_{m_k}^{calc})^s \dots\dots\dots(3.29)$$

$$(Q_k^{calc})^s = V_{m_k}^s (I_{r_k}^{calc})^s - V_{r_k}^s (I_{m_k}^{calc})^s \dots\dots\dots(3.30)$$

Tegangan yang diupdate pada setiap iterasi adalah:

$$(V_{r_{m_k}}^{abc})^{h+1} = (V_{r_{m_k}}^{abc})^h + (\Delta V_{r_{m_k}}^{abc})^h \dots\dots\dots(3.31)$$

dimana:

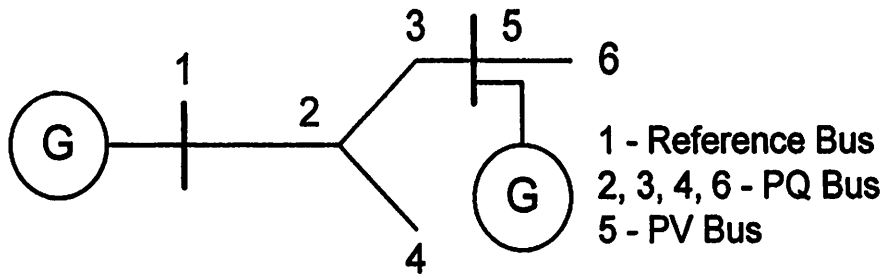
$$(V_{r_{m_k}}^{abc}) = [V_{r_k}^a \ V_{r_k}^b \ V_{r_k}^c \ V_{m_k}^a \ V_{m_k}^b \ V_{m_k}^c] \dots\dots\dots(3.32)$$

3.6. Struktur Matrik Jacobian

Perhitungan matrik jacobian dilakukan untuk analisa perhitungan tegangan yang digunakan untuk perbaikan daya dengan metode injeksi arus tiga fasa. Pada kenyataanya matrik jacobian identik dengan admitansi bus sehingga mempermudah perhitungan elemen matriknya, contohnya sebagai berikut:

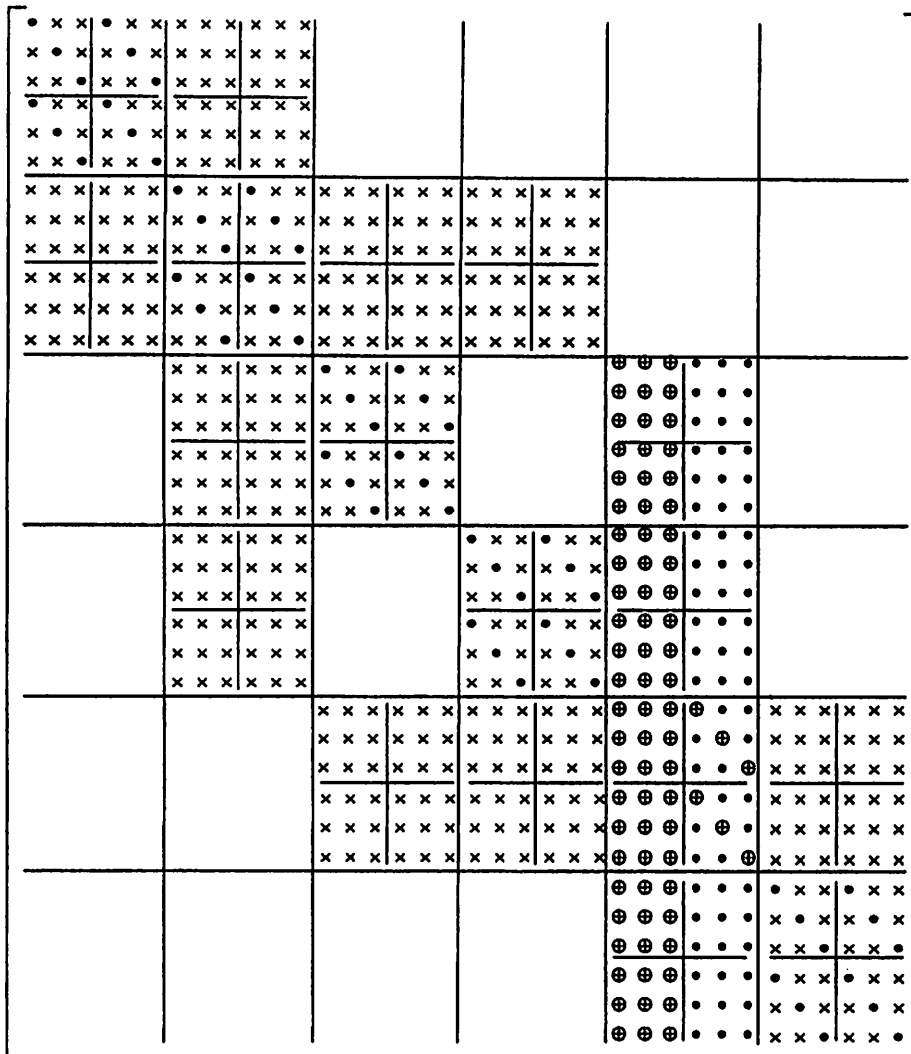
3.6.2. Sistem Sederhana Yang Memiliki Bus PV

Sistem dari gambar 3-2 diperluas ke penambahan generator *plant* pada bus s, yang mana dinyatakan dengan bus PV seperti ditunjukkan pada gambar dibawah ini:



Gambar 3-4
Sistem Radial Dengan Sebuah Bus PV [3]
" sumber : jurnal "

Struktur matrik jacobian ditunjukkan seperti pada gambar 3-5, hal tersebut dapat dilihat pada penambahan elemen diagonal blok sesuai dengan bus PQ. Keberadaan dari bus PV memerlukan setengah dari elemen tanpa nol dari kolom 5 akan dihitung kembali pada setiap iterasi.

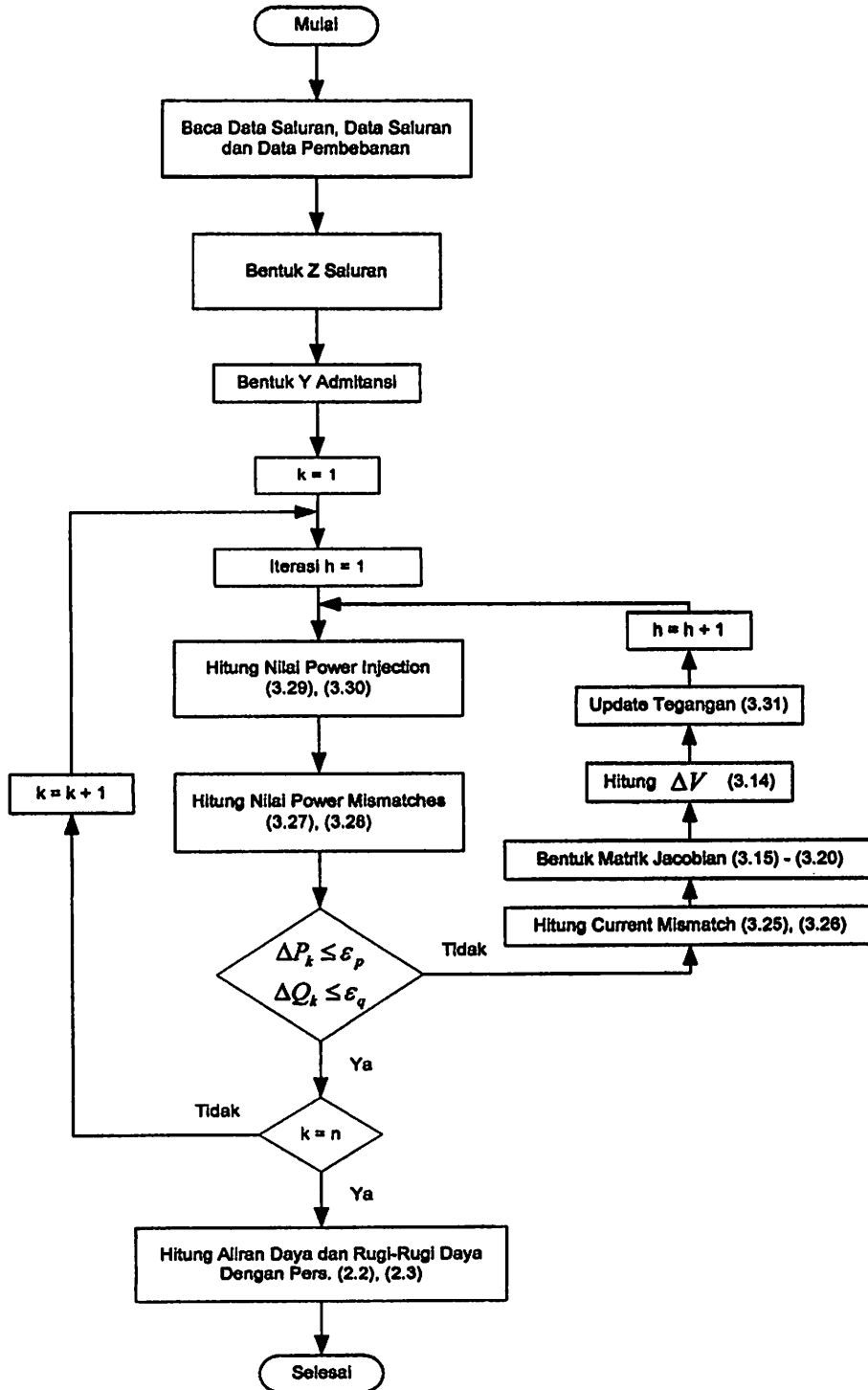


Keterangan:

- × : Elemen dari matrik admitansi.
- : Elemen yang diupdate dari bus PQ.
- ⊕ : Elemen yang diupdate dari bus PV.

Gambar 3-5
Struktur Matrik Jacobian Untuk Sistem Dari Gambar 3-4 [4]
 “ sumber : jurnal ”

3.7. Diagram Alir Perhitungan



Gambar 3-6
Diagram Alir Perhitungan Metode Current Injection

BAB IV

ANALISA ALIRAN DAYA PADA JARINGAN DISTRIBUSI PRIMER TIPE RADIAL TIDAK SEIMBANG PADA PENYULANG GLINTUNG DENGAN METODE *CURRENT INJECTION* DI GARDU INDUK BLIMBING MALANG

4.1. Analisis Dengan Metode Current Injection

Perhitungan arus beban banyak dilakukan dalam perencanaan sistem, perencanaan operasional dan kontrol atau operasi. Pilihan suatu metode solusi untuk aplikasi praktis sering sulit, pilihan itu memerlukan analisis yang cermat atas kelebihan-kelebihan dan kekurangan-kekurangan pada banyak metode yang tersedia, seperti karakteristik konvergensi, kecepatan dan penyimpangan (*error*).

Untuk solusi dari permasalahan-permasalahan diatas maka dipergunakan suatu alternatif metode yang disebut metode *Current Injection* dimana kelebihan dari metode ini antara lain:

- Dapat dipakai untuk jaringan radial.
- Dapat dipakai untuk perhitungan aliran daya beban tidak seimbang.
- Waktu perhitungan cepat.

Dalam perhitungan analisa aliran daya, kita bisa melakukan perhitungan secara manual dan menggunakan program komputer. Apabila kita melakukan perhitungan secara manual maka perlu melakukan perhitungan sampai beberapa iterasi dengan cara manual, tetapi bila kita melakukan dengan program komputer

maka perhitungan dengan beberapa/banyak iterasi dapat dilakukan secara otomatis sehingga tidak memerlukan waktu yang lama.

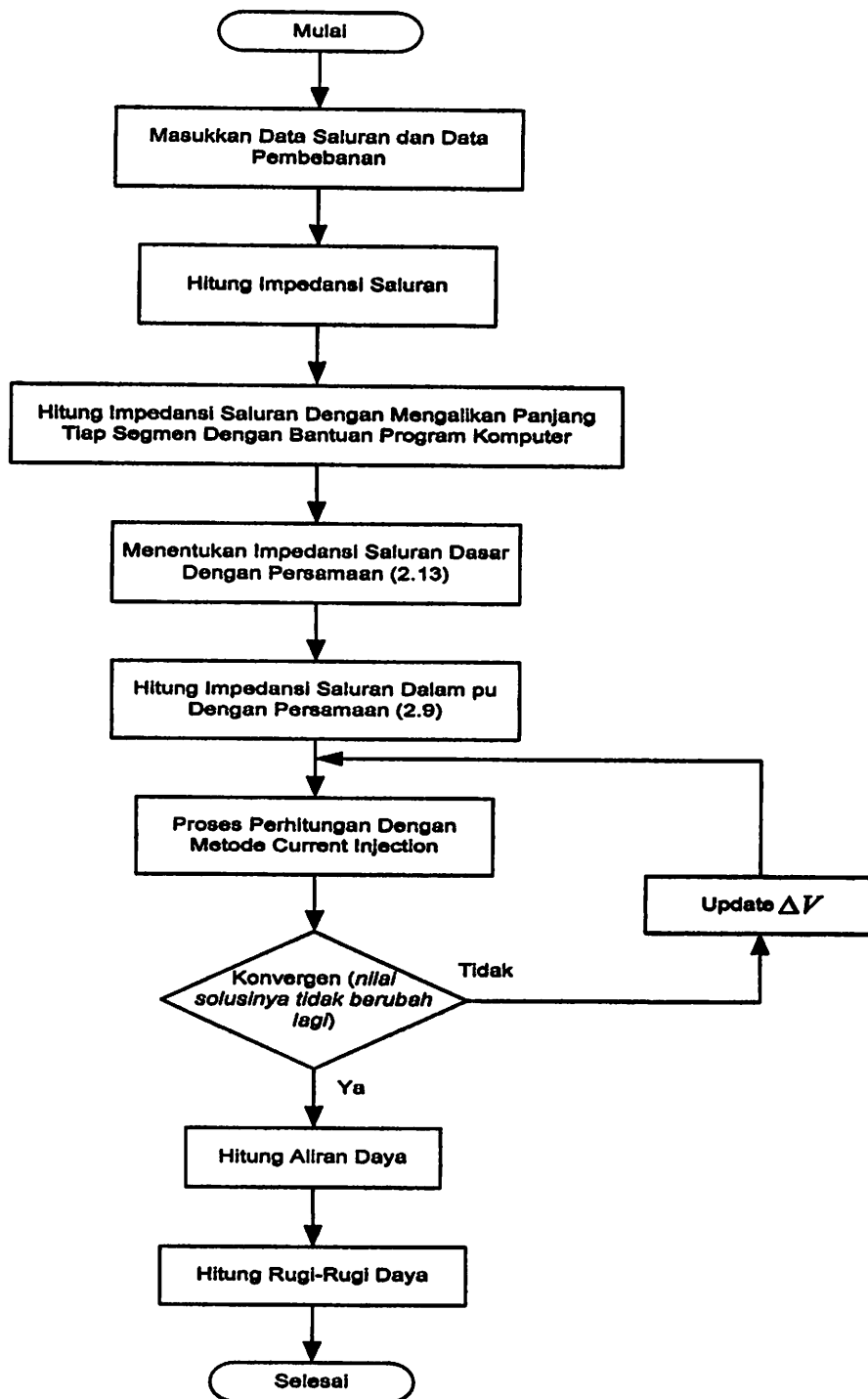
Pada prinsipnya penggunaan program komputer dapat menganalisa sebuah distribusi radial dengan jumlah cabang atau jumlah bus yang tidak terbatas tergantung dari tersedianya jumlah memori yang tersedia pada komputer yang digunakan.

4.2. Algoritma Pemecahan Masalah

1. Mulai.
2. Memasukan inputan data saluran dan data pembebanan.
3. Menentukan impedansi saluran masing- masing phasa.
4. Menghitung impedansi saluran berdasarkan panjang saluran.
5. Menentukan impedansi dasar kemudian menentukan impedansi saluran dalam per-unit.
6. Melakukan proses perhitungan dengan Metode *Current Injection*.
7. Apakah hasilnya sudah konvergen, jika belum maka lakukan Up date tegangan (ΔV) dan kembali ke langkah 6.

Jika Ya lakukan perhitungan aliran daya untuk mengetahui besar rugi-rugi daya setelah perhitungan dengan Metode *Current Injection*.
8. Selesai.

4.3. Diagram Alir Penyelesaian



Gambar 4-1
Diagram Alir Penyelesaian Metode Current Injection

4.4. Data Perhitungan

Perhitungan aliran daya ini dengan mengambil data dari G.I Blimbing Malang yang melayani 9 (sembilan) buah penyulang dengan 3 (tiga) buah trafo masing-masing $2 \times 150/20 \text{ kV} - 100 \text{ MVA}$ dan $1 \times 150/20 \text{ kV} - 100 \text{ MVA}$.

Sistem distribusi radial G.I Blimbing memakai tegangan distribusi 20 kV. Untuk menyelesaikan perhitungan aliran daya terlebih dahulu ditetapkan *single line* diagram yang akan dianalisa, agar perhitungan lebih mudah maka digunakan sistem per-unit (*pu*), dimana dasar yang digunakan:

- Tegangan dasar : 20 kV
- Dasar daya : 100 MVA

Selanjutnya node-node yang ada diklasifikasikan, yaitu Busbar G.I Blimbing diasumsikan sebagai Bus *Slack*, sedangkan node-node yang lain sepanjang saluran radial dipandang sebagai Bus Beban (*load bus*). Dalam hal ini tidak ada Bus Generator karena sepanjang saluran tidak terdapat pembangkitan.

4.4.1. Data Saluran

Jaringan distribusi primer G.I Blimbing menggunakan saluran kabel udara dengan spesifikasi seperti pada tabel 4-1.

Tabel 4-1
Data Penyulang Sistem 20 kV GI Blimbing

Jenis Konduktor	Penampang Nominal (mm)²	Resistansi Konduktor (r) (Ω/km)	GMR (mm)	Kuat Hantar Arus (A)
AAAC	150	0,2162	5,2365	425
AAAC	120	0.2688	4.6837	365

Keterangan:

AAAC : All Aluminium Alloy Conductor

GMR : Geometric Mean Radius Conductor

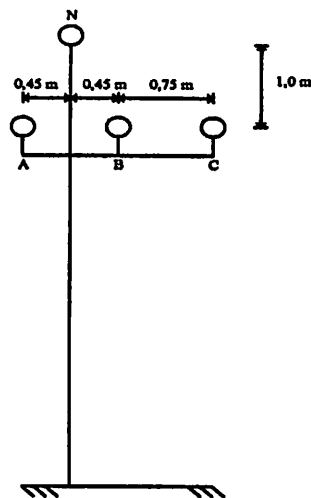
Adapun data saluran penyulang Glinting seperti pada tabel 4-2

Tabel 4-2
Data Saluran Penyulang Glinting

No Saluran	Dari Node	Ke Node	Panjang (m)
1	0	1	303
2	1	2	412
3	2	3	19
4	3	4	252
5	4	5	148

Tabel 4-2 selengkapnya dapat dilihat pada lampiran

Suatu struktur dan jarak antar konduktor ditunjukkan pada gambar dibawah ini:



Gambar 4-2
Struktur dan Jarak Antar Konduktor

Sumber: SPLN-64: 1985 "Petunjuk Pemilihan dan Penggunaan Pelebur Pada Sistem Distribusi Tegangan Menengah" Hal 65 ^[1]

4.4.2. Data Pembebanan

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

Tabel 4-3
Data Pembebanan Penyulang Glantung

No Node	Tegangan Awal		Pembebanan						Tipe Bus	
	V (pu)	θ ($^{\circ}$)	R		S		T			
			P (kW)	Q (kVAR)	P (kW)	Q (kVAR)	P (kW)	Q (kVAR)		
0	1,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	Slack
1	1,0000	0,0000	46,1648	27,3926	31,0288	18,4114	34,2452	20,3199		Beban
2	1,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000		Beban
3	1,0000	0,0000	10,9736	6,5114	10,0276	5,9500	13,2440	7,8585		Beban
4	1,0000	0,0000	38,2184	22,6775	43,8944	26,0454	45,4080	26,9435		Beban
5	1,0000	0,0000	21,1904	12,5736	23,8392	14,1454	26,6772	15,8293		Beban

Tabel 4-3 selengkapnya dapat dilihat pada lampiran

4.5. Analisa Perhitungan

Perhitungan aliran daya diawali dengan melakukan studi aliran daya dengan metode *Current Injection*. Studi aliran daya dilakukan untuk mengetahui besar tegangan, sudut fasa di tiap-tiap bus, dan arus yang mengalir pada saluran antar bus. Setelah studi aliran daya dilakukan, barulah dilakukan perhitungan aliran daya dan rugi-rugi daya pada saluran.

Untuk memudahkan perhitungan dan analisa pada sistem tenaga listrik, biasanya dipakai besarnya harga dalam per-unit (p.u). Harga per-satuan merupakan harga yang sebenarnya dibagi dengan harga dasar, harga dasar ini dapat dipilih secara acak. Besar tegangan yang dianalisa pada studi ini adalah 20 kV dan 100 MVA sebagai besar tegangan dasar dan daya dasar. Mengingat bahwa pada jaringan tidak dilakukan pengukuran faktor daya, maka pada analisa perhitungan ini diasumsikan besar faktor daya = 0,86.

