

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

**“REKONFIGURASI JARINGAN SISTEM DISTRIBUSI
UNTUK MENGURANGI RUGI-RUGI DAYA DENGAN
MENGUNAKAN METODE BRANCH EXCHANGES DI
GI. BLIMBING MALANG”**

SKRIPSI

Disusun Oleh :
AGUS JUNAEDI
NIM. 00.12.157

September 2006

LEMBAR PERSETUJUAN

**REKONFIGURASI JARINGAN SISTEM DISTRIBUSI UNTUK
MENGURANGI RUGI-RUGI DAYA DENGAN
MENGUNAKAN METODE BRANCH EXCHANGES DI G.I.
BLIMBING MALANG**

SKRIPSI

*Disusun dan Diajukan Untuk Memenuhi Persyaratan Guna Mencapai Gelar
Sarjana Teknik Elektro Strata Satu (S-1)*

Disusun Oleh :
AGUS JUNAEDI
NIM : 00.12.157



Mengetahui,
(Ir. F. Yudi Limpraptono, MT)
NIP.Y. 1039500274

Disetujui,
Dosen Pembimbing

(Ir. Djojo Priatmono, MT)
Nip.Y : 1018500107

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

LEMBAR PERSETUJUAN

**REKONFIGURASI JARINGAN SISTEM DISTRIBUSI UNTUK
MENGURANGI RUGI-RUGI DAYA DENGAN
MENGUNAKAN METODE BRANCH EXCHANGES DI G.I.
BLIMBING MALANG**

SKRIPSI

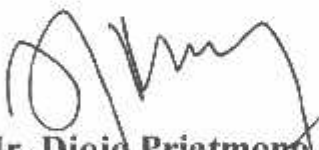
*Disusun dan Diajukan Untuk Memenuhi Persyaratan Guna Mencapai Gelar
Sarjana Teknik Elektro Strata Satu (S-1)*

Disusun Oleh :
AGUS JUNAEDI
NIM : 00.12.157

Mengetahui,
Ketua Jurusan Teknik Elektro S-1

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

Disetujui,
Dosen Pembimbing


(Ir. Djojo Priatmono, MT)
Nip.Y. 1018500107

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

KATA PENGANTAR

Dengan memanjatkan puji syukur atas kehadiran Allah SWT yang telah melimpahkan segala rahmat-Nya kepada penyusun sehingga penulis dapat menyelesaikan penulisan skripsi ini dengan baik.

Adapun tujuan penulisan skripsi ini adalah sebagai prasyarat untuk memperoleh gelar Sarjana Teknik di Jurusan Elektro Konsentrasi Energi Listrik Fakultas Teknologi Industri Institut Teknologi Nasional Malang.

Tak lupa penulis menyampaikan rasa terima kasih yang tak terhingga kepada :

1. Bapak Dr. Ir. Abraham Lomi, MSEE selaku Rektor ITN Malang.
2. Bapak Ir. Mochtar Asroni, MSME selaku Dekan FTI ITN Malang.
3. Bapak Ir. F. Yudi Limpraptono, MT selaku Ketua Jurusan Teknik Elektro S-1.
4. Bapak Ir. Djojo Priatmono, MT, selaku dosen pembimbing.
5. Semua pihak yang telah membantu sehingga penulisan skripsi ini bisa terselesaikan.

Dalam penyusunan skripsi ini penulis telah berusaha semaksimal mungkin namun penulis menyadari bahwa skripsi ini masih jauh dari sempurna mengingat keterbatasan kemampuan dari pengetahuan yang ada dalam diri penulis. Oleh karena itu segala bentuk saran, kritik maupun pendapat yang bersifat membangun akan sangat penulis harapkan

Akhirnya penulis berharap agar skripsi ini dapat berguna bagi mahasiswa/i jurusan teknik elektro pada khususnya dan mahasiswa jurusan lain pada umumnya.

Malang, September 2006

Penulis

ABSTRAKSI

“REKONFIGURASI JARINGAN SISTEM DISTRIBUSI UNTUK MENGURANGI RUGI-RUGI DAYA DENGAN MENGGUNAKAN METODE BRANCH EXCHANGES PADA DI GI. BLIMBING MALANG”

(AGUS JUNAEDI, NIM : 00.12.157, Teknik Elektro Energi Listrik, 57 Hal, 2006)
(Dosen Pembimbing : Ir. Djojo Priatmono, MT)

Kata Kunci : Perencanaan Sistem Distribusi, Branch Exchanges, Sistem Distribusi Radial.

Pada skripsi ini membahas bagaimana cara meminimalisasi rugi-rugi daya pada jaringan distribusi primer dengan rekonfigurasi jaringan menggunakan metode *Branch Exchanges*. Rekonfigurasi merupakan proses menata konfigurasi jaringan distribusi dengan memanfaatkan pengubahan status *switch normally open* (NO) dan *switch normally close* (NC).

Pada rekonfigurasi dengan metode *Branch Exchanges* merupakan pencarian rugi daya, yang dilakukan dengan beberapa *stage* dari konfigurasi awal (*initial condition*) jaringan menjadi suatu konfigurasi akhir (*optimum condition*) sehingga diperoleh kerugian daya sistem distribusi yang paling kecil.

Dengan melakukan rekonfigurasi jaringan menggunakan metode *Branch Exchanges*, yaitu mengubah status *switch* sebagai berikut :

bus 2 ke bus 3, bus 4 ke bus 5, bus 12 ke bus 13 dan bus 14 ke bus 15, maka didapatkan reduksi rugi-rugi daya sebesar 1.1 % dengan perincian sebagai berikut :

Rugi-rugi saluran sebelum rekonfigurasi : 0.193 MW dan 0,295 MVAR

Rugi-rugi saluran setelah rekonfigurasi : 0.186 MW dan 0.284 MVAR

Reduksi rugi-rugi saluran : 0,007 MW dan 0,011 MVAR

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PERSETUJUAN	ii
KATA PENGANTAR.....	iii
ABSTRAKSI.....	v
DAFTAR ISI.....	vi
DAFTAR GAMBAR.....	ix
DAFTAR TABEL.....	x
BAB I	PENDAHULUAN
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan Pembahasan.....	3
1.4. Batasan Masalah.....	3
1.5. Metode Pembahasan.....	4
1.6. Sistematika Penulisan.....	4
BAB II	SISTEM DISTRIBUSI TENAGA LISTRIK DAN TEORI
	DASAR METODE
2.1. Sistem Distribusi	6
2.1.1. Struktur Jaringan Radial.....	7
2.1.2. Struktur Jaringan Loop.....	8
2.1.3. Struktur Jaringan Mesh	8
2.2. Macam-macam Struktur Jaringan Distribusi Radial	9
2.3. Daya Sistem Distribusi.....	10
2.3.1. Daya Nyata (<i>Real Power</i>).....	10
2.3.2. Daya Reaktif (<i>Reactive Power</i>).....	11
2.3.3. Daya Semu (<i>Complex Power</i>)	11
2.4. Beban Sistem Distribusi	12
2.5. Rugi-rugi Jaringan.....	12
2.5.1. Mereduksi Rugi Daya Pada Sistem Distribusi	13
2.6. Konsep Dasar Rekonfigurasi.....	16

BAB III METODE REKONFIGURASI DENGAN BRANCH EXCHANGES

3.1.	Analisis Aliran Daya	20
3.1.1.	Tujuan.....	20
3.1.2.	Klasifikasi Bus	21
3.1.3	Pembentukan Matrik Admitansi Simpul	22
3.1.	Penyelesaian Aliran Daya	23
3.1.5.	Metode <i>Newton</i> Raphson	25
3.1.6.	Pernyataan Permasalahan	30
3.17	Metode Penyelesaian Studi Aliran Daya.....	32
3.18	Kalkulasi dari Terminologi Obyek.....	34
3.2	Metoda Backward and Forward Update.....	38
3.2.1.	Update Arus Daya	38
3.2.2.	Kalkulasi Pengurangan Kerugian Daya	40

BAB IV PENERAPAN METODE BRANCH EXCHANGES DALAM MEREKONFIGURASI JARINGAN DISTRIBUSI TENAGA LISTRIK

4.1.	Metode <i>Branch Exchanges</i>	41
4.1.1	Maksud dan tujuan	41
4.1.2.	<i>Branch Exchanges</i> Dalam Sistem Algoritma.....	41
4.2.	Bentuk Jaringan Sistem Distribusi	42
4.3.	Model Penyusunan Ulang Jaringan Distribusi	44
4.4.	Pengolahan Data.....	47
4.5.	Analisa Perhitungan Program.....	50
4.6.	Hasil Perhitungan	53
4.7.	Rugi-rugi daya.....	54

BAB V PENUTUP

5.1.	Kesimpulan.....	67
5.2.	Saran.....	68

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar

2-1	Diagram Satu Garis Sistem Tenaga Listrik.....	6
2-2	Struktur Jaringan Radial.....	7
2-3	Struktur Jaringan Loop.....	8
2-4	Struktur Jaringan Mesh	9
2-5	Rugi-rugi Pada Bagian Sistem Tenaga.....	13
2-6	Contoh Kondisi Gangguan Pada Sistem Distribusi Radial	15
3-1	Diagram satu garis dari suatu sistem distribusi kecil	31
3-2	Suatu diagram garis dari suatu jaringan radial	33
3-3	Pengulangan/Jerat dihubungkan dengan cabang terbuka b.....	37
4-1	Konfigurasi Jaringan Radial sebelum Rekonfigurasi.....	43
4-2	Flowchart Program.....	46
4-3	Tampilan Program Memasukkan Data Saluran	50
4-4	Tampilan Program Perhitungan Aliran Daya Sebelum Rekonfigurasi	51
4-5	Tampilan Hasil Load Flow Sebelum Rekonfigurasi	51
4-6	Tampilan hasil Program Besarnya Aliran Daya Setelah Rekonfigurasi	52
4-7	Tampilan Hasil Load Flow Setelah Rekonfigurasi	52
4-8	Tampilan Program Hasil Rekonfigurasi.....	52
4-9	Grafik Perbandingan Tegangan.....	53
4-10	Konfigurasi Jaringan Radial setelah Rekonfigurasi	53

DAFTAR TABEL

TABEL

4-1	Hasil Perhitungan Daya dan Impedansi Saluran Jaringan Distribusi GI Blimbing Malang	48
4-2	Hasil Perhitungan Aliran Daya Sebelum Rekonfigurasi.....	49
4-3	Hasil Perhitungan Rugi-rugi Daya Sebelum Rekonfigurasi.....	49
4-4	Hasil Perhitungan Rugi-rugi Daya Sebelum Rekonfigurasi.....	54
4-5	PERBANDINGAN RUGI-RUGI SALURAN	55
4-6	Perhitungan Beban Pada Saat Terjadi Pengalihan beban Sebelum Rekonfigurasi	56
4-7	Perhitungan Beban Pada Saat Terjadi Pengalihan beban Setelah Rekonfigurasi.	56
4-8	PERBANDINGAN RUGI-RUGI SALURAN	56

BAB I

PENDAHULUAN

1.1. Latar Belakang

Listrik merupakan bentuk energi yang paling cocok dan nyaman bagi manusia modern. Tanpa listrik infra struktur masyarakat sekarang tidak akan menyenangkan. Makin bertambahnya konsumsi listrik perkapita diseluruh dunia menunjuk kan kenaikan standar kehidupan manusia. Pemanfaatan secara optimum bentuk energi ini oleh masyarakat dapat dibantu dengan sistem distribusi yang efektif.

Dalam sistem distribusi yang efektif, tentunya harus benar-benar memperhatikan dan mengusahakan pelayanan yang baik kepada konsumen dengan cara penyaluran energi listrik yang lebih efektif, mulai dari sistem transmisi tenaga sampai dengan sistem distribusi.

Sistem distribusi perlu perhatian khusus sebab kesalahan-kesalahan sistem distribusi ini langsung berpengaruh langsung kepada konsumen. Sesungguhnya 90% dari terputusnya aliran kepada konsumen berasal dari sistem distribusi. Disamping itu sistem distribusi pada umumnya lebih rawan terutama lokasi beban yang bervariasi, sehingga banyak terjadi rugi-rugi daya dari sistem tersebut. Usaha untuk meminimalisasi rugi-rugi daya dari sistem distribusi dengan memformulasikan permasalahan sebagai persoalan rekonfigurasi jaringan distribusi.

Penggunaan metode *Branch Exchanges* adalah untuk penyelesaian yang efisien pada persoalan jaringan konfigurasi secara optimal. Metode ini direduksi dengan mengubah status sectionalizing tombol, penentuan pemecahan persamaan arus dan persamaan tegangan dari jaringan. Untuk menopang pencarian aliran daya dengan derajat ketelitian yang bermacam-macam, digunakan Metoda *Backward and Forward*. Penggunaan program komputer sangat efektif untuk mendukung metode ini..

1.2. Rumusan Masalah

Pada sistem distribusi yang besar dan kompleks, serta dengan semakin bertambahnya beban pada jaringan distribusi akan timbul masalah dalam menentukan penyulang maupun trafo mana yang akan dipakai untuk menyalurkan energi listrik secara radial dengan cepat agar sistem tetap terjaga dan meminimalkan rugi-rugi pada jaringan.

Terkait dengan hal tersebut, maka dalam pembahasan skripsi ini rumusan masalah yang akan dibahas adalah sebagai berikut :

1. Pengaturan beban pada trafo atau penyulang jaringan distribusi sehingga dapat meminimalkan rugi-rugi daya.
2. Menggunakan Metoda *Branch Exchanges*. Untuk menopang pencarian aliran daya dengan derajat ketelitian yang bermacam-macam, digunakan Metoda *Backward and Forward* sebagai figur untuk mendapatkan pengurangan kerugian yang lebih kecil.

Dari permasalahan yang timbul diatas, maka dalam skripsi ini dapat diambil judul:
“REKONFIGURASI JARINGAN SISTEM DISTRIBUSI UNTUK
MENGURANGI RUGI-RUGI DAYA DENGAN MENGGUNAKAN METODE
BRANCH EXCHANGES DI GI. BLIMBING MALANG”

1.3. Tujuan Pembahasan

Tujuan dari penulisan skripsi ini adalah untuk mensimulasikan *rekonfigurasi* penyulang distribusi dengan metode rekonfigurasi *Branch Exchanges* sehingga di dapat suatu kondisi rugi-rugi daya pada sistem distribusi seminimal mungkin.

1.4. Batasan Masalah

Batasan masalah dari skripsi ini adalah:

1. Beban diasumsikan sebagai beban tiga fasa seimbang.
2. Hanya membahas rugi-rugi daya pada saluran.
3. Tidak membahas penyebab terjadinya gangguan.
4. Tidak membahas cara pengontrolan *switching* yang dilakukan.
5. Metode rekonfigurasi yang digunakan adalah *Branch Exchanges*.
6. Pengambilan data dilakukan pada PT. PLN (Persero) Distribusi Jawa Timur Cabang Malang.

1.5. Metode Pembahasan

Metode yang digunakan dalam pembahasan skripsi ini adalah :

1. Metode literatur, yaitu dengan mempelajari hal-hal yang sesuai pembahasan melalui literatur yang sesuai.
2. Memperoleh data sebagai acuan dalam melengkapi parameter-parameter yang digunakan untuk menganalisis masalah.
3. Menganalisa data yang diperoleh dengan metode *Branch Exchanges*.

1.6. Sistematika Penulisan

Penulisan skripsi ini disusun dalam lima bab yang masing-masing membahas hal sebagai berikut:

- BAB I : Pada bab ini akan dibahas mengenai latar belakang, rumusan masalah, tujuan pembahasan, batasan masalah, metode pembahasan, dan sistematika penulisan skripsi ini.
- BAB II: Pada bab ini akan dibahas mengenai sistem jaringan distribusi radial dan loop, macam-macam daya, macam-macam beban, rugi-rugi jaringan dan Teori Dasar *Branch Exchanges*.
- BAB III: Bab ini membahas analisis aliran daya, rekonfigurasi jaringan dan metode rekonfigurasi jaringan *Branch Exchanges*.
- BAB IV: Bab ini berisi tentang penerapan metode *rekonfigurasi* jaringan *Branch Exchanges*.
- BAB V : Bab ini berisi kesimpulan dan saran-saran untuk pengembangan lebih lanjut.

BAB II

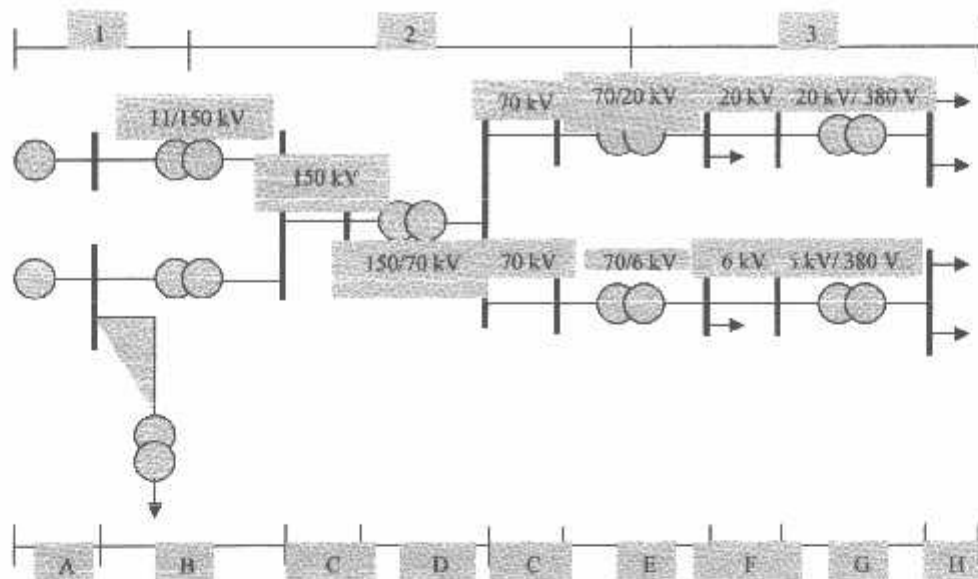
SISTEM DISTRIBUSI TENAGA LISTRIK

2.1. Dasar Teori

Sistem tenaga listrik merupakan suatu sistem terpadu yang terbentuk oleh hubungan-hubungan peralatan dan komponen-komponen listrik. Sistem tenaga listrik ini mempunyai peranan utama untuk menyalurkan energi listrik yang dibangkitkan oleh generator ke konsumen yang membutuhkan energi listrik tersebut.

