

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



**ANALISIS REKONFIGURASI JARINGAN UNTUK
MEREDUKSI RUGI-RUGI SISTEM DISTRIBUSI TIGA FASA
DENGAN MENGGUNAKAN METODE *SIMULATED
ANNEALING* DI GI BLIMBING MALANG**

SKRIPSI

Disusun Oleh :

DYAH AYU ANGGRAENI

NIM : 99.12.027

MARET 2005

REKAM JEKUTAN DAN KEMERDEKAAN
REKAM JEKUTAN DAN KEMERDEKAAN
REKAM JEKUTAN DAN KEMERDEKAAN
REKAM JEKUTAN DAN KEMERDEKAAN

REKAM JEKUTAN DAN KEMERDEKAAN
REKAM JEKUTAN DAN KEMERDEKAAN
REKAM JEKUTAN DAN KEMERDEKAAN
REKAM JEKUTAN DAN KEMERDEKAAN

REKAM JEKUTAN

REKAM JEKUTAN
REKAM JEKUTAN
REKAM JEKUTAN

REKAM JEKUTAN

LEMBAR PERSETUJUAN

**ANALISIS REKONFIGURASI JARINGAN UNTUK MEREDUKSI
RUGI-RUGI SISTEM DISTRIBUSI TIGA FASA DENGAN
MENGUNAKAN METODE *SIMULATED ANNEALING*
DI GI BLIMBING MALANG**

SKRIPSI

*Disusun dan Diajukan Untuk Melengkapi dan Memenuhi Syarat
Guna Mencapai Gelar Sarjana Teknik Elektro S-1*

Disusun Oleh :

**DYAH AYU ANGGRAENI
NIM. 99.12.027**



**Mengetahui,
Ketua Jurusan Teknik Elektro S-1**

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

**Menyetujui,
Dosen Pembimbing**

**(Ir. Taufik Hidayat, MT)
NIP. P. 108 700 151**

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

LEMBAR PERSEMBAHAN

*Kebaikan Dunia dan Akherat beserta Ilmu
dan Keburukan Dunia dan Akherat beserta Kebodohan
(H.R. Bukhori)*

Aku percaya bahwa segala sesuatu yang terjadi padaku hanya atas kehendak-Nya, karenanya rasa terima kasihku yang tertinggi kutujukan kepada Allah SWT atas segala nikmat yang telah dianugerahkan-Nya kepadaku.

Special thanks to my beloved Mum and Dad yang telah memberikan doa dan dukungan materil kepadaku selama kuliah sampai selesai mengerjakan skripsi ini.

Buat yayank ku (Aris Budi Sasono) yang terchayank, makasih buat semangat dan dukungannya sampai terselesaikannya skripsi ini. Makasih ya udah mau nemenin aku kompre sampe' jam 12 malem.

Untuk Mas Ugro, makasih ya programnya. Makasih juga buat Arif, Ubet, Farid Tewur, Soleh atas bantuan komputer 'n printernya serta dukungannya juga dan buat temen-temenku khususnya anak ST 1-2 yang terlalu banyak untuk disebutin atu-atu thanks ya.

KATA PENGANTAR

Dengan segala kerendahan hati, penulis mengucapkan rasa syukur kepada Allah SWT., karena hanya dengan lindungan, rahmat dan karunia-Nya penulis telah selesai menyusun skripsi ini dengan baik. Serta tidak lupa penulis panjatkan syukur kepada Nabi Muhammad SAW., yang telah membawa kita dari *Zaman Jahiliyah* ke *Zaman yang Terang Benderang*.

Adapun Skripsi ini dengan judul “**Analisis Rekonfigurasi Jaringan Untuk Mereduksi Rugi-rugi Sistem Distribusi Tiga Fasa Dengan Menggunakan Metode *Simulated Annealing* Di GI Blimbing Malang**”.

Dengan ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah membantu dan memberi dorongan baik secara *materiil* dan *spiritual*, sehingga Skripsi ini dapat terselesaikan tepat pada waktunya, terutama kepada:

1. Bapak Dr. Ir. Abraham Lomi, MSEE., selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. I Wayan Sujana, MT., selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang.
3. Bapak Ir. F. Yudi Limpraptono, MT., selaku Ketua Jurusan Teknik Elektro Institut Teknologi Nasional Malang.
4. Bapak Ir. Taufik Hidayat, MT., selaku Dosen Pembimbing yang telah banyak memberikan bimbingan, saran dan pemikiran serta *support*-nya sehingga terselesaikannya skripsi ini.

5. Kedua orang tuaku yakni Bapak Drs. Sumarto dan Ibu Siti Chotimah, yang mana telah memberikan banyak dorongan baik secara *spiritual* ataupun *material* yang tidak henti-hentinya.
6. Kedua kakak dan kedua adikku, yang selalu memberikan dukungan dalam segala hal.
7. Rekan-rekan mahasiswa Elektro ITN pada umumnya dan Elektro Energi Listrik Angkatan 1999 Kelas 1-2 pada khususnya, yang telah membantu dalam penulisan skripsi ini.

Pada akhirnya penulis mengharapkan saran dari pembaca demi kesempurnaan dari skripsi ini, karena penulis menyadari masih banyak kekurangan di dalam skripsi ini. Dan penulis berharap semoga skripsi ini dapat bermanfaat di kemudian hari. Amin.

Malang, Maret 2005

Penulis

ABSTRAKSI

“ANALISIS REKONFIGURASI JARINGAN UNTUK MEREDUKSI RUGI-RUGI DISTRIBUSI TIGA FASA DENGAN MENGGUNAKAN METODE *SIMULATED ANNEALING* DI GI BLIMBING MALANG”

**(Dyah Ayu Anggraeni, 99.12.027, Teknik Elektro Energi Listrik S-1)
(Dosen Pembimbing : Ir.Taufik Hidayat, MT.)**

Kata kunci : *Simulated Annealing, Fast Decoupled, Rekonfigurasi.*

Pada jaringan sistem distribusi mempunyai dua macam *switch* yaitu *tie switch* dan *sectionalizing switch* untuk menentukan efisiensi dalam penyaluran tenaga listrik. Oleh karena itu dicari konfigurasi yang paling efisien, dimana konfigurasi tersebut dapat mengurangi rugi-rugi daya saluran pada sistem distribusi tiga fasa.

Solusi yang diusulkan untuk masalah ini didasarkan pada pengkombinasian algoritma optimasi umum-khusus yang dikenal dengan *Simulated Annealing*. Solusi tersebut memanfaatkan hukum *Kirchoff* tegangan dan arus. Dan selanjutnya dapat dituliskan sebagai persamaan aliran daya tiga fasa.