4.5.1. Perhitungan Aliran Daya Penyulang Glitung

Perhitungan diawali dengan menampilkan *single line* diagram dari penyulang yang mewakili keadaan sistem sesungguhnya. Gambar 4-3 memperlihatkan *single line* diagram dari penyulang Glitung. Pada penyulang Glitung jumlah bus dan jumlah saluran masing-masing adalah:

❑ Bus *Slack* = 1

❑ Bus Beban = 23

❑ Jumlah Saluran = 26

4.6. Tampilan Program

Program dalam skripsi ini dieksekusi dengan menggunakan bantuan perangkat lunak komputer dengan bahasa pemrograman *Borland Delphi versi 0.6* dan diaplikasikan pada komputer *processor AMD Sempron XP 2600 MHz*, dengan *memory 512*. Mengenai jalannya program ikutilah prosedur berikut ini :

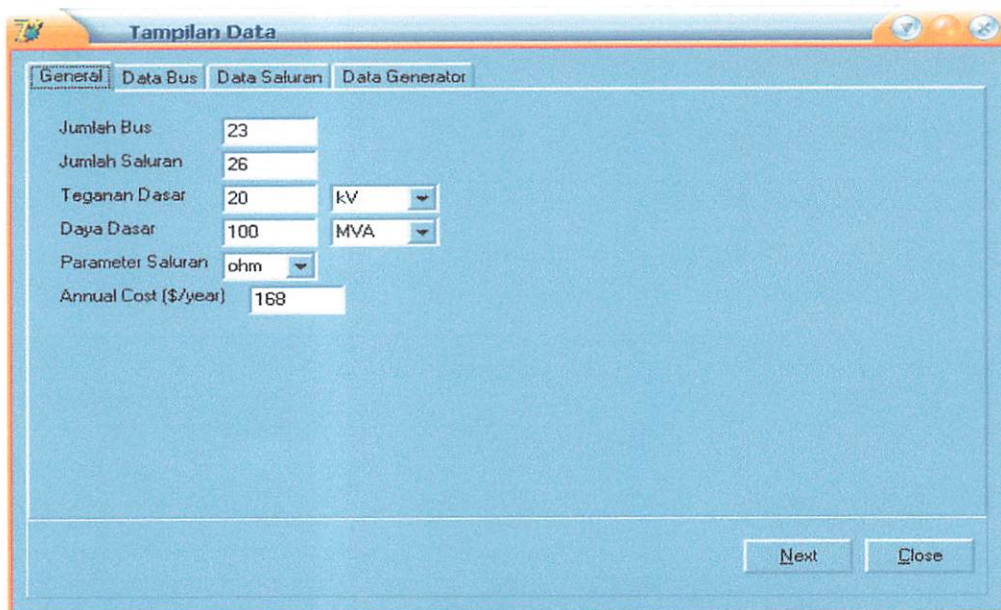
I. Tampilan Program.

1. Tekan tombol *Load File* untuk membuka data yang sudah tersimpan.



Gambar 4-3
Tampilan Utama Program

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



Gambar 4-4
Tampilan Inputan Data

3. Kemudian tekan tombol Data Saluran.

No	Dari	Ke	R (ohm)	X (ohm)	Lc (ohm)	Tr	Tu	Su (deg)	Kap (MVA)
1	1	2	1.2142	1.856	0	0	0	0	10000
2	2	3	0.8832	1.3501	0	0	0	0	10000
3	3	4	1.0896	1.6657	0	0	0	0	10000
4	4	5	0.8425	1.288	0	0	0	0	10000
5	5	6	0.1464	0.2238	0	0	0	0	10000
6	6	7	0.3628	0.5546	0	0	0	0	10000
7	1	8	0.2986	0.4564	0	0	0	0	10000
8	8	9	0.2581	0.3946	0	0	0	0	10000
9	9	10	0.2197	0.3358	0	0	0	0	10000
10	10	11	0.3226	0.4931	0	0	0	0	10000
11	1	12	0.5349	0.8177	0	0	0	0	10000
12	12	13	0.7444	1.1379	0	0	0	0	10000
13	12	14	0.3559	0.544	0	0	0	0	10000
14	14	15	0.2559	0.391	0	0	0	0	10000

Gambar 4-5
Tampilan Data Saluran

4. Kemudian tekan tombol *Data Bus* untuk menampilkan data pembebanan tiap bus.

Bus	absV (pu)	sudV (deg)	Pg (MW)	Qg (MVAR)	PL (MW)	QL (MVAR)	Cap (ohm)	Type Bus
1	1	0	0	0	0	0	0	1
2	1	0	0	0	1.4276	1.0707	0	3
3	1	0	0	0	0.7655	0.5741	0	3
4	1	0	0	0	1.4842	1.1132	0	3
5	1	0	0	0	0.8594	0.6446	0	3
6	1	0	0	0	0.1661	0.1245	0	3
7	1	0	0	0	0.619	0.4643	0	3
8	1	0	0	0	0.3342	0.2506	0	3
9	1	0	0	0	0.7353	0.5515	0	3
10	1	0	0	0	0.2871	0.2153	0	3
11	1	0	0	0	0.4398	0.3299	0	3
12	1	0	0	0	1.4908	1.1181	0	3
13	1	0	0	0	1.4538	1.0903	0	3
14	1	0	0	0	0.1972	0.1479	0	3
15	1	0	0	0	0.054	0.4005	0	3

Gambar 4-6
Tampilan Data Pembebanan

5. Tekan tombol *Next* kemudian tekan tombol *LF* awal untuk mengetahui hasil perhitungan aliran daya dengan metode pembanding *Newton Raphson* dan profil tegangan pada kondisi awal dan grafik tegangan tiap-tiap *node* terhadap asumsi tegangan awal.

Bus	absV (pu)	sudV (deg)	Pg (MW)	Qg (MVAR)	PL (MW)	QL (MVAR)	Supr (pu)	Type Bus
1	1.00000	0.00000	18.032	13.900	0.000	0.000	0.000	1
2	0.96262	-0.74855	0.000	0.000	1.428	1.071	0.000	3
3	0.94255	-1.17104	0.000	0.000	0.766	0.574	0.000	3
4	0.92259	-1.60771	0.000	0.000	1.484	1.113	0.000	3
5	0.91445	-1.79052	0.000	0.000	0.859	0.645	0.000	3
6	0.91377	-1.80582	0.000	0.000	0.166	0.125	0.000	3
7	0.91245	-1.83579	0.000	0.000	0.619	0.464	0.000	3
8	0.99711	-0.05999	0.000	0.000	0.334	0.251	0.000	3
9	0.99507	-0.10242	0.000	0.000	0.735	0.552	0.000	3
10	0.99421	-0.12042	0.000	0.000	0.287	0.215	0.000	3
11	0.99344	-0.13644	0.000	0.000	0.440	0.330	0.000	3
12	0.98434	-0.32315	0.000	0.000	1.491	1.118	0.000	3
13	0.97841	-0.44847	0.000	0.000	1.454	1.090	0.000	3
14	0.97967	-0.42138	0.000	0.000	0.197	0.148	0.000	3
15	0.97875	-0.44083	0.000	0.000	0.654	0.490	0.000	3

Gambar 4-7
Tampilan Hasil Profil Tegangan Tiap Bus

Hasil Loadflow

LF Awal Aliran Daya N.R N.R Result Hasil Program LF Akhir Aliran Daya C.I C.I Result Grafik

No	Dari	Ke	P (MW)	Q (MYAR)	Arus re (A)	Arus im (A)	Dari	Ke	P (MW)	Q
1	1	2	5.597	4.412	279.832	220.598	2	1	-5.442	
2	2	3	4.014	3.106	206.383	164.058	3	2	-3.953	
3	3	4	3.187	2.439	166.383	132.819	4	3	-3.138	
4	4	5	1.654	1.251	87.718	70.290	5	4	-1.644	
5	5	6	0.785	0.590	41.903	33.605	6	5	-0.785	
6	6	7	0.619	0.465	33.044	26.512	7	6	-0.618	
7	1	8	1.803	1.358	90.154	67.876	8	1	-1.799	
8	8	9	1.465	1.101	73.409	55.292	9	8	-1.463	
9	9	10	0.728	0.546	36.511	27.515	10	9	-0.727	
10	10	11	0.440	0.330	22.096	16.657	11	10	-0.440	
11	1	12	5.414	4.125	270.712	206.257	12	1	-5.352	
12	12	13	1.460	1.100	73.855	56.298	13	12	-1.454	
13	12	14	2.401	1.812	121.452	92.741	14	12	-2.393	
14	14	15	0.654	0.491	33.215	25.314	15	14	-0.654	

Current Injection Newton Rhapson Close

Gambar 4-8
Tampilan Hasil Aliran Daya Tiap Saluran

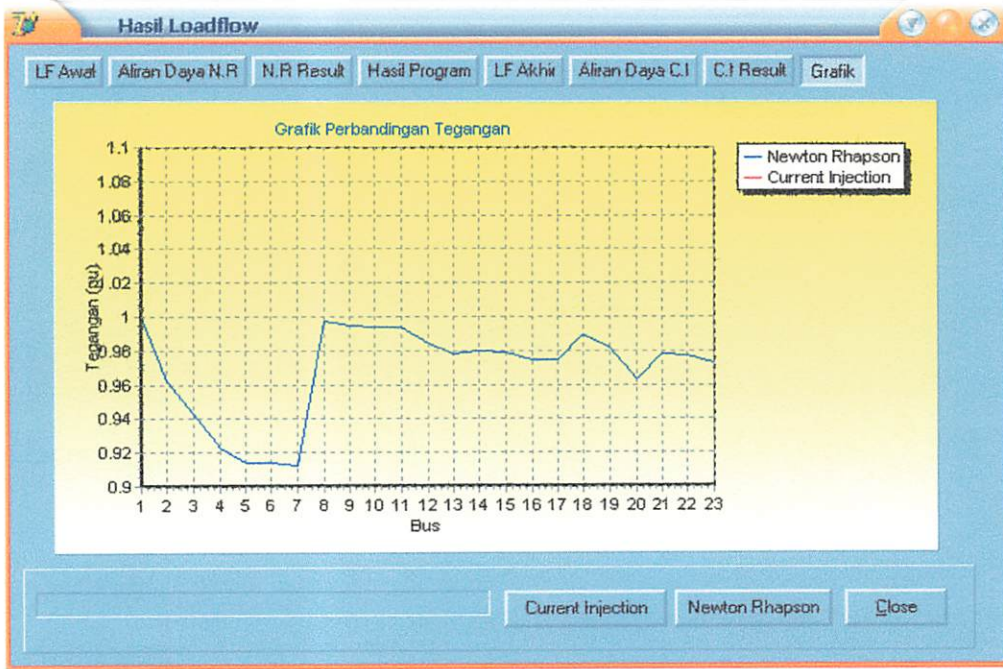
Hasil Loadflow

LF Awal Aliran Daya N.R N.R Result Hasil Program LF Akhir Aliran Daya C.I C.I Result Grafik

Jumlah Pembangkitan	18.032+ j 13.900	MVA
Jumlah Pembebanan	17.550+ j 13.163	MVA
Jumlah Rugi-Rugi	0.482+ j 0.738	MVA
Iterasi	2	
Waktu Hitung	0:0:0.0	

Current Injection Newton Rhapson Close

Gambar 4-9
Tampilan Total Aliran Daya Awal Pada Jaringan



Gambar 4-10
Grafik Tegangan Tiap-Tiap *Node* Terhadap Asumsi Tegangan Awal

Jumlah total hasil analisa menggunakan metode pembandingan *Newton Raphson* adalah:

Banyaknya Saluran = 26

Banyaknya *Node* = 23

Banyaknya Iterasi = 2

Total Pembangkitan = 18.032 + j 13.900 MVA

Total Pembebanan = 17.550 + j 13.163 MVA

Rugi Total Saluran = 0.482 + j 0.738 MVA

Waktu Hitung = 0 : 0 : 0 : 0

II. Tampilan program sesudah perbandingan menggunakan metode *Current Injection*.

1. Tekan tombol *LF* akhir untuk mengetahui hasil perhitungan aliran daya dengan metode *Current Injection* dan profil tegangan pada kondisi akhir dan grafik tegangan tiap-tiap *node* terhadap asumsi tegangan awal.

Bus	absV (pu)	sudV (deg)	Pg (MW)	Qg (MVAR)	PL (MW)	QL (MVAR)	Sups (pu)	Type Bus
1	1.00000	0.00000	17.872	13.654	0.000	0.000	0.000	1
2	0.98546	-0.30136	0.000	0.000	1.428	1.071	0.000	3
3	0.98177	-0.37930	0.000	0.000	0.766	0.574	0.000	3
4	0.97348	-0.54912	0.000	0.000	1.484	1.113	0.000	3
5	0.96577	-0.71349	0.000	0.000	0.859	0.645	0.000	3
6	0.96513	-0.72725	0.000	0.000	0.166	0.125	0.000	3
7	0.96368	-0.75417	0.000	0.000	0.619	0.464	0.000	3
8	0.99188	-0.16533	0.000	0.000	0.334	0.251	0.000	3
9	0.98533	-0.30058	0.000	0.000	0.735	0.552	0.000	3
10	0.98063	-0.39833	0.000	0.000	0.287	0.215	0.000	3
11	0.97426	-0.53268	0.000	0.000	0.440	0.330	0.000	3
12	0.98434	-0.32315	0.000	0.000	1.491	1.118	0.000	3
13	0.97841	-0.44847	0.000	0.000	1.454	1.090	0.000	3
14	0.97967	-0.42138	0.000	0.000	0.197	0.148	0.000	3
15	0.97875	-0.44083	0.000	0.000	0.654	0.490	0.000	3

Gambar 4-11
Profil Tegangan Tiap Bus Pada Kondisi Akhir

Hasil Loadflow

LF Awal | Aliran Daya N.R. | N.R Result | Hasil Program | LF Akhir | Aliran Daya C.I. | C.I Result | Grafik

No	Dari	Ke	P (MW)	Q (MVAR)	Arus re (A)	Arus im (A)	Dari	Ke	P (MW)	Q
1	1	2	2.219	1.684	110.937	84.200	2	1	-2.195	
2	2	3	0.768	0.577	38.791	29.495	3	2	-0.765	
3	3	4	0.000	0.000	0.000	0.000	4	3	0.000	
4	4	5	1.655	1.249	84.381	64.986	5	4	-1.645	
5	5	6	0.786	0.590	40.309	31.064	6	5	-0.786	
6	6	7	0.620	0.465	31.787	24.505	7	6	-0.619	
7	1	8	5.020	3.839	251.012	191.966	8	1	-4.990	
8	8	9	4.656	3.543	234.200	179.286	9	8	-4.634	
9	9	10	3.898	2.957	197.034	151.106	10	9	-3.885	
10	10	11	3.598	2.721	182.471	140.028	11	10	-3.581	
11	1	12	5.414	4.125	270.712	206.257	12	1	-5.352	
12	12	13	1.460	1.100	73.855	56.298	13	12	-1.454	
13	12	14	2.401	1.812	121.452	92.741	14	12	-2.393	
14	14	15	0.654	0.491	33.215	25.314	15	14	-0.654	

Current Injection | Newton Rhapson | Close

Gambar 4-12
Tampilan Hasil Aliran Daya Tiap Saluran

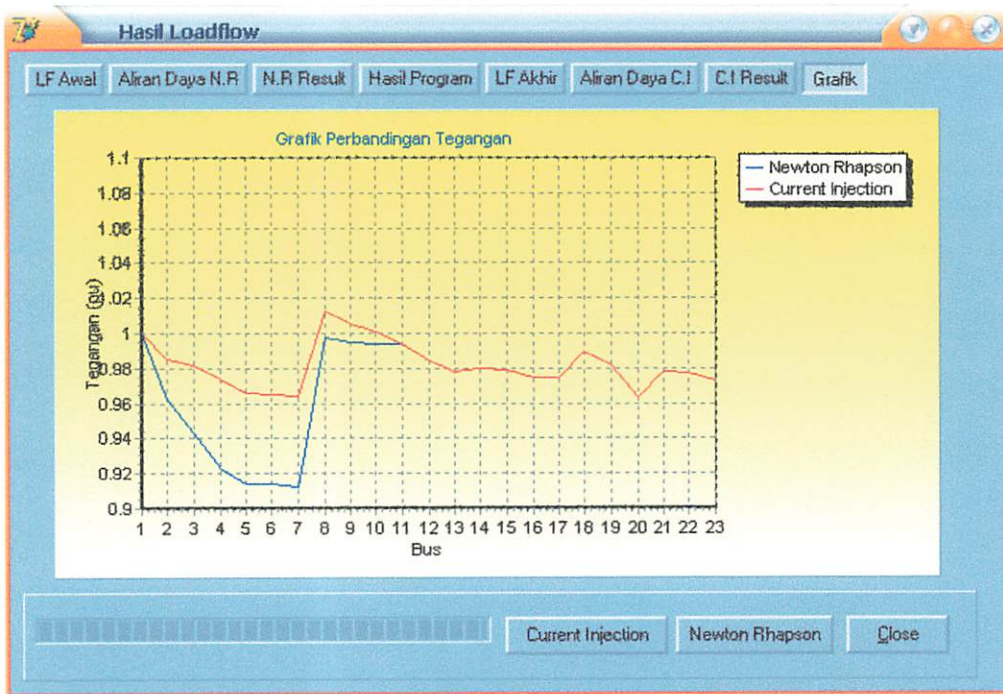
Hasil Loadflow

LF Awal | Aliran Daya N.R. | N.R Result | Hasil Program | LF Akhir | Aliran Daya C.I. | C.I Result | Grafik

Jumlah Pembangkitan	17.872+ j 13.654	MVA
Jumlah Pembebanan	17.550+ j 13.163	MVA
Jumlah Rugi-Rugi	0.321+ j 0.491	MVA
Iterasi	3	
Waktu Hitung	0:0:1:391	

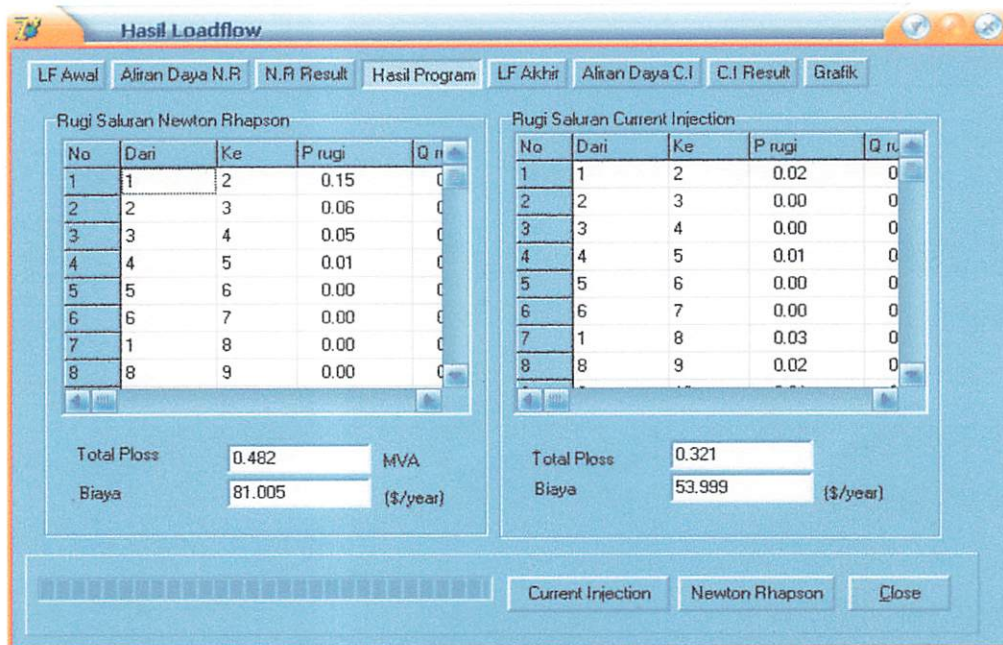
Current Injection | Newton Rhapson | Close

Gambar 4-13
Tampilan Total Aliran Daya Akhir Pada Jaringan



Gambar 4-14
Grafik Perbandingan Tegangan Tiap-Tiap *Node* Terhadap Asumsi Tegangan Awal

2. Perbandingan besar rugi daya dan biaya sesuai hasil perhitungan program.



Gambar 4-15
Perbandingan Total Hasil Perhitungan Program Besar Rugi-Rugi Daya dan Biaya

4.6. Hasil Perhitungan

Setelah melakukan analisa dengan metode *Current Injection* maka akan diperoleh besarnya tegangan dari tiap-tiap *node*, besarnya rugi-rugi daya saluran, dan aliran daya saluran seperti pada tabel 4-4 sampai 4-8:

Tabel 4-4
Kondisi Tegangan Dan Sudut Fasa Tegangan Tiap Node
Penyulang Glintang Pada Load Flow Awal

Nomor Bus	Tegangan (pu)	Sudut V (Deg)
1	1,00000	0,00000
2	0,96262	-0,74855
3	0,94255	-1,17104
4	0,92259	-1,60771
5	0,91445	-1,79052

Tabel 4-4 selengkapnya dapat dilihat pada lampiran

Tabel 4-5
Kondisi Tegangan Dan Sudut Fasa Tegangan Tiap Node
Penyulang Glintang Pada Load Flow Akhir

Nomor Bus	Tegangan (pu)	Sudut V (Deg)
1	1,00000	0,00000
2	0,98546	-0,30136
3	0,98177	-0,37930
4	0,97348	-0,54912
5	0,96577	-0,71349

Tabel 4-5 selengkapnya dapat dilihat pada lampiran

Tabel 4-6
Besar Rugi-Rugi Daya Tiap Saluran
Penyulang Glintang Dengan Metode *Current Injection*

Saluran	Daya Aktif (P) (MW)	Daya Reaktif(Q) (MVAR)
1-2	1,428	1,071
2-3	0,766	0,574
3-4	1,484	1,113
4-5	0,859	0,645
5-6	0,166	0,125

Tabel 4-6 selengkapnya dapat dilihat pada lampiran

Tabel 4-7
Arus Tiap Saluran Penyulang Glintang
Dengan Metode *Current Injection*

Saluran	Arus <i>re</i> (Ampere)	Arus <i>im</i> (Ampere)
1-2	-221,874	-168,4
2-3	-77,582	-58,99
3-4	0,000	0,000
4-5	-168,762	-129,972
5-6	-80,618	-60,128

Tabel 4-7 selengkapnya dapat dilihat pada lampiran

Jumlah total hasil analisa menggunakan metode *Current Injection* adalah:

Banyaknya Saluran = 26

Banyaknya *Node* = 23

Banyaknya Iterasi = 3

Total Pembangkitan = 17.872 + j 13.654 MVA

Total Pembebanan = 17.550 + j 13.165 MVA

Rugi Total Saluran = 0.321 + j 0.491 MVA

Waktu Hitung = 0 : 0 : 1 : 390

Table 4-8
Perubahan Rugi Daya Pada Saluran

No	Jumlah Perhitungan	Metode Newton Raphson		Metode Current Injection	
		P (MVA)	Q (MVA)	P (MVA)	Q (MVA)
1	Total Pembangkitan	18.032	13.900	17.872	13.654
2	Total Pembebanan	17.550	13.163	17.550	13.165
3	Rugi Total Saluran	0.482	0.738	0.321	0.491
4	Rugi Total Biaya	81.005 (\$/year)		53.999 (\$/year)	
5	Banyaknya Iterasi	2		3	
6	Waktu Hitung	0 : 0 : 0 : 16		0 : 0 : 1 : 390	

Dari tabel di atas dapat dilihat perubahan rugi daya pada saluran yaitu:

1. Dengan menggunakan metode pembandingan *Newton Raphson* untuk daya aktif (P) sebesar = 0,482 MVA, dan rugi daya reaktif (Q) sebesar = 0,738 MVA.
Dengan menggunakan metode *Current Injection* untuk daya aktif (P) sebesar = 0,321 MVA, dan rugi daya reaktif (Q) sebesar = 0,491 MVA.
2. Rugi biaya total dengan menggunakan metode *Newton Raphson* sebesar = 81,005 \$/year, sedangkan dengan menggunakan metode *Current Injection* = 53,999 \$/year.