Secara garis besar suatu sistem tenaga listrik dapat dikelompokkan menjadi 3 bagian sub sistem, yaitu :

1. Sistem Pembangkitan, berperan sebagai sumber daya tenaga listrik dan disebut juga sebagai Produtor Energi.
2. Sistem Transmisi atau Penyaluran, berfungsi sebagai penyalur daya listrik secara besar-besaran dari pembangkit ke bagian sistem distribusi atau konsumen.
3. Sistem Distribusi dan Beban, berperan sebagai distributor energi ke konsumen yang memerlukan energi tersebut.



Gambar 2.1. Diagram Satu Garis Sistem Tenaga Listrik

Keterangan :

- | | |
|------------------------|--------------------------------|
| 1. Sistem Pembangkitan | A. Generator |
| 2. Sistem Transmisi | B. Gardu Induk Pembangkitan |
| 3. Sistem Distribusi | C. Saluran Transmisi |
| | D. Gardu Induk |
| | E. Gardu Induk Distribusi |
| | F. Saluran Distribusi Primer |
| | G. Gardu Distribusi |
| | H. Saluran Distribusi Sekunder |

2.2. Sistem Distribusi

Sistem Distribusi merupakan bagian dari sistem tenaga listrik yang berperan dalam mendistribusikan tenaga listrik dari Gardu Induk (GI) sampai ke konsumen. Sistem distribusi pada hakekatnya dapat dibedakan menjadi 2 bagian yaitu:

1. Jaringan Distribusi Primer (Jaringan Tegangan Menengah)

Jaringan distribusi primer merupakan sistem tenaga listrik yang menyalurkan energi listrik antara Gardu Induk Distribusi sampai ke Gardu Distribusi dengan tegangan kerja 20 kV atau 6 kV.

2. Jaringan Distribusi Sekunder (Jaringan Tegangan Rendah)

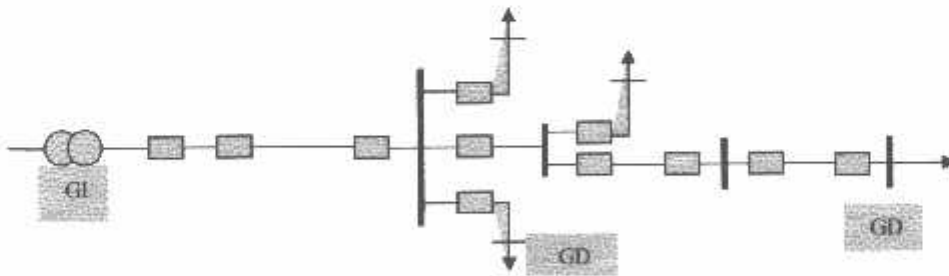
Jaringan distribusi sekunder merupakan jaringan tenaga listrik yang menyalurkan energi listrik antara Gardu Distribusi sampai ke konsumen dengan tegangan kerja 110 V atau 220 V.

Secara garis besar jaringan distribusi dapat diklasifikasikan menjadi:

- Struktur Jaringan Radial
- Struktur Jaringan Loop (Lingkaran)
- Struktur Jaringan Grid atau Mesh (Anyaman)

2.2.1. Struktur Jaringan Radial

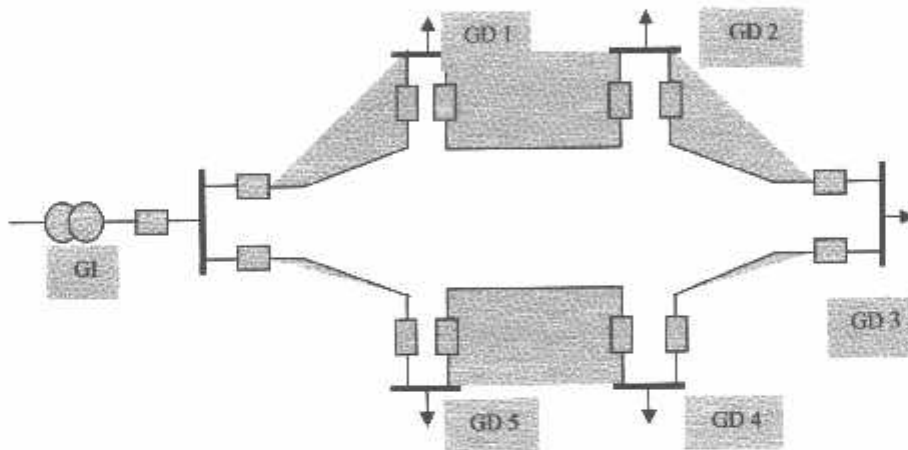
Struktur jaringan radial merupakan struktur jaringan yang paling sederhana dan paling murah biaya pembangunannya. Struktur jaringan ini dalam menyalurkan energi listrik, keandalannya kurang. Suatu gangguan pada penyulang dapat mengakibatkan gangguan dalam penyaluran energi listrik ke konsumen yang berada di belakang titik gangguan.



Gambar 2.2. Struktur Jaringan Radial


2.2.2. Struktur Jaringan Loop

Struktur jaringan loop merupakan gabungan dari dua struktur jaringan radial, dimana pada ujung kedua jaringan dipasang sebuah pemutus (PMT/CB) atau pemisah (PMS/DS). Pada saat terjadi gangguan, setelah gangguan dapat diisolir, maka pemutus atau pemisah ditutup sehingga aliran daya listrik ke bagian yang tidak terkena gangguan tidak berhenti. Dalam kondisi normal, struktur jaringan loop ini merupakan dua struktur jaringan radial. Struktur jaringan ini mempunyai keandalan yang cukup, sehingga biaya pembangunannya lebih mahal dibandingkan dengan biaya pembangunan struktur jaringan radial.



Gambar 2.3. Struktur Jaringan Loop

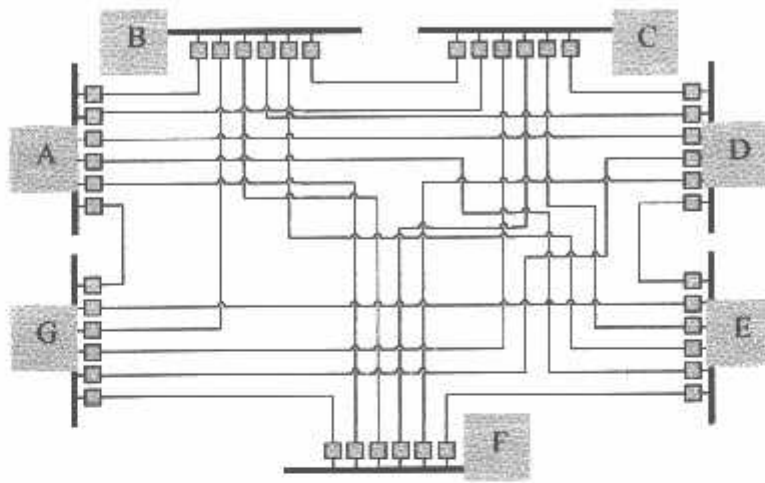
Keterangan :  = *Normally Open*

 = *Normally Close*

2.1.3. Struktur Jaringan Mesh

Struktur jaringan ini merupakan kombinasi antara struktur jaringan radial dengan struktur jaringan loop. Titik beban memiliki lebih banyak alternatif penyulang, sehingga bila salah satu penyulang terganggu maka dengan segera

dapat digantikan oleh penyulang yang lain. Dengan demikian kontinuitas penyaluran daya sangat terjamin. Gambar 2-4 menunjukkan bentuk dari struktur jaringan Mesh.



Gambar 2.4. Struktur Jaringan Mesh

2.3. Macam-Macam Struktur Jaringan Distribusi Radial

Tipe jaringan ini merupakan bentuk dasar, susunan maupun kebutuhan alat-alat penunjangnya paling sedikit dan paling sederhana. Tipe ini paling banyak digunakan untuk melayani konsumen terutama beban-beban rumah tangga yang tidak menuntut tingkat kontinuitas pelayanan yang tinggi. Sumber daya pada tipe ini hanya dari satu titik. Salurannya dicabang-cabang menuju ke titik-titik beban, dan antara titik sumber dengan titik beban hanya ada satu pilihan. Dengan demikian bila salah satu saluran cabang mengalami gangguan maka seluruh beban yang ada di saluran tersebut akan mengalami pemadaman total.

Jaringan distribusi radial mempunyai beberapa keuntungan dan kelemahan sebagai berikut :

Keuntungan jaringan radial :

1. Bentuknya sederhana dibandingkan dengan bentuk yang lain.
2. Biaya investasinya relatif lebih murah, karena saluran menuju ke tiap beban hanya tersedia satu jalur.

Kelemahan jaringan radial :

1. Kualitas pelayanan (penyaluran daya) dibanding dengan tipe yang lain lebih jelek, sebab jatuh tegangan dan rugi daya relatif besar. Kerugian ini terjadi pada saluran.
2. Kontinuitas penyaluran daya tidak terjamin. Dengan hanya mengandalkan satu saluran, maka bila terjadi gangguan pada saluran tersebut tidak dapat diharapkan suplai melalui saluran lain.

Jaringan radial ini dalam perkembangannya mengalami beberapa bentuk modifikasi sehingga dikenal beberapa macam jaringan distribusi radial, yaitu :

1. Sistem radial Pohon
2. Sistem radial dengan *Tie* dan *Switch* Pemisah
3. Sistem radial dengan Pusat Beban
4. Sistem radial dengan pembagian Daerah Fasa (*Phase Area*).

2.4. Daya Sistem Distribusi

Daya merupakan banyaknya perubahan tenaga terhadap waktu dalam besaran tegangan dan arus.

2.4.1. Daya Nyata (*Real Power*)

Daya nyata dinyatakan dalam persamaan :

$$P = |V| |I| \cos \varphi \quad (2.1)$$

Daya nyata untuk beban 3 fasa seimbang :

$$P = \sqrt{3} |V_{\text{jala-jala}}| |I_{\text{jala-jala}}| \cos \varphi \quad (2.2)$$

2.4.2. Daya Reaktif (*Reactive Power*)

Daya reaktif dinyatakan dalam persamaan :

$$Q = |V| |I| \sin \varphi \quad (2.3)$$

Daya reaktif untuk beban 3 fasa seimbang :

$$Q = \sqrt{3} |V_{\text{jala-jala}}| |I_{\text{jala-jala}}| \sin \varphi \quad (2.4)$$

2.4.3. Daya Semu (*Apparent Power*)

Daya semu dinyatakan dalam persamaan :

$$S = |V| |I| \quad (2.5)$$

Daya semu untuk beban 3 fasa seimbang :

$$S = \sqrt{3} |V_{\text{jala-jala}}| |I_{\text{jala-jala}}| \quad (2.6)$$

Persamaan (2.2), (2.4), (2.6) berlaku tanpa memandang apakah bebannya dihubung secara segitiga (Δ) atau bintang (Y).

2.5. Beban Sistem Distribusi

Secara garis besar beban dapat diklasifikasikan menjadi tiga, yaitu :

1. Beban Perumahan (Rumah Tangga)

Beban perumahan pada umumnya berupa penerangan, kipas angin, alat-alat rumah tangga dan lain-lain

2. Beban Komersial

Beban komersial pada umumnya berupa penerangan toko, reklame dan lain-lain.

3. Beban Industri

Beban industri diklasifikasikan menjadi 2 yaitu skala besar dan kecil.

2.6. Rugi-rugi Jaringan

Dalam proses penyaluran energi listrik ke pelanggan terjadi rugi-rugi teknis (*losses*), yaitu rugi daya dan rugi energi, dimulai dari Pembangkit, Transmisi dan Distribusi. Rugi-rugi teknis adalah pada penghantar saluran, adanya tahanan dari penghantar yang dialiri arus, timbulah rugi-rugi teknis (I^2R) pada jaringan tersebut. Pada gambar 2.5, rugi teknis tersebut terdapat pada saluran dan transformator. Rugi teknis pada penyaluran merupakan penjumlahan dari I^2R atau rugi tahanan. Rugi teknis dari jaringan tenaga listrik tergantung dari macam pembebanan pada saluran tersebut (beban merata, terpusat).

Namun demikian, sistem radial tersebut dioperasikan dengan beberapa "improvisasi". Biasanya terdapat beberapa *tie switch* yang tersedia untuk saling interkoneksi bagi beberapa bus beban (*load bus*) baik dalam satu *feeder* maupun berlainan *feeder*.

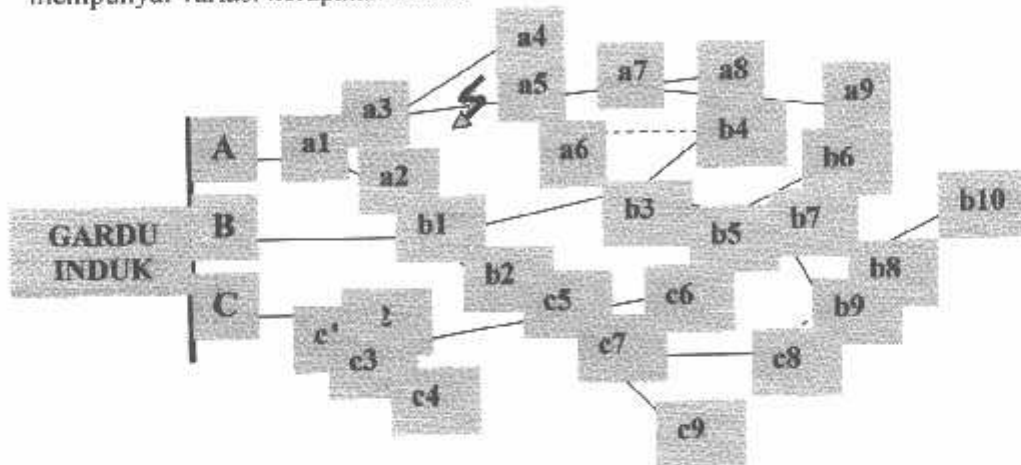
Kondisi improvisasi tersebut sangat diperlukan demi memenuhi persyaratan sebagai sistem yang cukup baik, terutama yaitu kontinuitas pelayanan. Hal ini dapat dijelaskan dengan melihat gambar 2.6.

Suatu sistem penyaluran distribusi radial dari gardu induk mempunyai tiga buah *feeder* (penyulang) yaitu *feeder* A, B dan C. *Feeder* A mempunyai bus-bus beban a1, a2, ..., a9. *Feeder* B mempunyai bus-bus beban b1, b2, ..., b10. *Feeder* C mempunyai bus-bus beban c1, c2, ..., c9. Jika pada *feeder* A terdapat gangguan pada saluran antara bus a3 dan a5 sehingga sampai menyebabkan putusnya saluran maka bus a5, a6, ..., a9 akan mengalami pemadaman total. Oleh karena itu bus-bus beban yang mengalami pemadaman total tersebut harus disupply dari *feeder* lain (bisa juga dari bus a4). Tanda (----) menunjukkan bahwa pada saluran itu terdapat *switch* (*Normally Open switch*). Dari gambar dapat dilihat bahwa NO *switch* terdapat antara bus a6 pada *feeder* A dengan bus b4 pada *feeder* B. Jika pada saat gangguan terjadi sehingga bus a3 dan a5 *open* maka NO *switch* a6-b4 dapat ditutup (*close*) sehingga dengan demikian bus-bus pada *feeder* A yang mengalami pemadaman total dapat diatasi. Hal yang sama juga dapat dilaksanakan antara *feeder* B dengan *feeder* C dengan NO *switch* b9-c8.

Pada kenyataan di lapangan, lokasi dari bus-bus itu sendiri mempunyai lokasi yang tidak sama sehingga akan menggunakan konduktor yang panjangnya

tidak sama. Kondisi ini tentunya akan membutuhkan resistansi dan reaktansi saluran yang berbeda-beda.

Bahkan bukan itu saja, hal sangat mungkin terjadi adalah bahwa besarnya beban-beban pada bus-bus tersebut tidak sama sehingga dapat dikatakan mempunyai variasi kerapatan beban.



Gambar 2.6. Contoh Kondisi Gangguan Pada Sistem Distribusi Radial

Kedua fenomena tersebut diatas itulah yang menyebabkan sulitnya sistem distribusi radial untuk memenuhi aspek teknis karena alasan lokasi beban dan variasi dari kerapatan beban itu sendiri. Pada akhirnya fenomena ini akan mengakibatkan rugi-rugi daya pada saluran distribusi menjadi tinggi.