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PERSETUJUAN	ii
KATA PENGANTAR	iii
ABSTRAKSI	v
DAFTAR ISI	vi
DAFTAR GAMBAR	x
DAFTAR TABEL	xii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan Pembahasan	3
1.4. Batasan Masalah	3
1.5. Metode Pembahasan	4
1.6. Sistematika Pembahasan	4
1.7. Kontribusi Penelitian	5
BAB II SISTEM DISTRIBUSI TENAGA LISTRIK	6
2.1. Sistem Distribusi	6
2.1.1. Sistem Distribusi Primer	6
2.1.2. Sistem Distribusi Sekunder	7

2.2.	Struktur Jaringan Distribusi Radial	7
2.2.1.	Struktur Jaringan Radial	7
2.2.2.	Struktur Jaringan Loop	8
2.2.3.	Struktur Jaringan Mesh	9
2.3.	Macam-macam Struktur Jaringan Radial	10
2.4.	Daya Sistem Distribusi	11
2.4.1.	Daya Nyata (<i>Real Power</i>)	12
2.4.2.	Daya Reaktif (<i>Reactive Power</i>)	12
2.4.3.	Daya Komplek (<i>Complex Power</i>)	12
2.5.	Beban Sistem Distribusi	13
2.6.	Rugi-rugi Daya Saluran Distribusi	13

BAB III METODE REKONFIGURASI..... 15

3.1.	Analisis Aliran Daya	15
3.1.1.	Tujuan	15
3.1.2.	Klasifikasi Bus.....	16
3.1.3.	Penomoran Bus dan Lateral	17
3.1.4.	Pemodelan Sistem	18
3.1.4.1.	Pemodelan Beban.....	20
3.1.4.2.	Pemodelan Saluran.....	20
3.1.4.3.	Pemodelan Pemutus	21
3.1.5.	Feeder Tunggal	21
3.1.6.	Struktur Umum Jaringan Distribusi Radial	22

3.1.7. Metode Penyelesaian Studi Aliran Daya	24
3.1.8. Metode <i>Newton Raphson</i>	24
3.1.9. Metode <i>Fast Decoupled</i>	29
3.1.9.1. Algoritma <i>Fast Decoupled</i>	31
3.1.9.2. <i>Flowchart Fast Decoupled</i>	33
3.2. Rekonfigurasi Jaringan	34
3.2.1. Metode rekonfigurasi Jaringan	35
3.2.2. Metode Rekonfigurasi Jaringan Menggunakan <i>Simulated Annealing (SA)</i>	35
3.3. Teori Dasar <i>Simulated Annealing (SA)</i>	36
3.3.1. Pengenalan Konsep <i>Simulated Annealing (SA)</i>	36
3.3.2. Kombinasi Optimasi	37
3.3.3. Analogi Pada Pemanasan Fisik (<i>Physical Annealing</i>)	38
3.3.4. Algoritma <i>Simulated Annealing (SA)</i>	39
3.3.5. <i>Flowchart Simulated Annealing (SA)</i>	41

BAB IV ANALISA PENGGUNAAN METODE REKONFIGURASI

JARINGAN	42
4.1. Program Komputer Metode Rekonfigurasi Jaringan	42
4.2. Algoritma Program	42
4.3. <i>Flowchart</i> Program	43
4.4. Data Jaringan	44
4.5. Tampilan Program	48

4.6. Analisa Perhitungan	52
BAB V PENUTUP	57
5.1. Kesimpulan	57
5.2. Saran	58
DAFTAR PUSTAKA	59
LAMPIRAN-LAMPIRAN	60

DAFTAR GAMBAR

Gambar 2.1. Struktur Jaringan Radial	8
Gambar 2.2. Struktur Jaringan Loop	9
Gambar 2.3. Struktur Jaringan Mesh	10
Gambar 2.4. Rugi-rugi Pada Bagian Sistem Tenaga	14
Gambar 3.1. Contoh Penomoran Bus dan Lateral	18
Gambar 3.2. Blok Diagram Dasar Sistem Distribusi Radial	19
Gambar 3.3. Feeder Tunggal	21
Gambar 3.4. Perhitungan Perbedaan Tegangan	23
Gambar 3.5. Sistem Distribusi Sederhana	26
Gambar 3.6. <i>Flowchart Fast Decoupled</i>	34
Gambar 3.7. Diagram Skematik Tipe (a) NC Switch (b) NO Switch	35
Gambar 3.8. Jaringan Distribusi	36
Gambar 3.9. <i>Flowchart Simulated Annealing</i>	41
Gambar 4.1. <i>Flowchart Program</i>	43
Gambar 4.2. Konfigurasi Jaringan Radial Sebelum Rekonfigurasi	46
Gambar 4.3. Konfigurasi Jaringan Mesh.....	47
Gambar 4.4. Menu Utama	48
Gambar 4.5. Data General	48
Gambar 4.6. Data Saluran.....	49
Gambar 4.7. Data Pembebanan	49
Gambar 4.8. Sebelum Rekonfigurasi	50

Gambar 4.9. *Simulated Annealing* 50

Gambar 4.10. Hasil *Simulated Annealing* (SA)..... 51

Gambar 4.11. Sesudah Rekonfigurasi 51

Gambar 4.12. Konfigurasi Jaringan Setelah Rekonfigurasi 54

DAFTAR TABEL

Tabel 3.1. Persamaan Perbaikan	20
Tabel 4.1. Data Jaringan Distribusi	45
Tabel 4.2. Hasil Perhitungan Sebelum Rekonfigurasi	52
Tabel 4.3. Hasil Perhitungan Setelah Rekonfigurasi	55
Tabel 4.4. Perbandingan Rugi-rugi Daya	56

BAB I

PENDAHULUAN

1.1. Latar Belakang

Kemajuan teknologi yang cepat memberikan dampak yang cukup besar pada kebutuhan dan pola konsumsi masyarakat akan energi listrik. Keadaan tersebut berpengaruh pada pola dan sistem penyediaan energi listrik, terutama pada sisi penyediaan tenaga dan pengaturan pendistribusian energi listrik agar didapat suatu sistem yang baik.

Berangkat dari hal di atas, produsen energi listrik harus benar-benar memperhitungkan dan mengusahakan penyaluran energi listrik yang baik kepada konsumen mulai dari sistem transmisi tenaga sampai dengan sistem distribusi. Dalam hal ini yang perlu ditekankan adalah sistem distribusi, karena sistem distribusi berhubungan langsung dengan pemakai atau konsumen energi listrik.

Sistem distribusi merupakan sistem penyaluran energi listrik yang paling dekat dengan konsumen. Rugi-rugi daya dalam sistem distribusi merupakan masalah yang klasik dan selalu dicari cara untuk mereduksi rugi-rugi. Dalam skripsi ini saya mencoba untuk mereduksi rugi-rugi daya dengan mengatur ulang konfigurasi (rekonfigurasi) penyulang-penyulang dengan suatu metoda pengaturan tertentu melalui pengaturan membuka dan menutup pemutus daya (*switch*) pada penyulang sehingga didapatkan rugi-rugi yang seminimal mungkin.

Metode rekonfigurasi pada skripsi ini berdasarkan metode *Simulated Annealing* sedangkan metode aliran dayanya menggunakan metode *Fast*

Decoupled. Metode *Simulated Annealing* adalah metode yang didasarkan pada proses fisik pendinginan logam dengan pengotoran, dimana suatu partikel cair didinginkan pelan-pelan dari temperatur yang sangat tinggi sampai mengeras pada temperatur yang rendah.^[4]

1.2. Rumusan Masalah

Pada sistem distribusi yang besar dan kompleks serta dengan semakin bertambahnya beban pada jaringan distribusi akan timbul masalah didalam 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.

Pada skripsi ini akan dikaji pengaturan beban pada trafo atau penyulang jaringan sistem distribusi sehingga dapat meminimalkan rugi-rugi jaringan dengan menggunakan metode rekonfigurasi untuk mengevaluasi dan memilih keputusan yang optimal dari berbagai alternatif yang ada.

Dari permasalahan di atas, maka judul yang diangkat adalah:

“ANALISIS REKONFIGURASI JARINGAN UNTUK MEREDUKSI RUGI-RUGI SISTEM DISTRIBUSI TIGA PHASA MENGGUNAKAN METODE *SIMULATED ANNEALING* DI GI BLIMBING-MALANG”

1.3. Tujuan Pembahasan

Tujuan dari penulisan skripsi ini adalah:

1. Untuk merekonfigurasi jaringan yaitu dengan cara mengubah status *switch*.
2. Untuk menghitung daya aktif dan daya reaktif sebelum dan sesudah rekonfigurasi.
3. Untuk menghitung rugi-rugi daya aktif dan rugi-rugi daya reaktif sebelum dan sesudah rekonfigurasi.