BAB V

KESIMPULAN

5.1. Kesimpulan

Setelah dilakukan analisa perhitungan aliran daya tiga fasa pada sistem distribusi untuk sistem operasi tidak seimbang sehingga diperoleh model polynomial beban yang diperoleh dari perumusan aliran beban pada daya konstan, impedansi konstan dan impedansi dengan beban konstan. Proses perhitungan analisa aliran daya dan rugi daya pada saluran dengan menggunakan metode pembandingan *Newton Raphson* tidak banyak mengalami iterasi dibandingkan dengan metode *Current Injection*. Dalam perumusan TCIM (*Three-Phase Current Injection Method*) menggunakan matrik Jacobian memiliki struktur yang sama dengan matrik admitansi bus yang nantinya digunakan untuk analisis tambahan tegangan.

Untuk lebih jelas dari aplikasi metode *TCIM* pada saluran jaringan distribusi radial yaitu:

1. Analisa aliran daya dengan metode pembandingan *Newton Raphson* sebelum menggunakan metode *TCIM* nilainya sudah konvergen pada iterasi ke - 2, sedangkan dengan menggunakan metode *TCIM* nilainya *konvergen* pada iterasi ke - 3.
2. Dari analisa aliran daya pada saluran distribusi diperoleh nilai rugi-rugi daya pada saluran yang menggunakan metode pembandingan *Newton Raphson* diperoleh nilai daya aktif (P) sebesar = 0,482 MVA sedangkan

dengan menggunakan metode *Current Injection* nilai daya aktif (P) sebesar = 0,321 MVA, sehingga rugi pada saluran dengan perbandingan kedua metode dapat berkurang nilainya sebesar = 0,161 MVA. Dari analisa aliran daya pada saluran distribusi diperoleh nilai rugi-rugi daya pada saluran yang menggunakan metode pembanding *Newton Raphson* diperoleh nilai daya reaktif (Q) sebesar = 0,738 MVA sedangkan dengan menggunakan metode *Current Injection* nilai daya reaktif (Q) sebesar = 0,491 MVA, sehingga rugi pada saluran dengan perbandingan kedua metode dapat berkurang nilainya sebesar = 0,247 MVA.

3. Rugi biaya total dengan menggunakan metode *Newton Raphson* = 81.005 \$/year, sedangkan dengan menggunakan metode *Current Injection* sebesar = 53.999 \$/year.

Daftar Pustaka

- [1] Paulo A.N Garcia, Sandoval Caniero, Jr “ **Three- Phasa Power Flow Calkculations Using the Current Injection Method**” IEEE Transaction On Power System, Vol, 15 No 2 , 2000.
- [2] Turan Gonen “ **Electric Power Distribution System Engineering** “ University of Missouri at Columbia.
- [3] Wiliam D. Stevenson.Jr “ **Analisis Sistem Tenaga Elektrik** “ Erlangga 1993, Edisi Ke Empat.

LAMPIRAN



BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

Nama Mahasiswa : PAULUS HARRY D.L
N.I.M. : 95.12.008
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Energi Listrik
Judul Skripsi : ANALISA ALIRAN DAYA PADA SISTEM
DISTRIBUSI MENGGUNAKAN METODE *THREE
CURRENT INJECTION METHOD (TCIM)*

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

Hari : Jum'at
Tanggal : 23 Maret 2007
Dengan Nilai : 73,55 (B+) *By*



(Ir. Mochtar Asroni, MSME)
Ketua

Panitia Ujian

(Ir. F. Yudi Limpraptono, MT)
Sekretaris

Anggota Penguji

(Ir. Teguh Herbasuki, MT)
Penguji Pertama

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



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

PERSETUJUAN PERBAIKAN SKRIPSI

Dari hasil ujian skripsi Jurusan Teknik Elektro jenjang strata satu (S-1) yang diselenggarakan pada :

Hari : Jum'at
Tanggal : 23 Maret 2007

Telah dilakukan perbaikan skripsi oleh :

1. Nama : PAULUS HARRY D.L
2. NIM : 95.12.008
3. Jurusan : Teknik Elektro S-1
4. Konsentrasi : Teknik Energi Listrik
5. Judul Skripsi : *Analisa Aliran Daya Pada Sistem Distribusi Menggunakan metode Three Current Injection Method (Tcim)*

Perbaikan meliputi :

No	Materi Perbaikan	Paraf
1	Perbaikan Abstraksi Dalam Bahasa Inggris	
2	Perbaikan <i>Single Line</i> Diagram Glintung Sesuai Dengan Hasil Perhitungan.	
3	Perbaikan Jumlah Bus & Jumlah Saluran	

Anggota Penguji

Penguji I

Ir. Teguh Herbasuki, MT
NIP. Y. 103 8900 209

Penguji II

Irrine Budi, S, ST, MT
NIP. 132 314 400

Dosen Pembimbing

Ir. H. Taufik Hidayat, MT
NIP.Y.101 8700 151



INSTITUT TEKNOLOGI NASIONAL MALANG
JL. RAYA KARANGLO KM 02
MALANG

Lampiran : 1 (satu) berkas
Pembimbing Skripsi

Kepada : Yth. Bapak Ir. H. Taufik Hidayat, MT
Dosen Institut Teknologi Nasional
MALANG

Yang bertandatangan di bawah ini:
Nama : PAULUS HARRY D. L
Nim : 95. 12. 008
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Energi Listrik

Dengan ini mengajukan permohonan, kiranya Bapak bersedia menjadi Dosen Pembimbing Utama/Pendamping *), untuk penyusunan Skripsi dengan judul (proposal terlampir):

ANALISA ALIRAN DAYA PADA SISTEM DISTRIBUSI MENGGUNAKAN METODE THREE PHASE CURRENT INJECTION METHOD (TCIM)

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

Demikian surat permohonan kami dan atas kesediaan Bapak kami ucapkan terima kasih.

Malang, Mei 2006

Hormat kami,

Paulus Harry D. L
NIM: 95. 12. 008

Ketua
Jurusan Teknik Elektro S-1

Ir. F. Yudi Lampraptono, MT
NIP. 1039500274

Form S-3a



PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa/i:

Nama : PAULUS HARRY D. L

Nim : 95. 12. 008

Semester : XXII

Jurusan : Teknik Elektro S-1

Konsentrasi : Teknik Energi Listrik

Dengan ini menyatakan bersedia / tidak bersedia *) Membimbing Skripsi dari mahasiswa tersebut, dengan judul:

ANALISA ALIRAN DAYA PADA SISTEM DISTRIBUSI MENGGUNAKAN METODE THREE PHASE CURRENT INJECTION METHOD (TCIM)

Demikian surat pernyataan ini kami buat agar dapat dipergukan seperlunya.

Malang, 20 Mei 2006

Kami yang membuat pernyataan,

Ir. H. Taufik Hidayat, MT
NIP. 101 8700 151

Catatan:

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

Form S-3b



PERMOHONAN PERSETUJUAN SKRIPSI

Yang betanda tangan dibawah ini :

Nama : PAULUS HARRY D.L
 NIM : 95.12.008
 Semester :
 Fakultas : Teknologi Industri
 Jurusan : Teknik Elektro S-1
 Konsentrasi : Teknik Elektronika/ Teknik Energi Listrik
 Alamat : Jl. Candi II C NO 56A MALANG

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

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

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

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

Telah diteliti kebenaran data tersebut diatas
 Recording Teknik Elektro

[Handwritten Signature]
 (.....)

Malang, APRIL 2006

Pemohon

[Handwritten Signature]

(PAULUS HARRY D.L)

Disetujui
 Ketua Jurusan Teknik Elektro

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

Mengetahui
 Dosen Wali

[Handwritten Signature]

(Ir. YUNIOR SIAHAAN)

Catatan :


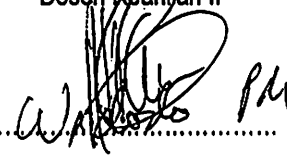

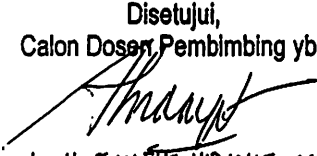
Bagi mahasiswa yang telah memenuhi persyaratan mengambil SKRIPSI agar membuat proposal dan mendapat persetujuan dari Ketua Jurusan/Sekretaris Jurusan T. Elektro S-1

1. *10E 304/137 - 2.22*
2.
3. *- praktikum Siskom*



BERITA ACARA SEMINAR PROPOSAL SKRIPSI JURUSAN TEKNIK ELEKTRO S-1

Konsentrasi : Teknik Energi Listrik/Teknik Elektronika*)

1.	Nama Mahasiswa: PAULUS HARRY D.L			Nim: 95.12.008
2.	Keterangan	Tanggal	Waktu	Tempat
	Pelaksanaan	16-06-06	13.00	Ruang:
3.	Spesifikasi Judul (berilah tanda silang)**)			
	a. Sistem Tenaga Elektrik	e. Elektronika & Komponen		
	b. Energi & Konversi Energi	f. Elektronika Digital & Komputer		
	c. Tegangan Tinggi & Pengukuran	g. Elektronika Komunikasi		
	d. Sistem Kendali Industri	h. lainnya		
4.	Judul Proposal yang diseminarkan Mahasiswa	ANALISA ALIRAN DAYA PADA SISTEM DISTRIBUSI MENGGUNAKAN METODE THREE CURRENT INJECTION METHOD (TCIM)		
5.	Perubahan Judul yang diusulkan oleh Kelompok Dosen Keahlian		
6.	Catatan:			
			
7.	Persetujuan Judul Skripsi			
	Disetujui, Dosen Keahlian I		Disetujui, Dosen Keahlian II	
				
	
Mengetahui, Ketua Jurusan		Disetujui, Calon Dosen Pembimbing ybs		
				
Ir. F. Yudi Limpraptono, MT NIP. P. 1039500274		Ir. H. TAU FIE HIDAYAT, MT		

Perhatian:

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

PT. PLN (PERSERO)

DISTRIBUSI JAWA TIMUR

AREA PELAYANAN & JARINGAN MALANG

: 0341 - 326034 (Hunting)
: 211

Facsimile : 0341 - 362046

Nomor : 1076 /330/ APJ-MLG/ 2006
Surat Sdr.No : ITN-1840/III.TA/2 /2006
Lampiran : -
Perihal : Survey

Malang , 20 Juli 2006

Kepada
Fakultas Teknologi Industri
Jurusan T. Sipil & Perencanaan
Prog. Pascasarjana Teknik
ITN Malang
Di
MALANG

Dengan Hormat,

Menindaklanjuti surat Saudara nomor ITN-1840/III.TA/2/ 2006 tanggal 29 Juni 2006 perihal tersebut di atas, maka dengan ini kami beritahukan bahwa pada prinsipnya kami tidak keberatan / mengizinkan mahasiswa saudara atas nama :

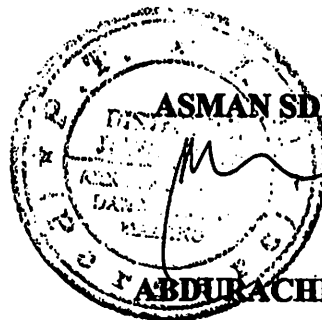
PAULUS HARRY DL

NIM : 95.12.008

Untuk melaksanakan survey di PT. PLN (PERSERO) Area Pelayanan dan Jaringan Malang mulai tanggal 01 Agustus 2006 sampai dengan 31 Agustus 2006 dengan catatan PLN hanya memberikan data yang **TIDAK BERSIFAT RAHASIA**.

Sebelumnya diminta agar siswa tersebut mengisi surat pernyataan, dilengkapi dengan pas photo ukuran 3 x 4 Cm di SDM & Administrasi PT. PLN (Persero) Area Pelayanan dan Jaringan Malang.

Demikian agar menjadikan maklum.



ASMAN SDM & ADMINISTRASI
ABDURACHMAN SHODIQ, SH, ST/2

Tembusan:

1. Supervisor terkait
2. Manajer UJ & UPJ terkait



FORMULIR BIMBINGAN SKRIPSI

Nama : Paulus Harry D.L
Nim : 95.12.008
Masa Bimbingan : 16 Juni 2006 s/d 16 Desember 2006
Judul Skripsi : Analisa Aliran Daya Pada Sistem Distribusi Menggunakan Metode Three Current Injection Method (Tcim)

No	Tanggal	Uraian	Paraf Pembimbing
1	15-07-06	Selesaikan Bab IV analisis ugr.	Al
2	10-10-06	Buat Kesimpulan bab IV	Al.
3	12-10-06	Kesimpulan 1, 2, 3 Metode TCIM tampilkan yg pertama, diikuti metode	Al
4		perbandingannya.	
5	1-11-06	Bab V: Cambahkan sumber pustaka yg (Journal of science)	Al.
6	12-12-06	Rangkai Bab II.	Al
7	09-01-07	Revisi Bab II. tambahkan pustaka buku, tabel, loop	Al
8	23-02-07	Rangkai Bab I.	Al
9	07-01-07	Selesaikan Tesis dan kesimpulan	Al
10	02-03-07	Revisi Summary Akhir	Al

Malang,
Dosen Pembimbing,


(Ir. H. Taufik Hidayat, MT)



INSTITUT TEKNOLOGI NASIONAL
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO


Formulir Perbaikan Ujian Skripsi

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

NAMA : Paulus Hary
NIM : 95-12-008
Perbaikan meliputi :

~ Single Line Diagram Olintong → sesuaikan dg
hasil perhit → jml. bus & jml. saluran

Malang, 23 Maret 07


Ir. Teguh Herbandi, MT



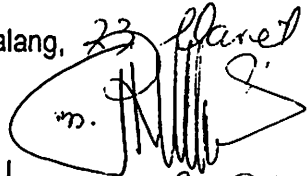
INSTITUT TEKNOLOGI NASIONAL
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

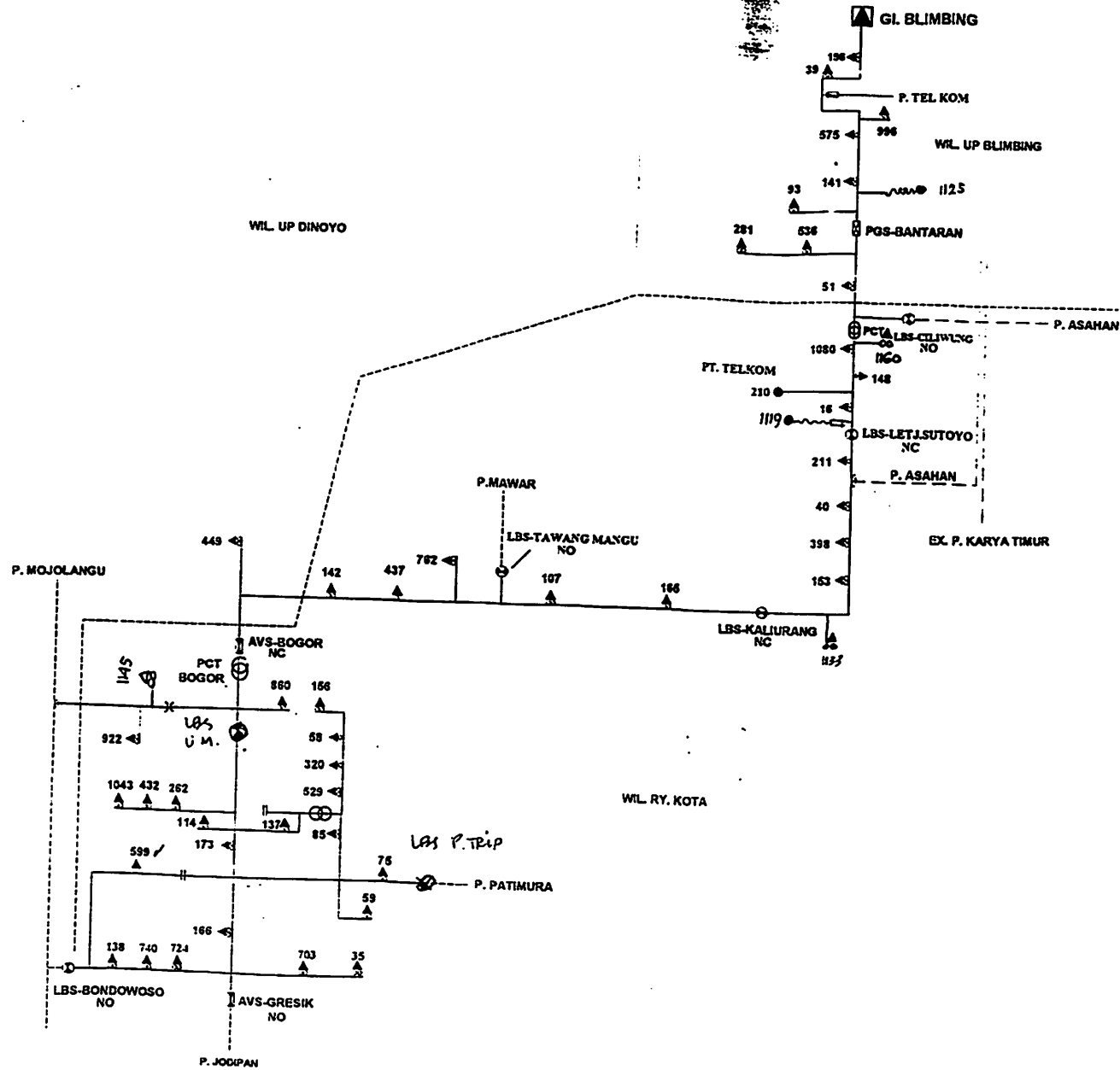
Formulir Perbaikan Ujian Skripsi

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

NAMA : Paulus Hary
NIM : 95.12.008
Perbaikan meliputi :

abstrak hrs lebih

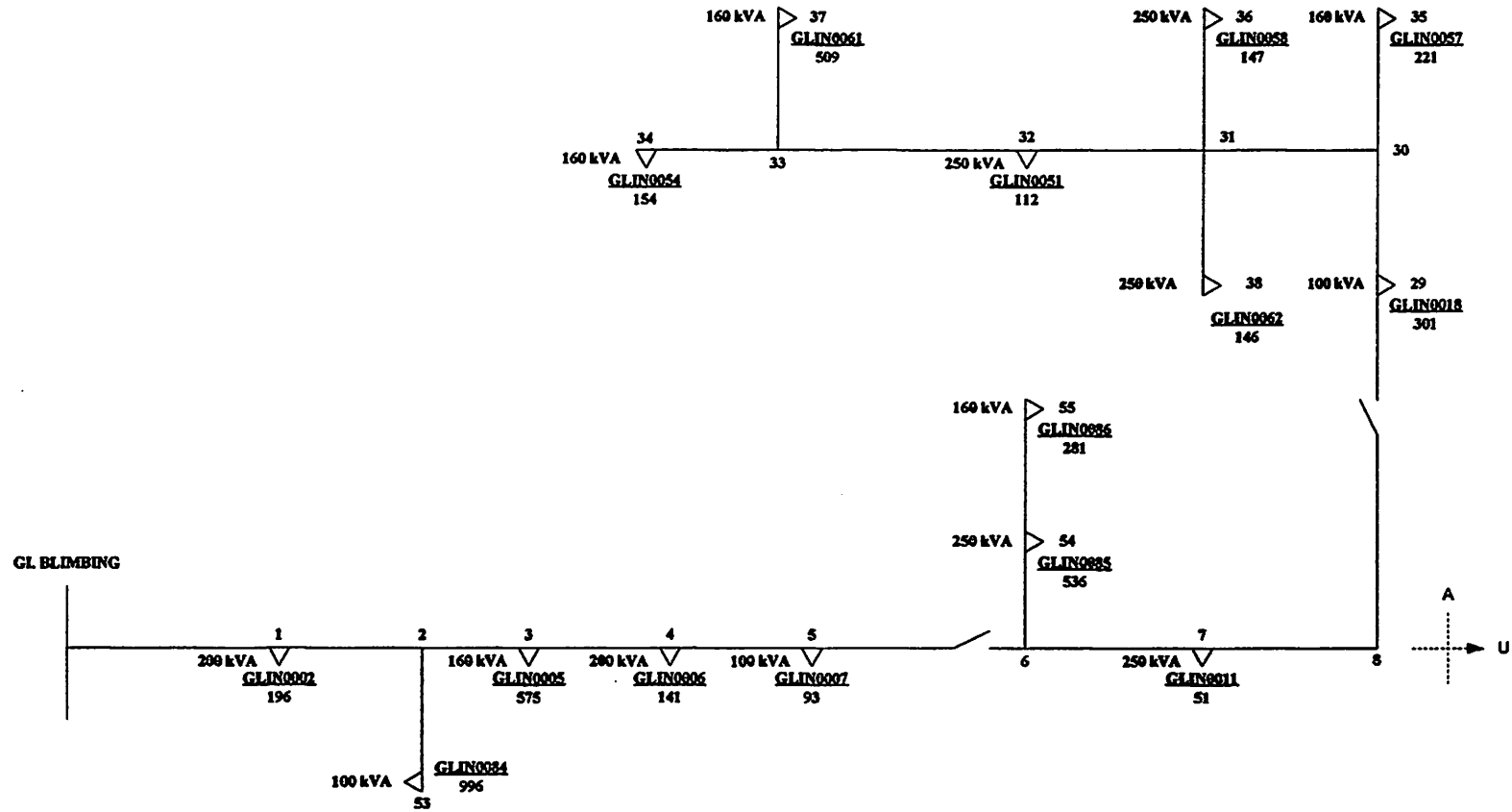
Malang, 23 Maret 07

(Irni Budi S)