Dari permasalahan tersebut diatas, maka diperlukan suatu upaya untuk dapat mengurangi kerugian daya yang terjadi. Dalam prakteknya, sebenarnya ada dua macam cara dalam upaya untuk mengurangi/meminimalisasi rugi-rugi daya pada sistem distribusi tenaga listrik yaitu :

1. Pemasangan/penginstalasian kapasitor.
2. Rekonfigurasi jaringan distribusi.

dipisahkan ke cabang yang lainnya dalam satu saluran. Semua switch akan dikembalikan pada waktu kondisi normal setelah gangguannya diperbaiki. Para ahli distribusi kadangkala menyusun saluran distribusi menggunakan switch pembuka dan penutup (operasi saklar / switching operation) agar meningkatkan ketahanan dan menghilangkan rugi-rugi saluran. Hasil dari jaringan harus tetap berbentuk radial dan dapat menampung semua beban. Koordinasi skema pengaman yang baru sama dengan yang lama.

Pada skripsi ini, rumusan umum dari masalah penyusunan ulang penyulang untuk mengurangi kerugian dan keseimbangan beban memberikan suatu solusi metoda baru untuk diperkenalkan. Solusi yang mempekerjakan sebuah pencarian atas perbedaan bentuk susunan radial yang diciptakan dengan mempertimbangkan pertukaran cabang jenis switching.

Untuk memandu pencarian, dua perbedaan metoda pendekatan aliran daya dengan derajat ketelitian yang berbeda-beda telah dikembangkan dan diuji. Metoda ini digunakan untuk mengkalkulasi aliran daya baru pada sistem dan menggunakan pengembangan persamaan aliran daya untuk sistem distribusi radial.

Ketelitian analisa dan hasil yang terbaik menunjukkan bahwa :

- Metoda penilaian computationally sangat efisien dalam memberi hasil konservatif secara umum. Juga dalam memberikan pertimbangan aliran daya reaktif dan nyata. Oleh karena itu, metoda tersebut dapat digunakan untuk mencari dan mengatur kembali sistem yang ditentukan jika sistem

tidak bagus penyusunan ulangnya dan melibatkan pemindahan beban antara cabang stasiun yang berbeda.

- Metoda pencarian yang diperkenalkan pada skripsi ini, mempunyai properti menarik sebagai berikut : hal tersebut tidak menyeluruh, itu pesanan m^2 (m adalah banyaknya dari tombol yang terbuka), dan itu melibatkan solusi tentang aliran daya m . Karakteristik pemusatannya bisa diterima meskipun tidak menjamin pemusatan jumlah optimum global. Bagaimanapun, modifikasi pada pencarian dasar diusulkan untuk meningkatkan computational dan karakteristik pemusatan ketika diperlukan.

Bila kedua metoda dibandingkan untuk keseimbangan beban, metoda Distflow yang disederhanakan nampak lebih menarik dengan pertimbangan sebagai berikut .

- Sejak index dari kesetimbangan beban relatif, ketelitian dari metoda Distflow yang disederhanakan adalah cukup.
- Metoda Distflow yang disederhanakan menyediakan suatu perkiraan kasar dan cepat menyangkut aliran daya tanpa menuntut data pada parameter jaringan.
- Pada metoda *backward and forward update* menjadi lebih nyata daripada metoda Distflow yang disederhanakan dalam menaksir pengurangan kerugian daya berkaitan dengan suatu pertukaran cabang, terutama sebagai figur pengurangan kerugian yang lebih kecil. Oleh karena itu, keputusan untuk memilih metoda harus dibuat dengan pertimbangan yang besar

untuk menggambarkan pengurangan kerugian. Suatu rencana yang compromi antara perhitungan dan ketelitian akan dimulai dengan metoda Distflow yang disederhanakan dan kemudian tombol untuk metoda backward and forward update sebagai figur dari pengurangan kerugian untuk mendapatkan yang lebih kecil.

- Rencana pencarian memberikan solusi yang bisa diterima untuk tujuan yang praktis sejak solusi terpusat pada suatu titik optimal lokal, perbedaan antara solusi lokal dan solusi global akan menjadi lebih kecil. Lagipula, karakteristik pemusatan pencarian dapat ditingkatkan dengan memeriksa tempat solusi. Suatu rencana yang mungkin adalah melakukan pencarian cepat yang lain dengan menerapkan lebih dari satu pertukaran cabang dengan pengurangan kerugian yang besar pada masing-masing tingkatan pencarian. Kemudian dua solusi dapat dibandingkan untuk melihat jika mereka memusatkan pada titik yang sama.

Untuk menjaga keseimbangan beban, suatu index kesetimbangan beban menunjukkan bahwa pencarian dan metoda penilaian aliran daya dikembangkan untuk mengurangi kerugian daya dan dapat juga digunakan untuk menjaga keseimbangan beban karena dua permasalahan yang serupa. Antara dua metoda penilaian, kedua metoda Distflow yang disederhanakan, sepertinya lebih sesuai menjaga keseimbangan beban karena dari relatif alam, beban menjaga keseimbangan konsep.

BAB III

METODE RECONFIGURASI DENGAN BRANCH EXCHANGES

3.1. Analisis Aliran Daya

Dalam analisa metode *rekonfigurasi* diperlukan suatu proses aliran daya untuk mengetahui tegangan pada bus beban dan rugi-rugi daya pada saluran. Oleh karena itu pembahasan selanjutnya sebelum membahas proses metode rekonfigurasi adalah analisis aliran daya menggunakan metode *backward and forward update*.

3.1.1. Tujuan

Dalam analisa aliran daya dilakukan perhitungan tegangan, arus, daya nyata dan daya reaktif yang terdapat pada berbagai titik dalam suatu jala-jala listrik pada keadaan pengoperasian normal untuk sekarang dan akan datang.

Tujuan dari aliran daya adalah:

- Untuk menentukan daya nyata dan daya reaktif.
- Untuk mengetahui apakah semua peralatan pada sistem memenuhi batas-batas yang telah ditetapkan untuk operasi penyaluran daya.
- Untuk mengetahui kondisi awal pada perencanaan sistem yang baru.
- Untuk menentukan daya yang mengalir di tiap saluran jaringan tenaga listrik.

Di dalam analisa aliran daya terdapat 3 jenis variabel :

1. Variabel bebas, misalnya $|V|$ dan δ pada bus beban atau δ dan Q pada bus generator.

2. Variabel tidak bebas, misalnya P dan $|V|$ pada bus generator.
3. Variabel yang tidak dapat diatur, misalnya kebutuhan konsumen.

3.1.2. Klasifikasi Bus

Tujuan aliran daya pada sistem tenaga listrik adalah untuk menghitung besar (*magnitude*) tegangan $|V|$ dan sudut fasa tegangan δ pada semua bus, sehingga dengan diketahuinya parameter-parameter tersebut akan dapat dihitung besar daya yang mengalir beserta rugi-ruginya.

Pada setiap bus dari jaringan terdapat parameter sebagai berikut :

1. Daya nyata, dinyatakan dengan P satuannya MW.
2. Daya reaktif, dinyatakan dengan Q satuannya MVAR.
3. Besar (*magnitude*) tegangan, dinyatakan dengan $|V|$ satuannya kV.
4. Sudut fasa tegangan, dinyatakan dengan δ satuannya derajat.

1 dan 2 menyatakan daya yang dibangkitkan oleh generator yang mengalir ke bus. Jika pada bus terdapat beban, maka daya tersebut menyatakan selisih antara daya yang dibangkitkan dengan daya beban.

Dalam aliran daya, pada setiap busnya perlu diketahui 2 parameter dari 4 parameter yang diperhitungkan. Dengan melihat kedua parameter tersebut dapat diketahui jenis busnya, yaitu:

1. Bus beban

Pada bus ini daya nyata P dan daya reaktif Q diketahui, sementara *magnitude* tegangan $|V|$ dan sudut fasa δ dicari (dihitung).

2. Bus generator (rel tegangan konstan)

Pada bus ini daya nyata P dan magnitudo tegangan $|V|$ diketahui, sementara daya reaktif Q dan sudut fasa δ dicari.

3. Bus berayun

Pada bus ini magnitudo tegangan $|V|$ dan sudut fasa δ diketahui besarnya, sementara daya nyata P dan daya reaktif Q dicari. Sedangkan sudut fasa tegangan δ berharga nol, karena pada bus ini fasor tegangan dari bus dipakai sebagai referensi.

3.1.3 Pembentukan Matrik Admitansi Simpul

Suatu rangkaian yang mempunyai n simpul (termasuk simpul referensi) dapat ditulis dalam bentuk persamaan arus :

$$I_1 = Y_{11}V_1 + Y_{12}V_2 + \dots + Y_{1n}V_n$$

$$I_2 = Y_{21}V_1 + Y_{22}V_2 + \dots + Y_{2n}V_n$$

$$I_n = Y_{n1}V_1 + Y_{n2}V_2 + \dots + Y_{nn}V_n$$

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

Dimana : $p = 1, 2, 3, \dots, n$.

I_p = arus yang masuk ke simpul p.

V_q = tegangan simpul q terhadap referensi.

Y_{pq} = admitansi antara simpul p dan q.

Persamaan diatas, dapat ditulis dalam bentuk matriks sebagai berikut :

$$I_{bus} = Y_{bus} \cdot V_{bus} \quad (3.2)$$

Dimana : I_{bus} = matriks kolom vektor arus simpul.

Y_{bus} = matriks kolom vektor admitansi simpul.

V_{bus} = matriks kolom vektor tegangan simpul.

Elemen dari Y_{bus} matriks adalah arus yang mengalir pada simpul dari jaringan dimana terdapat unit sumber tegangan atau simpul dan referensi.

Untuk elemen diagonal Y_{pp} , unit sumber tegangan dipasang pada simpul terhadap referensi dan diukur arus yang melalui elemen tersebut, dimana semua simpul yang lain dihubung singkat, sedangkan elemen bukan diagonal Y_{pq} diukur arus pada simpul p dimana unit sumber tegangan dipasang pada simpul q sedangkan simpul yang lain dihubung singkat, jika p dan q tidak dihubungkan secara langsung maka arus yang mengalir akan sama dengan nol, atau Y_{pq} sama dengan nol.

3.1.4. Penyelesaian Aliran Daya

Untuk penyelesaian masalah aliran daya diperlukan suatu persamaan daya aktif dan daya reaktif yang dinyatakan dalam besaran kompleks. Dimana untuk jaringan sistem tenaga listrik pada tiap-tiap bus dapat dituliskan sebagai berikut :

$$I_1 = Y_{11}V_1 + Y_{12}V_2 + \dots + Y_{1n}V_n$$

$$I_2 = Y_{21}V_1 + Y_{22}V_2 + \dots + Y_{2n}V_n$$

$$I_p = Y_{p1}V_1 + Y_{p2}V_2 + \dots + Y_{pn}V_n$$

Dimana persamaan tersebut dapat dituliskan menjadi bentuk umum :

$$I_p = \sum_{q=1}^n Y_{pq} V_{pq} \quad (3.3)$$

dimana : $p = 1, 2, 3, \dots, n$.

I_p = arus dalam bentuk kompleks yang masuk bus p .

V_q = tegangan bus q dalam bentuk kompleks terhadap referensi.

Y_{pq} = admitansi antara simpul p dan q.

Dalam bentuk matriks persamaan diatas dapat ditulis sebagai berikut :

$$\begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_n \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} & \dots & Y_{1n} \\ Y_{21} & Y_{22} & Y_{23} & \dots & Y_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ Y_{n1} & Y_{n2} & Y_{n3} & \dots & Y_{nn} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_n \end{bmatrix}$$

atau dapat juga ditulis

$$I_{bus} = Y_{bus} \cdot V_{bus} \tag{3.4}$$

dimana : I_{bus} = matriks kolom vektor arus bus.

Y_{bus} = matriks kolom vektor admitansi bus.

V_{bus} = matriks kolom vektor tegangan bus.

Persamaan daya nyata dan daya reaktif yang masuk ke bus adalah :

$$P_p - jQ_p = V_p^* I_p \tag{3.5}$$

Dengan mensubstitusikan persamaan (3.3) ke persamaan (3.5) maka diperoleh persamaan :

$$P_p - jQ_p = V_p^* \sum_{q=1}^n Y_{pq} V_{pq} \tag{3.6}$$

$$V_p^* = e_p - jf_p = |V_p| \angle -\delta_p$$

$$V_q = e_q - jf_q = |V_q| \angle \delta_q$$

$$Y_{pq} = G_{pq} - jB_{pq} = |Y_{pq}| \angle -\theta_{pq}$$

Dengan demikian persamaan ini dapat dinyatakan dalam bentuk *rectangular* sebagai berikut :

$$P_p - jQ_p = (e_p - jf_p) \sum_{q=1}^n (G_{pq} - jB_{pq}) (e_q + jf_q) \quad (3.7)$$

dimana : $p = 1, 2, 3, \dots, n$.

Persamaan-persamaan terakhir menyatakan dua buah persamaan aliran daya pada sistem dengan n bus, dimana tiap-tiap bus digolongkan dalam $4n$ variabel dan $2n$ variabel yang lain telah ditentukan terlebih dahulu.

Variabel dari bus berayun telah ditentukan terlebih dahulu sehingga persamaan untuk bus berayun ini dapat dihilangkan dan dengan demikian hanya dibutuhkan $(n-1)$ buah persamaan untuk studi aliran daya. Bentuk persamaan dua terakhir + 1 menyatakan hubungan matematis yang diperlukan untuk analisa aliran daya dengan komputer.

Selain itu persamaan diatas hanya berlaku untuk sistem satu fasa dan untuk sistem tiga fasa yang seimbang, dengan satuan yang dipakai adalah perunit, sebab sistem perunit tidak akan terpengaruh oleh perbedaan tegangan pada sistem jaringan yang sesungguhnya.

3.1.5. Metode *Newton Raphson*

Metode *Newton Raphson* dibentuk berdasarkan matriks admitansi simpul (Y_{bus}), sehingga merupakan suatu prosedur yang langsung dan sederhana.

Pada admitansi simpul elemen diagonalnya (Y_{pp}) adalah jumlah admitansi dari semua elemen-elemen jaringan yang terhubung dengan simpul p tersebut. Untuk elemen bukan diagonal (Y_{pq}) adalah sama dengan negatif admitansi dari elemen jaringan yang menghubungkan simpul p dan simpul q .

Karena pada jaringan sistem tenaga listrik tidak semua simpul saling berhubungan satu dengan yang lainnya, maka Y_{bus} akan membentuk matriks

yang terdiri dari elemen-elemen yang mempunyai nilai tidak sama dengan nol (diantara simpul-simpul tersebut mempunyai hubungan saluran) dan elemen-elemen yang bernilai sama dengan nol (diantara simpul-simpul tersebut tidak mempunyai hubungan saluran). Secara matematis persamaan aliran daya metode *Newton Raphson* dapat menggunakan koordinat rectangular, koordinat polar atau bentuk hibrid (gabungan antara bentuk kompleks dengan bentuk polar). Dalam pembahasan ini digunakan bentuk koordinat polar.

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

$$I_p = \sum_{q=1}^n Y_{pq} V_{pq} \quad (3.8)$$

Injeksi daya pada simpul p adalah :

$$S_p = P_p - jQ_p = V_p^* \cdot I_p \quad (3.9)$$

$$= V_p^* \sum_{q=1}^n Y_{pq} V_q \quad (3.10)$$

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

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

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

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

Maka persamaan (3.6) dapat ditulis :

$$P_p - jQ_p = \sum_{q=1}^n |V_p V_q Y_{pq}| e^{-j(\delta_p - \delta_q - \theta_{pq})} \quad (3.11)$$

Dengan memisahkan bagian riil dan bagian imajiner maka diperoleh :

$$P_p = \sum_{q=1}^n |V_p V_q Y_{pq}| \cos(\delta_p - \delta_q + \theta_{pq}) \quad (3.12)$$

$$Q_p = \sum_{q=1}^n |V_p V_q Y_{pq}| \sin(\delta_p - \delta_q + \theta_{pq}) \quad (3.13)$$

Kedua persamaan diatas akan menghasilkan suatu kumpulan persamaan serempak (*simultan*) yang tidak linier untuk setiap simpul sistem tenaga listrik. Untuk mengetahui magnitudo tegangan (V) dan sudut fasa (δ) di setiap simpul dapat diselesaikan dengan menggunakan persamaan (3.12) dan (3.13) yang dilinierkan dengan metode *Newton Raphson* seperti persamaan berikut ini :

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} H & N \\ M & L \end{bmatrix} \begin{bmatrix} \Delta \delta \\ \Delta |V| \end{bmatrix} \quad (3.14)$$

Dimana :

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

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

$\Delta \delta$ = vektor koreksi sudut fasa tegangan.

$\Delta|V|$ = vektor koreksi magnitudo tegangan.