1.4. Batasan Masalah

Adapun batasan masalah dari skripsi ini adalah:

1. Tidak membahas koordinasi pengaman jaringan listrik.
2. Hanya membahas rugi-rugi daya pada saluran.
3. Tidak membahas penyebab terjadinya gangguan.
4. Tidak membahas cara pengontrolan *swithing* yang dilakukan.
5. Metode studi aliran daya yang digunakan adalah *Fast Decoupled*.
6. Metode rekonfigurasi yang digunakan adalah *Simulated Annealing*.
7. Tidak membahas optimasi pemilihan rekonfigurasi jaringan.
8. Pembagian seksi berdasarkan adanya *NO Switch* pada *single line* GI Blimbing Malang.
9. Pengambilan data dilakukan pada PT. PLN (Persero) Distribusi Jawa Timur Cabang Malang.

1.5. Metode Pembahasan

Metode yang digunakan dalam penulisan skripsi ini adalah:

1. Metode literatur, yaitu dengan mempelajari hal-hal yang sesuai dengan 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 menggunakan metode yang telah ditentukan.
4. Penyusunan buku.

1.6. Sistematika Pembahasan

Bab I : Pendahuluan

Pada bab ini meliputi latar belakang, permasalahan, batasan masalah, tujuan yang akan dicapai dalam skripsi ini, metode pembahasan, sistematika pembahasan, dan kontribusi penelitian.

Bab II : Sistem Distribusi Tenaga Listrik

Bab ini menjelaskan tentang sistem dasar tenaga listrik terutama sistem distribusi bertegangan AC yang mempunyai tipe-tipe saluran daya tertentu menurut kebutuhan bebannya.

Bab III : Metode Rekonfigurasi Jaringan

Bab ini membahas pengenalan awal rekonfigurasi jaringan dalam upaya untuk mengurangi rugi-rugi daya pada sistem distribusi radial.

Bab IV: Analisa Rekonfigurasi Jaringan

Bab ini memuat pembahasan mengenai permasalahan rekonfigurasi jaringan dengan menggunakan metode *Simulated Annealing* dan hasil simulasinya.

Bab V : Penutup

Dalam bab ini akan diuraikan kesimpulan yang dapat diambil dari hasil pembahasan pada bab-bab sebelumnya.

1.7. Kontribusi Penelitian

Dalam skripsi ini diharapkan dapat memberikan informasi mengenai analisa penyelesaian persoalan rekonfigurasi jaringan sistem distribusi primer untuk meminimalisasi rugi daya pada jaringan dengan menggunakan metode *Simulated Annaling*. Diharapkan dengan metode penyelesaian rekonfigurasi jaringan memperoleh hasil yang paling optimum dan akhirnya dapat diaplikasikan pada instansi-instansi yang terkait dengan sistem penyaluran energi listrik.

BAB II

SISTEM DISTRIBUSI TENAGA LISTRIK

2.1. Sistem Distribusi^[1]

Jaringan setelah keluar dari Gardu Induk (GI) biasanya disebut dengan jaringan distribusi. Setelah tenaga listrik disalurkan melalui jaringan distribusi primer maka kemudian tenaga listrik diturunkan tegangannya dalam gardu-gardu distribusi sehingga menjadi tegangan rendah. Setelah itu tenaga listrik disalurkan melalui JTR untuk selanjutnya disalurkan ke rumah-rumah pelanggan (konsumen) PLN melalui sambungan rumah.

Dalam pendistribusian tenaga listrik ke konsumen tegangan yang digunakan bervariasi tergantung dari jenis konsumen yang membutuhkan. Untuk konsumen industri biasanya digunakan tegangan menengah 20 kV sedangkan untuk konsumen perumahan digunakan tegangan 220/380 V yang merupakan tegangan siap pakai untuk peralatan-peralatan rumah tangga. Dengan demikian berdasarkan tingkat tegangan distribusi maka sistem distribusi tenaga listrik dapat diklasifikasikan menjadi 2 (dua) bagian sistem, yaitu:

1. Sistem Distribusi Primer
2. Sistem Distribusi Sekunder

2.1.1. Sistem Distribusi Primer^[1]

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

2.1.2. Sistem Distribusi Sekunder^[1]

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

Sistem jaringan yang digunakan untuk menyalurkan dan mendistribusikan tenaga listrik tersebut dapat menggunakan sistem 1 (satu) fasa dengan 2 (dua) kawat maupun sistem 3 (tiga) fasa dengan 4 (empat) kawat.

2.2. Struktur Jaringan Distribusi Tenaga Listrik^[1]

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

1. Sistem Jaringan Distribusi Radial
2. Sistem Jaringan Distribusi Rangkaian Tertutup (Loop)
3. Sistem Jaringan Distribusi Mesh

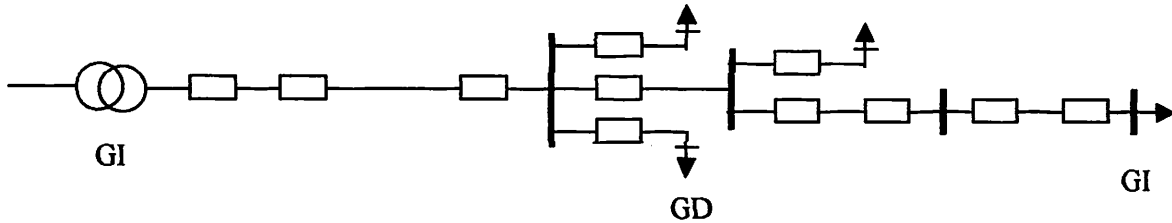
2.2.1. Struktur Jaringan Radial^[1]

Bentuk jaringan ini merupakan bentuk dasar yang paling sederhana dan paling banyak digunakan. Sistem ini dikatakan radial karena dalam kenyataannya jaringan ini ditarik secara radial dari GI ke pusat-pusat beban atau konsumen yang dilayaninya. Sistem ini terdiri dari saluran utama (*tunk line*) dan saluran cabang (*lateral*).

Pelayanan tenaga listrik untuk suatu daerah beban tertentu dilaksanakan dengan memasang transformator pada sambungan titik pada jaringan yang sedekat mungkin dengan daerah beban yang dilayani. Untuk daerah beban yang

menyimpang jauh dari saluran utama maupun saluran cabang, maka akan ditarik lagi saluran tambahan yang dicabangkan pada saluran tersebut.

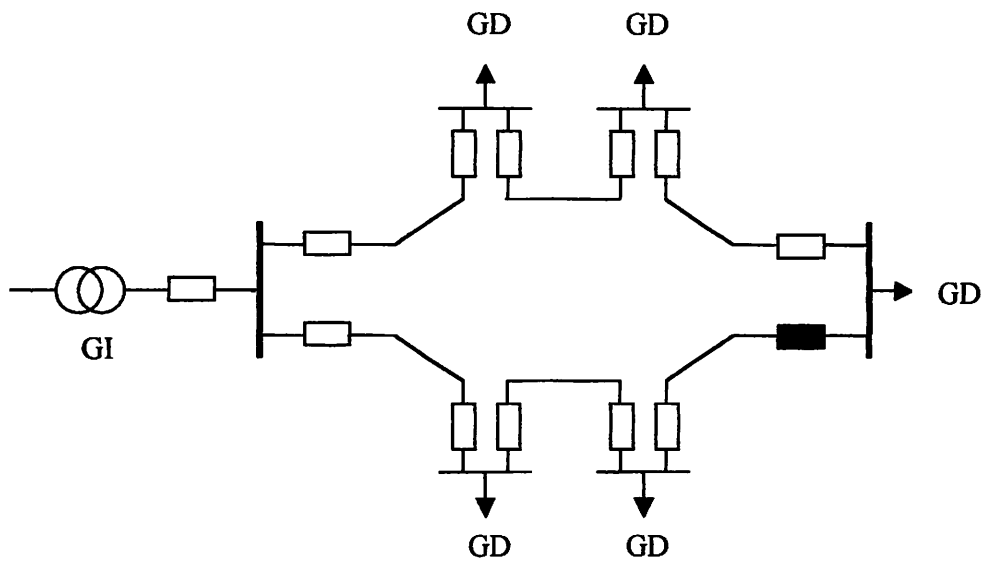
Kelemahan yang dimiliki oleh sistem radial adalah jatuh tegangan yang cukup besar dan bila terjadi gangguan pada sistem akan mengakibatkan jatuhnya tegangan sebagian atau bahkan keseluruhan beban sistem.