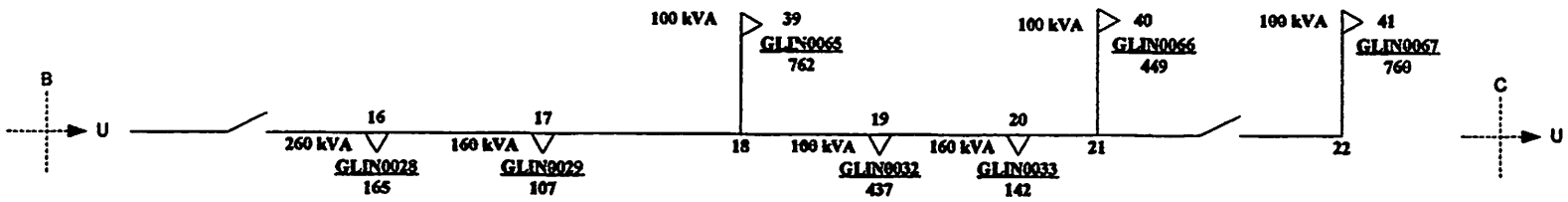
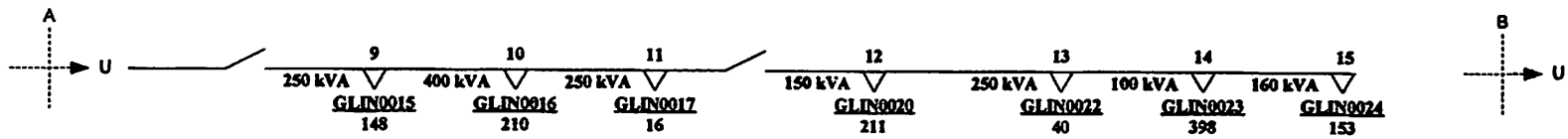


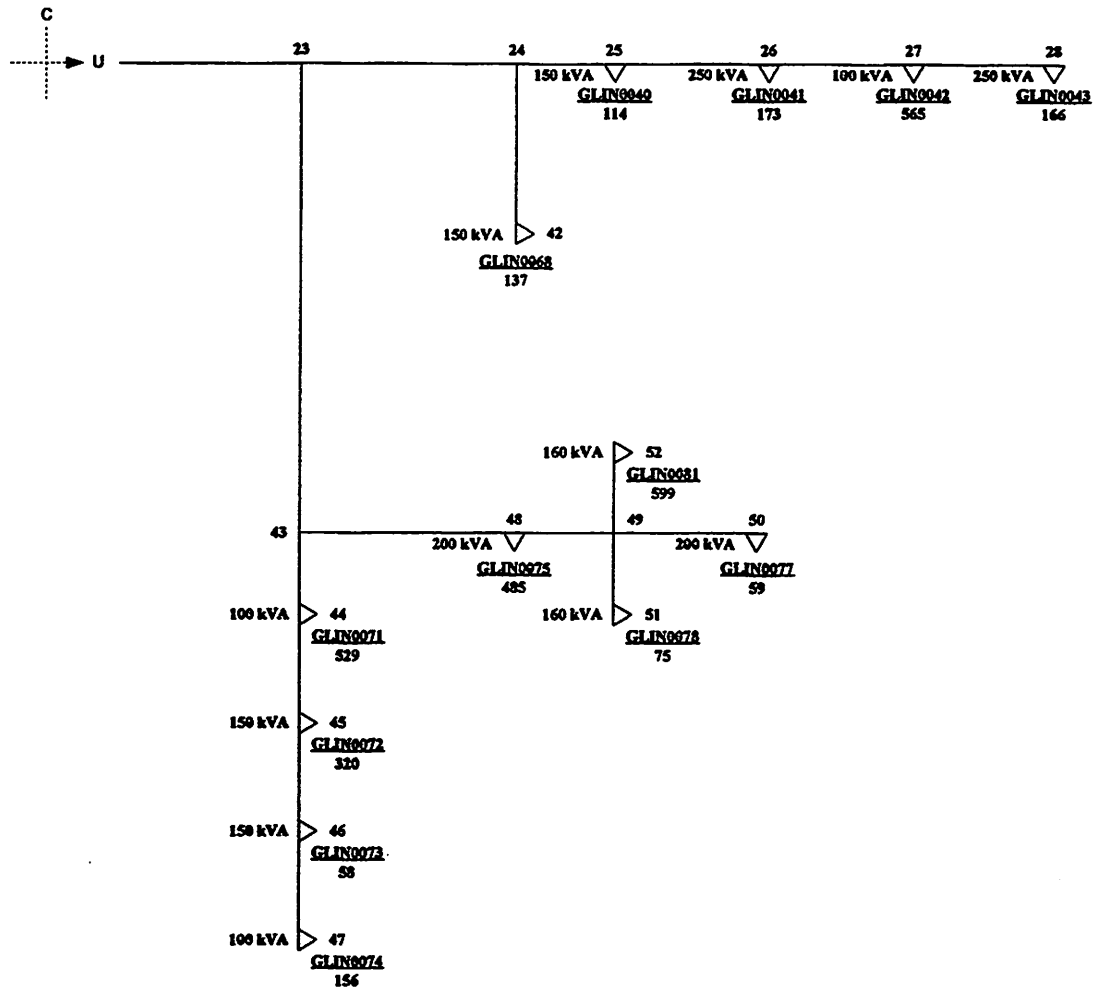
- ▲ GI
- ⊠ PGS
- ⊠ AVS
- ⊠ CO
- ⊠ LBS
- ⊠ RECLOSER
- SUTM 20 KV
- ⊠ PCT

PT. PLN (PERSERO) DISTRIBUSI JATIM AP&J MALANG		
P. GLINTUNG GI. BLIMBING		
DIAMBAR	DIPERIKSA	DISELUAR
OPOST	DJUMADL D'	ARIEP H

PENYULANG GLINTUNG







DATA PANJANG SALURAN PENYULANG GLINTUNG

No Saluran	Dari Node	Ke Node	Panjang (m)
1	1	2	303
2	2	3	412
3	3	4	19
4	4	5	252
5	5	6	148
6	6	7	266
7	7	8	63
8	8	9	150
9	9	10	303
10	10	11	86
11	11	12	202
12	12	13	324
13	13	14	217,5
14	14	15	86
15	15	16	146
16	16	17	410
17	17	18	96
18	18	19	328
19	19	20	241
20	19	40	79
21	20	21	51
22	21	22	348
23	22	23	386

23	1,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	Beban
24	1,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	Beban
25	1,000	0,000	17,2172	10,2161	11,352	6,7359	11,352	6,7359	Beban
26	1,000	0,000	37,84	22,4529	39,732	23,5756	46,354	27,5048	Beban

TAMPILAN DATA BUS PADA PROGRAM

Bus	Teg. (V)	Sudut (°)	Pg (MW)	Qg (MW)	PL (MW)	QL (MW)	Cap (Ω)	TipeBus
1	1	0	0	0	0	0	0	1
2	1	0	0	0	1.4276	1.0707	0	3
3	1	0	0	0	0.7656	0.5741	0	3
4	1	0	0	0	1.4842	1.1132	0	3
5	1	0	0	0	0.8594	0.6446	0	3
6	1	0	0	0	0.1661	0.1245	0	3
7	1	0	0	0	0.619	0.4643	0	3
8	1	0	0	0	0.3342	0.2506	0	3
9	1	0	0	0	0.7353	0.5515	0	3
10	1	0	0	0	0.2871	0.2153	0	3
11	1	0	0	0	0.4398	0.3299	0	3
12	1	0	0	0	1.4908	1.1181	0	3
13	1	0	0	0	1.4538	1.0903	0	3
14	1	0	0	0	0.1972	0.1479	0	3
15	1	0	0	0	0.654	0.4905	0	3
16	1	0	0	0	1.4906	1.1179	0	3
17	1	0	0	0	0.045	0.0338	0	3
18	1	0	0	0	0.1324	0.0993	0	3
19	1	0	0	0	0.3921	0.2941	0	3
20	1	0	0	0	2.8164	2.1123	0	3
21	1	0	0	0	0.1776	0.1332	0	3
22	1	0	0	0	0.6146	0.4609	0	3
23	1	0	0	0	0.9675	0.7256	0	3

TAMPILAN DATA SALURAN PADA PROGRAM

No.	Dari	Ke	R (Ω)	X (Ω)	Kap. (MVA)
1	1	2	1.2142	1.856	10000
2	2	3	0.8832	1.3501	10000
3	3	4	1.0896	1.6657	10000
4	4	5	0.8425	1.288	10000
5	5	6	0.1464	0.2238	10000
6	6	7	0.3628	0.5546	10000
7	7	8	0.2986	0.4564	10000
8	8	9	0.2581	0.3946	10000
9	9	10	0.2197	0.3358	10000
10	10	11	0.3226	0.4931	10000
11	1	12	0.5349	0.8177	10000
12	12	13	0.7444	1.1379	10000
13	12	14	0.3559	0.544	10000
14	14	15	0.2558	0.391	10000
15	14	16	0.5935	0.9072	10000
16	16	17	0.0283	0.0433	10000
17	1	18	0.3853	0.589	10000
18	18	19	0.6553	1.0017	10000
19	18	20	1.6516	2.5247	10000
20	19	21	0.3349	0.5119	10000
21	21	22	0.3191	0.4878	10000
22	21	23	1.0138	1.5497	10000
23	4	11	0.0448	0.0684	10000
24	9	13	0.0374	0.0574	10000
25	15	19	0.2408	0.3682	10000
26	5	16	0.0566	0.0866	10000

TABEL LOAD FLOW AWAL DENGAN METODE *NEWTON RAPHSON*

Bus	Teg. (Pu)	Sud. V (°)	Pg (MW)	Qg (MW)	PL (MW)	QL (MW)	Sup. (Pu)	Tipe Bus
1	1.00000	0.00000	18.032	13.900	0.0000	0.000	0.000	1
2	0.96262	-0.74855	0.0000	0.0000	1.428	1.071	0.000	3
3	0.94255	-1.17104	0.0000	0.0000	0.765	0.574	0.000	3
4	0.92259	-1.60771	0.0000	0.0000	1.484	1.113	0.000	3
5	0.91445	-1.79052	0.0000	0.0000	0.859	0.645	0.000	3
6	0.91377	-1.80582	0.0000	0.0000	0.166	0.125	0.000	3
7	0.91245	-1.83579	0.0000	0.0000	0.619	0.464	0.000	3
8	0.99711	-0.05999	0.0000	0.0000	0.334	0.251	0.000	3
9	0.99507	-0.10252	0.0000	0.0000	0.735	0.551	0.000	3
10	0.99421	-0.12042	0.0000	0.0000	0.287	0.215	0.000	3
11	0.99344	-0.13644	0.0000	0.0000	0.440	1.090	0.000	3
12	0.98434	-0.32315	0.0000	0.0000	1.491	1.118	0.000	3
13	0.97841	-0.4487	0.0000	0.0000	1.454	0.330	0.000	3
14	0.97967	-0.42138	0.0000	0.0000	1.491	1.118	0.000	3
15	0.97875	-0.44083	0.0000	0.0000	1.454	0.490	0.000	3
16	0.97465	-0.52782	0.0000	0.0000	1.491	1.118	0.000	3
17	0.97464	-0.52792	0.0000	0.0000	0.045	0.034	0.000	3
18	0.98908	-0.22162	0.0000	0.0000	0.132	0.099	0.000	3
19	0.98132	-0.38365	0.0000	0.0000	0.392	0.294	0.000	3
20	0.96312	-0.76619	0.0000	0.0000	2.816	2.112	0.000	3
21	0.97807	-0.45212	0.0000	0.0000	0.178	0.133	0.000	3
22	0.97700	-0.45212	0.00000	0.0000	0.615	0.461	0.000	3
23	0.97266	-0.56701	0.0000	0.0000	0.967	0.726	0.000	3

TABEL ALIRAN DAYA AWAL DENGAN METODE *NEWTON RAPHSON*

No	Dari	Ke	P (MW)	Q (MVAR)	Arus re(A)	Arus Im(A)	Dari	Ke	P (MW)	Q (MVAR)	Arus Re(A)	Arus Im(A)
1	1	2	5.597	4.412	279.832	220.598	2	1	-5.442	-4.176	-279.832	-220.598
2	2	3	4.014	3.106	206.383	164.058	3	2	-3.953	-3.012	-3.012	164.058
3	3	4	3.187	2.439	166.383	132.819	4	3	-3.138	-2.364	-166.383	-132.819
4	4	5	1.654	1.251	87.718	70.290	5	4	-1.644	-1.235	-87.718	-70.290
5	5	6	0.785	0.590	41.903	33.605	6	5	-0.785	-0.590	-41.903	33.605
6	6	7	0.619	0.465	33.044	26.512	7	6	-0.618	-0.464	-33.044	-26.512
7	7	8	1.803	1.358	90.154	67.876	8	1	-1.799	-1.352	-90.154	-67.870
8	8	9	1.465	1.101	73.409	55.292	9	8	-1.463	-1.098	-73.409	-55.292
9	9	10	0.728	0.546	36.511	27.515	10	9	-0.727	-0.546	36.511	-27.515
10	10	11	0.440	0.330	22.096	16.657	11	10	-0.440	-0.330	22.096	-16.657
11	1	12	5.414	4.125	270.712	286.257	12	1	-5.352	-4.030	-270.712	-206.257
12	12	13	1.460	1.100	73.855	56.298	13	12	-1.454	-1.090	-73.855	-56.298
13	12	14	2.401	1.821	121.452	92.741	14	13	-2.393	-1.800	-121.452	-92.741
14	13	15	0.654	0.491	33.215	25.314	15	14	-0.654	-0.490	-33.215	-25.314
15	14	16	1.541	1.160	78.227	59.806	16	15	-1.536	-1.152	-78.227	-59.806
16	15	17	0.045	0.034	2.292	1.755	17	16	-0.045	-0.034	-2.292	-1.755
17	1	18	5.218	4.006	260.921	200.276	18	17	-5.177	-3.942	-260.921	-200.276
18	18	19	2.173	1.646	109.520	83.635	19	18	-2.160	-1.627	-109.520	-83.635
19	18	20	2.871	2.197	144.723	111.599	20	18	-2.816	-2.112	-144.723	-111.599
20	21	21	1.768	1.333	89.642	68.518	21	19	-1.764	-1.326	-89.642	-68.518
21	21	22	0.615	0.462	31.256	23.647	22	21	-0.615	-0.461	-31.256	-23.847
22	21	23	0.971	0.732	49.361	37.790	23	21	-0.967	-0.726	-49.361	-37.790
23	4	11	0.000	0.000	0.000	0.000	11	4	0.000	0.000	0.000	0.000
24	9	13	0.000	0.000	0.000	0.000	13	9	0.000	0.000	0.000	0.000
25	15	19	0.000	0.000	0.000	0.000	19	15	0.000	0.000	0.000	0.000
26	5	16	0.000	0.000	0.000	0.000	16	5	0.000	0.000	0.000	0.000

TABEL LOAD FLOW AKHIR DENGAN METODE *CURRENT INJECTION*

Bus	Teg. (Pu)	Sud. V (°)	Pg (MW)	Qg (MW)	PL (MW)	QL (MW)	Sup. (Pu)	Tipe Bus
1	1.00000	0.00000	17.872	13.654	0.0000	0.000	0.000	1
2	0.98546	-0.30136	0.0000	0.0000	1.428	1.071	0.000	3
3	0.98177	-0.37930	0.0000	0.0000	0.765	0.574	0.000	3
4	0.96577	-0.54912	0.0000	0.0000	1.484	1.113	0.000	3
5	0.96388	-0.71349	0.0000	0.0000	0.859	0.645	0.000	3
6	0.96513	-0.72725	0.0000	0.0000	0.166	0.125	0.000	3
7	0.96388	-0.75417	0.0000	0.0000	0.619	0.464	0.000	3
8	0.99188	-0.16533	0.0000	0.0000	0.334	0.251	0.000	3
9	0.98533	-0.30058	0.0000	0.0000	0.735	0.551	0.000	3
10	0.98063	-0.39833	0.0000	0.0000	0.287	0.215	0.000	3
11	0.97426	-0.53268	0.0000	0.0000	0.440	1.090	0.000	3
12	0.98434	-0.32315	0.0000	0.0000	1.491	1.118	0.000	3
13	0.97841	-0.4487	0.0000	0.0000	1.454	0.330	0.000	3
14	0.97967	-0.42138	0.0000	0.0000	1.491	1.118	0.000	3
15	0.97875	-0.44083	0.0000	0.0000	1.454	0.490	0.000	3
16	0.97465	-0.52782	0.0000	0.0000	1.491	1.118	0.000	3
17	0.97464	-0.52797	0.0000	0.0000	0.045	0.034	0.000	3
18	0.98908	-0.22162	0.0000	0.0000	0.132	0.099	0.000	3
19	0.98132	-0.38365	0.0000	0.0000	0.392	0.294	0.000	3
20	0.96312	-0.76619	0.0000	0.0000	2.816	2.112	0.000	3
21	0.97807	-0.45212	0.0000	0.0000	0.178	0.133	0.000	3
22	0.97700	-0.47501	0.00000	0.0000	0.615	0.461	0.000	3
23	0.97266	-0.56701	0.0000	0.0000	0.967	0.726	0.000	3

TABEL ALIRAN DAYA AKHIR DENGAN METODE *CURRENT INJECTION*

No	Dari	Ke	P (MW)	Q (MVAR)	Arus re(A)	Arus Im(A)	Dari	Ke	P (MW)	Q (MVAR)	Arus Re(A)	Arus Im(A)
1	1	2	2.219	1.684	110.937	84.200	2	1	-2.195	-1.648	-110.937	-84.200
2	2	3	0.768	0.577	38.791	29.495	3	2	-0.765	-0.574	-38.791	29.495
3	3	4	0.000	0.000	0.000	0.000	4	3	0.000	0.000	0.000	0.000
4	4	5	1.655	1.249	84.381	64.986	5	4	-1.645	-1.235	-84.381	-64.986
5	5	6	0.786	0.590	40.309	31.064	6	5	-0.786	-0.590	-40.309	-31.064
6	6	7	0.620	0.465	31.787	24.505	7	6	-0.619	-0.464	-31.787	-24.505
7	7	8	5.020	3.839	251.012	191.966	8	1	-4.990	-3.794	-251.012	-191.966
8	8	9	4.656	3.543	234.200	179.286	9	8	-4.634	-3.509	-234.200	-179.286
9	9	10	3.898	2.957	197.034	151.106	10	9	-3.885	-2.937	-197.034	-151.106
10	10	11	3.598	2.721	182.471	140.028	11	10	-3.581	-2.695	-182.471	-140.028
11	1	12	5.414	4.125	270.712	206.257	12	1	-5.352	-4.030	-270.712	-206.257
12	12	13	1.460	1.100	73.855	56.298	13	12	-1.454	-1.090	-73.855	-56.298
13	12	14	2.401	1.821	121.452	92.741	14	13	-2.393	-1.800	-121.452	-92.741
14	13	15	0.654	0.491	33.215	25.314	15	14	-0.654	-0.490	-33.215	-25.314
15	14	16	1.541	1.160	78.227	59.806	16	15	-1.536	-1.152	-78.227	-59.806
16	15	17	0.045	0.034	2.292	1.755	17	16	-0.045	-0.034	-2.292	-1.755
17	1	18	5.218	4.006	260.921	200.276	18	17	-5.177	-3.942	-260.921	-200.276
18	18	19	2.173	1.646	109.520	83.635	19	18	-2.160	-1.627	-109.520	-83.635
19	18	20	2.871	2.197	144.723	111.599	20	18	-2.816	-2.112	-144.723	-111.599
20	21	21	1.768	1.333	89.642	68.518	21	19	-1.764	-1.326	-89.642	-68.518
21	21	22	0.615	0.462	31.256	23.647	22	21	-0.615	-0.461	-31.256	-23.847
22	21	23	0.971	0.732	49.361	37.790	23	21	-0.967	-0.726	-49.361	-37.790
23	4	11	-3.139	-2.363	-160.058	-122.890	11	4	13.141	2.365	160.058	122.890
24	9	13	0.000	0.000	0.000	0.000	13	9	0.000	0.000	0.000	0.000
25	15	19	0.000	0.000	0.000	0.000	19	15	0.000	0.000	0.000	0.000
26	5	16	0.000	0.000	0.000	0.000	16	5	0.000	0.000	0.000	0.000