H, L, M, N merupakan elemen-elemen *off* diagonal dan diagonal dari sub matriks Jaqobian yang dibentuk dengan mendefinisikan persamaan (3.12) dan (3.13),
dimana :

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

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

Adapun rumus dari elemen matriks Jaqobian adalah :

Untuk H :

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

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

$q \neq p$

Untuk N :

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

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

$q \neq p$

Untuk M :

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

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

Untuk L :

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

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

Untuk menghitung selisih daya, maka mula-mula ditentukan nilai awal tegangan simpul dan sudut fasanya. Kemudian daya nyata dan daya reaktif dihitung dengan menggunakan persamaan (3.12) dan (3.13). Selisih antara daya yang telah ditentukan dengan daya hasil perhitungan ini merupakan perubahan daya yang terjadi pada simpul.

$$\Delta P = P_{pembangkitan} - P_{beban} - P_{perhitungan} \quad (3.15)$$

$$\Delta Q = Q_{pembangkitan} - Q_{beban} - Q_{perhitungan} \quad (3.16)$$

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

Persamaan (3.14) diselesaikan untuk menghitung vektor koreksi magnitudo tegangan $\Delta(|V|)$ dan sudut fasa tegangan ($\Delta\delta$) yang baru. Sehingga diperoleh harga magnitudo tegangan dan sudut fasa tegangan yang baru, yaitu :

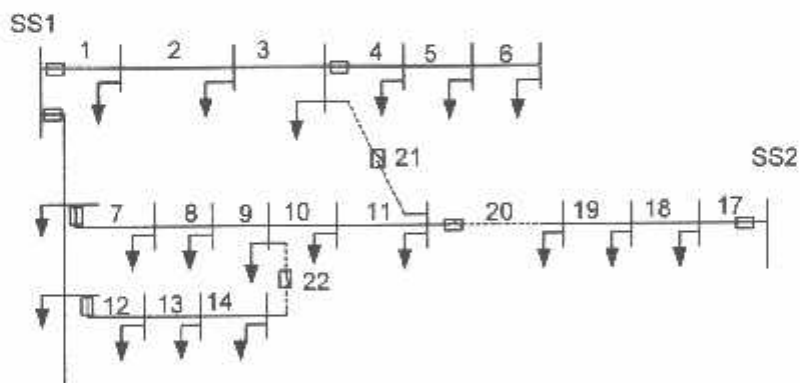
$$|V|^{k+1} = |V|^k + \Delta|V|^k \quad (3.17)$$

$$\delta_{k+1} = \delta_k + \Delta\delta_k \quad (3.18)$$

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

3.1.6. Pernyataan Permasalahan

Untuk menyederhanakan presentasi, kita akan menghadirkan sistem pada suatu basis per tahap dan beban sepanjang suatu pensuplai adalah tetap sedangkan penempatan beban P, Q pada ujung baris. Kita juga mengasumsikan bahwa tiap-tiap tombol dihubungkan dengan satu baris pada suatu sistem. Sebagai contoh, kita berasumsi bahwa sistem gambar 3-1 dapat diterjemahkan untuk suatu jaringan padanan yang ditunjukkan pada gambar 3-2.



Gambar 3-1: Diagram satu garis dari suatu sistem distribusi kecil

Pada gambar, cabang padat merepresentasikan busbar busbar anak-anak bahwa pada layanan dan ketentuan dasar susunan radial. Cabang yang diberi titik (cabang 20,21,22) menghadirkan busbar busbar anak-anak dengan tombol terbuka.

Jaringan dasar dapat diatur kembali oleh penutupan pertama suatu cabang terbuka, katakan cabang 21 pada gambar. Sejak penekanan tombol ini akan

menciptakan sebuah jaringan di dalam sistem, (yang terdiri atas cabang 1,2,3,21,11,10,9,8,7, dan 15), sebuah cabang di dalam jaringan yang berisi suatu tombol harus selalu terbuka, katakan pada cabang 7, untuk mengembalikan struktur radial dari sistem. Sebagai hasil dari switching, beban antara cabang 7-11 akan ditransfer dari satu pensuplai ke pensuplai lainnya. Kita gunakan istilah yang sama untuk cabang [7] dan menamainya operasi dasar *switching* yang merupakan suatu pertukaran cabang antara cabang 21 dan 7. Secara umum, seperti digambarkan pada pengenalan, rencana switching mungkin lebih rumit; kita akan meniru kasus dengan menerapkan beberapa pertukaran cabang berturut-turut.

Perpindahan beban antar feeder yang berbeda dapat ditirukan oleh pertukaran cabang jenis *switching* juga. Dalam hal ini, titik cabang stasiun (titik SS1 dan SS2 pada gambar) akan diperlakukan seperti umumnya walaupun mereka bukan pada titik yang sama. Metoda yang diperkenalkan pada makalah ini mampu menangani seluruh kasus. Ini adalah sesuatu yang penting dari metoda yang diusulkan.

Permasalahan penyusunan ulang jaringan untuk pengurangan kerugian dan menjaga keseimbangan beban akan melibatkan jenis operasi yang sama, yakni beban yang berpindah antara *feeder* atau cabang stasiun oleh perubahan posisi tombol. Perbedaannya hanya pada sasaran. Faktor lain, seperti profil tegangan dari sistem, kapasitas *line/transformers*, penghambat keandalan dapat berfungsi sebagai penghambat.

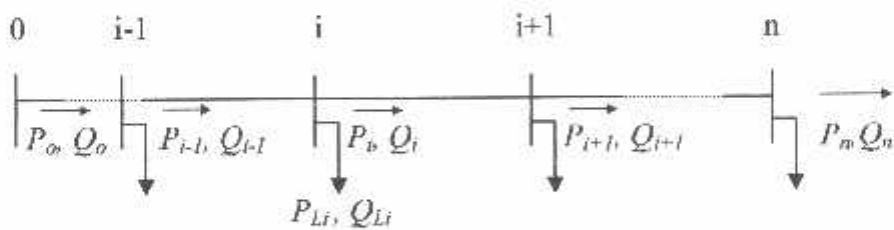
Untuk menyatakan permasalahan ini sebagai permasalahan optimisasi, catat bahwa susunan yang radial sesuai dengan suatu "perputaran pohon" dari suatu grafik yang mewakili jaringan topologi. Sehingga, kita menyebutnya

sebagai masalah perputaran pohon minimal yang dapat dicontohkan sebagai berikut. Berikan sebuah grafik, tentukan sebuah perputaran pohon bahwa fungsi sasaran diperkecil pada saat penghambat mencakup: (i) penghambat tegangan, (ii) penghambat kapasitas lines/transformers, (iii) penghambat keandalan.

Ini merupakan suatu masalah optimisasi *combinatorial* sejak solusi melibatkan pertimbangan dari segala kemungkinan perputaran pohon.

3.1.7 Metode Penyelesaian Studi Aliran Daya

Untuk mengkalkulasi istilah pada masalah optimisasi yang digambarkan pada bagian sebelumnya, kita akan menggunakan suatu persamaan aliran daya yang dari sudut bangunannya mewah dan berguna bagi rencana solusi *computationally* yang efisien [13]. Sebagai gambarannya, perhatikan jaringan radial pada Gambar 3-2.



Gambar 3-2: Suatu diagram garis dari suatu jaringan radial

Kita mendapatkan baris dengan impedansi $z_1 = r_1 + jx_1$, dan beban sebagai daya tetap,

$$S_L = P_L + jQ_L.$$

Aliran daya pada suatu jaringan distribusi radial dapat diuraikan oleh satu persamaan berulang, disebut *cabang persamaan Distflow*, bahwa penggunaan daya nyata, daya reaktif, dan tegangan yang besar di pengiriman akhir dari suatu cabang- P_i , Q_i , V_i berturut-turut pernyataan jumlah yang sama tentang penerimaan akhir dari cabang sebagai berikut.

$$P_{i-1} = P_i - r_i \frac{P_i^2 + Q_i^2}{V_i^2} - P_{Li-1} \quad (3.19)$$

$$Q_{i+1} = Q_i - x_i \frac{P_i^2 + Q_i^2}{V_i^2} - Q_{Li+1} \quad (3.20)$$

$$V_{i+1}^2 = V_i^2 - 2(r_i P_i + x_i Q_i) + (r_i^2 + x_i^2) \frac{P_i^2 + Q_i^2}{V_i^2} \quad (3.21)$$

Karenanya, jika P_0 , Q_0 , V_0 di tangkai pohon pertama dari jaringan dikenal atau diperkirakan, kemudian jumlah yang sama di tangkai pohon yang lain dapat dihitung dengan menerapkan pada persamaan cabang yang berturut-turut. Kita akan mengacu pada prosedur pembaharuan ini.

Persamaan cabang Distflow dapat ditulis mundur, yaitu dengan menggunakan daya riil, daya reaktif, dan tegangan yang besar pada penerimaan akhir suatu cabang, P , Q_i , V_i untuk menyatakan jumlah yang sama di pengiriman akhir cabang itu. Hasil dari persamaan berulang tersebut, disebut *backward branch equations* (persamaan cabang mundur),

$$P_{i-1} = P_i + r_i \frac{P_i^2 + Q_i^2}{V_i^2} + P_{Li} \quad (3.22)$$

$$Q_{i-1} = Q_i + x_i \frac{P_i^2 + Q_i^2}{V_i^2} - Q_{Li} \quad (3.23)$$

$$V_{i-1}^2 = V_i^2 + 2(r_i P_i' + x_i Q_i') + (r_i^2 + x_i^2) \frac{P_i'^2 + Q_i'^2}{V_i^2} \quad (3.24)$$

di mana, $P_i' = P_i + PL_i$, $Q_i' = Q_i + QL_i$

Serupa untuk perbaruan ke depan, suatu perbaruan ke belakangpun dapat digambarkan: awal perbaruan dari tangkai pohon terakhir mengumpamakan variabel jaringan itu P_n , Q_n , V_n pada titik itu diberi dan diproses mundur menghitung jumlah yang sama di tangkai pohon yang lain dengan menerapkan Persamaan (2) berturut-turut. Pembaharuan proses berakhir di tangkai pohon yang pertama (tangkai pohon 0) dan akan menyediakan perkiraan baru dari suntikan daya ke dalam jaringan itu, P_0 , Q_0 .

Catat bahwa dengan rencana penerapan backward and forward update berturut-turut seseorang bisa mendapatkan suatu solusi arus daya sebagaimana diterangkan pada [13]

3.1.8 Kalkulasi dari Terminologi Obyek

Setelah suatu model jaringan, sekarang kita dapat menyatakan kerugian daya dan mengukur keseimbangan beban pada sistem dalam hal variabel sistem.

Untuk mengurangi kerugian, obyek akan memperkecil total i^2r kerugian pada sistem, yang mana dapat dihitung sebagai berikut.

$$LP = \sum_{i=0}^{n-1} r_i \frac{P_i'^2 + Q_i'^2}{V_i^2} \quad \text{p.u.} \quad (3.25)$$

Ini akan menjadi fungsi obyek, cp dari penyusunan ulang jaringan untuk mengurangi kerugian.

Untuk keseimbangan beban, kita akan menggunakan perbandingan daya kompleks pada pengiriman akhir dari suatu cabang. S_i di atas kapasitas kVA, S_i^{max}

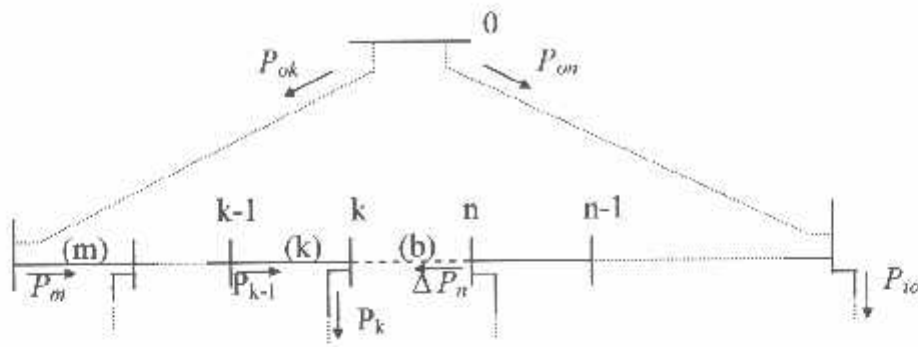
sebagai satu ukuran dari berapa banyak cabang itu terisi. Cabang bisa merupakan suatu trafo, suatu tie-line dengan suatu tombol yang sectionalizing atau sederhananya satu bagian baris. Kemudian kita menggambarkan index keseimbangan beban untuk keseluruhan sistem sebagai penjumlahan dari ukuran ini, yaitu.

$$c_b = \sum \left(\frac{S_i}{S_{i \max}} \right)^2 = \sum \frac{P_i^2 + Q_i^2}{S_{i \max}^2} \quad (3.26)$$

Ini akan menjadi fungsi obyektif, c_b dari keseimbangan beban

Seperti catatan sebelumnya, dua masalah serupa. Keduanya memerlukan data yang sama (parameter sistem dan beban) dan kalkulasi aliran beban untuk mengevaluasi sasaran hasil untuk suatu topologi jaringan yang ditentukan.

Seperti contoh pada bagian sebelumnya, pertukaran cabang dapat digunakan untuk menciptakan perputaran pohon relevan mulai dari perputaran dasar. Secara umum, suatu perputaran pohon T_0 , adalah menghubungkan suatu loop dengan tiap-tiap cabang terbuka pada jaringan dengan mempertimbangkan seolah-olah cabang tertutup. Gambar 3-3 menunjukkan sebuah loop dihubungkan dengan cabang b yang terbuka. Perubahan cabang menciptakan suatu pohon baru dengan menutup suatu cabang terbuka (cabang b pada gambar) dan dengan membuka suatu cabang tertutup pada loop (katakan cabang m pada gambar).



Gambar 3-3: Pengulangan/Jerat dihubungkan dengan cabang terbuka b

Gagasan dasar rencana pencarian yang menggunakan pertukaran cabang akan dimulai dengan kemungkinan suatu pohon dan kemudian menciptakan berturut-turut sesuatu yang baru dengan menerapkan satu pergantian cabang untuk ditetapkan menjadi "satu yang terbaik" (satu yang berfungsi untuk meningkatkan sasaran paling besar tanpa pelanggaran batasan) kemungkinan antara semua pohon (cabang) bahwa dapat diturunkan dari pejabat yang sekarang yang perputaran pohon (induk) dengan pertukaran cabang. Metoda dapat diuraikan sebagai suatu algoritma dengan langkah-langkah sebagai berikut :

Langkah 1: Berikan dengan kemungkinan sebuah pohon (induk)

menjalankan Aliran Daya untuk menentukan titik operasi

Langkah 2: Menguji semua cabang dari induk sebagai berikut.

untuk masing-masing cabang terbuka b

- temukan suatu calon pohon baru, T dengan cara

- mengidentifikasi loop
- memutuskan atas cabang, m untuk dikembalikan

- untuk calon pohon, T
- mengkalkulasi pengurangan pada sasaran, Δc_{bm}

Langkah 3: Sort; jenis cabang (pengujian pohon) dengan menggunakan Δc_{bm} 'S

Langkah 4: Temukan pohon T^* yang mana mempunyai $\Delta c_{bm} > 0$ yang terbesar dan memuaskan batasan kelayakan.

Langkah 5: Jika ada seperti sebuah T^* ,

Kemudian memilih T^* sebagai T_0 dan melangkah ke 1; selain itu stop.

Kita mencatat komentar berikut tentang pencarian.

- Pencarian ini tidak menguji kemungkinan semua pohon dan karenanya solusi akan terjadi di tempat optimal.
- Efisiensi computational dari engsel algoritma ini pada dua hal-hal; pemilihan cabang m untuk; menjadi terbuka, sejak secepatnya mempengaruhi banyaknya pencarian untuk dilakukan, dan kalkulasi dari terminologi sasaran, milik c , karena masing-masing kalkulasi memerlukan suatu aliran daya baru. Meskipun solusi aliran daya dapat diperoleh melalui Distflow yang secara efisien, adalah diinginkan untuk; menjadi mampu menaksir aliran daya lebih cepat tanpa benar-benar mempertimbangkan untuk menjalankan suatu Distflow untuk masing-masing pertukaran cabang. Ini akan mengurangi solusi Distflow ke satu untuk mencari tingkatan yang lain (berulang-ulang)

- Perkiraan aliran daya digunakan pada urutan cabang terbuka. Oleh karena itu, kesalahan perkiraan pada figur bisa mendorong kearah suatu pencarian berbeda dibandingkan dengan penggunaan suatu aliran daya yang tepat.

Pada bagian berikutnya, dua aliran daya dengan metoda pendekatan berbeda, dengan derajat ketelitian yang berbeda, diberikan untuk pengurangan kerugian.

3.2 Metoda Backward and Forward Update

3.2.1 Update Arus Daya

Metoda yang kedua menggunakan Backward and Forward Update dari Distflow, pada bagian 2 diperkenalkan Update aliran daya di sekitar loop dari suatu pertukaran cabang. Untuk pertukaran cabang nominal b-k dari gambar 4, langkah-langkah metoda menjadi seperti berikut :

Langkah 1: Backward Update

Update aliran daya di sekitar loop oleh Backward Update dimulai dari titik k dan n pada loop dan dengan menyelesaikan Update tegangan dan daya secara terpisah (yaitu, menggunakan Persamaan (2.i) dan Persamaan (2.ii) dengan tegangan asli, V_i untuk perbaruan daya, dan menggunakan persamaan (2.iii) untuk perbaruan tegangan. Dengan mengabaikan perbaruan daya

$$\hat{P}_i, \hat{Q}_i, \quad i=k, \dots, ok \quad ; \quad \hat{P}_i, \hat{Q}_i, \quad i=n, \dots, on \quad (3.27)$$

dan Update tegangan pada tangkai pohon yang umum adalah V_{on}^k dan V_{ok}^k .

Langkah 2: Forward Update

Bandingkan perbedaan tegangan pada titik o (perbedaan antara V_o dan V_{on}^k, V_{ok}^k). Jika perbedaan tegangan terlalu besar (lebih besar dari suatu nilai yang sudah dikenal, ϵ^{max}), berhasil; keliling suatu forward update untuk mengurangi kesalahan (waktu ini dimulai dari titik yang umum o dan penggunaan V_o, P_{ok}^k, P_{on}^k , seperti memaraf, pemberian nilai-nilai dan penerapan perbaruan depan). Maka update daya adalah :

$$P_i^k, Q_i^k, \quad i=ok-1, \dots, k ; \quad P_i^k, Q_i^k, \quad i=on+1, \dots, n \quad (3.28)$$

Langkah 3: Mengoreksi aliran daya di titik yang umum

Menggunakan perbedaan antara perbaruan P_k^k, P_n^k and P_k^k, P_n^k seperti daya tak serasi dan mengoreksi P_{on}^k and P_{ok}^k oleh penambahan tidak sepadan mereka, yaitu.,

$$P_{ok}^k = P_{ok}^k + (P_k^k - P_k^k) ; \quad P_{on}^k = P_{on}^k + (P_n^k - P_n^k) \quad (3.29)$$

Rincian pengembangan dari algoritma ini disampaikan pada catatan tambahan C.