Gambar 2.1. Struktur Jaringan Radial

2.2.2. Struktur Jaringan Loop^[1]

Struktur jaringan loop merupakan gabungan dari 2 (dua) sistem jaringan radial, dimana pada ujung kedua jaringan dipasang sebuah pemutus (CB) atau pemisah (DS). Pada saat terjadi gangguan dapat diisolir, maka pemisah atau pemutus ditutup sehingga aliran daya listrik ke bagian yang tidak terkena gangguan tidak berhenti. Dalam kondisi normal, struktur jaringan loop ini merupakan 2 (dua) struktur jaringan radial. Struktur jaringan ini mempunyai keandalan yang cukup, sehingga biaya pembangunannya relatif lebih mahal dibandingkan dengan biaya pembangunan struktur jaringan radial.

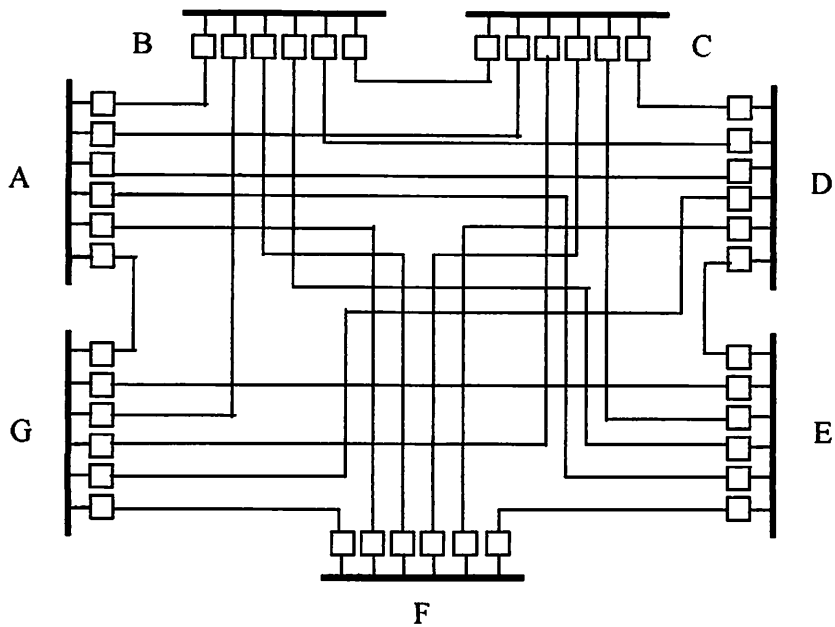


Keterangan: = Normally Open
 = Normally Closed

Gambar 2.2. Struktur Jaringan Loop

2.2.3. Struktur Jaringan Mesh^[1]

Struktur jaringan ini merupakan kombinasi antara sistem jaringan radial dengan sistem jaringan loop. Titik beban memiliki lebih banyak alternatif penyaluran, sehingga bila salah satu penyulang terganggu maka dengan segera dapat digantikan oleh penyulang yang lain. Dengan demikian kontinuitas penyaluran daya sangat terjamin.



Gambar 2.3. Struktur Jaringan Mesh

2.3. Macam-macam Struktur Jaringan Radial^[1]

Jaringan distribusi radial mempunyai keuntungan dan kelemahan, sebagai berikut:

- Keuntungan:
 1. Bentuknya sederhana dibandingkan dengan bentuk yang lain.
 2. Biaya investasinya relatif lebih murah, karena saluran menuju ke tiap beban hanya tersedia 1 (satu) jalur.
 3. Mudah dikembangkan.

- Kelemahan:

1. Kualitas pelayanan (penyaluran daya) dibandingkan 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 1 (satu) saluran, maka bila terjadi gangguan pada saluran tersebut tidak dapat diharapkan *supply* melalui saluran lain.

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

1. Struktur jaringan pohon
2. Struktur jaringan dengan tie dan switch pemisah
3. Struktur jaringan dengan pusat beban
4. Struktur jaringan dengan pembagian daerah phasa (phasa area)

2.4. Daya Sistem Distribusi^[1]

Daya merupakan banyaknya perubahan tenaga terhadap waktu dalam besaran tegangan dan arus. Daya dalam Watt yang diserap oleh suatu beban pada setiap saat adalah hasil kali jatuh tegangan sesaat diantara beban dalam Volt dengan arus sesaat yang mengalir dalam beban tersebut dalam Ampere.

Guna keperluan analisa, daya dalam sirkuit arus bolak-balik, dirinci lagi sesuai tipe dari daya tersebut.

2.4.1. Daya Nyata (*Real Power*)^[1]

Daya nyata dinyatakan dalam persamaan:

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

Daya nyata untuk beban tiga fasa seimbang:

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

2.4.2. Daya Reaktif (*Reactive Power*)^[1]

Daya reaktif dinyatakan dalam persamaan:

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

Daya reaktif untuk beban tiga fasa seimbang:

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

2.4.3. Daya Kompleks (*Complex Power*)^[1]

Daya kompleks dinyatakan dalam persamaan:

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

Daya kompleks untuk beban tiga fasa seimbang:

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

Persamaan di atas berlaku tanpa memandang apakah beban dihubungkan secara segitiga atau bintang.

2.5. Beban Sistem Distribusi^[1]

Secara umum beban dapat diklasifikasikan menjadi:

1. Beban Perumahan (Rumah Tangga)

Beban perumahan berupa penerangan, kipas angin, televisi, alat-alat rumah tangga, dll.

2. Beban Komersial

Beban komersial pada umumnya berupa penerangan toko, reklame, dll.

3. Beban industri

Beban industri diklasifikasikan menjadi:

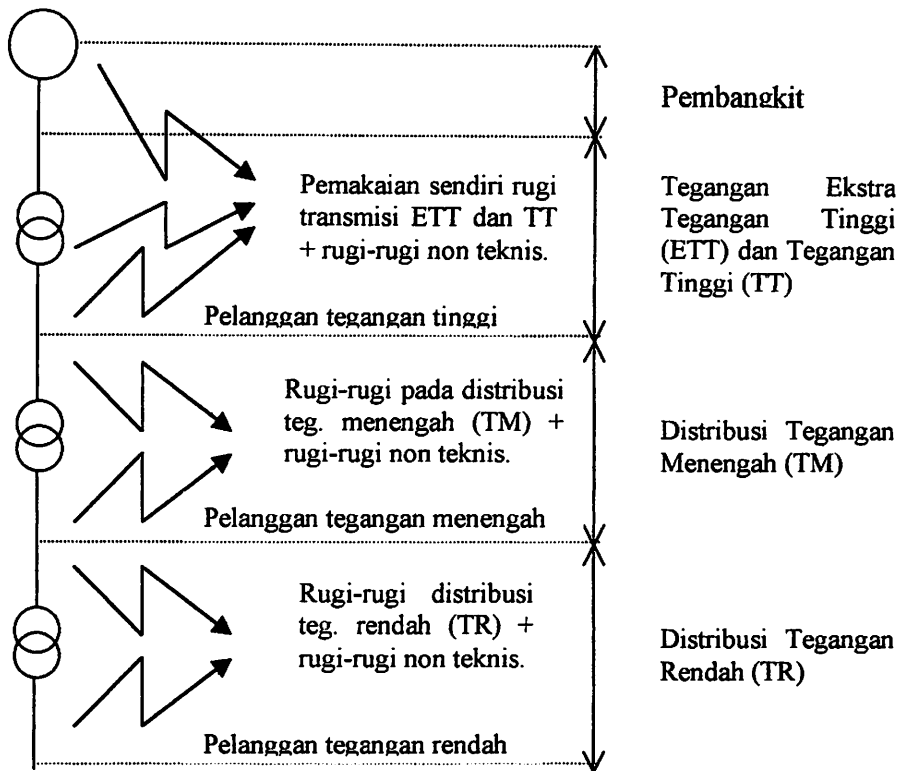
- skala kecil
- skala menengah
- skala besar