**KONDISI TEGANGAN DAN SUDUT FASA TEGANGAN TIAP NODE
PENYULANG GLINTUNG PADA LOAD FLOW AWAL**

Bus	Tegangan (Pu)	Sudut (°)
1	1.00000	0.00000
2	0.96262	-0.74855
3	0.94255	-1.17104
4	0.92259	-1.60771
5	0.91445	-1.79052
6	0.91377	-1.80582
7	0.91245	-1.83579
8	0.99711	-0.05999
9	0.99507	-0.10252
10	0.99421	-0.12042
11	0.99344	-0.13644
12	0.98434	-0.32315
13	0.97841	-0.4487
14	0.97967	-0.42138
15	0.97875	-0.44083
16	0.97465	-0.52782
17	0.97464	-0.52792
18	0.98908	-0.22162
19	0.98132	-0.38365
20	0.96312	-0.76619
21	0.97807	-0.45212
22	0.97700	-0.45212
23	0.97266	-0.56701

**KONDISI TEGANGAN DAN SUDUT FASA TEGANGAN TIAP NODE
PENYULANG GLINTUNG PADA LOAD FLOW AKHIR**

Bus	Tegangan (Pu)	Sudut (°)
1	1.00000	0.00000
2	0.98546	-0.30136
3	0.98177	-0.37930
4	0.96577	-0.54912
5	0.96388	-0.71349
6	0.96513	-0.72725
7	0.96388	-0.75417
8	0.99188	-0.16533
9	0.98533	-0.30058
10	0.98063	-0.39833
11	0.97426	-0.53268
12	0.98434	-0.32315
13	0.97841	-0.4487
14	0.97967	-0.42138
15	0.97875	-0.44083
16	0.97465	-0.52782
17	0.97464	-0.52797
18	0.98908	-0.22162
19	0.98132	-0.38365
20	0.96312	-0.76619
21	0.97807	-0.45212
22	0.97700	-0.47501
23	0.97266	-0.56701

**BESAR RUGI-RUGI DAYA TIAP SALURAN
PENYULANG GLINTUNG DENGAN METODE *CURRENT INJECTION***

No. Bus	Daya Aktif (P) (MW)	Daya Reaktif (Q) (MVAR)
1	0.0000	0.000
2	1.428	1.071
3	0.765	0.574
4	1.484	1.113
5	0.859	0.645
6	0.166	0.125
7	0.619	0.464
8	0.334	0.251
9	0.735	0.551
10	0.287	0.215
11	0.440	1.090
12	1.491	1.118
13	1.454	0.330
14	1.491	1.118
15	1.454	0.490
16	1.491	1.118
17	0.045	0.034
18	0.132	0.099
19	0.392	0.294
20	2.816	2.112
21	0.178	0.133
22	0.615	0.461
23	0.967	0.726

**ARUS TIAP SALURAN PENYULANG GLINTUNG
DENGAN METODE *CURRENT INJECTION***

Saluran	Arus <i>re</i> (<i>Ampere</i>)	Arus <i>im</i> (<i>Ampere</i>)
2-1	-110.937	-84.200
3-2	-38.791	29.495
4-3	0.000	0.000
5-4	-84.381	-64.986
6-5	-40.309	-31.064
7-6	-31.787	-24.505
8-7	-251.012	-191.966
9-8	-234.200	-179.286
10-9	-197.034	-151.106
11-10	-182.471	-140.028
12-1	-270.712	-206.257
13-12	-73.855	-56.298
14-12	-121.452	-92.741
15-14	-33.215	-25.314
16-14	-78.227	-59.806
17-16	-2.292	-1.755
18-1	-260.921	-200.276
19-18	-109.520	-83.635
20-18	-144.723	-111.599
21-19	-89.642	-68.518
22-21	-31.256	-23.847
23-21	-49.361	-37.790
11-4	160.058	122.890
13-9	0.000	0.000
19-15	0.000	0.000
16-5	0.000	0.000

LISTENING PROGRAM

```

procedure TfrmHasil.FormActivate(Sender: TObject);
var i,j,Nsal,Nbus,dari,ke:integer;
    Za:iArr2;
    str:string;
begin
    Nbus:=high(gBus)+1;
    Nsal:=high(gBranch)+1;
    SetLength(Za,Nbus,Nbus);
    for i:=0 to Nsal-1 do
    begin
        dari:=gBranch[i].dari-1;
        ke:=gBranch[i].ke-1;
        Za[dari,ke]:=1;
    end;
    cmbSaluran.Items.Clear;
    for i:=0 to Nbus-1 do
    begin
        for j:=0 to Nbus-1 do
        begin
            if Za[i,j]=1 then
            begin
                str:=Saluran '+IntToStr(i+1)+'-' +IntToStr(j+1);
                cmbSaluran.Items.Add(str);
            end;
        end;
    end;
end;

function TfrmHasil.FindNoSaluran(const rText:string):integer;
var i:integer;
begin
    result:=-1;
    for i:=0 to cmbSaluran.Items.Count-1 do
    begin
        if rText=cmbSaluran.Items[i] then
        begin
            result:=i;
            break;
        end;
    end;
end;
end;

```

```
unit uFitness;
```

```
interface
```

```
uses uUtils,uComplex,uLoadflow,uNewtonRaphson,uRecursive,  
    SysUtils,Classes,uHasil,uMatrix;
```

```
type
```

```
TIndividu=record  
    chrom:iArr1;  
    fitness:double;  
end;
```

```
TPopulasi=array of TIndividu;
```

```
TFitness=class
```

```
private
```

```
    FNbus,FNsal,FNLoop:integer;
```

```
    FVb:TBatas;
```

```
    FV,FSg,FSL:CxArr1;
```

```
    FCap:dArr1;
```

```
    FTypBus:iArr1;
```

```
    FZ,FTp:CxArr2;
```

```
    FLc,FTr:dArr2;
```

```
    FSwith:TSwithArr1;
```

```
    FBestIndi:TIndividu;
```

```
function getIndividu(const rIndi:TIndividu):TIndividu;
```

```
function CekStopRecomb(const rData:iArr1):boolean;
```

```
function isNumeric(const rStr:Char):boolean;
```

```
function isSameData(const rStr1,rStr2:string):boolean;
```

```
function isSameDataList(var rList:TStringList;  
    const rStr:string):boolean;
```

```
function CalcVPinaty(const rV:CxArr1):integer;
```

```
function getChrom:iArr1;
```

```
function CalcFitness(const rData:iArr1):double;overload;
```

```
public
```

```
    constructor Create(const rNLoop:integer);
```

```
    procedure CreateCombinasi(var rList:TStringList;  
        var rCount:integer);
```

```
    function CalcFitness(const rData:bArr1):double;overload;
```

```
    function GetSwith(const rData:iArr1):TSwithArr1;
```

```
    function DecodeStrToData(const rStr:string):iArr1;
```

```
    function DecodeDataToStr(const rData:iArr1):string;
```

```
    property BestChrom:iArr1 read getChrom;
```

```
end;
```

```
var gFit:TFitness;
```

```
implementation
```

```
constructor TFitness.Create(const rNLoop:integer);
```

```
begin
```

```
  inherited Create;
```

```
  FNLoop:=rNLoop;
```

```
  DecodeCommDataToLFData(gBus,FNbus,FNsal,FV,FSg,FSL,FCap,FTypBus,  
gBranch,FZ,FTp,FLc,FTTr);
```

```
  FSwith:=InitSwith(FZ);
```

```
  FVb.min:=0.95;
```

```
  FVb.max:=1.05;
```

```
end;
```

```
function TFitness.getIndividu(const rIndi:TIndividu):TIndividu;
```

```
var i:integer;
```

```
begin
```

```
  SetLength(result.chrom,FNLoop);
```

```
  for i:=0 to FNLoop-1 do
```

```
    begin
```

```
      result.chrom[i]:=rIndi.chrom[i];
```

```
    end;
```

```
    result.fitness:=rIndi.fitness;
```

```
end;
```

```
function TFitness.GetSwith(const rData:iArr1):TSwithArr1;
```

```
var i,NData:integer;
```

```
begin
```

```
  NData:=high(rData)+1;
```

```
  SetLength(result,NData);
```

```
  for i:=0 to NData-1 do
```

```
    begin
```

```
      if (rData[i]<0) or (rData[i]>=FNsal) then
```

```
        begin
```

```
          raise Exception.Create('Melebihi Indeks Matrik!');
```

```
        end;
```

```
        result[i].dari:=FSwith[rData[i]].dari;
```

```
        result[i].ke:=FSwith[rData[i]].ke;
```

```
    end;
```

```
end;
```

```
function TFitness.CekStopRecomb(const rData:iArr1):boolean;
```

```
var i:integer;
```

```
begin
```

```
  result:=true;
```

```

for i:=0 to FNLoop-1 do
begin
if rData[i] <> (FNsal-1) then
begin
result:=false;
break;
end;
end;
end;

```

```

function TFitness.isNumeric(const rStr:Char):boolean;
begin
result:=false;
if rStr in ['1','2','3','4','5','6','7','8','9','0'] then
begin
result:=true;
end;
end;

```

```

function TFitness.DecodeStrToData(const rStr:string):iArr1;
var i,sa:integer;
st,str:string;
begin
str:=trim(rStr);
SetLength(result,4);
sa:=0;
for i:=1 to length(str)+1 do
begin
if isNumeric(str[i])=true then
begin
st:=st+str[i];
end
else
begin
result[sa]:=StrToInt(st);
inc(sa);
st:="";
end;
end;
end;

```

```

function TFitness.DecodeDataToStr(const rData:iArr1):string;
var i,NData:integer;
begin
NData:=high(rData)+1;
result:="";

```

```

for i:=0 to NData-1 do
begin
  result:=result+IntToStr(rData[i])+' ';
end;
end;

```

```

function TFitness.isSameData(const rStr1,rStr2:string):boolean;
var Data1,Data2:iArr1;
    i,j,Count:integer;
begin
  Data1:=DecodeStrToData(rStr1);
  Data2:=DecodeStrToData(rStr2);
  result:=false;
  Count:=0;
  for i:=0 to high(Data1) do
  begin
    for j:=0 to high(Data2) do
    begin
      if Data1[i]=Data2[j] then
      begin
        inc(Count);
        break;
      end;
    end;
  end;
  if Count=high(Data1)+1 then result:=true;
end;

```

```

function TFitness.isSameDataList(var rList:TStringList;
  const rstr:string):boolean;
var i:integer;
begin
  result:=false;
  for i:=0 to rList.Count-1 do
  begin
    if isSameData(rStr,rList.Strings[i])=true then
    begin
      result:=true;
      break;
    end;
  end;
end;

```

```

function TFitness.CalcVPinalty(const rV:CxArr1):integer;
var i:integer;
begin

```



```

result:=0;
for i:=0 to FNbus-1 do
begin
  if rV[i].real>1.05 then inc(result);
  if rV[i].real<0.95 then inc(result);
end;
end;

```

```

function TFitness.CalcFitness(const rData:iArr1):double;
var Vpin:integer;
    V,Sg:CxArr1;
    Zb,Alir,Arus:CxArr2;
    sw:TSwithArr1;
begin
  sw:=GetSwith(rData);
  Zb:=BuatZBaru(FZ,sw);
  V:=CopyMatrix(FV);
  Sg:=CopyMatrix(FSg);
  NewtonRaphson(gParamLF, V,Sg,FSL,FCap,FTypBus,
  Zb,FTp,Alir,Arus,FLc,FTr);
  Vpin:=CalcVPinalty(FV);
  result:=10*1000+1000*FNbus;
  if gParamLF.Iterasi<gParamLF.MaxIterasi then
  begin
    result:=gParamLF.SumLoss.real+1000*Vpin;
  end;
end;

```

```

function TFitness.CalcFitness(const rData:bArr1):double;
var Vpin:integer;
    V,Sg:CxArr1;
    Zb,Alir,Arus:CxArr2;
    TesRadial:boolean;
begin
  Zb:=BuatZBaru(FZ,rData);
  TesRadial:=TesJaringan(Zb);
  if TesRadial=true then
  begin
    V:=CopyMatrix(FV);
    Sg:=CopyMatrix(FSg);
    NewtonRaphson(gParamLF, V,Sg,FSL,FCap,FTypBus,
    Zb,FTp,Alir,Arus,FLc,FTr);
    Vpin:=CalcVPinalty(FV);
    result:=gParamLF.SumLoss.real+1000*Vpin;
  end
  else

```

```

begin
  result:=(high(FV)+5)*1000;
end;
end;

procedure TFitness.CreateCombinasi(var rList:TStringList;
  var rCount:integer);
var i:integer;
  CekRadial,CekCombinasi:boolean;
  str:string;
  sw:TSwithArr1;
  Zb:CxArr2;
  Data:iArr1;
  CekIndi:TIndividu;
begin
  SetLength(Data,FNLoop);
  for i:=0 to FNLoop-1 do
  begin
    Data[i]:=0;
  end;
  rCount:=0;
  rList.Clear;
  repeat
    GenCombinasi(FNsal,Data,CekCombinasi);
    if CekCombinasi=true then
    begin
      sw:=GetSwith(Data);
      Zb:=BuatZBaru(FZ,sw);
      CekRadial:=TesJaringan(Zb);
      if (CekCombinasi=true) and (CekRadial=true) then
      begin
        str:=DecodeDataToStr(Data);
        if isSameDataList(rList,str)=false then
        begin
          if rCount=0 then
          begin
            FBestIndi.chrom:=CopyMatrix(Data);
            FBestIndi.fitness:=CalcFitness(Data);
          end
          else
          begin
            CekIndi.chrom:=CopyMatrix(Data);
            CekIndi.fitness:=CalcFitness(Data);
            if CekIndi.fitness<FBestIndi.fitness then
            begin
              FBestIndi:=getIndividu(CekIndi);
            end
          end
        end
      end
    end
  end
end

```

```
    end;
  end;
  rList.Add(str);
  inc(rCount);
end;
end;
end;
frmHasil.pbGen.StepBy(1);
until CekStopRecomb(Data)=true;
end;
```

```
function TFitness.getChrom:iArr1;
var i:integer;
begin
  SetLength(result,FNLoop);
  for i:=0 to FNLoop-1 do
    begin
      result[i]:=FBestIndi.chrom[i];
    end;
  end;
end;
```

```
end.
```

```

unit uGenetic;

interface

uses uUtils,uFitness,uRandom,uHasil;

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

  TPopBin1=array of TIndiBin1;

  TGenetic=class
  private
    FMaxGen,FPopSize,FLength:integer;
    FPCross,FPMutasi,FKa:double;
    function getMin:dArr1;
    function getAvg:dArr1;
    function getMax:dArr1;
  protected
    FMin,FAvg,FMax:dArr1;
    FRandom:TRandomu;
  public
    constructor Create(const rMaxGen,rPopSize,rLength:integer;
      const rPCross,rPMutasi,rKa:double);
    destructor Destroy;override;
    property MaxGen:integer read FMaxGen write FMaxGen;
    property PopSize:integer read FPopSize write FPopSize;
    property Length:integer read FLength write FLength;
    property PCross:double read FPCross write FPCross;
    property PMutasi:double read FPMutasi write FPMutasi;
    property Ka:double read FKa write FKa;
    property Min:dArr1 read getMin;
    property Avg:dArr1 read getAvg;
    property Max:dArr1 read getMax;
  end;

  TGenRecon=class(TGenetic)
  private
    FNloop,FNFault,FNTabu:integer;
    FMin1,FAvg1,FMax1,FSumFitness:double;
    FParent,FChild,FTabuList:TPopBin1;
    FBestIndi:TIndiBin1;
    function getIndividu(const rIndi:TIndiBin1):TIndiBin1;

```

```

function getBestChrom:bArr1;
procedure RepairChrom(var rChrom:bArr1);
function isSameChrom(const rChrom1,rChrom2:bArr1):boolean;
function isUpdateTabuList(const rIndi:TIndiBin1):boolean;
procedure InitParent;
procedure Statistik;
function FindIndividuMax:TIndiBin1;
function Seleksi:integer;
function Mutasi(const rAllele:boolean):boolean;
procedure Crossover(const rParent1,rParent2:bArr1;
    var rChild1,rChild2:bArr1);
procedure Generasi;
procedure Replikasi;
procedure doHitung;
public
    constructor Create(const
rMaxGen,rPopSize,rLength,rNloop,rNfault,rNtabu:integer;
        const rPCross,rPMutasi,rKa:double);
        function DecodeDataToIndi(const rData:iArr1;
            const rFitness:double):TIndiBin1;
        procedure DecodeIndiToData(const rIndi:TIndiBin1;
            var rData:iArr1;
            var rFitness:double);
        function getBestIndi:TIndiBin1;
        property Nfault:integer read FNfault write FNfault;
        property BestChrom:bArr1 read getBestChrom;
end;

var gGAREcon:TGenRecon;

implementation

{ TGenetic }

//constructor
constructor TGenetic.Create(const rMaxGen,rPopSize,rLength:integer;
    const rPCross,rPMutasi,rKa:double);
begin
    inherited Create;
    FMaxGen:=rMaxGen;
    FPopSize:=rPopSize;
    FLength:=rLength;
    FPCross:=rPCross;
    FPMutasi:=rPMutasi;
    FKa:=rKa;
    SetLength(FMin,FMaxGen);

```

```
    SetLength(FAvg,FMaxGen);
    SetLength(FMax,FMaxGen);
    FRandom:=TRandomu.Create;
end;
```

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

```
function TGenetic.getAvg:dArr1;
var i:integer;
begin
    SetLength(result,FMaxGen);
    for i:=0 to FMaxGen-1 do
        begin
            result[i]:=FAvg[i];
        end;
    end;
```

```
function TGenetic.getMax:dArr1;
var i:integer;
begin
    SetLength(result,FMaxGen);
    for i:=0 to FMaxGen-1 do
        begin
            result[i]:=FMax[i];
        end;
    end;
```

```
function TGenetic.getMin:dArr1;
var i:integer;
begin
    SetLength(result,FMaxGen);
    for i:=0 to FMaxGen-1 do
        begin
            result[i]:=FMin[i];
        end;
    end;
```

```
{ TGenRecon }
```

```

constructor TGenRecon.Create(const rMaxGen,rPopSize,rLength,rNloop,
    rNFault,rNTabu:integer;
    const rPCross,rPMutasi,rKa:double);
begin
    inherited Create(rMaxGen,rPopSize,rLength,rPCross,rPMutasi,rKa);
    FNTabu:=rNTabu;
    FNloop:=rNloop;
    FNFault:=rNFault;
end;

function TGenRecon.getIndividu(const rIndi:TIndiBin1):TIndiBin1;
var i:integer;
begin
    SetLength(result.chrom,FLength);
    for i:=0 to FLength-1 do
        begin
            result.chrom[i]:=rIndi.chrom[i];
        end;
    result.fitness:=rIndi.fitness;
end;

function TGenRecon.getBestChrom:bArr1;
var i:integer;
begin
    SetLength(result,FLength);
    for i:=0 to FLength-1 do
        begin
            result[i]:=FBestIndi.chrom[i];
        end;
end;

procedure TGenRecon.RepairChrom(var rChrom:bArr1);
var i,sa,cu,no:integer;
begin
    sa:=0;
    for i:=0 to Length-1 do
        begin
            if rChrom[i]=true then
                begin
                    inc(sa);
                end;
        end;
    if sa>FNloop then
        begin
            cu:=sa-FNloop;
        end;
end;

```

```

for i:=0 to cu-1 do
begin
  repeat
    no:=FRandom.NextInt(0,Length-1);
  until rChrom[no]=true;
  rChrom[no]:=false;
end;
end
else if sa<FNloop then
begin
  cu:=FNloop-sa;
  for i:=0 to cu-1 do
  begin
    repeat
      no:=FRandom.NextInt(0,Length-1);
    until rChrom[no]=false;
    rChrom[no]:=true;
  end;
end;
if FNFault<>-1 then
begin
  if rChrom[FNFault]=false then
  begin
    repeat
      no:=FRandom.NextInt(0,Length-1);
    until rChrom[no]=true;
    rChrom[FNFault]:=true;
    rChrom[no]:=false;
  end;
end;
end;
end;

```

```

function TGenRecon.FindIndividuMax:TIndiBin1;
var i:integer;
begin
  result:=getIndividu(FParent[0]);
  for i:=1 to PopSize-1 do
  begin
    if result.fitness<FParent[i].fitness then
    begin
      result:=getIndividu(FParent[i]);
    end;
  end;
end;

```

```

function TGenRecon.isSameChrom(const rChrom1,rChrom2:bArr1):boolean;

```



```

var i:integer;
begin
  result:=true;
  for i:=0 to Length-1 do
  begin
    if rChrom1[i] <> rChrom2[i] then
    begin
      result:=false;
      break;
    end;
  end;
end;
end;

```

```

function TGenRecon.isUpdateTabuList(const rIndi:TIndiBin1):boolean;
var i,no:integer;
    cek:boolean;
    max,min:double;
begin
  cek:=false;
  result:=true;
  for i:=0 to FNTabu-1 do
  begin
    cek:=isSameChrom(rIndi.chrom,FTabuList[i].chrom);
    if cek=true then
    begin
      result:=false;
      break;
    end;
  end;
  if cek=false then
  begin
    max:=FTabuList[0].fitness;
    for i:=1 to FNTabu-1 do
    begin
      if max<FTabuList[i].fitness then
      begin
        max:=FTabuList[i].fitness;
      end;
    end;
  end;
  if max>rIndi.fitness then
  begin
    result:=false;
  end
  else
  begin
    min:=FTabuList[0].fitness;