Catatan backward and forward update mendasari suatu iterasi dari solusi aliran daya yang menggunakan cabang persamaan Distflow. Di sini, kita

memanfaatkan metoda dengan melokalisir loop pertukaran cabang dan melakukan suatu iterasi khusus. Oleh karena itu::

- ketelitian dari metoda sebagian besar tergantung pada pemindahan beban P_k, Q_k .

3.2.2 Kalkulasi Pengurangan Kerugian Daya

Karena penilaian kerugian daya, maka :

$$P_{ok} - P_{ek}^{\Delta} \approx \Delta P_k + \Delta LP_L ; P_{on} - P_{en}^{\Delta} \approx -\Delta P_k + \Delta LP_R \quad (3.30)$$

di mana, ΔLP_R dan ΔLP_L menyebabkan pengurangan kerugian daya pada R dan L sisi loop berturut-turut. Oleh karena itu, jumlah pengurangan kerugian daya dapat mendekati seperti :

$$\Delta LP = \Delta LP_L + \Delta LP_R = (P_{ek} - P_{ek}^{\Delta}) + (P_{en} - P_{en}^{\Delta}) \quad (3.31)$$

BAB IV

PENERAPAN METODE *BRANCH EXCHANGES* DALAM MEREKONFIGURASI JARINGAN DISTRIBUSI TENAGA LISTRIK

4.1. Metode *Branch Exchanges*

4.1.1 Maksud dan tujuan

Metode *Branch Exchanges* adalah suatu aturan yang tersusun dari proses dengan berdasarkan pada pengalaman, intuisi dan pertimbangan-pertimbangan. Dengan kata lain metode *Branch Exchanges* erat hubungannya dengan cara fikir seorang insinyur yang berbasis pada kemampuan analisis dan pengalaman, jadi bukan suatu bentuk optimasi yang berbasis murni matematika.

Metode *Branch Exchanges* merupakan salah satu metode yang mendekati baik dalam menemukan solusi dengan waktu perhitungan yang singkat, hal ini disebabkan metode *Branch Exchanges* mempunyai aturan-aturan (*rules*) yang mudah dan efisien sehingga mengurangi berbagai alternatif pencarian solusi. Dengan meminimalkan jumlah kandidat suatu solusi, maka metode *Branch Exchanges* mempunyai nilai tambah tersendiri dibandingkan dengan metode lain.

4.1.2. *Branch Exchanges* Dalam Sistem Algoritma

Pada dasarnya algoritma sering digunakan sebagai alat untuk mempermudah pemecahan suatu permasalahan yang dinyatakan dalam bentuk matematis. Sedangkan *Branch Exchanges* adalah suatu kumpulan algoritma-algoritma, maka sangatlah penting untuk menetapkan *Branch Exchanges* dalam bentuk algoritma sehingga memiliki aturan-aturan yang jelas, apabila tidak diperoleh suatu metode yang efisien.

Untuk mendapat solusi yang baik pada suatu masalah ada beberapa strategi yang perlu dipertimbangkan yaitu :

- ▶ Mencari algoritma yang efisien
- ▶ Menunjukkan bahwa hanya kasus khusus dari masalah yang akan diselesaikan dengan membuat algoritma yang efisien
- ▶ Melonggarkan beberapa pembatas yang ada dan mengembangkan algoritma masalahnya sejauh tidak menyimpang dari aturan-aturan yang telah ditetapkan.

Untuk pemecahan masalah *rekonfigurasi* jaringan digunakan bantuan program komputer. Program komputer ini sangat berguna untuk mempercepat proses perhitungan yang membutuhkan ketelitian tinggi dan sering melibatkan iterasi yang membutuhkan waktu yang lama bila dikerjakan manual.

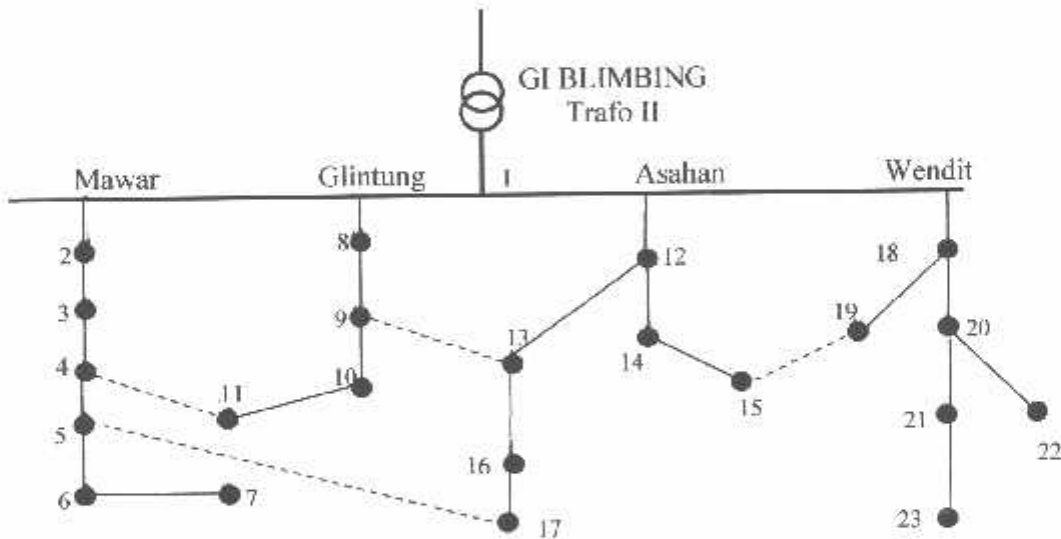
Program yang dipergunakan pada bahasa pemrograman *Borland Delphi* versi 7.0 merupakan bahasa pemrograman terstruktur yang relatif mudah untuk dipelajari dan mudah penggunaannya.

4.2. Bentuk Jaringan Sistem Distribusi

Sebagai sarana untuk mengilustrasikan pelaksanaan program penyusunan ulang jaringan distribusi dengan metode *Branch Exchanges* dipergunakan contoh sistem jaringan distribusi primer yang dalam keadaan awalnya seperti ditunjukkan pada gambar 4.1 dengan data impedansi seperti pada tabel 4.2.

Sistem distribusi yang pada awalnya diasumsikan bersifat radial yaitu dalam kelompok beban dilayani oleh satu sumbu dan beban-beban untuk tiap node

ditetapkan besarnya serta ditetapkan besarnya serta terdapat *switch-switch* yang statusnya dapat diubah sesuai dengan karakteristik dan bentuk jaringan.. Pada jaringan distribusi ini menggunakan transformator II GI Blimbing sebagai sumber tenaga dan mensuplai daerah yang kepadatan bebannya sangat bervariasi, dimana semua itu di gunakan hanya sebagai contoh untuk menunjukkan bagaimana program penyusunan ulang jaringan dapat di tampilkan. Tujuan dari program penyusunan ulang jaringan dapat ditampilkan. Tujuan utama pada program ini untuk mencari solusi yang mendekati *optimal* dalam mengatasi beban lebih pada jaringan dan tegangan abnormal pada jaringan, begitu pula secara bersama untuk mereduksi rugi-rugi daya yang ada pada suatu jaringan.



Gambar 4.1. Konfigurasi Jaringan Radial sebelum Rekonfigurasi

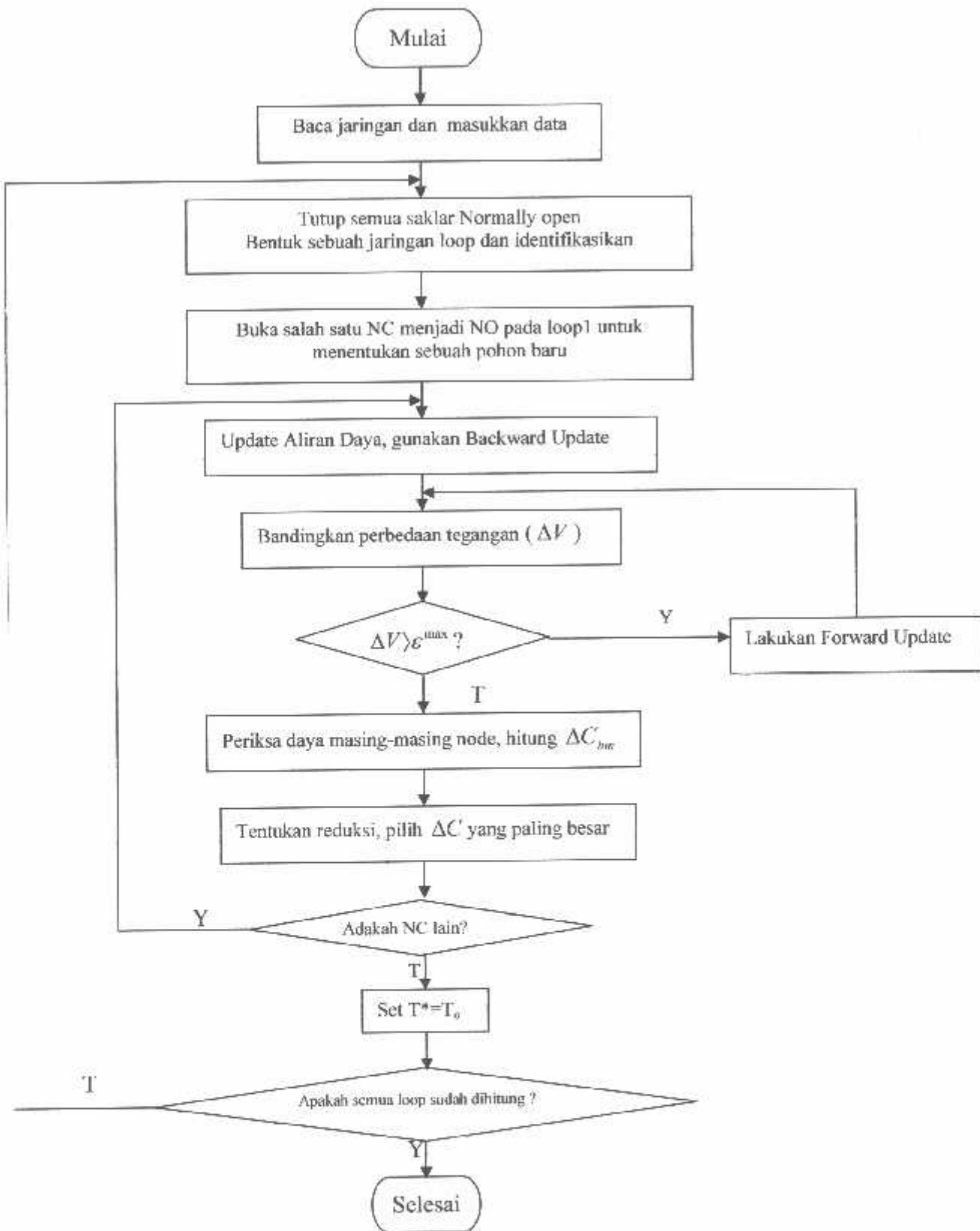
- Keterangan :
- = Switch Penghubung (NC)
 - = Switch Pembatas (NO)
 - = Titik beban

4.3. Model Penyusunan Ulang Jaringan Distribusi

Dalam strategi penyusunan ulang jaringan distribusi ada beberapa aturan untuk mendapat konfigurasi baru yang bisa mengatasi pelanggaran-pelanggaran diatas. Diagram alir (*flow-chart*) pengoperasian program penyusunan ulang jaringan secara *branch exchanges* ditunjukkan pada gambar 4.2. dan secara garis besarnya dapat dijelaskan sebagai berikut :

1. Masukkan data-data jaringan.
2. Tutup semua Normally Open, bentuk jaringan *loop*.
3. Tentukan pohon awal (T_0) dan hitung aliran daya.
4. Identifikasi loop.
5. Tentukan pohon baru (T) dengan membuka salah satu Normally Close menjadi Normally Open..
6. Update aliran daya dengan menggunakan Backward Update. Gunakan persamaan 3.4 dan 3.5 dengan tegangan yang asli dan gunakan persamaan 3.6 untuk meng-update tegangan
7. Bandingkan perbedaan tegangan pada ode awal dengan kedua tegangan ujung (V_0 dengan V_{ok}, V_{on}). Jika perbedaan tegangan besar sekali, (lebih besar dari ϵ^{max}), lakukan Forward Update untuk menguji error.
8. Periksa daya masing-masing node, dan hitung besarnya reduksi (ΔC_{hm}).
9. Periksa apakah ada NC pada loop ini. Jika ada, buka NC dan tutup NO. Kembali ke langkah 6. Jika tidak ada, tentukan reduksi pada loop 1 ($T^{\#}$) dengan memilih ΔC yang paling besar, kemudian lanjutkan ke langkah 10.
10. Set $T^*=T_0$

11. Periksa apakah semua loop sudah dihitung, jika belum kembali ke langkah 3 dan jika sudah perhitungan selesai.



Gambar 4.2. Flowchart Program

4.4. Pengolahan Data

Data teknis dari Trafo II G.I Blimbing Malang adalah sebagai berikut :

- Tegangan Dasar : 70/20 kV
- Daya Dasar : 30 MVA
- Penghantar SUTM 20 kV : AAAC 150 mm²
- Impedansi Saluran : 0.2162 + j 0.3305 Ω / km
- Masing-masing penyulang di seting 300 A

Dari data beban untuk masing-masing penyulang (lampiran A1-A7) dan data teknis, nilai P (MW), Q (MVAR), R (Ω) dan X (Ω) dari bus 1-2 Penyulang Mawar dapat dihitung sebagai berikut :

Dengan nilai $\cos \varphi$ diasumsikan sebesar 0.86

❖ Untuk bus 1-2 :

$$P = 1533,64 \cos \varphi \rightarrow P = 1533,64 \times 0.86 = 1.3190 \text{ MW}$$

$$Q = 1533,64 \sin \varphi \rightarrow Q = 1533,64 \times 0.51 = 0.7820 \text{ MVAR}$$

$$R = 5,616 \text{ km} \times R \text{ saluran} \quad R = 5,616 \text{ km} \times 0.2162 \text{ } \Omega/\text{km} = 1,2142 \text{ } \Omega$$

$$X = 5,616 \text{ km} \times X \text{ saluran} \quad X = 5,616 \text{ km} \times 0.3305 \text{ } \Omega/\text{km} = 1,856 \text{ } \Omega$$

Dengan cara menghitung yang sama, maka daya P, Q, R dan X untuk masing-masing bus akan diperoleh hasil seperti pada tabel 4-1 berikut :

Tabel 4.1
Hasil Perhitungan Daya dan Impedansi Saluran
Jaringan Distribusi GI Blimbing Malang

Bus	P Bus Ujung (MW)	Q Bus Ujung (MVAR)	Jarak (km)	Resistansi R (ohm)	Reaktansi X (ohm)
1 – 2	1.319	0.782	5,616	1,2142	1,8560
2 – 3	1.091	0.647	4,085	0,8832	1,3501
3 – 4	1.303	0.773	5,04	1,0896	1,6657
4 – 5	1.025	0.608	3,897	0,8425	1,2880
5 – 6	0.24	0.142	0,677	0,1464	0,2238
6 – 7	0.675	0.401	1,678	0,3628	0,5546
1 – 8	0.449	0.266	1,381	0,2986	0,4564
8 – 9	0.561	0.333	1,194	0,2581	0,3946
9 – 10	0.5	0.297	1,016	0,2197	0,3358
10 – 11	1.607	0.953	1,492	0,3226	0,4931
1 – 12	0.19	0.113	2,474	0,5349	0,8177
12 – 13	0.089	0.527	3,443	0,7444	1,1379
12 – 14	0.428	0.254	1,646	0,3559	0,5440
14 – 15	0.427	0.253	1,183	0,2558	0,3910
13 – 16	0.543	0.322	2,745	0,5935	0,9072
16 – 17	0.137	0.081	0,131	0,0283	0,0433
1 – 18	0.288	0.171	1,782	0,3853	0,5890
18 – 19	0.058	0.034	3,031	0,6553	1,0017
18 – 20	0.699	0.414	7,639	1,6516	2,5247
20 – 21	0.153	0.091	1,549	0,3349	0,5119
20 – 22	0.343	0.203	1,476	0,3191	0,4878
21 – 23	0.248	0.147	4,689	1,0138	1,5497
4 – 11	-	-	0,207	0,0448	0,0684
9 – 13	-	-	0,173	0,0374	0,0572
15 – 19	-	-	1,114	0,2408	0,3682
5 – 17	-	-	0,262	0,0566	0,0866

Keterangan : Bus 1 merupakan slack bus, bus 2 sampai dengan 23 merupakan bus beban

Dari data pada tabel 4.1 (data jaringan distribusi) dimasukkan ke dalam software rekonfigurasi, maka hasil dari perhitungan tersebut dapat dilihat pada table 4.2 dibawah ini.