2.6. Rugi-rugi Daya Saluran Distribusi^[1]

Rugi daya adalah besar daya yang hilang dalam penyaluran daya elektrik. Dalam proses penyaluran energi listrik ke pelanggan terjadi rugi-rugi teknis, yaitu rugi daya dan rugi energi dimulai dari Pembangkit, Saluran Transmisi, dan Saluran Distribusi. Rugi-rugi teknis adalah pada penghantar saluran, adanya tahanan dari penghantar yang dialiri arus maka timbullah rugi-rugi teknis ($I^2.R$) pada jaringan tersebut.

$$P_{loss} = I^2 \cdot Z \quad (2.7)$$

Rugi teknis dari jaringan tenaga listrik tergantung dari macam pembebanan pada saluran tersebut (beban merata, beban terpusat), seperti terlihat pada gambar 2.4.



Gambar 2.4. Rugi-rugi Pada Bagian Sistem Tenaga

Keterangan:

- : aliran daya dan energi
- : pembangkit
- ⊗ : trafo
- ⚡ : rugi-rugi

BAB III

METODE REKONFIGURASI

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 *Fast Decoupled*.

3.1. Analisis Aliran Daya^[2]

3.1.1. Tujuan^[2]

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 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, yaitu:

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 konsumen.

3.1.2. Klasifikasi Bus^[2]

Tujuan aliran daya pada sistem tenaga listrik adalah untuk menghitung besar (magnitude) tegangan $|V|$ dan sudut phasa 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 phasa 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 (*load bus*)

Bus beban/*load bus* adalah suatu bus yang hanya ada penyerapan daya. Pada bus ini daya nyata P dan daya reaktif Q diketahui, sementara magnitude tegangan $|V|$ dan sudut phasa δ dicari (dihitung).

2. Bus generator (*generator bus*)

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

3. Bus berayun (*slack bus*)

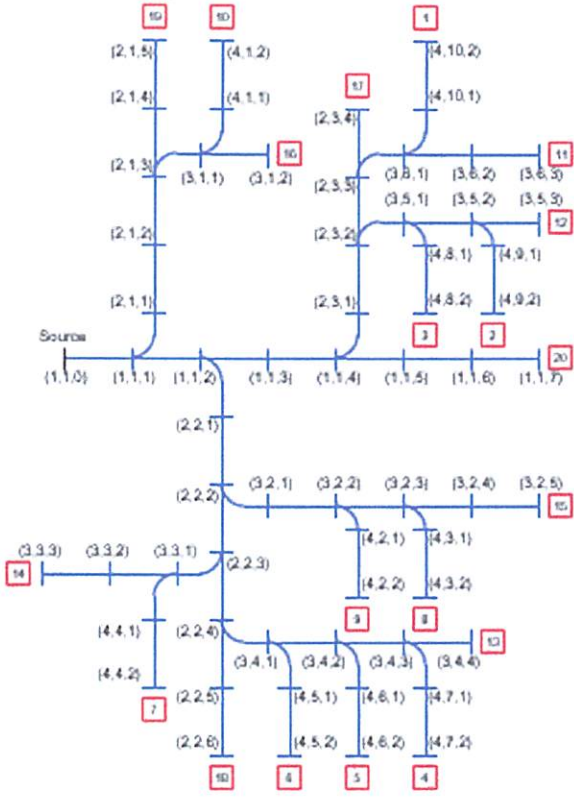
Pada bus ini magnitude tegangan $|V|$ dan sudut fasa δ diketahui besarnya, sementara daya nyata P dan daya reaktif Q dicari. Sedangkan sudut fasa tegangan berharga nol, karena pada bus ini fasor tegangan dari bus dipakai sebagai referensi.

3.1.3. Penomoran Bus dan Lateral^[3]

Sebuah sistem radial dapat dianggap sebagai sebuah feeder utama dengan lateral-lateral. Lateral ini mungkin masih mempunyai sub-lateral, dimana lateral tadi mungkin masih mempunyai sub-lateral-sub-lateral dan seterusnya. Jadi pertama didefinisikan bahwa level lateral i adalah jumlah lateral-lateral yang akan ditelusuri dari ujung lateral i sampai ke sumber. Sebagai contoh feeder utama didefinisikan sebagai level 1, sub lateralnya adalah level 2, sub-sub-lateral dari sub-lateral tadi adalah level 3 dan seterusnya.

Lateral-lateral dalam level l diindeksasi menurut urutan yang ditemui selama penelusuran jaringan. Tiap-tiap lateral didefinisikan sebagai pasangan terurut/ordered pair (l, m) dimana l adalah level lateral dan m adalah indeks lateral di dalam level l . Bus-bus juga diindeksasi dalam masing-masing lateral sehingga setiap bus secara unik diidentifikasi sebagai *triple-terurut/ordered triple* (l, m, n) dimana n adalah indeks dari bus. Jadi (l, m, n) menunjuk pada bus ke- n , indeks ke- m pada lateral level ke- l sedangkan bus sumber diberikan notasi indeks $(1, 0,$

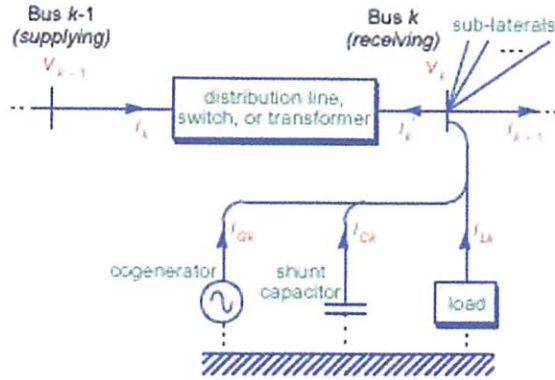
0). Gambar 3.1 menunjukkan sebuah contoh indeksasi pada penyusunan pembalikan pertama-muncul (*reverse breadth-first/RBF*). Angka di dalam kotak menunjukkan penyusunan lateral secara *RBF* yaitu dengan cara menyusun lateral dengan urutan terbalik, pertama dari levelnya kemudian dari indeks lateralnya.



Gambar 3.1. Contoh Penomoran Bus dan Lateral

3.1.4. Pemodelan Sistem

Untuk tujuan studi aliran daya, sistem distribusi radial dimodelkan sebagai bus-bus terhubung dengan saluran distribusi, pemutus-pemutus atau transformator-transformator ke tegangan sumber yang telah diketahui. Pada setiap bus mungkin juga terdapat beban, kapasitor shunt, dan cogenerator. Model tersebut dapat direpresentasikan seperti gambar 3.2.



Gambar 3.2. Blok Diagram Dasar Sistem Distribusi Radial

Konsep dasar dari rumusan ini adalah bahwa tegangan dan arus pada suatu bus dapat dinyatakan sebagai sebuah fungsi dari tegangan dan arus pada bus berikutnya. Jika dinyatakan bahwa:

$$w_k = \begin{bmatrix} V_k \\ I_{k+1} \end{bmatrix} \quad (3.1)$$

maka untuk bus sebelumnya dapat dinyatakan sebagai:

$$w_{k-1} = g_k(w_k) \quad (3.2)$$

fungsi g_k ditentukan oleh sub-lateral yang keluar dari bus k dan juga model untuk saluran distribusi, pemutus, transformator, beban, kapasitor shunt dan cogenerator.