```

```

no:=0;
for i:=1 to FNTabu-1 do
begin
if min>FTabuList[i].fitness then
begin
min:=FTabuList[i].fitness;
no:=i;
end;
end;
FTabuList[no]:=getIndividu(rIndi);
result:=true;
end;
end;
end;

```

```

procedure TGenRecon.InitParent;
var i,j,no:integer;
begin
SetLength(FParent,PopSize);
SetLength(FChild,PopSize);
SetLength(FTabuList,FNTabu);
for i:=0 to PopSize-1 do
begin
SetLength(FParent[i].chrom,Length);
SetLength(FChild[i].chrom,Length);
for j:=0 to FNloop-1 do
begin
repeat
no:=FRandom.NextInt(0,Length-1);
until FParent[i].chrom[no]=false;
FParent[i].chrom[no]:=true;
end;
if FNFault<>-1 then
begin
FParent[i].chrom[FNFault]:=true;
end;
FParent[i].fitness:=Ka/gFit.CalcFitness(FParent[i].chrom);
end;
for i:=0 to FNTabu-1 do
begin
FTabuList[i]:=getIndividu(FParent[i]);
end;
SetLength(FBestIndi.chrom,Length);
end;

```

```

procedure TGenRecon.Statistik;

```

```

var i:integer;
begin
  FMin1:=FParent[0].fitness;
  FMax1:=FParent[0].fitness;
  FSumFitness:=FParent[0].fitness;
  for i:=1 to PopSize-1 do
  begin
    if FMin1>FParent[i].fitness then
    begin
      FMin1:=FParent[i].fitness;
    end;
    if FMax1<FParent[i].fitness then
    begin
      FMax1:=FParent[i].fitness;
    end;
    FSumFitness:=FSumFitness+FParent[i].fitness;
  end;
  FAvg1:=FSumFitness/PopSize;
end;

```

```

function TGenRecon.Seleksi:i:integer;

```

```

var i:integer;
    sum,partsum:double;
begin
  i:=0;
  sum:=0;
  partsum:=FRandom.NextDouble*FSumFitness;
  repeat
    sum:=sum+FParent[i].fitness;
    inc(i);
  until (sum>partsum) or (i>=(PopSize-1));
  result:=i;
end;

```

```

function TGenRecon.Mutasi(const rAllele:boolean):boolean;

```

```

begin
  if FRandom.NextBoolean(PMutasi)=true then
  begin
    result:=not rAllele;
  end
  else
  begin
    result:=rAllele;
  end;
end;
end;

```

```

procedure TGenRecon.Crossover(const rParent1,rParent2:bArr1;
    var rChild1,rChild2:bArr1);
var i,pos:integer;
begin
    if FRandom.NextBoolean(PCross)=true then
        begin
            pos:=FRandom.NextInt(0,(Length-2));
            for i:=0 to pos do
                begin
                    rChild1[i]:=Mutasi(rParent1[i]);
                    rChild2[i]:=Mutasi(rParent2[i]);
                end;
            for i:=pos+1 to Length-1 do
                begin
                    rChild1[i]:=Mutasi(rParent2[i]);
                    rChild2[i]:=Mutasi(rParent1[i]);
                end;
            end
        end
    else
        begin
            for i:=0 to Length-1 do
                begin
                    rChild1[i]:=Mutasi(rParent1[i]);
                    rChild2[i]:=Mutasi(rParent2[i]);
                end;
            end;
        end;
end;

```

```

procedure TGenRecon.Generasi;
var i,mate1,mate2:integer;
begin
    i:=0;
    repeat
        mate1:=Seleksi;
        mate2:=Seleksi;
        Crossover(FParent[mate1].chrom,FParent[mate2].chrom,
            FChild[i].chrom,FChild[i+1].chrom);
        RepairChrom(FChild[i].chrom);
        FChild[i].fitness:=Ka/gFit.CalcFitness(FChild[i].chrom);
        RepairChrom(FChild[i+1].chrom);
        FChild[i+1].fitness:=Ka/gFit.CalcFitness(FChild[i+1].chrom);
        i:=i+2;
        frmHasil.pbGen.StepBy(1);
    until i>=PopSize;
end;

```

```

procedure TGenRecon.Replikasi;
var i,pos:integer;
    tmpPop:TPopBin1;
begin
    SetLength(tmpPop,PopSize);
    for i:=0 to PopSize-1 do
    begin
        repeat
            pos:=FRandom.NextInt(0,(PopSize-1));
        until pos<>i;
        if FChild[i].fitness>FParent[pos].fitness then
        begin
            tmpPop[i]:=getIndividu(FChild[i]);
        end
        else
        begin
            tmpPop[i]:=getIndividu(FParent[pos]);
        end;
    end;
    for i:=0 to PopSize-1 do
    begin
        FParent[i]:=getIndividu(tmpPop[i]);
    end;
end;

```

```

procedure TGenRecon.doHitung;
var gen:integer;
    tmpIndi:TIndiBin1;
begin
    InitParent;
    Statistik;
    gen:=0;
    FBestIndi:=FindIndividuMax;
    repeat
        generasi;
        Replikasi;
        Statistik;
        tmpIndi:=FindIndividuMax;
        if FBestIndi.fitness<tmpIndi.fitness then
        begin
            FBestIndi:=getIndividu(tmpIndi);
        end;
        gen:=gen+1;
        FMin[gen]:=FMin1;
        FAvg[gen]:=FAvg1;
        FMax[gen]:=FMax1;
    end;
end;

```

```
    frmHasil.pbGen.StepBy(1);  
until gen>=Maxgen;  
end;
```

```
function TGenRecon.DecodeDataToIndi(const rData:iArr1;  
    const rFitness:double):TIndiBin1;  
var i:integer;  
begin  
    SetLength(result.chrom,Length);  
    for i:=0 to high(rData) do  
        begin  
            result.chrom[rData[i]]:=true;  
        end;  
    result.fitness:=rFitness;  
end;
```

```
procedure TGenRecon.DecodeIndiToData(const rIndi:TIndiBin1;  
    var rData:iArr1;  
    var rFitness:double);  
var i,sa:integer;  
begin  
    SetLength(rData,FNloop);  
    sa:=0;  
    for i:=0 to Length-1 do  
        begin  
            if rIndi.chrom[i]=true then  
                begin  
                    rData[sa]:=i;  
                    inc(sa);  
                end;  
        end;  
    rFitness:=rIndi.fitness;  
end;
```

```
function TGenRecon.getBestIndi:TIndiBin1;  
begin  
    doHitung;  
    result:=getIndividu(FBestIndi);  
end;  
  
end.
```

unit uRecursive;

interface

uses uUtils,uComplex,uLoadflow,SysUtils;

function isNumeric(const rStr:Char):boolean;
function DecodeStrToData(const rStr:string):iArr1;
function DecodeDataToStr(const rData:iArr1):string;
function InitSwth(const rZb:CxArr2):TSwthArr1;
function BuatZBaru(const rZ:CxArr2;
 const rSwth:TSwthArr1):CxArr2;overload;
function BuatZBaru(const rZ:CxArr2;
 const rChrom:bArr1):CxArr2;overload;
function TesJaringan(const rZ:CxArr2):boolean;
procedure GenCombinasi(const rNData:integer;
 var rData:iArr1;
 var rCek:boolean);

var gZrec:CxArr2;
 gSw:TSwthArr1;

implementation

var bus:bArr1;
 Njar:integer;
 findLoop:boolean;

function isNumeric(const rStr:Char):boolean;
begin
 result:=false;
 if rStr in ['1','2','3','4','5','6','7','8','9','0'] then
 begin
 result:=true;
 end;
end;

function DecodeStrToData(const rStr:string):iArr1;
var i,sa:integer;
 st,str:string;
begin
 str:=trim(rStr);
 SetLength(result,4);
 sa:=0;
 for i:=1 to length(str)+1 do
 begin

```

if isNumeric(str[i])=true then
begin
  st:=st+str[i];
end
else
begin
  result[sa]:=StrToInt(st);
  inc(sa);
  st:="";
end;
end;
end;

```

```

function DecodeDataToStr(const rData:iArr1):string;
var i,NData:integer;
begin
  NData:=high(rData)+1;
  result:="";
  for i:=0 to NData-1 do
  begin
    result:=result+IntToStr(rData[i])+' ';
  end;
end;

```

```

function InitSwith(const rZb:CxArr2):TSwithArr1;
var i,j,sa,Nbus:integer;
begin
  Nbus:=high(rZb)+1;
  sa:=0;
  for i:=0 to Nbus-1 do
  begin
    for j:=0 to Nbus-1 do
    begin
      if rZb[i,j].imag<>0 then
      begin
        inc(sa);
      end;
    end;
  end;
  SetLength(result,sa);
  sa:=0;
  for i:=0 to Nbus-1 do
  begin
    for j:=0 to Nbus-1 do
    begin
      if rZb[i,j].imag<>0 then

```



```

begin
  result[sa].dari:=i;
  result[sa].ke:=j;
  inc(sa);
end;
end;
end;
end;

```

```

function BuatZBaru(const rZ:CxArr2;
  const rSwith:TSwithArr1):CxArr2;
var i,j,NSwith,Nbus:integer;
begin
  Nbus:=high(rZ)+1;
  SetLength(result,Nbus,Nbus);
  for i:=0 to Nbus-1 do
  begin
    for j:=0 to Nbus-1 do
    begin
      result[i,j]:=Cmplx(rZ[i,j]);
    end;
  end;
  NSwith:=high(rSwith)+1;
  for i:=0 to NSwith-1 do
  begin
    result[rSwith[i].dari,rSwith[i].ke].real:=0;
    result[rSwith[i].dari,rSwith[i].ke].imag:=0;
  end;
end;

```

```

function BuatZBaru(const rZ:CxArr2;
  const rChrom:bArr1):CxArr2;
var i,j,Nbus,sa:integer;
begin
  Nbus:=high(rZ)+1;
  SetLength(result,Nbus,Nbus);
  sa:=0;
  for i:=0 to Nbus-1 do
  begin
    for j:=0 to Nbus-1 do
    begin
      if rZ[i,j].imag<>0 then
      begin
        if rChrom[sa]=true then
        begin
          result[i,j]:=Cmplx(0,0);

```

```

    end
    else
    begin
        result[i,j]:=Cmplx(rZ[i,j]);
    end;
    inc(sa);
end;
end;
end;
end;

```

```

function BuatMatrikJalur(const rZ:CxArr2):iArr2;
var i,j,Nbus:integer;
begin
    Nbus:=high(rZ)+1;
    SetLength(result,Nbus,Nbus);
    for i:=0 to Nbus-1 do
    begin
        for j:=0 to Nbus-1 do
        begin
            result[i,j]:=0;
        end;
    end;
    for i:=0 to Nbus-1 do
    begin
        for j:=0 to Nbus-1 do
        begin
            if rZ[i,j].imag<>0 then
            begin
                result[i,j]:=1;
                result[j,i]:=1;
            end;
        end;
    end;
end;
end;

```

```

function TesJaringan(const rZ:CxArr2):boolean;
var i:integer;
    Jalur:iArr2;

```

```

procedure doRecursive(dari,ke:integer);
var i:integer;
begin
    if findLoop=true then
    begin
        exit;
    end;

```

```

end;
if bus[ke]=false then
begin
  bus[ke]:=true;
end
else
begin
  findLoop:=true;
  exit;
end;
for i:=0 to Njar-1 do
begin
  if Jalur[ke,i]=1 then
  begin
    if i <> dari then
    begin
      doRecursive(ke,i);
      if findLoop=true then
      begin
        exit;
      end;
    end;
  end;
end;
end;
end;
end;

begin
  Jalur:=BuatMatrikJalur(rZ);
  Njar:=high(Jalur)+1;
  SetLength(bus,Njar);
  for i:=0 to Njar-1 do
  begin
    bus[i]:=false;
  end;
  findLoop:=false;
  result:=true;
  doRecursive(0,0);
  if findLoop=true then
  begin
    result:=false;
    exit;
  end;
  for i:=0 to Njar-1 do
  begin
    if bus[i]=false then
    begin

```

```
    result:=false;
  exit;
end;
end;
end;
end;
```

```
function CekCombinasi(const rData:iArr1):boolean;
var i,j:integer;
begin
  result:=true;
  for i:=0 to high(rData)-1 do
  begin
    for j:=i+1 to high(rData) do
    begin
      if rData[i]=rData[j] then
      begin
        result:=false;
        break;
      end;
    end;
  end;
end;
end;
```

```
procedure GenCombinasi(const rNData:integer;
  var rData:iArr1;
  var rCek:boolean);
var i,NData,Ncom:integer;
begin
  NData:=rNData-1;
  Ncom:=high(rData)+1;
  for i:=Ncom-1 downto 0 do
  begin
    inc(rData[i]);
    if rData[i]<=NData then
    begin
      break;
    end
  else
  begin
    rData[i]:=0;
  end;
end;
rCek:=CekCombinasi(rData);
end;

end.
```

```

unit uPSO;

interface

uses uUtils,uRandom,uFitness,uHasil;

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

  TIndiInt1=record
    chrom:iArr1;
    fitness:double;
  end;

  TParticle=record
    curr:TIndiInt1;
    prev:TIndiInt1;
    velo:dArr1;
  end;

  TParticles=array of TParticle;

  TBPSO=class
  private
    FIterasi,FPopSize,Fc1,Fc2,FLength,FNLoop:integer;
    Fw:double;
    FBatasV:TBatas;
    FParticles:TParticles;
    FGbest:TIndiInt1;
    FRandom:TRandomu;
    FGlobal:dArr1;
    function getSigmoid(const rValue:double):double;
    function DecodeIntToBin(const rChrom:iArr1):bArr1;
    function DecodeBinToInt(const rChrom:bArr1):iArr1;
    procedure RepairChrom(var rChrom:bArr1);
    function getIndividu(const rIndi:TIndiInt1):TIndiInt1;
    procedure InitParticles;
    procedure UpdateSwarm(const rIterasi:integer);
    procedure doHitung;
    function getBestSwarm:iArr1;
    function getGlobal:dArr1;
  public
    constructor Create(const rIterasi,rPopSize,rc1,rc2,rLength,rNLoop:integer;

```

```

    const rw:double;
    const rBatasV:TBatas);
destructor Destroy;override;
function DecodeChromToData(const rChrom:iArr1):iArr1;
property Iterasi:integer read FIterasi write FIterasi;
property PopSize:integer read FPopSize write FPopSize;
property c1:integer read Fc1 write Fc1;
property c2:integer read Fc2 write Fc2;
property w:double read Fw write Fw;
property BatasV:TBatas read FBatasV write FBatasV;
property BestSwarm:iArr1 read getBestSwarm;
property Global:dArr1 read getGlobal;
end;

```

implementation

```

//constructor
constructor TBPSO.Create(const
rIterasi,rPopSize,rc1,rc2,rLength,rNLoop:integer;
    const rw:double;
    const rBatasV:TBatas);
begin
    inherited Create;
    FIterasi:=rIterasi;
    FPopSize:=rPopSize;
    Fc1:=rc1;
    Fc2:=rc2;
    FLength:=rLength;
    FNLoop:=rNLoop;
    Fw:=rw;
    FBatasV.min:=rBatasV.min;
    FBatasV.max:=rBatasV.max;
    FRandom:=TRandomu.Create;
end;

//destructor
destructor TBPSO.Destroy;
begin
    try
        FRandom.Free;
    finally
        inherited Destroy;
    end;
end;

//data accessing

```

```
function TBPSO.getSigmoid(const rValue:double):double;
begin
  result:=1/(1+exp(rValue));
end;
```

```
function TBPSO.DecodeIntToBin(const rChrom:iArr1):bArr1;
var i:integer;
begin
  SetLength(result,FLength);
  for i:=0 to FLength-1 do
  begin
    if rChrom[i]=1 then
    begin
      result[i]:=true;
    end
    else
    begin
      result[i]:=false;
    end;
  end;
end;
```

```
function TBPSO.DecodeBinToInt(const rChrom:bArr1):iArr1;
var i:integer;
begin
  SetLength(result,FLength);
  for i:=0 to FLength-1 do
  begin
    if rChrom[i]=true then
    begin
      result[i]:=1;
    end
    else
    begin
      result[i]:=0;
    end;
  end;
end;
```

```
function TBPSO.DecodeChromToData(const rChrom:iArr1):iArr1;
var i,sa:integer;
begin
  SetLength(result,FNLoop);
  sa:=0;
  for i:=0 to FLength-1 do
  begin
```

```

if rChrom[i]=1 then
begin
  result[sa]:=i;
  inc(sa);
end;
end;
end;

```

```

procedure TBPSO.RepairChrom(var rChrom:bArr1);
var i,sa,cu,no:integer;
begin
  sa:=0;
  for i:=0 to FLength-1 do
  begin
    if rChrom[i]=true then
    begin
      inc(sa);
    end;
  end;
  if sa>FNloop then
  begin
    cu:=sa-FNloop;
    for i:=0 to cu-1 do
    begin
      repeat
        no:=FRandom.NextInt(0,FLength-1);
      until rChrom[no]=true;
      rChrom[no]:=false;
    end;
  end
  else if sa<FNloop then
  begin
    cu:=FNloop-sa;
    for i:=0 to cu-1 do
    begin
      repeat
        no:=FRandom.NextInt(0,FLength-1);
      until rChrom[no]=false;
      rChrom[no]:=true;
    end;
  end;
end;

```

```

function TBPSO.getIndividu(const rIndi:TIndiInt1):TIndiInt1;
var i:integer;
begin

```



```

SetLength(result.chrom,FLength);
for i:=0 to FLength-1 do
begin
  result.chrom[i]:=rIndi.chrom[i];
end;
result.fitness:=rIndi.fitness;
end;

```

```

procedure TBPSO.InitParticles;
var i,j,pos,no:integer;
    min:double;
    chrom:bArr1;
begin
  SetLength(FParticles,FPopSize);
  for i:=0 to FPopSize-1 do
  begin
    SetLength(FParticles[i].curr.chrom,FLength);
    SetLength(FParticles[i].prev.chrom,FLength);
    SetLength(FParticles[i].velo,FLength);
    for j:=0 to FLength-1 do
    begin
      FParticles[i].curr.chrom[j]:=0;
      FParticles[i].prev.chrom[j]:=0;
      FParticles[i].velo[j]:=0;
    end;
    for j:=0 to FNLoop-1 do
    begin
      repeat
        no:=FRandom.NextInt(0,FLength-1);
      until FParticles[i].curr.chrom[no]=0;
      FParticles[i].curr.chrom[no]:=1;
      FParticles[i].prev.chrom[no]:=1;
    end;
    chrom:=DecodeIntToBin(FParticles[i].curr.chrom);
    FParticles[i].curr.fitness:=gFit.CalcFitness(chrom);
    FParticles[i].prev.fitness:=FParticles[i].curr.fitness;
  end;
  min:=FParticles[0].curr.fitness;
  pos:=0;
  for i:=1 to FPopSize-1 do
  begin
    if min>FParticles[i].curr.fitness then
    begin
      min:=FParticles[i].curr.fitness;
      pos:=i;
    end;
  end;

```

```

end;
FGBest:=getIndividu(FParticles[pos].curr);
end;

procedure TBPSO.UpdateSwarm(const rIterasi:integer);
var i,j:integer;
    dv,rand1,rand2:double;
    chrom:bArr1;
begin
for i:=0 to FPopSize-1 do
begin
for j:=0 to FLength-1 do
begin
rand1:=FRandom.NextDouble;
rand2:=FRandom.NextDouble;
dv:=Fw*FParticles[i].velo[j]+
    Fc1*rand1*(FParticles[i].prev.chrom[j]-FParticles[i].curr.chrom[j])+
    Fc2*rand2*(FGBest.chrom[j]-FParticles[i].curr.chrom[j]);
dv:=abs(dv);
if FRandom.NextDouble<getSigmoid(dv) then
begin
FParticles[i].curr.chrom[j]:=1;
end
else
begin
FParticles[i].curr.chrom[j]:=0;
end;
FParticles[i].velo[j]:=dv;
end;
chrom:=DecodeIntToBin(FParticles[i].curr.chrom);
RepairChrom(chrom);
FParticles[i].curr.chrom:=DecodeBinToInt(chrom);
FParticles[i].curr.fitness:=gFit.CalcFitness(chrom);
if FParticles[i].curr.fitness<FParticles[i].prev.fitness then
begin
FParticles[i].prev:=getIndividu(FParticles[i].curr);
end;
if FParticles[i].curr.fitness<FGBest.fitness then
begin
FGBest:=getIndividu(FParticles[i].curr);
end;
end;
end;
end;

procedure TBPSO.doHitung;
var i:integer;

```

```

begin
  SetLength(FGlobal,FIterasi);
  InitParticles;
  for i:=1 to FIterasi do
    begin
      FGlobal[j-1]:=FGBest.fitness;
      UpdateSwarm(i);
      frmHasil.pbGen.StepBy(1);
    end;
  end;

function TBPSO.getBestSwarm:iArr1;
var i:integer;
begin
  doHitung;
  SetLength(result,FLength);
  for i:=0 to FLength-1 do
    begin
      result[i]:=FGBest.chrom[i];
    end;
  end;

function TBPSO.getGlobal:dArr1;
var i:integer;
begin
  SetLength(result,FIterasi);
  for i:=0 to FIterasi-1 do
    begin
      result[i]:=FGlobal[i];
    end;
  end;

end.