Tabel 4.2
Hasil Perhitungan Aliran Daya Sebelum Rekonfigurasi

Bus	Tegangan		Daya Pembangkitan		Daya Pembebanan	
	V (pu)	Sudut(°)	MW	MVAR	MW	MVAR
1	1.00000	0.00000	12.566	8.107	0.000	0.00
2	0.97962	-0.53497	0.00	0.00	1.319	0.782
3	0.97050	-0.77170	0.00	0.00	1.091	0.647
4	0.96511	-0.89808	0.00	0.00	1.303	0.773
5	0.96636	-0.83851	0.00	0.00	1.025	0.608
6	0.96569	-0.85773	0.00	0.00	0.240	0.142
7	0.96444	-0.89148	0.00	0.00	0.675	0.401
8	0.99551	-0.12525	0.00	0.00	0.449	0.266
9	0.99219	-0.21867	0.00	0.00	0.561	0.333
10	0.98995	-0.28181	0.00	0.00	0.500	0.297
11	0.98745	-0.35287	0.00	0.00	1.607	0.953
12	0.98836	-0.26754	0.00	0.00	0.190	0.113
13	0.97588	-0.54616	0.00	0.00	0.089	0.527
14	0.98688	-0.31060	0.00	0.00	0.428	0.254
15	0.98635	-0.32565	0.00	0.00	0.427	0.253
16	0.96732	-0.80831	0.00	0.00	0.543	0.322
17	0.96698	-0.81874	0.00	0.00	0.137	0.081
18	0.99666	-0.09274	0.00	0.00	0.288	0.171
19	0.99648	-0.09791	0.00	0.00	0.058	0.034
20	0.98511	-0.41808	0.00	0.00	0.699	0.414
21	0.98446	-0.43663	0.00	0.00	0.153	0.091
22	0.98459	-0.43322	0.00	0.00	0.343	0.203
23	0.98325	-0.47144	0.00	0.00	0.248	0.147

Keterangan : Bus 1 merupakan slack bus, bus 2 sampai dengan 23 merupakan bus beban.

Tabel 4.3
Hasil Perhitungan Rugi-rugi Daya Sebelum Rekonfigurasi

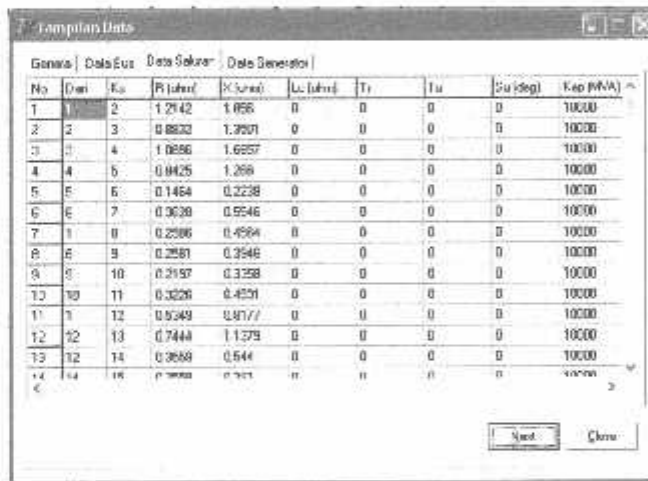
Total Pembangkitan		Total Pembebanan		Total Rugi-rugi Daya	
P (MW)	Q (MVAR)	P (MW)	Q (MVAR)	P (MW)	Q (MVAR)
12.566	8.107	12.373	7.812	0.193	0.295

4.5. Analisa Perhitungan Program

Analisa hasil perhitungan program rekonfigurasi sistem jaringan distribusi ini dilakukan untuk mereduksi rugi-rugi seoptimal mungkin. Dalam melakukan perhitungan dibutuhkan suatu bahasa pemrograman agar perhitungan dapat dilakukan dengan cepat dan efisien. Dalam skripsi di gunakan bahasa pemrograman Delphi 7. Adapun langkah-langkah dalam menjalankan pemrograman rekonfigurasi dengan metode *Branch Exchanges* adalah sebagai berikut

1. Masukkan data jaringan

data jaringan yang dimasukkan berupa data bus dan data saluran



Genera		Data Bus		Data Saluran		Data Generator			
No	Dari	Ke	R (ohm)	X (ohm)	L (ohm)	Tr	Tu	Cu (deg)	Kap (MVA)
1	1	2	1.2142	1.456	0	0	0	0	10000
2	2	3	0.8832	1.2501	0	0	0	0	10000
3	3	4	1.0256	1.6257	0	0	0	0	10000
4	4	5	0.9425	1.269	0	0	0	0	10000
5	5	6	0.1464	0.2228	0	0	0	0	10000
6	6	7	0.3638	0.9546	0	0	0	0	10000
7	1	8	0.2286	0.4564	0	0	0	0	10000
8	6	9	0.2581	0.3546	0	0	0	0	10000
9	5	10	0.2157	0.3258	0	0	0	0	10000
10	10	11	0.3226	0.4531	0	0	0	0	10000
11	1	12	0.5249	0.9177	0	0	0	0	10000
12	12	13	0.7444	1.1279	0	0	0	0	10000
13	12	14	0.3869	0.544	0	0	0	0	10000
14	14	15	0.3869	0.544	0	0	0	0	10000

Gambar 4-3 Tampilan Program Memasukkan Data Saluran

2. Melakukan Perhitungan Aliran Daya Sebelum Rekonfigurasi

Dengan data jaringan yang ada maka dilakukan perhitungan untuk mencari aliran daya awal yang merupakan aliran daya dari sitem sebelum rekonfigurasi. Hasil dari perhitungan ini akan dibandingkan dengan hasil rekonfigurasi sehingga diketahui besar reduksi rugi daya

No	Dari	Ke	P (MW)	Q (MVAR)	Angle (A)	Angle (R)	Dan	Ke	P (MW)	Q (MVAR)
1	1	2	3.400	2.179	163.984	109.989	2	1	-3.850	-
2	2	3	2.031	1.320	103.036	68.431	3	2	-2.018	-
3	3	4	0.927	0.654	47.284	34.343	4	3	-0.922	-
4	4	5	0.380	0.265	19.930	14.761	5	4	-0.380	-
5	5	6	0.916	0.645	46.075	33.972	6	5	-0.916	-
6	6	7	0.576	0.402	34.661	21.334	7	6	-0.676	-
7	7	8	3.140	1.894	136.901	94.179	8	7	-1.130	-
8	8	9	2.591	1.800	134.409	93.769	9	8	-2.674	-
9	9	10	2.113	1.360	136.252	93.881	10	9	-2.110	-
10	10	11	1.010	0.657	81.072	48.706	11	10	-1.602	-
11	11	12	4.331	2.928	211.051	147.921	12	11	-4.180	-
12	12	13	0.139	0.380	158.277	116.225	13	12	-3.111	-
13	13	14	0.858	0.508	43.187	25.936	14	13	-0.895	-
14	14	15	0.427	0.303	21.572	12.948	15	14	-0.427	-

Gambar 4-4 Tampilan Program Perhitungan Aliran Daya Awal.

Summary Loadflow		
Jumlah Pembangkitan	12.588+ 8.108	MVA
Jumlah Pembebanan	12.379+ 7.912	MVA
Jumlah Tumpukan	0.162+ 0.285	MVA
Waktu	5	
Waktu Hitung	00.00	

Gambar 4-5 Tampilan Hasil Load Flow Sebelum Rekonfigurasi

3. Melakukan Pencarian Rekonfigurasi

Tahap ini merupakan tahap akhir program rekonfigurasi dengan metoda *Branch Exchanges*. Pada tahap ini dilakukan penceraian dari metode *Branch Exchanges* untuk melakukan rekonfigurasi jaringan dengan memilih switch-switch mana yang akan di buka atau ditutup sehingga diperoleh hasil reduksi yang paling besar.

No	Bus	Bus	P (MW)	Q (MVAR)	Active (P)	Reactive (Q)	Voltage (kV)	Loss (MW)
1	1	2	1.325	0.733	85.312	38.624	2	1.215
2	2	3	0.000	0.000	0.000	0.000	3	0.000
3	3	4	-1.081	-0.647	-55.019	-24.205	4	1.095
4	4	5	0.000	0.000	0.000	0.000	5	0.000
5	5	6	0.915	0.540	46.620	20.551	6	0.915
6	6	7	0.676	0.402	34.547	21.086	7	0.675
7	7	8	5.507	3.415	280.227	170.753	8	5.574
8	8	9	5.125	3.100	257.746	157.258	9	5.102
9	9	10	4.541	2.731	225.401	140.166	10	4.525
10	10	11	4.125	2.410	204.756	124.786	11	4.006
11	11	12	2.357	1.203	147.072	114.167	12	2.335
12	12	13	2.743	1.742	148.345	100.439	13	2.725
13	13	14	0.000	0.000	0.000	0.000	14	0.000
14	14	15	0.428	-0.254	-21.543	-12.915	15	0.425

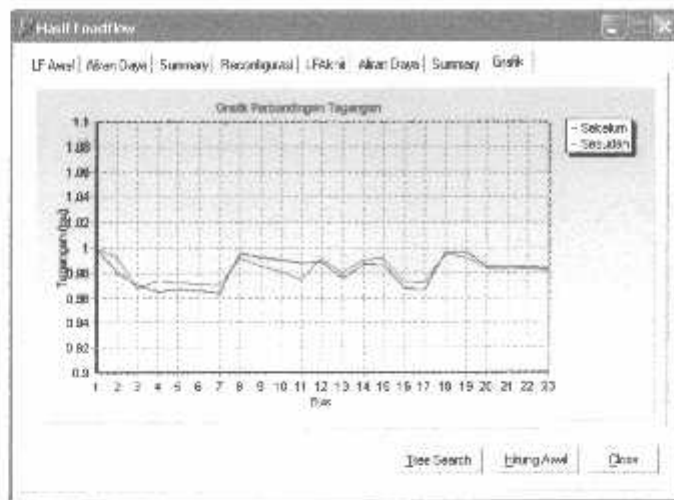
Gambar 4-6 Tampilan hasil Program Besarnya Aliran Daya Setelah Rekonfigurasi

Summary Loadflow	Value	Unit
Jumlah Perulangan	12.553	MVA
Jumlah Pembatasan	12.372	MVA
Jumlah Plug-Plug	0.188	MVA
Iterasi	2	
Waktu Hujung	0.217.05	

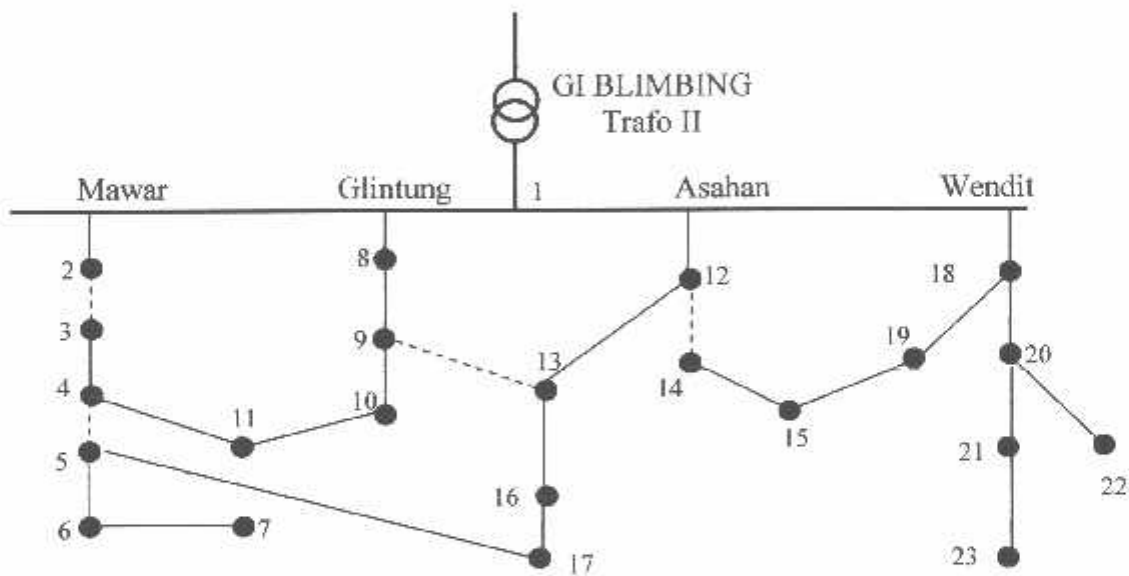
Gambar 4-7 Tampilan Hasil Load Flow Setelah Rekonfigurasi

Mo	Asal	Tujuan
1	2	3
2	4	5
3	9	13
4	12	14

Gambar 4-8 Tampilan Program Hasil Rekonfigurasi



Gambar 4-9 Grafik Perbandingan Tegangan



Gambar 4.10 Konfigurasi Jaringan Radial setelah Rekonfigurasi

4.6. Hasil Perhitungan

Setelah melakukan analisa dan perhitungan dengan Metode *Branch Exchanges*, maka akan diperoleh suatu sistem jaringan hasil rekonfigurasi dimana hasil rekonfigurasi ini dapat mereduksi besar kerugian daya pada sistem.

Tabel 4-4 menunjukkan hasil aliran daya setelah proses rekonfigurasi.

TABEL 4.4
HASIL PERHITUNGAN ALIRAN DAYA SETELAH REKONFIGURASI

Bus	Tegangan		Daya Pembangkitan		Daya Pembebanan	
	V (pu)	Sudut (°)	MW	MVAR	MW	MVAR
1	1.00000	0.00000	12.559	8.096	0.000	0.000
2	0.99230	-0.21632	0.00	0.00	1.319	0.782
3	0.99230	-0.90356	0.00	0.00	1.091	0.647
4	0.97380	-0.73454	0.00	0.00	1.303	0.773
5	0.97162	-0.64153	0.00	0.00	1.025	0.608
6	0.97096	-0.66055	0.00	0.00	0.240	0.142
7	0.96975	-0.69537	0.00	0.00	0.675	0.401
8	0.99193	-0.22225	0.00	0.00	0.449	0.266
9	0.98551	-0.40137	0.00	0.00	0.561	0.333
10	0.98066	-0.53843	0.00	0.00	0.500	0.297
11	0.97432	-0.71942	0.00	0.00	1.607	0.953
12	0.99138	-0.17294	0.00	0.00	0.190	0.113
13	0.98008	-0.39900	0.00	0.00	0.089	0.527
14	0.99067	-0.26152	0.00	0.00	0.428	0.254
15	0.99120	-0.24658	0.00	0.00	0.427	0.253
16	0.97244	-0.61760	0.00	0.00	0.543	0.322
17	0.97215	-0.62594	0.00	0.00	0.137	0.081
18	0.99507	-0.13726	0.00	0.00	0.288	0.171
19	0.99219	-0.21851	0.00	0.00	0.058	0.034
20	0.98350	-0.46380	0.00	0.00	0.699	0.414
21	0.98288	-0.48153	0.00	0.00	0.153	0.091
22	0.98294	-0.47975	0.00	0.00	0.343	0.203
23	0.98166	-0.51647	0.00	0.00	0.248	0.147

4.7. Rugi-rugi daya

Dengan dilakukannya *rekonfigurasi* jaringan ternyata mempengaruhi besar kerugian pada saluran, setelah dilakukan *rekonfigurasi* rugi-rugi daya akan turun. seperti dalam tabel 4.5 dimana rugi-rugi dayanya :

4.5.1. Perhitungan Beban Tiap Penyulang Pada Kondisi Setelah Rekonfigurasi

Dari lampiran tabel A5 pengelompokan *switch* dapat diketahui jumlah beban tiap penyulang setelah rekonfigurasi sebagai berikut :

Jumlah beban penyulang Mawar 2286,80 kVA

$$I \text{ nominalnya adalah } I_n = \frac{VA}{\sqrt{3} \cdot V} = \frac{2286,80 \cdot 10^3}{\sqrt{3} \cdot 20 \cdot 10^3} = 66,01 \text{ Ampere}$$

$$\text{Persentasenya sebesar : } \frac{66,01}{300} \times 100\% = 22\%$$

Jumlah beban penyulang Glintung 3661,42 kVA

$$I \text{ nominalnya adalah } I_n = \frac{VA}{\sqrt{3} \cdot V} = \frac{3661,42 \cdot 10^3}{\sqrt{3} \cdot 20 \cdot 10^3} = 105,7 \text{ Ampere}$$

$$\text{Persentasenya sebesar : } \frac{105,7}{300} \times 100\% = 35,23\%$$

Jumlah beban penyulang Asahan 4104,43 kVA

$$I \text{ nominalnya adalah } I_n = \frac{VA}{\sqrt{3} \cdot V} = \frac{4104,43 \cdot 10^3}{\sqrt{3} \cdot 20 \cdot 10^3} = 118,5 \text{ Ampere}$$

$$\text{Persentasenya sebesar : } \frac{118,5}{300} \times 100\% = 39,5\%$$

Jumlah beban penyulang Wendit 2822,96 kVA

$$I \text{ nominalnya adalah } I_n = \frac{VA}{\sqrt{3} \cdot V} = \frac{2822,96 \cdot 10^3}{\sqrt{3} \cdot 20 \cdot 10^3} = 81,5 \text{ Ampere}$$

$$\text{Persentasenya sebesar : } \frac{81,5}{300} \times 100\% = 27,16\%$$

Dilihat dari hasil perhitungan, beban pada penyulang-penyulang setelah rekonfigurasi tidak sampai melebihi 80% dari I nominal, paling tinggi adalah 39,5 % yang terdapat pada penyulang Asahan.:

Tabel 4.6
Perhitungan Beban Pada Saat Terjadi Pengalihan beban Sebelum Rekonfigurasi

Penyulang	Beban (MVA)	Arus (Amp)	%
Penyulang Mawar	6574,77	189,79	63,26 %
Penyulang Glintung	3625,65	104,66	34,88 %
Penyulang Asahan	2110,11	60,91	20,30 %
Penyulang Wendit	2080,33	60,05	20,017 %

Tabel 4.7
Perhitungan Beban Pada Saat Terjadi Pengalihan beban Setelah Rekonfigurasi

Penyulang	Beban (MVA)	Arus (Amp)	%
Penyulang Mawar	2286,80	66,01	22 %
Penyulang Glintung	3661,42	105,7	35,23 %
Penyulang Asahan	4104,43	118,5	39,5 %
Penyulang Wendit	2822,96	81,5	27,16 %

TABEL 4.8

PERBANDINGAN RUGI-RUGI SALURAN

Rugi-rugi Sebelum		Rugi-rugi Sesudah	
MW	MVAR	MW	MVAR
0.193	0.295	0.186	0.284
Reduksi Rugi-rugi			
MW		MVAR	
0,007		0,011	

Dengan prosentase reduksi rugi-rugi saluran adalah 1.1 %.