Dari nilai V_k maka dapat dihitung arus yang diinjeksikan oleh beban dari persamaan (3.1). Dengan I_{k+1} dan arus-arus I_j yang diinjeksikan ke dalam sub-lateral yang keluar dari bus k didapatkan:

$$I_k' = IG_k + IC_k + IL_k - I_{k+1} - \sum_{j \in A_k} I_j \quad (3.3)$$

dimana A_k adalah bus-bus yang berdekatan dengan bus k pada sub-lateral yang keluar dari bus k .

Jadi untuk setiap bus, tegangan dan arus pada bus $k-i$ dapat dinyatakan sebagai fungsi tegangan dan arus pada bus k dalam bentuk persamaan (3.2) menggunakan persamaan (3.3) dan persamaan yang bersesuaian dari tabel 3.1.

TABEL 3.1
PERSAMAAN PERBAIKAN

Elemen	Persamaan Perbaikan
Beban	$I_{Lk} = (\bar{S}_{Lk} / V_k)^*$ (3.4)
Saluran	$V_{k-1} = V_k + Z_k \left(\frac{1}{2} Y_k V_k - I_k \right)$ (3.5)
	$I_k = \frac{1}{2} Y_k (V_k + V_{k-1}) - I_k$ (3.6)
Pemutus	$V_{k-1} = V_k$ (3.7)
	$I_k = -I_k$ (3.8)

3.1.4.1. Pemodelan Beban^[3]

Pemodelan beban yang digunakan adalah pemodelan beban secara umum dimana masing-masing beban dapat terhubung bintang atau segitiga. Arus injeksi untuk beban terhubung dihitung dengan persamaan (3.4).

3.1.4.2. Pemodelan Saluran^[3]

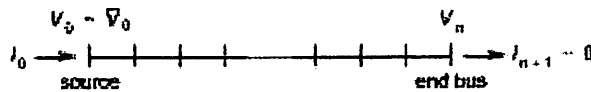
Pemodelan saluran yang digunakan adalah model π . Impedansi saluran distribusi k digambarkan sebagai impedansi Z_k dan efek *line charging* dibagi antara dua lengan shunt, masing-masing dengan sebuah admitansi $Y_k/2$. Tegangan dan arus pada ujung kirim saluran dihitung melalui persamaan (3.5) dan (3.6).

3.1.4.3. Pemodelan Pemutus^[3]

Pemutus/pemisah dimodelkan sebagai saluran dengan impedansi nol. Nilai tegangan dan arus diberikan oleh persamaan (3.7) dan (3.8).

3.1.5. Feeder Tunggal^[3]

Dengan diketahuinya tegangan dan arus pada bus ujung feeder atau bus sumber kita dapat menghitung tegangan dan arus pada bus-bus lainnya. Ada dua kondisi tertentu dalam hal ini, arus I_{n+1} pada ujung feeder adalah sama dengan nol dan tegangan V_0 pada sumber adalah tegangan V_0 konstan yang telah ditentukan. Jadi dapat diambil variabel I_0 pada sumber atau tegangan V_0 pada ujung feeder sebagai variabel yang belum diketahui. Tidak seperti I_0 , asumsi awal untuk V_n sudah ada tanpa perhitungan (1 pu seimbang) sehingga V_n yaitu tegangan ujung feeder dipilih sebagai variabel independen.



Gambar 3.3. Feeder Tunggal

Dengan menggunakan persamaan (3.2) dan fakta bahwa I_{n+1} sama dengan nol, maka tegangan sumber (dan arus sumber) dapat dinyatakan sebagai fungsi V_n .

$$w_{n-1} = g_n(w_n) = g_n\left(\begin{bmatrix} V_n \\ 0 \end{bmatrix}\right) \quad (3.9)$$

$$w_{n-2} = g_n(w_{n-1}) = g_{n-1}\left(\begin{bmatrix} V_n \\ 0 \end{bmatrix}\right) \quad (3.10)$$

$$w_0 = \begin{bmatrix} V_0 \\ I_1 \end{bmatrix} = g_1(w_1) = g_1 \cdot \dots \cdot g_{n-1} \cdot g_n\left(\begin{bmatrix} V_n \\ 0 \end{bmatrix}\right) \quad (3.11)$$

$$w_{n-2} = g_n(w_{n-1}) = g_{n-1} \left(\begin{bmatrix} V_n \\ 0 \end{bmatrix} \right) \quad (3.10)$$

$$w_0 = \begin{bmatrix} V_0 \\ I_1 \end{bmatrix} = g_1(w_1) = g_1 \bullet \dots \bullet g_{n-1} \bullet g_n \left(\begin{bmatrix} V_n \\ 0 \end{bmatrix} \right) \quad (3.11)$$

Apabila bagian variabel tegangan pada fungsi gabungan $g_1 \bullet \dots \bullet g_{n-1} \bullet g_n$ ditentukan sebagai V_0 maka perbedaan tegangan (*voltage mismatch*) antara tegangan sumber yang diketahui V_0 dan tegangan sumber yang dihitung \tilde{V}_0 (V_n) harus sama dengan nol. Persamaan aliran dayanya dapat ditentukan sebagai berikut:

$$f(V_n) = \tilde{V}_0(V_n) - \bar{V}_0 = 0 \quad (3.12)$$

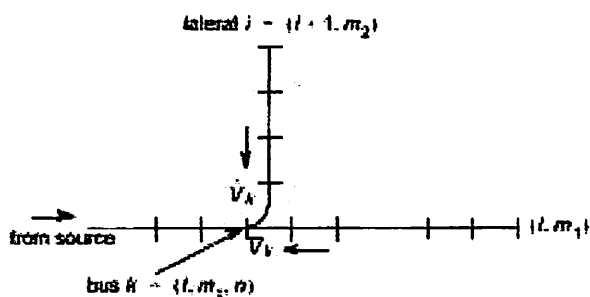
Cabang *Jacobian* G_k untuk cabang bus *incoming* k adalah *Jacobian* dari persamaan (3.2).

$$G_k = \frac{\partial g_k}{\partial W_k} = \begin{bmatrix} \left(\frac{\partial V_{k-1}}{\partial V_k} \right) & \left(\frac{\partial V_{k-1}}{\partial I_{k+1}} \right) \\ \left(\frac{\partial I_k}{\partial V_k} \right) & \left(\frac{\partial I_k}{\partial I_{k+1}} \right) \end{bmatrix} \quad (3.13)$$

3.1.6. Struktur Umum Jaringan Distribusi Radial^[3]

Untuk menyederhanakan keseluruhan rumusan dari struktur jaringan radial maka diambil pedoman bahwa tegangan pada setiap permulaan lateral dapat dihitung sebagai fungsi dari tegangan ujung akhir lateral jika arus-arus yang diinjeksikan ke dalam setiap sub-lateral diketahui. Di dalam system distribusi radial dengan level L , lateral level L tidak mempunyai sub-lateral sehingga lateral inilah yang dihitung pertama kali. Setelah semua lateral L dihitung, arus-arus yang

Untuk menyelesaikan perhitungan perbedaan tegangan (*voltage mismatch*) untuk setiap lateral, maka diambil perbedaan tegangan yang telah dihitung untuk masing-masing bus cabang. Misalnya lateral (l, m, n) mempunyai sub-lateral i level $l+1$ yang keluar dari bus k , dimana i adalah pasangan terurut $(l+1, m_2)$ dan k adalah *triple* terurut (l, m_1, n) . Perbedaan tegangan untuk lateral i dinyatakan sebagai perbedaan tegangan antara V_k yang dihitung dari tegangan ujung dari sub-lateral, dan V_k yang dihitung dari tegangan ujung dari lateral penyuplai (l, m_1) .