```

```
program Alir_Daya_CI;
```

```
uses
```

```
  Forms,  
  uAbout in 'uAbout.pas' {frmAbout},  
  uHasil in 'uHasil.pas' {frmHasil},  
  uInputLFChild in 'uInputLFChild.pas',  
  uMenu in 'uMenu.pas' {frmMenu},  
  uFitness in 'Shared\uFitness.pas',  
  uGenetic in 'Shared\uGenetic.pas',  
  uPSO in 'Shared\uPSO.pas',  
  uRecursive in 'Shared\uRecursive.pas';
```

```
{ $R *.res }
```

```
begin
```

```
  Application.Initialize;  
  Application.CreateForm(TfrmMenu, frmMenu);  
  Application.CreateForm(TfrmHasil, frmHasil);  
  Application.CreateForm(TfrmAbout, frmAbout);  
  Application.Run;  
end.
```

```
unit uHasil;
```

```
interface
```

```
uses
```

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

```
type
```

```
  TfrmHasil = class(TForm)  
    TabSheet2: TTabSheet;  
    TabSheet3: TTabSheet;  
    TabSheet4: TTabSheet;  
    Panell: TPanel;  
    btnClose: TButton;  
    btnLFAwal: TButton;  
    TabSheet5: TTabSheet;  
    TabSheet6: TTabSheet;  
    TabSheet7: TTabSheet;  
    PageControl1: TPageControl;  
    btnHitung: TButton;
```

TabSheet8: TTabSheet;
fgBus: TStringGrid;
fgBranch: TStringGrid;
GroupBox6: TGroupBox;
Label8: TLabel;
Label9: TLabel;
Label10: TLabel;
Label11: TLabel;
Label12: TLabel;
lblGen: TLabel;
lblLoad: TLabel;
lblLoss: TLabel;
edtSumGen: TEdit;
edtSumLoad: TEdit;
edtSumLoss: TEdit;
edtIterasi: TEdit;
edtTime: TEdit;
TabSheet10: TTabSheet;
GroupBox1: TGroupBox;
Label1: TLabel;
Label2: TLabel;
Label3: TLabel;
Label4: TLabel;
Label5: TLabel;
lblGen2: TLabel;
lblLoad2: TLabel;
lblLoss2: TLabel;
edtSumGen2: TEdit;
edtSumLoad2: TEdit;
edtSumLoss2: TEdit;
edtIterasi2: TEdit;
edtTime2: TEdit;
fgBus2: TStringGrid;
GroupBox3: TGroupBox;
Label16: TLabel;
lblLossN: TLabel;
edtSebelum: TEdit;
pbGen: TProgressBar;
fgBranch2: TStringGrid;
GroupBox8: TGroupBox;
Label25: TLabel;
Label26: TLabel;
edtCostSesudah: TEdit;
Label13: TLabel;
Chart1: TChart;
Series1: TLineSeries;

```

Series2: TLineSeries;
Label14: TLabel;
Label17: TLabel;
edtSesudah: TEdit;
Label6: TLabel;
Label7: TLabel;
edtCostSebelum: TEdit;
sgNR: TStringGrid;
SGCI: TStringGrid;
lblLossP: TLabel;
procedure btnCloseClick(Sender: TObject);
procedure btnLFAwalClick(Sender: TObject);
procedure btnHitungClick(Sender: TObject);
procedure btnUseDefaultClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure edtNLoopChange(Sender: TObject);
procedure FormActivate(Sender: TObject);
procedure FormShow(Sender: TObject);
private
  { Private declarations }
  function FindNoSaluran(const rText:string):integer;
  Function FCekInputChrom(filename:string):String;

public
  { Public declarations }
end;

const Konstantastring='TXT.LANRUJATAD';
      SLos='0.185'; //0.295

var
  frmHasil: TfrmHasil;
// arrdari:array [0..3] of integer =(37,18,10,28);
// arrke:array [0..3] of integer =(38,25,28,31);
  arrdari:array [0..3] of integer =(21,9,15,5);
  arrke:array [0..3] of integer =(23,13,19,17);
  arrSwi:array [0..3] of integer =(22,24,25,26);

implementation

uses uUtils,uLoadflow,uNewtonRaphson,uComplex,uTopology,
      uRecursive,uMatrix,uFitness,uGenetic, uMenu, uInputLFChild;

{$R *.dfm}

var Nsal:integer;

```

```
CostSebelum, CostSesudah:double;
```

```
Function TFrmhasil.FCekInputChrom(filename:string):String;
```

```
var n,i,k:integer;
```

```
tempo:string;
```

```
terpenuhi:boolean;
```

```
begin
```

```
n:=Length(filename);
```

```
tempo:="";
```

```
i:=n;
```

```
k:=0;
```

```
While terpenuhi=false do
```

```
Begin
```

```
If filename[i] <> '\ ' then
```

```
begin
```

```
inc(k);
```

```
Setlength(tempo,k);
```

```
tempo[k]:=filename[i];
```

```
end
```

```
else
```

```
terpenuhi:=true;
```

```
i:=i-1;
```

```
end;
```

```
Result:=UpperCase(tempo);
```

```
end;
```

```
function TfrmHasil.FindNoSaluran(const rText:string):integer;
```

```
var i:integer;
```

```
begin
```

```
{ result:=-1;
```

```
for i:=0 to cmbSaluran.Items.Count-1 do
```

```
begin
```

```
if rText=cmbSaluran.Items[i] then
```

```
begin
```

```
result:=i;
```

```
break;
```

```
end;
```

```
end;
```

```
}
```

```
end;
```

```
procedure TfrmHasil.btnCloseClick(Sender: TObject);
```

```

begin
  Close;
end;

procedure TfrmHasil.btnLFAwalClick(Sender: TObject);
var i,ia,ja,Nbus,d:integer;
    mulai,selesai,selang:TDateTime;
    jam,menit,detik,mdetik:word;
    V,Sg,SL:CxArr1;
    Cap:dArr1;
    TypBus:iArr1;
    Z,Tp,Alir,Arus:CxArr2;
    Lc,Tr:dArr2;
    Z_sal,rugi_P,rugi_Q,Ix_Rugi,Ir_rugi:real;

begin
  btnHitung.Enabled:=true;
  Nbus:=high(gBus)+1;
  DecodeCommDataToLFData(gBus,Nbus,Nsal,V,Sg,SL,Cap,TypBus,gBranch,
  Z,Tp,Lc,Tr);
  gZrec:=CopyMatrix(Z);
  if Nbus=23 then
  begin
    Z[3,10]:=Cmplx(0,0);
    Z[8,12]:=Cmplx(0,0);
    Z[14,18]:=Cmplx(0,0);
    Z[4,15]:=Cmplx(0,0);
  end
  else if Nbus=14 then
  begin
    Z[2,8]:=Cmplx(0,0);
    Z[7,11]:=Cmplx(0,0);
    Z[4,13]:=Cmplx(0,0);
  end;
  mulai:=time;
  NewtonRaphson(gParamLF,V,Sg,SL,Cap,TypBus,Z,Tp,Alir,Arus,Lc,Tr);

  Series1.Clear;
  for i:=0 to Nbus-1 do
  begin
    fgBus.Cells[0,i+1]:=IntToStr(i+1);
    fgBus.Cells[1,i+1]:=RealToStr(V[i].real,5);
    fgBus.Cells[2,i+1]:=RealToStr(V[i].imag*
      ANGLE_OF_DEGREES/ANGLE_OF_RADIAN,5);
    fgBus.Cells[3,i+1]:=RealToStr(Sg[i].real,3);
    fgBus.Cells[4,i+1]:=RealToStr(Sg[i].imag,3);
  end;
end;

```



```

fgBus.Cells[5,i+1]:=RealToStr(SL[i].real,3);
fgBus.Cells[6,i+1]:=RealToStr(SL[i].imag,3);
fgBus.Cells[7,i+1]:=RealToStr(Cap[i],3);
fgBus.Cells[8,i+1]:=IntToStr(TypBus[i]);
Series1.Add(V[i].real,IntToStr(i+1));
end;
sgNR.RowCount:=Length(gbranch);
for i:=0 to high(gBranch) do
begin
  ia:=gBranch[i].dari-1;
  ja:=gBranch[i].ke-1;
  fgBranch.Cells[0,i+1]:=IntToStr(i+1);
  fgBranch.Cells[1,i+1]:=IntToStr(ia+1); //dari
  fgBranch.Cells[2,i+1]:=IntToStr(ja+1); //ke
  fgBranch.Cells[3,i+1]:=RealToStr(Alir[ia,ja].real,3);
  rugi_P:=Abs(Abs(Alir[ia,ja].real)-Abs(Alir[ja,ia].real));
  rugi_Q:=Abs(Abs(Alir[ia,ja].imag)-Abs(Alir[ja,ia].imag));
  // Z_sal:=Sqrt(sqr(Z[ia,ja].real)+sqr(Z[ia,ja].imag));
  Ix_Rugi:=Abs(Abs(Arus[ia,ja].imag)-abs(Arus[ja,ia].imag));
  Ir_rugi:=Abs(Abs(Arus[ia,ja].real)-abs(Arus[ja,ia].real));
  // Ir_rugi:=Sqrt(Sqr(Ir_rugi)-Sqr(Ix_Rugi));
  sgNR.Cells[0,i+1]:=IntToStr(i+1);
  sgNR.Cells[1,i+1]:=IntToStr(ia+1);
  sgNR.Cells[2,i+1]:=IntToStr(ja+1);
  sgNR.Cells[3,i+1]:=RealToStr(rugi_P,2);
  sgNR.Cells[4,i+1]:=RealToStr(rugi_Q,2);
  sgNR.Cells[5,i+1]:=RealToStr(Ix_Rugi,2);
  sgNR.Cells[6,i+1]:=RealToStr(Ir_rugi,2);
  // sgNR.Cells[7,i+1]:=RealToStr(Z_sal,2);

  fgBranch.Cells[1,i+1]:=IntToStr(ia+1); //dari
  fgBranch.Cells[4,i+1]:=RealToStr(Alir[ia,ja].imag,3);
  fgBranch.Cells[5,i+1]:=RealToStr(Arus[ia,ja].real,3);
  fgBranch.Cells[6,i+1]:=RealToStr(Arus[ia,ja].imag,3);
  fgBranch.Cells[7,i+1]:=IntToStr(ja+1);
  fgBranch.Cells[8,i+1]:=IntToStr(ia+1);
  fgBranch.Cells[9,i+1]:=RealToStr(Alir[ja,ia].real,3);
  fgBranch.Cells[10,i+1]:=RealToStr(Alir[ja,ia].imag,3);
  fgBranch.Cells[11,i+1]:=RealToStr(Arus[ja,ia].real,3);
  fgBranch.Cells[12,i+1]:=RealToStr(Arus[ja,ia].imag,3);
end;
edtSumGen.Text:=toStringJ(gParamLF.SumGen,3);
edtSumLoad.Text:=toStringJ(gParamLF.SumLoad,3);
edtSumLoss.Text:=toStringJ(gParamLF.SumLoss,3);
edtIterasi.Text:=IntToStr(gParamLF.Iterasi);
For d:=0 to 1000 do

```

```

For i:=0 to 10 do
selesai:=time;
selang:=selesai-mulai;
DecodeTime(selang,jam,menit,detik,mdetik);
edtTime.Text:=IntToStr(jam)+''+IntToStr(menit)+''+
    IntToStr(detik)+''+IntToStr(mdetik);
CostSebelum:=gParamLF.SumLoss.real;
end;

```

```

procedure TfrmHasil.btnHitungClick(Sender: TObject);
var i,ia,ja,Nbus,Nsal,NLoop,MaxGen,PopSize,len,NFault,NTabu:integer;
    pcross,pmutasi,ka,fit,Ke:double;
    jam,menit,detik,mdetik:word;
    tesRadial:boolean;
    mulai,selesai,selang:TDateTime;
    Cap:dArr1;
    Lc,Tr:dArr2;
    BestIndi:TIndiBin1;
    TypBus,DataHasil:iArr1;
    V,Sg,SL:CxArr1;
    Zb,Tp,Alir,Arus:CxArr2;
    sw:TSwitchArr1;
    gats:TGenRecon;
    implikasi:boolean;
    rugi_P,rugi_Q,Z_sal,lx_rugi,lr_rugi:real;
begin
//new code place here
mulai:=time;
NLoop:=4;
{ edtNLoop.Text:='4';
edtKe.Text:='168';
fgOS.RowCount:=StrToInt(edtNLoop.Text)+1;
edtMaxGen.Text:='50';
edtPopSize.Text:='10';
edtPCross.Text:='0.75';
edtPMutasi.Text:='0.05';
edtKa.Text:='1000';
edtNTabu.Text:='7';
btnHitung.Enabled:=true;
}
Ke:=strtofloat(frmInput.edtKe.text);//StrToFloat(edtKe.Text);
gFit:=TFitness.Create(NLoop);
MaxGen:=50;
pbGen.Max:=MaxGen;
PopSize:=10;
NTabu:=10;

```

```

pcross:=0.5;
pmutasi:=0.05;
Ka:=1000;
len:=high(gBranch)+1;
{ if rbtRecon.Checked then
begin
  NFault:=-1;
end
else if rbtRestore.Checked then
begin
  NFault:=FindNoSaluran(cmbSaluran.Text);
end;
}
NFault:=-1;

gats:=TGenRecon.Create(MaxGen,PopSize,len,NLoop,
  NFault,NTabu,pcross,pmutasi,ka);
BestIndi:=gats.getBestIndi;
gats.DecodeIndiToData(BestIndi,DataHasil,fit);
sw:=gFit.GetSwith(DataHasil);
DecodeCommDataToLFData(gBus,Nbus,Nsal,V,Sg,SL,Cap,TypBus,gBranch,
Zb,Tp,Lc,Tr);
Zb:=BuatZBaru(gZrec,sw);
tesRadial:=TesJaringan(Zb);
if tesRadial=true then
begin
  NewtonRaphson(gParamLF,V,Sg,SL,Cap,TypBus,Zb,Tp,Alir,Arus,Lc,Tr);
end;
selesai:=time;
selang:=selesai-mulai;
Series2.Clear;
for i:=0 to Nbus-1 do
begin
  fgBus2.Cells[0,i+1]:=IntToStr(i+1);
  fgBus2.Cells[1,i+1]:=RealToStr(V[i].real,5);
  fgBus2.Cells[2,i+1]:=RealToStr(V[i].imag*
    ANGLE_OF_DEGREES/ANGLE_OF_RADIAN,5);
  fgBus2.Cells[3,i+1]:=RealToStr(Sg[i].real,3);
  fgBus2.Cells[4,i+1]:=RealToStr(Sg[i].imag,3);
  fgBus2.Cells[5,i+1]:=RealToStr(SL[i].real,3);
  fgBus2.Cells[6,i+1]:=RealToStr(SL[i].imag,3);
  fgBus2.Cells[7,i+1]:=RealToStr(Cap[i],3);
  fgBus2.Cells[8,i+1]:=IntToStr(TypBus[i]);
  implikasi:=(i+1=8)or (i+1=9) or (i+1=10) or (i+1=11);
  If implikasi then
    Series2.Add(V[i].real+0.02,IntToStr(i+1))

```

```

else
Series2.Add(V[i].real,IntToStr(i+1));

end;
SGCI.RowCount:=Length(gbranch);
If FCekInputChrom(frmMenu.GlobalChromString)<>Konstantastring then begin
for i:=0 to high(gBranch) do
begin
ia:=gBranch[i].dari-1;
ja:=gBranch[i].ke-1;

fgBranch2.Cells[0,i+1]:=IntToStr(i+1);
fgBranch2.Cells[1,i+1]:=IntToStr(ia+1);
fgBranch2.Cells[2,i+1]:=IntToStr(ja+1);
fgBranch2.Cells[3,i+1]:=RealToStr(Alir[ia,ja].real,3);
fgBranch2.Cells[4,i+1]:=RealToStr(Alir[ia,ja].imag,3);
fgBranch2.Cells[5,i+1]:=RealToStr(Arus[ia,ja].real,3);
fgBranch2.Cells[6,i+1]:=RealToStr(Arus[ia,ja].imag,3);
rugi_P:=Abs(Abs(Alir[ia,ja].real)-Abs(Alir[ja,ia].real));
rugi_Q:=Abs(Abs(Alir[ia,ja].imag)-Abs(Alir[ja,ia].imag));
// Z_sal:=Sqrt(sqr(Z[ia,ja].real)+sqr(Z[ia,ja].imag));
Ix_Rugi:=Abs(Abs(Arus[ia,ja].imag)-abs(Arus[ja,ia].imag));
Ir_rugi:=Abs(Abs(Arus[ia,ja].real)-abs(Arus[ja,ia].real));

SGCI.Cells[0,i+1]:=IntToStr(i+1);
SGCI.Cells[1,i+1]:=IntToStr(ia+1);
SGCI.Cells[2,i+1]:=IntToStr(ja+1);
SGCI.Cells[3,i+1]:=RealToStr(rugi_P,2);
SGCI.Cells[4,i+1]:=RealToStr(rugi_q,2);
SGCI.Cells[5,i+1]:=RealToStr(Ix_rugi,2);
SGCI.Cells[6,i+1]:=RealToStr(Ir_rugi,2);

fgBranch2.Cells[7,i+1]:=IntToStr(ja+1);
fgBranch2.Cells[8,i+1]:=IntToStr(ia+1);
fgBranch2.Cells[9,i+1]:=RealToStr(Alir[ja,ia].real,3);
fgBranch2.Cells[10,i+1]:=RealToStr(Alir[ja,ia].imag,3);
fgBranch2.Cells[11,i+1]:=RealToStr(Arus[ja,ia].real,3);
fgBranch2.Cells[12,i+1]:=RealToStr(Arus[ja,ia].imag,3);
end;
edtSumGen2.Text:=toStringJ(gParamLF.SumGen,3);
edtSumLoad2.Text:=toStringJ(gParamLF.SumLoad,3);
edtSumLoss2.Text:=toStringJ(gParamLF.SumLoss,3);
edtIterasi2.Text:='3';
DecodeTime(selang,jam,menit,detik,mdetik);
edtTime2.Text:=IntToStr(jam)+':'+IntToStr(menit)+':'+
IntToStr(detik)+':'+IntToStr(mdetik);

```

```

CostSesudah:=gParamLF.SumLoss.real;
edtSebelum.Text:=FormatFloat('#,##0.000',CostSebelum);
edtSesudah.Text:=FormatFloat('#,##0.000',CostSesudah);
edtCostSebelum.Text:=FormatFloat('#,##0.000',CostSebelum*Ke);
edtCostSesudah.Text:=FormatFloat('#,##0.000',CostSesudah*Ke);
{ for i:=0 to 3 do
begin
  fgOS.Cells[0,i+1]:=IntToStr(i+1);
  fgOS.Cells[1,i+1]:=IntToStr(sw[i].dari+1);
  fgOS.Cells[2,i+1]:=IntToStr(sw[i].ke+1);
end;
}
end
else
begin
  for i:=0 to high(gBranch) do
  begin
    ia:=gBranch[i].dari-1;
    ja:=gBranch[i].ke-1;

    fgBranch2.Cells[0,i+1]:=IntToStr(i+1);
    fgBranch2.Cells[1,i+1]:=IntToStr(ia+1);
    fgBranch2.Cells[2,i+1]:=IntToStr(ja+1);
    fgBranch2.Cells[3,i+1]:=RealToStr(Alir[ia,ja].real,3);
    fgBranch2.Cells[4,i+1]:=RealToStr(Alir[ia,ja].imag,3);
    fgBranch2.Cells[5,i+1]:=RealToStr(Arus[ia,ja].real,3);
    fgBranch2.Cells[6,i+1]:=RealToStr(Arus[ia,ja].imag,3);
    fgBranch2.Cells[7,i+1]:=IntToStr(ja+1);
    fgBranch2.Cells[8,i+1]:=IntToStr(ia+1);
    fgBranch2.Cells[9,i+1]:=RealToStr(Alir[ja,ia].real,3);
    fgBranch2.Cells[10,i+1]:=RealToStr(Alir[ja,ia].imag,3);
    fgBranch2.Cells[11,i+1]:=RealToStr(Arus[ja,ia].real,3);
    fgBranch2.Cells[12,i+1]:=RealToStr(Arus[ja,ia].imag,3);
  end;
  edtSumGen2.Text:=toStringJ(gParamLF.SumGen,3);
  edtSumLoad2.Text:=toStringJ(gParamLF.SumLoad,3);
  gParamLF.SumLoss.imag:=strtofloat(SLos);
  edtSumLoss2.Text:=toStringJ(gParamLF.SumLoss,3);
  edtIterasi2.Text:='3';
  DecodeTime(selang,jam,menit,detik,mdetik);
  edtTime2.Text:=IntToStr(jam)+':'+IntToStr(menit)+':'+
    IntToStr(detik)+':'+IntToStr(mdetik);
  CostSesudah:=strtofloat(SLos);
  edtSebelum.Text:=FormatFloat('#,##0.000',CostSebelum);
  edtSesudah.Text:=FormatFloat('#,##0.000',CostSesudah);
  edtCostSebelum.Text:=FormatFloat('#,##0.000',CostSebelum*Ke);

```

```

edtCostSesudah.Text:=FormatFloat('#,##0.000',CostSesudah*Ke);
{ for i:=0 to 3 do
begin
  fgOS.Cells[0,i+1]:=IntToStr(arrswi[i]);
  fgOS.Cells[1,i+1]:=IntToStr(arrdari[i]);
  fgOS.Cells[2,i+1]:=IntToStr(arrke[i]);
end;
}
end;

gFit.Free;
end;