BAB V

PENUTUP

5.1. Kesimpulan

Setelah dilakukan analisa mengenai metode *rekonfigurasi* jaringan, maka dapat diambil kesimpulan sebagai berikut :

1. Analisa *rekonfigurasi* yang dikembangkan dalam skripsi ini
 - Menggunakan metode *rekonfigurasi Branch Exchanges*
 - Mereduksi rugi-rugi daya hanya dengan merubah status *switch* tanpa menambah biaya investasi.
2. Setelah dilakukan perhitungan dalam program Delphi yang terlihat pada Gambar 4-8, maka akan diketahui open switch yang terbuka, yaitu dari: bus 2 ke bus 3, bus 4 ke bus 5, bus 12 ke bus 13 dan bus 14 ke bus 15 diperoleh reduksi rugi-rugi daya pada saluran.
3. Total rugi-rugi daya pada saluran turun dari 0,193MW dan 0,295 MVAR menjadi 0,186MW dan 0,284MVAR terjadi optimasi sebesar 0,007MW dan 0,011MVAR atau sekitar 1,1 %
4. Setelah dilakukan perhitungan saat terjadi pengalihan beban pada tiap-tiap penyulang, ternyata tiap-tiap penyulang masih mampu menahan beban.

5.2. Saran

Setelah dilakukan *Rekonfigurasi*, ternyata profile tegangan bisa diperbaiki dan rugi-rugi daya bisa direduksi. Jadi, sebaiknya PLN perlu untuk melakukan *rekonfigurasi* supaya bisa meningkatkan kualitas dan efektifitas terutama pada distribusi kepada pelanggan.

DAFTAR PUSTAKA

1. Hasan Basri, "**Sistem Distribusi**", Balai Penerbit dan Humas ISTN, Jakarta, 1996.
2. *A.S. Pabla*, "**Sistem Distribusi Daya Listrik**", Penerbit Erlangga, Jakarta, 1994
3. Hand Out, "Sistem Distribusi", Oleh Institut Teknologi Nasional, Malang, 2001
4. *Mesut E. Baran, Felix F. Wu*, "**Network Reconfiguration In Distribution Systems For Loss Reduction And Load Balancing**", IEEE Transactions on Power Delivery, Vol. 4, No. 2, April 1997.

LAMPIRAN

Tabel A-1
Data Beban Tiap Seksi Penyulang Mawar

Seksi	No. Trafo	Nominal (kVA)	Persen Beban (%)	Beban Trafo (kVA)
1	186	150	53.87	80.81
	764	160	80.43	124.06
	588	160	65.84	105.34
	424	200	69.53	139.04
	270	150	103.52	155.29
	775	250	77.53	124.06
	744	160	80.43	128.68
	306	160	61.05	78.67
	370	150	0.00	0.00
	455	250	53.92	134.81
	369	150	71.61	107.42
	425	100	39.53	39.53
	605	100	50.5	50.5
	542	150	52.46	78.59
	2	305	200	61.05
230		160	69.87	139.74
543		150	77.61	116.41
741		160	45.38	72.6
358		150	0.00	0.00
38		160	54.84	87.75
237		250	92.43	231.08
3	240	160	52.32	83.71
	22	150	97.12	145.67
	323	150	46.94	70.41
	150	100	65.48	65.48
	369	150	71.61	107.42
	777	250	59.04	147.61
	53	150	90.74	136.1
	265	150	53.86	80.78
	570	160	55.76	89.21
	100	100	23.98	23.98
	412	160	14.44	23.10
	122	150	98.33	147.49
4	615	250	66.42	166.06
	585	200	68.69	137.39
	284	150	62.77	94.16
	737	250	60.91	152.28
	920	50	77.09	38.54
	193	250	81.66	204.16
	253	150	85.68	128.52
5	706	100	7.45	7.45
	15	200	51.7	129.25
	21	200	60.54	121.08

	244	160	52.32	83.71
	36	250	81.33	203.32
6	133	150	0.00	0.00
	584	160	74.06	118.5
	230	160	69.87	139.74
	110	150	0.00	0.00
	358	150	0.00	0.00
	231	160	76.36	122.17
	32	100	74.99	74.99
	335	75	0.00	0.00
	235	200	35.86	71.72
	315	150	84.51	126.77
	729	160	0.00	0.00

Tabel A-2
Data Beban Tiap Seksi Penyulang Glantung

Seksi	No. Trafo	Nominal (kVA)	Persen Beban (%)	Beban Trafo (kVA)
1	196	250	69.47	138.94
	996	200	0.00	0.00
	575	160	51.03	51.03
	141	200	71.76	143.52
	93	250	46.22	46.22
	281	160	60.49	96.78
	51	250	49.50	123.75
2	148	250	49.50	123.36
	210	400	0.00	0.00
	16	250	67.94	169.84
	211	150	73.58	110.28
	40	250	85.76	214.4
	398	160	25.55	40.98
	153	160	69.35	110.96
3	165	200	70.3	140.61
	107	100	45.9	45.9
	762	100	0.00	0.00
	432	160	0.00	0.00
	142	250	66.71	166.77
	449	100	0.00	0.00
4	860	200	36.03	72.05
	114	160	0.00	0.00

Tabel A-3
Data Beban Tiap Seksi Penyulang Asahan

Seksi	No. Trafo	Nominal (kVA)	Persen Beban (%)	Beban Trafo (kVA)
1	267	100	80.24	109.44
	125	160	137.56	206.34
	124	100	33.99	67.98
	128	160	19.95	19.95
	757	630	0.00	0.00
	623	200	0.00	0.00
	622	100	0.00	0.00
	735	160	33.34	53.33
2	285	25	4.8	1.2
	752	200	34.2	68.3
	582	160	0.00	0.00
3	667	160	0.00	0.00
	354	160	85.46	128.18
	815	630	0.00	0.00
	234	100	66.83	66.83
	314	150	85.17	27.75
	3	200	59.63	119.25
	906	160	0.00	0.00
	354	160	85.46	128.18
	2	200	78.59	147.17
	910	100	0.00	0.00
	234	100	66.83	66.83
4	624	630	0.00	0.00
	279	250	37.07	74.15
5	528	800	0.00	0.00
	668	250	75	187.5
	865	200	75	150
	654	100	23	36.9
	88	100	75	187.5
	621	100	12.6	20.1
	660	160	6.8	10.9
6	445	630	75	120
	924	200	0.00	0.00
	726	160	29.56	47.84
	735	160	33.34	53.33
	913	100	0.00	0.00
	960	160	0.00	0.00
	372	250	102.24	264.02
	373	160	62.93	100.72
	697	160	93.05	150.22
	434	160	0.00	0.00
	694	160	0.00	0.00
767	160	0.00	0.00	

	734	160	8.90	13.39
	735	160	33.34	53.33
	875	160	31.24	40.59
	938	200	12.35	21.74
	939	100	0.00	0.00
	899	1250	0.00	0.00

Tabel A-4
Data Beban Tiap Seksi Penyulang Wendit

Seksi	No. Trafo	Nominal (kVA)	Persen Beban (%)	Beban Trafo (kVA)
1	224	200	49.50	79.20
2	606	160	43.24	64.86
	28	160	49.35	77.38
3	763	160	126.26	202.02
	844	200	36.54	73.08
4	215	160	43.96	87.91
	577	250	89.50	223.74
5	727	160	73.84	118.14
	185	150	67.61	101.42
	254	150	64.08	0.00
	428	100	57.98	57.98
	469	630	0.00	0.00
	705	50	15.77	7.88
	714	160	34.53	55.25
	715	160	73.84	118.14
6	854	100	43.39	43.99
	537	200	60.06	120.12
	699	100	0.00	0.00
	955	200	18.09	36.19
	956	100	88.66	88.66
	822	100	45.15	90.31
	768	160	37.91	60.65
	769	160	63.85	102.15
	844	200	36.54	73.08
	864	200	27.50	55.00
	974	160	17.17	27.47

Tabel A-5
KHA Penghantar Campuran Alumunium Telanjang (AAAC)

Luas Penampang (mm²)	KHA terus menerus (A)
16	105
25	135
35	170
50 (7 kawat)	210
50 (19 kawat)	210
70	255
95	320
120	365
150	425
185	490
240	585
300	670
400	810
500	930
630	1075
800	1255
1000	1450

Tabel A-5
Pengelompokan Switch

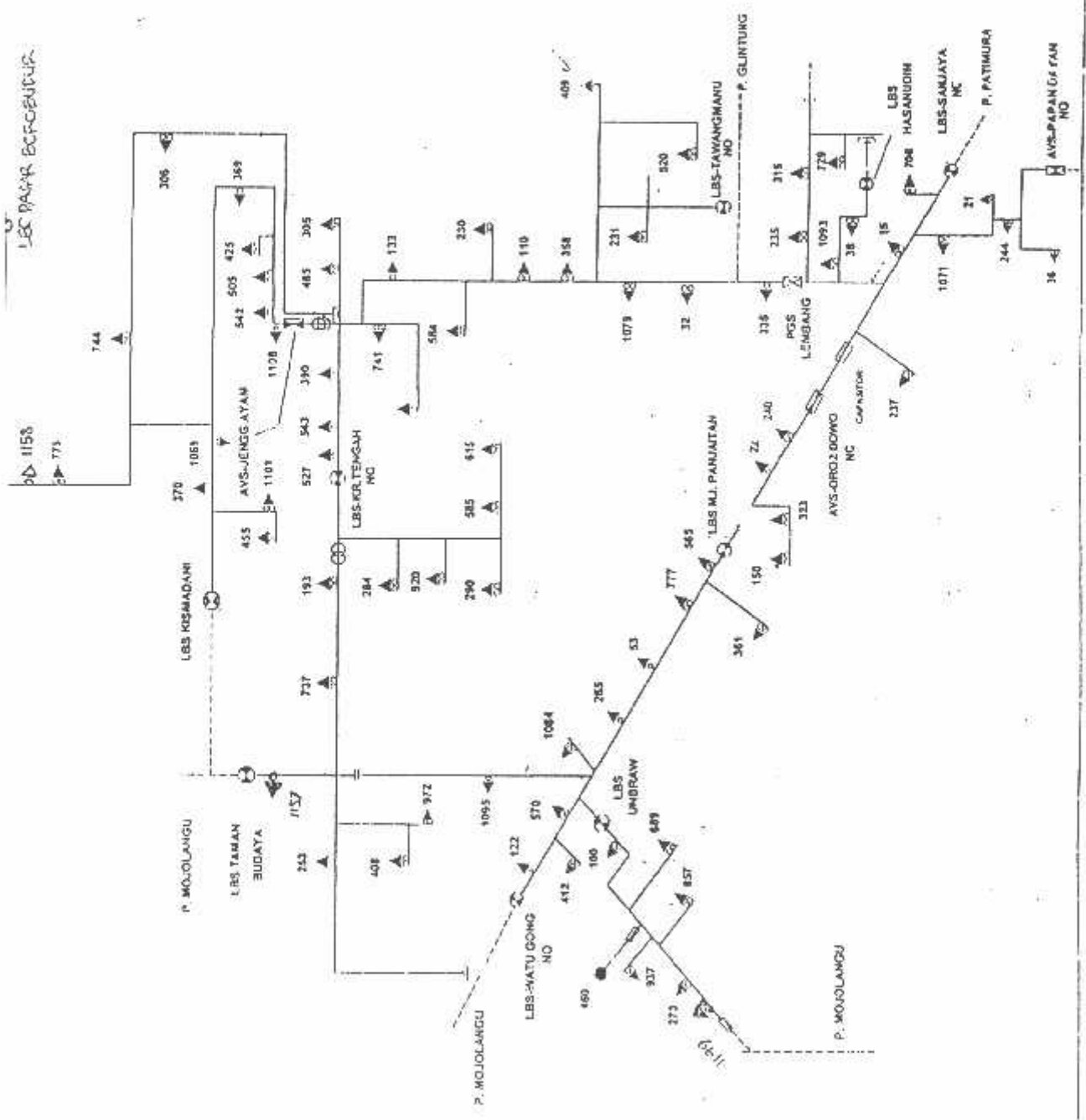
Penyulang	Seksi	Switch	Jumlah Beban (kVA)
Mawar	1	1	2286.80
	2	2	745.25
	3	3	1120.78
	4	4	921.11
	5	5	544.81
	6	6	653.89
Glantung	1	7	600.24
	2	8	769.82
	3	9	353.28
	4	10	72.05
Asahan	1	11	457.04
	2	12	69.5

	3	13	784.19
	4	14	74.15
	5	15	712.90
	6	16	745.18
Wendit	1	17	79.2
	2	18	142.24
	3	19	275.1
	4	20	311.65
	5	21	458.81
	6	22	697.62

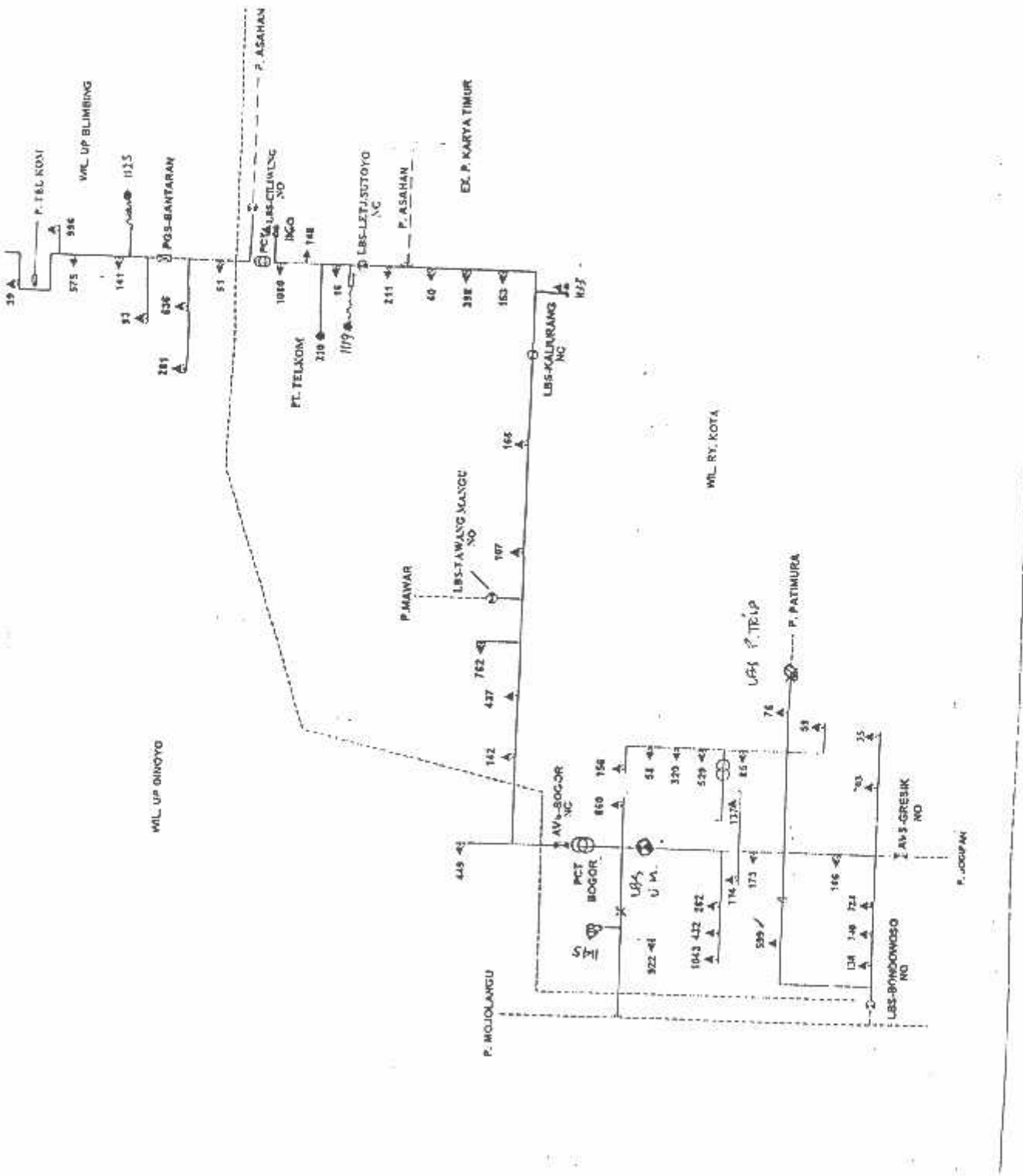
Tabel A-6
Penghantar AAC

Penampang Nominal (mm ²)	Jari-jari (mm)	Urat	GMR (mm)	Impedansi Urutan Positif (Ω/km)	Impedansi Urutan Nol (Ω/km)
16	2.2563	7	1.6380	2.0161 + j0.4036	2.1641 + j1.6911
25	2.8203	7	2.0475	1.2903 + j0.3895	1.4384 + j1.6770
35	3.3371	7	2.4227	0.9217 + j0.3790	1.0697 + j1.6665
50	3.9886	7	2.8957	0.6452 + j0.3678	0.7932 + j1.6553
70	4.7193	7	3.4262	0.4608 + j0.3572	0.6088 + j1.6447
95	5.4979	19	4.1674	0.3396 + j0.449	0.4876 + j1.6324
120	6.1791	19	4.6837	0.2688 + j0.3376	0.4168 + j1.6251
150	6.9084	19	5.2368	0.2162 + j0.3305	0.3631 + j1.6180
185	7.6722	19	5.8155	0.1744 + j0.3239	0.3224 + j1.6114
240	8.7386	19	6.6238	0.1344 + j0.3158	0.2824 + j1.6033