Gambar 3.4. Perhitungan |Perbedaan Tegangan

Untuk lateral i yang dimulai dari bus k , ditemukan persamaan yang serupa dengan persamaan (3.12), tetapi dengan V_n diganti dengan vektor i yang merupakan tegangan ujung dari semua lateral

$$F_1(x) = \tilde{V}_k(x) - \bar{V}_k(x) = 0 \quad (3.14)$$

Fungsi f_i tidak tergantung pada semua elemen dari x , tetapi hanya tegangan ujung bus yang disuplai lewat bus k karena tegangan ujung tersebut mempengaruhi arus yang diinjeksikan ke dalam sub-lateral dari lateral i .

Dengan menempatkan persamaan ini bersama dalam susunan *RBF* kita dapat menuliskan persamaan aliran daya yang baru sebagai:

$$F(x) = 0 \quad (3.15)$$

sebagai catatan bahwa $\bar{V}_k(x)$ tidak lagi bernilai konstan kecuali jika i adalah feeder/penyulang utama dan k adalah bus sumber (*source bus*) yaitu persamaan terakhir di dalam susunan *RBF*.

3.1.7. Metode Penyelesaian Studi Aliran Daya^[3]

Dalam studi aliran daya, beberapa teknik yang umum digunakan untuk menyelesaikan persoalan-persoalan aliran daya adalah:

1. Metode *Newton Raphson*
2. Metode *Fast Decoupled*

Metode yang ditinjau disini adalah metode *Fast Decoupled* yang penurunan matrik *Jacobiamya* diturunkan dari metode *Newton Raphson*. Dengan asumsi dan pendekatan yang digunakan tersebut diharapkan metode *Fast Decoupled* lebih efisien didalam perhitungan perbaikan (*up-date*).

3.1.8. Metode Newton Raphson^[3]

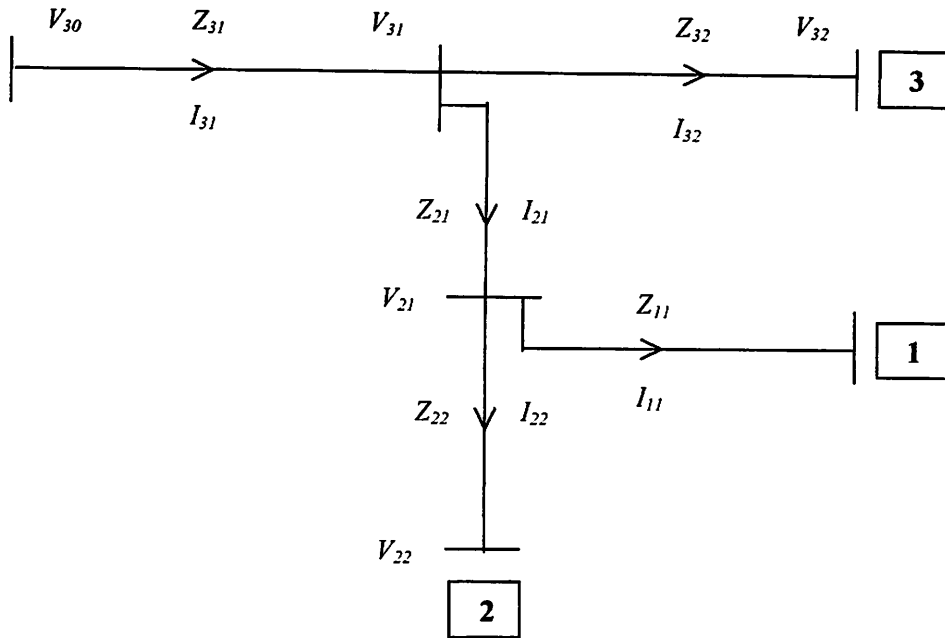
Persamaan aliran daya dapat diselesaikan dengan metode iterasi untuk mencari nilai nol dari persamaan non-linier. Prosedur metode *Newton Raphson* yang sudah dikenal luas untuk pemecahan persamaan non-linier adalah sebagai berikut:

1. Pilih sebuah nilai awal untuk x_0 .
2. Tetapkan $k = 0$.
3. Evaluasi nilai $Fk = F(xk)$.
4. Berhenti jika $\|Fk\| \leq \text{toleransi}$.

5. Evaluasi nilai Jacobian, $Jk = \left. \frac{\partial F}{\partial x} \right|$.
6. Hitung $Jk sk = - Fk$.
7. $x_{k+1} = x_k + s_k$.
8. $k = k + 1$ dan kembali ke langkah 3.

Sebagai sebuah ciri khas dalam metode *Newton Raphson*, mayoritas waktu perhitungan berada dalam langkah-5 : perhitungan nilai-nilai *Jacobian* dan langkah-6 yaitu menyelesaikan langkah-langkah perbaikan. Dengan adanya pendekatan logis yang menghasilkan perhitungan yang lebih sedikit dalam kedua langkah tersebut akan sangat mempengaruhi kecepatan perhitungan dengan hanya menambah jumlah iterasinya.

Untuk lebih memudahkan pemahaman terhadap pendekatan-pendekatan yang dilakukan pada metode *Fast Decoupled* sistem distribusi radial yang dibahas disini, maka akan diberikan suatu contoh sistem distribusi radial dengan 3 buah lateral.



Gambar 3.5. Sistem Distribusi Sederhana

Sistem distribusi radial di atas terdiri dari 3 (tiga) buah lateral sehingga didapatkan 3 (tiga) buah persamaan aliran daya:

$$F_{1(x)} = \tilde{V}_{10} - \bar{V}_{21} = 0 \quad (3.16)$$

$$F_{2(x)} = \tilde{V}_{20} - \bar{V}_{31} = 0 \quad (3.17)$$

$$F_{3(x)} = \tilde{V}_{30} - \bar{V}_0 = 0 \quad (3.18)$$

dimana:

\tilde{V}_{10} adalah tegangan ujung dari cabang ke-1. \tilde{V}_{20} adalah tegangan ujung dari lateral ke-2 dan \tilde{V}_{30} adalah tegangan ujung awal dari lateral ke-3. \bar{V}_0 adalah tegangan sumber bus lateral ke-3 (feeder utama) yang nilainya telah ditentukan dan besar masing-masing tegangan adalah:

$$\begin{aligned}
 V_{10} &= V_{11} + Z_{11} (\frac{1}{2} Y_{12} V_{11} + I_{11}) \\
 I_{11} &= - \left[\frac{S_{L1}}{V_{11}} \right] \\
 V_{21} &= V_{22} + Z_{22} (\frac{1}{2} Y_{22} V_{22} + I_{22}) \\
 I_{22} &= - \left[\frac{S_{L2}}{V_{22}} \right] \\
 V_{20} &= V_{21} + Z_{11} (\frac{1}{2} Y_{21} V_{21} + I_{21}) \\
 &= V_{22} + Z_{22} (\frac{1}{2} Y_{22} V_{22} - \left[\frac{S_{L2}}{V_{22}} \right]) + \frac{1}{2} Z_{21} Y_{21} (V_{22} + Z_{22} (\frac{1}{2} Y_{22} V_{22} - \left[\frac{S_{L2}}{V_{22}} \right])) \\
 &\quad - Z_{21} \left[\frac{S_{L1}}{V_{11}} \right] - Z_{21} \left[\frac{V_{22}}{S_{L2}} \right] \\
 V_{31} &= V_{32} + Z_{32} (\frac{1}{2} Y_{32} V_{32} + I_{32}) \\
 I_{32} &= - \left[\frac{S_{L3}}{V_{32}} \right] \\
 V_{30} &= V_{31} + Z_{31} (\frac{1}{2} Y_{31} V_{31} + I_{31}) \\
 &= V_{22} + Z_{22} (\frac{1}{2} Y_{22} V_{22} - \left[\frac{S_{L2}}{V_{22}} \right]) + \frac{1}{2} Z_{21} Y_{21} (V_{22} + Z_{22} (\frac{1}{2} Y_{22} V_{22} - \left[\frac{S_{L2}}{V_{22}} \right])) \\
 &\quad - Z_{31} \left[\frac{S_{L3}}{V_{32}} \right] - Z_{31} \left[\frac{V_{22}}{S_{L2}} \right] - Z_{31} \left[\frac{S_{L1}}{V_{11}} \right]
 \end{aligned}$$