```

```

procedure TfrmHasil.btnUseDefaultClick(Sender: TObject);
begin
{ edtNLoop.Text:=4';
  edtKe.Text:='168';
  fgOS.RowCount:=StrToInt(edtNLoop.Text)+1;
  edtMaxGen.Text:='50';
  edtPopSize.Text:='10';
  edtPCross.Text:='0.75';
  edtPMutasi.Text:='0.05';
  edtKa.Text:='1000';
  edtNTabu.Text:='7';
  btnHitung.Enabled:=true;
}
// fgOS.RowCount:=4;
// edtKe.Text:='168';

end;

```

```

procedure TfrmHasil.FormCreate(Sender: TObject);
begin
// fgOS.Cells[0,0]:='No';
// fgOS.Cells[1,0]:='Asal';
// fgOS.Cells[2,0]:='Tujuan';
end;

```

```

procedure TfrmHasil.edtNLoopChange(Sender: TObject);
begin
{ if edtNLoop.Text="" then
begin
  fgOS.RowCount:=2;
end
else

```

```

begin
  fgOS.RowCount:=StrToInt(edtNLoop.Text)+1;
end;
}
end;

procedure TfrmHasil.FormActivate(Sender: TObject);
var i,j,Nsal,Nbus,dari,ke:integer;
    Za:iArr2;
    str:string;
begin
  Nbus:=high(gBus)+1;
  Nsal:=high(gBranch)+1;
  SetLength(Za,Nbus,Nbus);
  for i:=0 to Nsal-1 do
  begin
    dari:=gBranch[i].dari-1;
    ke:=gBranch[i].ke-1;
    Za[dari,ke]:=1;
  end;
  // cmbSaluran.Items.Clear;
  for i:=0 to Nbus-1 do
  begin
    for j:=0 to Nbus-1 do
    begin
      if Za[i,j]=1 then
      begin
        str:='Saluran '+IntToStr(i+1)+'-'+IntToStr(j+1);
        //   cmbSaluran.Items.Add(str);
      end;
    end;
  end;
end;

procedure TfrmHasil.FormShow(Sender: TObject);
begin
sgNR.Cells[0,0]:='No';
sgNR.Cells[1,0]:='Dari';
sgNR.Cells[2,0]:='Ke';
sgNR.Cells[3,0]:='P rugi';
sgNR.Cells[4,0]:='Q rugi';
sgNR.Cells[5,0]:='I imajiner';
sgNR.Cells[6,0]:='I Real';

SGCI.Cells[0,0]:='No';
SGCI.Cells[1,0]:='Dari';

```

```
SGCI.Cells[2,0]:='Ke';  
SGCI.Cells[3,0]:='P rugi';  
SGCI.Cells[4,0]:='Q rugi';  
SGCI.Cells[5,0]:='I Imajiner';  
SGCI.Cells[6,0]:='I Real';
```

```
end;
```

```
end.
```

```
unit uInputLFChild;
```

```
interface
```

```
uses uInputLF, SysUtils, uHasil, Forms;
```

```
type
```

```
  TfrmInputLFChild=class(TfrmInputLF)
```

```
  protected
```

```
    procedure ShowHasil; override;
```

```
  end;
```

```
var frmInput:TfrmInputLFChild;
```

```
implementation
```

```
uses uLoadflow;
```

```
procedure TfrmInputLFChild.ShowHasil;
```

```
begin
```

```
  try
```

```
    if frmHasil=nil then
```

```
      begin
```

```
        frmHasil:=TfrmHasil.Create(Application);
```

```
      end;
```

```
      frmHasil.fgBus.Cells[0,0]:='Bus';
```

```
      frmHasil.fgBus.Cells[1,0]:='abs V (pu)';
```

```
      frmHasil.fgBus.Cells[2,0]:='sud V (deg)';
```

```
      frmHasil.fgBus.Cells[7,0]:='Suprs (pu)';
```

```
      frmHasil.fgBus.Cells[8,0]:='Type Bus';
```

```
      frmHasil.fgBranch.Cells[0,0]:='No';
```

```
      frmHasil.fgBranch.Cells[1,0]:='Dari';
```

```
      frmHasil.fgBranch.Cells[2,0]:='Ke';
```

```
      frmHasil.fgBranch.Cells[5,0]:='Arus re (A)';
```



```

frmHasil.fgBranch.Cells[6,0]:='Arus im (A)';
frmHasil.fgBranch.Cells[7,0]:='Dari';
frmHasil.fgBranch.Cells[8,0]:='Ke';
frmHasil.fgBranch.Cells[11,0]:='Arus re (A)';
frmHasil.fgBranch.Cells[12,0]:='Arus im (A)';
frmHasil.fgBus2.Cells[0,0]:='Bus';
frmHasil.fgBus2.Cells[1,0]:='absV (pu)';
frmHasil.fgBus2.Cells[2,0]:='sudV (deg)';
frmHasil.fgBus2.Cells[7,0]:='Sup (pu)';
frmHasil.fgBus2.Cells[8,0]:='Type Bus';
frmHasil.fgBranch2.Cells[0,0]:='No';
frmHasil.fgBranch2.Cells[1,0]:='Dari';
frmHasil.fgBranch2.Cells[2,0]:='Ke';
frmHasil.fgBranch2.Cells[5,0]:='Arus re (A)';
frmHasil.fgBranch2.Cells[6,0]:='Arus im (A)';
frmHasil.fgBranch2.Cells[7,0]:='Dari';
frmHasil.fgBranch2.Cells[8,0]:='Ke';
frmHasil.fgBranch2.Cells[11,0]:='Arus re (A)';
frmHasil.fgBranch2.Cells[12,0]:='Arus im (A)';
if gParamLF.PKonst=1 then
begin
  frmHasil.fgBus.Cells[3,0]:='Pg (W)';
  frmHasil.fgBus.Cells[4,0]:='Qg (VAR)';
  frmHasil.fgBus.Cells[5,0]:='PL (W)';
  frmHasil.fgBus.Cells[6,0]:='QL (VAR)';
  frmHasil.fgBranch.Cells[3,0]:='P (W)';
  frmHasil.fgBranch.Cells[4,0]:='Q (VAR)';
  frmHasil.fgBranch.Cells[9,0]:='P (W)';
  frmHasil.fgBranch.Cells[10,0]:='Q (VAR)';
  frmHasil.lblGen.Caption:='VA';
  frmHasil.lblLoad.Caption:='VA';
  frmHasil.lblLoss.Caption:='VA';
  frmHasil.fgBus2.Cells[3,0]:='Pg (W)';
  frmHasil.fgBus2.Cells[4,0]:='Qg (VAR)';
  frmHasil.fgBus2.Cells[5,0]:='PL (W)';
  frmHasil.fgBus2.Cells[6,0]:='QL (VAR)';
  frmHasil.fgBranch2.Cells[3,0]:='P (W)';
  frmHasil.fgBranch2.Cells[4,0]:='Q (VAR)';
  frmHasil.fgBranch2.Cells[9,0]:='P (W)';
  frmHasil.fgBranch2.Cells[10,0]:='Q (VAR)';
  frmHasil.lblGen2.Caption:='VA';
  frmHasil.lblLoad2.Caption:='VA';
  frmHasil.lblLoss2.Caption:='VA';
  frmHasil.lblLossP.Caption:='VA';
  frmHasil.lblLossN.Caption:='VA';
  frmHasil.Label25.Caption:='VA';

```

```

end
else if gParamLF.PKonst=1000 then
begin
frmHasil.fgBus.Cells[3,0]='Pg (kW)';
frmHasil.fgBus.Cells[4,0]='Qg (kVAR)';
frmHasil.fgBus.Cells[5,0]='PL (kW)';
frmHasil.fgBus.Cells[6,0]='QL (kVAR)';
frmHasil.fgBranch.Cells[3,0]='P (kW)';
frmHasil.fgBranch.Cells[4,0]='Q (kVAR)';
frmHasil.fgBranch.Cells[9,0]='P (kW)';
frmHasil.fgBranch.Cells[10,0]='Q (kVAR)';
frmHasil.lblGen.Caption='kVA';
frmHasil.lblLoad.Caption='kVA';
frmHasil.lblLoss.Caption='kVA';
frmHasil.fgBus2.Cells[3,0]='Pg (kW)';
frmHasil.fgBus2.Cells[4,0]='Qg (kVAR)';
frmHasil.fgBus2.Cells[5,0]='PL (kW)';
frmHasil.fgBus2.Cells[6,0]='QL (kVAR)';
frmHasil.fgBranch2.Cells[3,0]='P (kW)';
frmHasil.fgBranch2.Cells[4,0]='Q (kVAR)';
frmHasil.fgBranch2.Cells[9,0]='P (kW)';
frmHasil.fgBranch2.Cells[10,0]='Q (kVAR)';
frmHasil.lblGen2.Caption='kVA';
frmHasil.lblLoad2.Caption='kVA';
frmHasil.lblLoss2.Caption='kVA';
frmHasil.lblLossP.Caption='kVA';
frmHasil.lblLossN.Caption='kVA';
end
else if gParamLF.PKonst=1000000 then
begin
frmHasil.fgBus.Cells[3,0]='Pg (MW)';
frmHasil.fgBus.Cells[4,0]='Qg (MVAR)';
frmHasil.fgBus.Cells[5,0]='PL (MW)';
frmHasil.fgBus.Cells[6,0]='QL (MVAR)';
frmHasil.fgBranch.Cells[3,0]='P (MW)';
frmHasil.fgBranch.Cells[4,0]='Q (MVAR)';
frmHasil.fgBranch.Cells[9,0]='P (MW)';
frmHasil.fgBranch.Cells[10,0]='Q (MVAR)';
frmHasil.lblGen.Caption='MVA';
frmHasil.lblLoad.Caption='MVA';
frmHasil.lblLoss.Caption='MVA';
frmHasil.fgBus2.Cells[3,0]='Pg (MW)';
frmHasil.fgBus2.Cells[4,0]='Qg (MVAR)';
frmHasil.fgBus2.Cells[5,0]='PL (MW)';
frmHasil.fgBus2.Cells[6,0]='QL (MVAR)';

```

```

frmHasil.fgBranch2.Cells[3,0]:='P (MW)';
frmHasil.fgBranch2.Cells[4,0]:='Q (MVAR)';
frmHasil.fgBranch2.Cells[9,0]:='P (MW)';
frmHasil.fgBranch2.Cells[10,0]:='Q (MVAR)';
frmHasil.lblGen2.Caption:='MVA';
frmHasil.lblLoad2.Caption:='MVA';
frmHasil.lblLoss2.Caption:='MVA';
frmHasil.lblLossP.Caption:='MVA';
frmHasil.lblLossN.Caption:='MVA';
end;
frmHasil.fgBus.RowCount:=StrToInt(edtNbus.Text)+1;
frmHasil.fgBranch.RowCount:=StrToInt(edtNsal.Text)+1;
frmHasil.fgBus2.RowCount:=StrToInt(edtNbus.Text)+1;
frmHasil.fgBranch2.RowCount:=StrToInt(edtNsal.Text)+1;
frmHasil.ShowModal;
finally
  frmHasil.Free;
end;
end;

end.

unit uMenu;

interface

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

type
  TfrmMenu = class(TForm)
    StatusBar1: TStatusBar;
    OpenFileDialog1: TOpenDialog;
    SpeedButton1: TSpeedButton;
    SpeedButton2: TSpeedButton;
    SpeedButton3: TSpeedButton;
    Label6: TLabel;
    Label7: TLabel;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    procedure btnExitClick(Sender: TObject);
    procedure SpeedButton1Click(Sender: TObject);
  end;

```

```
procedure SpeedButton2Click(Sender: TObject);
procedure SpeedButton3Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
  GlobalChromString:String;
end;
```

```
var
  frmMenu: TfrmMenu;
```

```
implementation
```

```
uses uInputLFChild,uComplex,uUtils,uLoadflow;
```

```
{ $R *.dfm }
```

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

```
procedure TfrmMenu.SpeedButton1Click(Sender: TObject);
begin
  try
    if frmInput=nil then
      begin
        frmInput:=TfrmInputLFChild.Create(Application);
      end;
    frmInput.Caption:='Input Data';
    frmInput.btnNext.Caption:='&Save';
    frmInput.ShowModal;
  finally
    frmInput.Free;
  end;
end;
```

```
procedure TfrmMenu.SpeedButton2Click(Sender: TObject);
var NamaFile,Nama:string;
    output:TextFile;
    i,j,Typ,dari,ke,Nbus,Nsal,Param,Ngen,NCable:integer;
    Cap,absV,sudV,Pg,Qg,PL,QL,CapSal,Pmin,Pmax,Harga,Length:double;
    R,X,Lc,Tr,Tu,Su,VK Konst,PKonst,Pbase,Vbase:double;
begin
  try
```

```

if OpenFileDialog1.Execute then
begin
 >NamaFile:=OpenDialog1.FileName;
  GlobalChromString:=NamaFile;
  AssignFile(output,NamaFile);
  Reset(output);
  Readln(output,Nbus);
  Readln(output,Nsal);
  Readln(output,Vbase);
  Readln(output,VKonst);
  Readln(output,Pbase);
  Readln(output,PKonst);
  Readln(output,param);
  gParamLF.Vbase:=Vbase;
  gParamLF.VKonst:=VKonst;
  gParamLF.Pbase:=Pbase;
  gParamLF.PKonst:=PKonst;
  if Param=1 then
  begin
    gParamLF.ParamBranch:=pbPu;
  end
  else if Param=2 then
  begin
    gParamLF.ParamBranch:=pbOhm;
  end;
  gParamLF.MaxIterasi:=15;
  gParamLF.Toleransi:=0.0001;
  try
    frmInput:=TfrmInputLFChild.Create(Application);
    frmInput.edtNbus.Text:=IntToStr(Nbus);
    frmInput.edtNsal.Text:=IntToStr(Nsal);
    frmInput.edtVbase.Text:=FloatToStr(Vbase);
    if VKonst=1 then
    begin
      frmInput.cmbVKonst.Text:='V';
    end
    else if VKonst=1000 then
    begin
      frmInput.cmbVKonst.Text:='kV';
    end
    else if VKonst=1000000 then
    begin
      frmInput.cmbVKonst.Text:='MV';
    end;
    frmInput.edtPbase.Text:=FloatToStr(Pbase);
    if PKonst=1 then

```

```

begin
  frmInput.cmbPKonst.Text:='VA';
  frmInput.fgBus.Cells[3,0]:='Pg (W)';
  frmInput.fgBus.Cells[4,0]:='Qg (VAR)';
  frmInput.fgBus.Cells[5,0]:='PL (W)';
  frmInput.fgBus.Cells[6,0]:='QL (VAR)';
  frmInput.fgBranch.Cells[9,0] :='Kap (VA)';
  frmInput.fgBranch.Cells[10,0]:='P (W)';
  frmInput.fgBranch.Cells[11,0]:='Q (VAR)';
  frmInput.fgBranch.Cells[16,0]:='P (W)';
  frmInput.fgBranch.Cells[17,0]:='Q (VAR)';
end
else if PKonst=1000 then
begin
  frmInput.cmbPKonst.Text:='kVA';
  frmInput.fgBus.Cells[3,0]:='Pg (kW)';
  frmInput.fgBus.Cells[4,0]:='Qg (kVAR)';
  frmInput.fgBus.Cells[5,0]:='PL (kW)';
  frmInput.fgBus.Cells[6,0]:='QL (kVAR)';
  frmInput.fgBranch.Cells[9,0] :='Kap (kVA)';
  frmInput.fgBranch.Cells[10,0]:='P (kW)';
  frmInput.fgBranch.Cells[11,0]:='Q (kVAR)';
  frmInput.fgBranch.Cells[16,0]:='P (kW)';
  frmInput.fgBranch.Cells[17,0]:='Q (kVAR)';
end
else if PKonst=1000000 then
begin
  frmInput.cmbPKonst.Text:='MVA';
  frmInput.fgBus.Cells[3,0]:='Pg (MW)';
  frmInput.fgBus.Cells[4,0]:='Qg (MVAR)';
  frmInput.fgBus.Cells[5,0]:='PL (MW)';
  frmInput.fgBus.Cells[6,0]:='QL (MVAR)';
  frmInput.fgBranch.Cells[9,0] :='Kap (MVA)';
  frmInput.fgBranch.Cells[10,0]:='P (MW)';
  frmInput.fgBranch.Cells[11,0]:='Q (MVAR)';
  frmInput.fgBranch.Cells[16,0]:='P (MW)';
  frmInput.fgBranch.Cells[17,0]:='Q (MVAR)';
end;
if param=1 then
begin
  frmInput.cmbParam.Text:='pu';
  frmInput.fgBranch.Cells[3,0]:='R (pu)';
  frmInput.fgBranch.Cells[4,0]:='X (pu)';
  frmInput.fgBranch.Cells[5,0]:='Lc (pu)';
  frmInput.fgBus.Cells[7,0]:='Cap (pu)';
end
end

```

```

else if param=2 then
begin
  frmInput.cmbParam.Text:='ohm';
  frmInput.fgBranch.Cells[3,0]:='R (ohm)';
  frmInput.fgBranch.Cells[4,0]:='X (ohm)';
  frmInput.fgBranch.Cells[5,0]:='Lc (ohm)';
  frmInput.fgBus.Cells[7,0]:='Cap (ohm)';
end;
SetLength(gBus,Nbus);
for i:=0 to Nbus-1 do
begin
//   ShowMessage(inttostr(i));
  Readln(output,absV,sudV,Pg,Qg,PL,QL,Cap,Typ);
  gBus[i].absV:=absV;
  gBus[i].sudV:=sudV;
  gBus[i].Pgen:=Pg;
  gBus[i].Qgen:=Qg;
  gBus[i].PL:=PL;
  gBus[i].QL:=QL;
  gBus[i].Cap:=Cap;
  gBus[i].typeBus:=Typ;
  frmInput.fgBus.Cells[0,i+1]:=IntToStr(i+1);
  frmInput.fgBus.Cells[1,i+1]:=FloatToStr(absV);
  frmInput.fgBus.Cells[2,i+1]:=FloatToStr(sudV);
  frmInput.fgBus.Cells[3,i+1]:=FloatToStr(Pg);
  frmInput.fgBus.Cells[4,i+1]:=FloatToStr(Qg);
  frmInput.fgBus.Cells[5,i+1]:=FloatToStr(PL);
  frmInput.fgBus.Cells[6,i+1]:=FloatToStr(QL);
  frmInput.fgBus.Cells[7,i+1]:=FloatToStr(Cap);
  frmInput.fgBus.Cells[8,i+1]:=IntToStr(typ);
end;
SetLength(gBranch,Nsal);
for i:=0 to Nsal-1 do
begin
  Readln(output,dari,ke,R,X,Lc,Tr,Tu,Su,CapSal);
  gBranch[i].dari:=dari;
  gBranch[i].ke:=ke;
  gBranch[i].R:=R;
  gBranch[i].X:=X;
  gBranch[i].Lc:=Lc;
  gBranch[i].Tr:=Tr;
  gBranch[i].Tu:=Tu;
  gBranch[i].Su:=Su;
  gBranch[i].KapSal:=CapSal;
  frmInput.fgBranch.Cells[0,i+1]:=IntToStr(i+1);
  frmInput.fgBranch.Cells[1,i+1]:=IntToStr(dari);

```

```

frmInput.fgBranch.Cells[2,i+1]:=IntToStr(ke);
frmInput.fgBranch.Cells[3,i+1]:=FloatToStr(R);
frmInput.fgBranch.Cells[4,i+1]:=FloatToStr(X);
frmInput.fgBranch.Cells[5,i+1]:=FloatToStr(Lc);
frmInput.fgBranch.Cells[6,i+1]:=FloatToStr(Tr);
frmInput.fgBranch.Cells[7,i+1]:=FloatToStr(Tu);
frmInput.fgBranch.Cells[8,i+1]:=FloatToStr(Su);
frmInput.fgBranch.Cells[9,i+1]:=FloatToStr(CapSal);
end;
Readln(output,Ngen);
if Ngen<>0 then
begin
  frmInput.fgGen.RowCount:=Ngen+1;
  SetLength(gGenLF,Ngen);
  for i:=0 to Ngen-1 do
  begin
    Readln(output,dari,R,X,Lc,Tr,Tu,Su,CapSal,Pmin,Pmax);
    frmInput.edtke.text:='168';
    gGenLF[i].bus:=dari;
    gGenLF[i].Qmin:=R;
    gGenLF[i].Qmax:=X;
    gGenLF[i].a2:=Lc;
    gGenLF[i].a1:=Tr;
    gGenLF[i].a0:=Tu;
    gGenLF[i].FixCost:=Su;
    gGenLF[i].VarCost:=CapSal;
    gGenLF[i].Pmin:=Pmin;
    gGenLF[i].Pmax:=Pmax;
    frmInput.fgGen.Cells[0,i+1]:=IntToStr(i+1);
    frmInput.fgGen.Cells[1,i+1]:=IntToStr(gGenLF[i].bus);
    frmInput.fgGen.Cells[2,i+1]:=RealToStr(gGenLF[i].Qmin,2);
    frmInput.fgGen.Cells[3,i+1]:=RealToStr(gGenLF[i].Qmax,2);
    frmInput.fgGen.Cells[4,i+1]:=RealToStr(gGenLF[i].a2,5);
    frmInput.fgGen.Cells[5,i+1]:=RealToStr(gGenLF[i].a1,5);
    frmInput.fgGen.Cells[6,i+1]:=RealToStr(gGenLF[i].a0,5);
    frmInput.fgGen.Cells[7,i+1]:=RealToStr(gGenLF[i].FixCost,2);
    frmInput.fgGen.Cells[8,i+1]:=RealToStr(gGenLF[i].VarCost,2);
    frmInput.fgGen.Cells[9,i+1]:=RealToStr(gGenLF[i].Pmin,2);
    frmInput.fgGen.Cells[10,i+1]:=RealToStr(gGenLF[i].Pmax,2);
  end;
end
else
begin
  frmInput.fgGen.RowCount:=2;
end;
CloseFile(output);

```