LBS RAKAR BERBENTUK



PT. PLN (PERSERO) DISTRIBUSI JATIM AP&J MALANG	
P. MAWAR GI. BLIMBING	
DISKUSI OPUS	DESKUSI DAPAK D ANG P



- GI
- PGS
- AVS
- CO
- LBS
- SIS PAKLOSER
- SUTM 20 KV
- PCT

PT. PLN (PERSERO)
DISTRIBUSI JATIM
APLJ MALANG

P. GLINTUNG
GL BLIMBING

DIBANGUN	DIREKSI	DISELEKSI
----------	---------	-----------



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

BERITA ACARA UJIAN SKRIPSI

Nama : AGUS JUNAED
N.I.M : 00.12.157
Jurusan : TEKNIK ELEKTRO
Konsentrasi : ENERGI LISTRIK
Judul Skripsi : **REKONFIGURASI JARINGAN SISTEM DISTRIBUSI
UNTUK MENGURANGI RUGI-RUGI DAYA DENGAN
MENGUNAKAN METODE BRANCH EXCHANGES DI
GL. BLIMBING MALANG**

Dipertahankan dihadapan majelis penguji jenjang strata satu (S-1),

Hari : Rabu

Tanggal : 27 September 2006

Nilai : 74,75 (B+)



Ketua
Ir. Mochtar Astroni, MSME
NIP.Y. 101.810.0036

Panitia Ujian

Anggota Penguji



Sekretaris
Ir. F. Yudi Limpraptono, MT
NIP.Y. 103.950.0274



Penguji I
Ir. Eko Nurcahyo
NIP.Y. 102.870.0172

Penguji II



Ir. Choirul Saleh, MT
NIP.Y. 101.8800.190




FORMULIR BIMBINGAN SKRIPSI

Nama : AGUS JUNAEDI
Nim : 00.12.157
Masa Bimbingan : 15 Agustus 2006 s/d 15 Februari 2007
Judul Skripsi : REKONFIGURASI JARINGAN SISTEM DISTRIBUSI UNTUK MENGURANGI RUGI-RUGI DAYA DENGAN MENGGUNAKAN METODE BRANCH EXCHANGES DI GI. BLIMBING MALANG

No	Tanggal	Uraian	Paraf pembimbing
1	27-08-2006	BAKSI PENDAHULUAN	
2	27-08-2006	BAKSI II	
3	27-08-2006	BAKSI III	
4	11-9-2006	BAKSI IV	
5	16-9-2006	BAKSI V: -Revisi -Tabel -kesimpulan	
6	16-9-2006	ACC seminar hasil	
7	25-9-2006	ACC Ujian Kompe	
8			
9			
10			

Malang.
Dosen Pembimbing



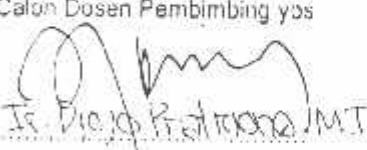

Ir. Djojo Priatmono, MT
Nip. Y : 1018500107

Form. S - 4b



BERITA ACARA SEMINAR PROPOSAL SKRIPSI JURUSAN TEKNIK ELEKTRO S-1

Konsentrasi : Teknik Energi Listrik/Teknik Elektronika*)

1.	Nama Mahasiswa: <u>AGUS SUHAEDI</u>			Nim: <u>001215</u>
2.	Keterangan	Tanggal	Waktu	Tempat
	Pelaksanaan	<u>14-08-2008</u>		Ruang:
Spesifikasi Judul (berilah tanda silang)**)				
3.	a. Sistem Tenaga Elektrik	e. Elektronika & Komponen		
	b. Energi & Konversi Energi	f. Elektronika Digital & Komputer		
	c. Tegangan Tinggi & Pengukuran	g. Elektronika Komunikasi		
	d. Sistem Kendali Industri	h. Lainnya		
4.	Judul Proposal yang diseminarkan Mahasiswa	<u>Pertanggung-jawaban Jaringan Sistem Distribusi Untuk Menguurangi Rugi-rugi Daya Dengan Mengurangi Metode Branch Crosses 17.6 I Blubine Malang</u>		
5.	Perubahan Judul yang diusulkan oleh Kelompok Dosen Keahlian			
6.	Catatan:			
7.	Persetujuan Judul Skripsi			
	Disetujui, Dosen Keahlian I	Disetujui, Dosen Keahlian II		
				
	Mengetahui, Ketua Jurusan.	Disetujui, Calon Dosen Pembimbing I		
<u>Ir. F. Yudi Limpratonc, MT</u> NIP. P. 1039500274		 <u>Is. Djojo Pratomo, MT.</u>		

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



Certificate No.: **QSC**
00475

0341 - 326034 (Hunting)
s : 211

Facsimile : 0341 - 362046

Nomor : **1289 / 330 / APJ-MLG / 2006**
Sarat Sdr.No : **ITN-1937/III.TA/2/06**
Lampiran :
Perihal : **Ijin Survey**

Malang, 24 Agustus 2006

Kepada
Fakultas Teknologi Industri
Institut Teknologi Malang
Jl. Raya Karanglo, Km 2
Di
MALANG

Dengan Hormat,

Menindaklanjuti surat saudara nomor ITN-1937/III.TA/2/2006 tanggal 5 Agustus 2006 perihal tersebut di atas, maka dengan ini kami beritahukan bahwa pada prinsipnya kami tidak keberatan / mengizinkan mahasiswa saudara atas nama :

Agus Junaedi

Nim: 00.12.157

Untuk melaksanakan Survey di PT. PLN (PERSERO) Area Pelayanan dan Jaringan Malang mulai tanggal 28 Agustus 2006 sampai dengan 28 September 2006 dengan catatan PLN hanya memberikan data yang **TIDAK BERSIFAT RAHASIA**.

Sebelumnya diminta agar mahasiswa 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, BE. ST.

Tembusan :

1. Supervisor Terkait
2. Manajer UJ dan UPJ Terkait



BERITA ACARA SEMINAR SKRIPSI JURUSAN TEKNIK ELEKTRO S-1

Konsentrasi : Teknik Energi Listrik/Teknik Elektronika*)

1.	Nama Mahasiswa: AGUS JUNAEI	Nim:
2.	Keterangan	Tanggal
	Pelaksanaan	18-09-2006
Waktu		
Tempat:		
Ruang: SIDANG		
Spesifikasi Judul (berilah tanda silang)**)		
3.	a. Sistem Tenaga Elektrik	e. Elektronika & Komponen
	b. Energi & Konversi Energi	f. Elektronika Digital & Komputer
	c. Tegangan Tinggi & Pengukuran	g. Elektronika Komunikasi
	d. Sistem Kendali Industri	h. lainnya
4.	Judul Proposal yang diseminarkan Mahasiswa	REKONFIGURASI JARINGAN SISTEM DISTRIBUSI UNTUK MENURAHSI RUGI-RUGI DATA DENGAN MENGGUNAKAN METODE BRANCH EXCHANGE & D.C. RUMBITZ
5.	Perubahan Judul yang diusulkan oleh Kelompok Dosen Keahlian/Pengamat
6.	Keputusan, Dari hasil penilaian sejumlahorang dosen keahlian danorang dosen pengamat sesuai format penilaian terlampir, peserta seminar tersebut diatas (1) dengan judul skripsi (4) dinyatakan LULUS/TIDAK LULUS *) dengan nilai Komutatif(angka) atau(huruf)	
Persetujuan Judul Skripsi		
Disetujui, Dosen Keahlian I		Disetujui, Dosen Keahlian II
.....	
Disetujui, Dosen Pengamat I  Ir. TECHA HERBASUKI, MT		Disetujui, Dosen Pengamat II  Ir. WIDODO PUDJI M, MT
Mengetahui, Ketua Jurusan Ir. F. Yudi Limpraptono, MT NIP. P. 1039500274		Disetujui, Dosen Pembimbing Ir. DJOJO PRATMONO, MT

Perhatian:

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

Form S-4c



FORMULIR PENDAFTARAN UJIAN SKRIPSI

Nama	: AGUS JUTAE DI
Nim	: 0012157
Fakultas	: Teknologi Industri
Jurusan/Konsentrasi	: Teknik Elektro / T. Energi Listrik / T. Elektronika-
Jenjang Program	: Strata Satu (S-1)
Alamat di Malang	: Jln Simpang Candi Jewu 25
Masa penulisan Skripsi	
Dosen pembimbing	: Ir. Yudi Limpraptono
Judul Skripsi	: REKONFIGURASI JARINGAN SISTEM DISTRIBUSI UNTUK MENGURANGI RUGI-RUGI DAYA DENGAN METODE BRATEN EXCHANGES DI G.I. BLIMBITG MALANG

Persyaratan yang harus dipenuhi **)

No	Persyaratan	Paraf*)
1.	Telah mengumpulkan 140 sks dengan IPK ≥ 2	
2.	Tidak ada nilai E	
3.	Telah menyelesaikan/mengumpulkan Laporan Praktek Kerja	
4.	Telah menempuh semua praktikum yang di syratkan Jurusan	
5.	Mengumpulkan Buku Skripsi yang telah ditandatangani Dosen pembimbing rangkap 3	
6.	Menyerahkan Surat Puas Bimbingan Skripsi dari pembimbing Ybs.	
7.	Menyerahkan kartu seminar yg di tanda tangani Jurusan	
8.	Telah melunasi persyaratan administrasi (kwitansi warna kuning Rp. 100.000)	

Malang, _____ 200

Mengetahui
Kajur/Sekjur

Mahasiswa ybs,

(Ir. F. YUDI LIMPRAPTONO, MT)
NIP. Y. 1039500274

(AGUS JUTAE DI)

Catatan

- *) Diparaf Sekretaris Jurusan
- **) Lampirkan bukti persyaratan yang diperlukan

L I S T I N G



```
unit uAbout;

interface

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

type
  TfrmAbout = class(TForm)
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmAbout: TfrmAbout;

implementation

{$R *.dfm}

end.
```

unit uHasil;

interface

uses

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

type

```
TfrmHasil = class(TForm)
  TabSheet2: TTabSheet;
  fgBus: TStringGrid;
  TabSheet3: TTabSheet;
  fgBranch: TStringGrid;
  TabSheet4: TTabSheet;
  GroupBox6: TGroupBox;
  Label8: TLabel;
  Label9: TLabel;
  Label10: TLabel;
  Label11: TLabel;
  Label12: TLabel;
  lblGen: TLabel;
  lblLoad: TLabel;
  lblLoss: TLabel;
  edtSumGen: TEdit;
  edtSumLoad: TEdit;
  edtSumLoss: TEdit;
  edtIterasi: TEdit;
  edtTime: TEdit;
  TabSheet9: TTabSheet;
  Panel1: TPanel;
  btnClose: TButton;
  btnHitung: TButton;
  TabSheet1: TTabSheet;
  TabSheet5: TTabSheet;
  TabSheet6: TTabSheet;
  TabSheet7: TTabSheet;
  Chart1: TChart;
  Series1: TLineSeries;
  Series2: TLineSeries;
  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;
fgBranch2: TStringGrid;
fgBus2: TStringGrid;
PageControl1: TPageControl;
GroupBox2: TGroupBox;
Label6: TLabel;
edtNLoop: TEdit;
btnUseDefault: TButton;
GroupBox3: TGroupBox;
Label16: TLabel;
Label17: TLabel;
edtSebelum: TEdit;
edtSesudah: TEdit;
btnCreate: TButton;
GroupBox5: TGroupBox;
fgOS: TStringGrid;
pbCount: TProgressBar;
Label7: TLabel;
Label13: TLabel;
Label14: TLabel;
edtSelisih: TEdit;
Label15: TLabel;
procedure btnCloseClick(Sender: TObject);
procedure btnHitungClick(Sender: TObject);
procedure btnCreateClick(Sender: TObject);
procedure btnUseDefaultClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure edtNLoopChange(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  frmHasil: TfrmHasil;

implementation

uses uUtils, uLoadflow, uNewtonRaphson, uComplex, uTopology,
    uRecursive, uMatrix, uFitness;

{$R *.dfm}

var Nsal: integer;
    CostSebelum, CostSesudah: double;

```

```

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

procedure TfrmHasil.btnHitungClick(Sender: TObject);
var ia,ja,i,Nbus:integer;
    mulai,selesai,selang:TDateTime;
    jam,menit,detik,mdetik:word;
    V,Sg,SL:CxArr1;
    Cap:dArr1;
    TypBus:iArr1;
    Z,Ip,Alir,Arus:CxArr2;
    Lc,Tr:dArr2;
begin
  DecodeCommDataToLFData(gBus,Nbus,Nsal,V,Sg,SL,Cap,TypBus,gBranch,
  Z,Ip,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;
  if Nbus=14 then
  begin
    Z[2,8]:=Cmplx(0,0);
    Z[4,13]:=Cmplx(0,0);
    Z[7,11]:=Cmplx(0,0);
  end;
  btnUseDefault.Enabled:=true;
  mulai:=time;
  NewtonRaphson(gParamLF,V,Sg,SL,Cap,TypBus,Z,Ip,Alir,Arus,Lc,Tr);
  selesai:=time;
  selang:=selesai-mulai;
  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);
    fgBus.Cells[5,i+1]:=RealToStr(SL[i].real,3);
    fgBus.Cells[6,i+1]:=RealToStr(SI[i].imag,3);
    fgBus.Cells[7,i+1]:=RealToStr(Cap[i],3);
    fgBus.Cells[8,i+1]:=IntToStr(TypBus[i]);
  end;
end;

```

```

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);
edtSelisih.Text:=FormatFloat('#,##0.000',(CostSebelum-CostSesudah));
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;

procedure TfrmHasil.btnUseDefaultClick(Sender: TObject);
begin
    edtNLoop.Text:='4';
    fgOS.RowCount:=StrToInt(edtNLoop.Text)+1;
    btnCreate.Enabled:=true;
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;

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] := 'absV (pu)';
    frmHasil.fgBus.Cells[2,0] := 'sudV (deg)';
    frmHasil.fgBus.Cells[7,0] := 'Sup (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';
  end;
end;

```

```

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

```

```

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

end.

```

```

unit uRecursive;

interface

uses uUtils,uComplex,uloadflow;

function InitSwith(const rZb:CxArr2):TSwithArr1;
function BuatZBaru(const rZ:CxArr2;
    const rSwith:TSwithArr1):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:TSwithArr1;

implementation

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

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;
            end;
        end;
    end;
end;

```

```

    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:hArr1):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;  
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;  
    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;  
  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;  
  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[kc,i]=1 then
  begin
    if i<>dari then
    begin
      doRecursive(kc,i);
      if findLoop=true then
      begin
        exit;
      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;

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 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;
        FKp,FCostSwith:double;
        FVb:TBatas;
        FV,FSg,FSL:CxArr1;
        FCap:dArr1;
        FTypBus:iArr1;
        FZ,FTp:CxArr2;
        FLc,FTr:dArr2;
        FSwith:TSwithArr1;
        FBestIndi,FSecondIndi:TIndividu;
        function getIndividu(const rIndi:TIndividu):TIndividu;
        function CekStopTreeSearch(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 CalcVPinalty(const rV:CxArr1):integer;
        function CalcFitness(const rData:iArr1):double;
        function getChrom:iArr1;
    public
        constructor Create(const rNLoop:integer;
            const rKp,rCostSwith:double);
        procedure CreateTreeSearch(var rList:TStringList;
            var rCount:integer);
        function GetSwith(const rData:iArr1):TSwithArr1;
        function DecodeStrToData(const rStr:string):iArr1;
        function DecodeDataToStr(const rData:iArr1):string;
        property BestChrom:iArr1 read getChrom;
    end;

implementation

constructor TFitness.Create(const rNLoop:integer;

```

```

    const rKp,rCostSwith:double);
begin
    inherited Create;
    FNLoop:=rNLoop;
    FKp:=rKp;
    FCostSwith:=rCostSwith;
    DecodeCommDataToLFData(gBus,FNbus,FNsal,FV,FSg,FSL,FCap,FTypBus,
    gBranch,FZ,FTp,FLc,FTr);
    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;
end;

```

```

function TFitness.CekStopTreeSearch(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;
end;

```

```
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;  
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;  
  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;  
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;
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;
end;

```

```

function TFitness.CalcFitness(const rData:iArr1):double;
var Vpin:integer;
V,Sg:CxArr1;
Zb,Alir,Arus:CxArr2;
sw:TSwithArr1;
begin

```