$$\begin{aligned}
 \frac{\partial F_3}{\partial V_{22}} \frac{\partial V_{22}}{\partial Z} &= \frac{\partial F_3}{\partial V_{22}} \frac{\partial V_{22}}{\partial Z} \\
 \frac{\partial F_3}{\partial V_{11}} \frac{\partial V_{11}}{\partial Z} &= \frac{\partial F_3}{\partial V_{11}} \frac{\partial V_{11}}{\partial Z} \\
 \frac{\partial F_2}{\partial V_{32}} &= - \left(1 + \frac{\partial F_2}{\partial V_{32}} \frac{\partial V_{32}}{\partial Z} + \frac{\partial F_2}{\partial V_{32}} \frac{\partial V_{32}}{\partial Z} \right) \frac{\partial V_{32}}{\partial Z} \\
 \frac{\partial F_2}{\partial V_{22}} &= 1 + \frac{\partial F_2}{\partial V_{22}} \frac{\partial V_{22}}{\partial Z} + \frac{\partial F_2}{\partial V_{22}} \frac{\partial V_{22}}{\partial Z} + \frac{\partial F_2}{\partial V_{22}} \frac{\partial V_{22}}{\partial Z} + \frac{\partial F_2}{\partial V_{22}} \frac{\partial V_{22}}{\partial Z} + \frac{\partial F_2}{\partial V_{22}} \frac{\partial V_{22}}{\partial Z} \\
 \frac{\partial F_2}{\partial V_{11}} \frac{\partial V_{11}}{\partial Z} &= \frac{\partial F_2}{\partial V_{11}} \frac{\partial V_{11}}{\partial Z} \\
 \frac{\partial F_1}{\partial V_{32}} &= 0 \\
 \frac{\partial F_1}{\partial V_{22}} &= - \left(1 + \frac{\partial F_1}{\partial V_{22}} \frac{\partial V_{22}}{\partial Z} + \frac{\partial F_1}{\partial V_{22}} \frac{\partial V_{22}}{\partial Z} \right) \frac{\partial V_{22}}{\partial Z} \\
 \frac{\partial F_1}{\partial V_{11}} &= 1 + \frac{\partial F_1}{\partial V_{11}} \frac{\partial V_{11}}{\partial Z} + \frac{\partial F_1}{\partial V_{11}} \frac{\partial V_{11}}{\partial Z}
 \end{aligned}$$

dimana :

$$J^k = \begin{bmatrix} \frac{\partial F_1}{\partial V_{11}} & \frac{\partial F_1}{\partial V_{22}} & \frac{\partial F_1}{\partial V_{32}} \\ \frac{\partial F_2}{\partial V_{11}} & \frac{\partial F_2}{\partial V_{22}} & \frac{\partial F_2}{\partial V_{32}} \\ \frac{\partial F_3}{\partial V_{11}} & \frac{\partial F_3}{\partial V_{22}} & \frac{\partial F_3}{\partial V_{32}} \end{bmatrix} \tag{3.19}$$

Matrik *Jacobian* dari persamaan di atas adalah:

$$\frac{\partial F_3}{\partial V_{32}} = 1 + \frac{1}{2} Z_{32} Y_{32} + Z_{32} \frac{S_{L3}^*}{|V_{32}|^2} + \frac{1}{2} Z_{31} Y_{31} + \frac{1}{4} Z_{31} Z_{32} Y_{31} Y_{32} + \frac{1}{2} Z_{31} Y_{31} Z_{32} \frac{S_{L3}^*}{|V_{32}|^2}$$

3.1.9. Metode Fast Decoupled^[3]

Pada sistem distribusi nilai *line charging* diabaikan karena selain saluran distribusi pendek-pendek juga dalam perhitungan *pu* nilai *line charging* (Y) adalah kecil dibandingkan nilai tegangan (1 *pu*), sehingga komponen-komponen yang mengandung perkalian dengan variabel Y pada turunan partial di atas akan bernilai nol, sehingga selanjutnya *Jacobian* pada sistem di atas akan menjadi:

$$J_k = \begin{bmatrix} 1 + Z_{11} \frac{S_{L1}^*}{|V_{11}|^2} & - \left(1 + Z_{22} \frac{S_{L2}^*}{|V_{22}|^2} \right) & 0 \\ Z_{21} \frac{S_{L1}^*}{|V_{11}|^2} & 1 + Z_{22} \frac{S_{L2}^*}{|V_{22}|^2} & - \left(1 + Z_{32} \frac{S_{L3}^*}{|V_{32}|^2} \right) \\ Z_{31} \frac{S_{L1}^*}{|V_{11}|^2} & Z_{31} \frac{S_{L2}^*}{|V_{22}|^2} & 1 + Z_{32} \frac{S_{L3}^*}{|V_{32}|^2} \end{bmatrix} \quad (3.20)$$

Karena blok-blok yang tidak nol yang berada di bawah diagonal memiliki suku $\frac{S_{L_i}^*}{|V_i|^2}$ yang bernilai kecil serta dikalikan dengan Z_i ($\leq 10^{-4}$) yang bernilai kecil pula maka blok-blok ini akan bernilai mendekati nol sehingga dapat diabaikan. Dengan melakukan ini dapat pula dikatakan bahwa perbedaan tegangan untuk lateral ke- i hanya diakibatkan oleh tegangan ujung akhir dari lateral ke- i dan tegangan ujung akhir dari induk lateral ke- i sedangkan pengaruh tegangan dari lateral yang lain diabaikan.

Maka *Jacobiannya* menjadi:

$$J_k = \begin{bmatrix} 1 + Z_{11} \frac{S_{L1}^*}{|V_{11}|^2} & -\left(1 + Z_{22} \frac{S_{L2}^*}{|V_{22}|^2}\right) & 0 \\ 0 & 1 + Z_{22} \frac{S_{L2}^*}{|V_{22}|^2} & -\left(1 + Z_{32} \frac{S_{L3}^*}{|V_{32}|^2}\right) \\ 0 & 0 & 1 + Z_{32} \frac{S_{L3}^*}{|V_{32}|^2} \end{bmatrix} \quad (3.21)$$

Pendekatan lebih lanjut dilakukan dengan memperhatikan nilai dari: $\frac{S_{L1}^*}{|V_1|^2}$

yang kecil sehingga blok bernilai mendekati "1" dapat diganti dengan nilai pasti yaitu sama dengan "1" dan blok-blok yang bernilai "-1" diganti dengan nilai pasti "1", sehingga *Jacobiannya* menjadi:

$$J_K = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 1 & 0 & 1 \end{bmatrix} \quad (3.22)$$

Maka langkah ke-6 dari metode *Newton Raphson* dapat disederhanakan menjadi substitusi terbalik:

$$J_{k sk} = -F_k$$

$$\begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S_{11} \\ S_{22} \\ S_{32} \end{bmatrix} = - \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} \quad (3.23)$$

$$S_{32} = -F_3$$

$$S_{22} = -F_2 + S_{32}$$

$$S_{11} = -F_1 + S_{22}$$

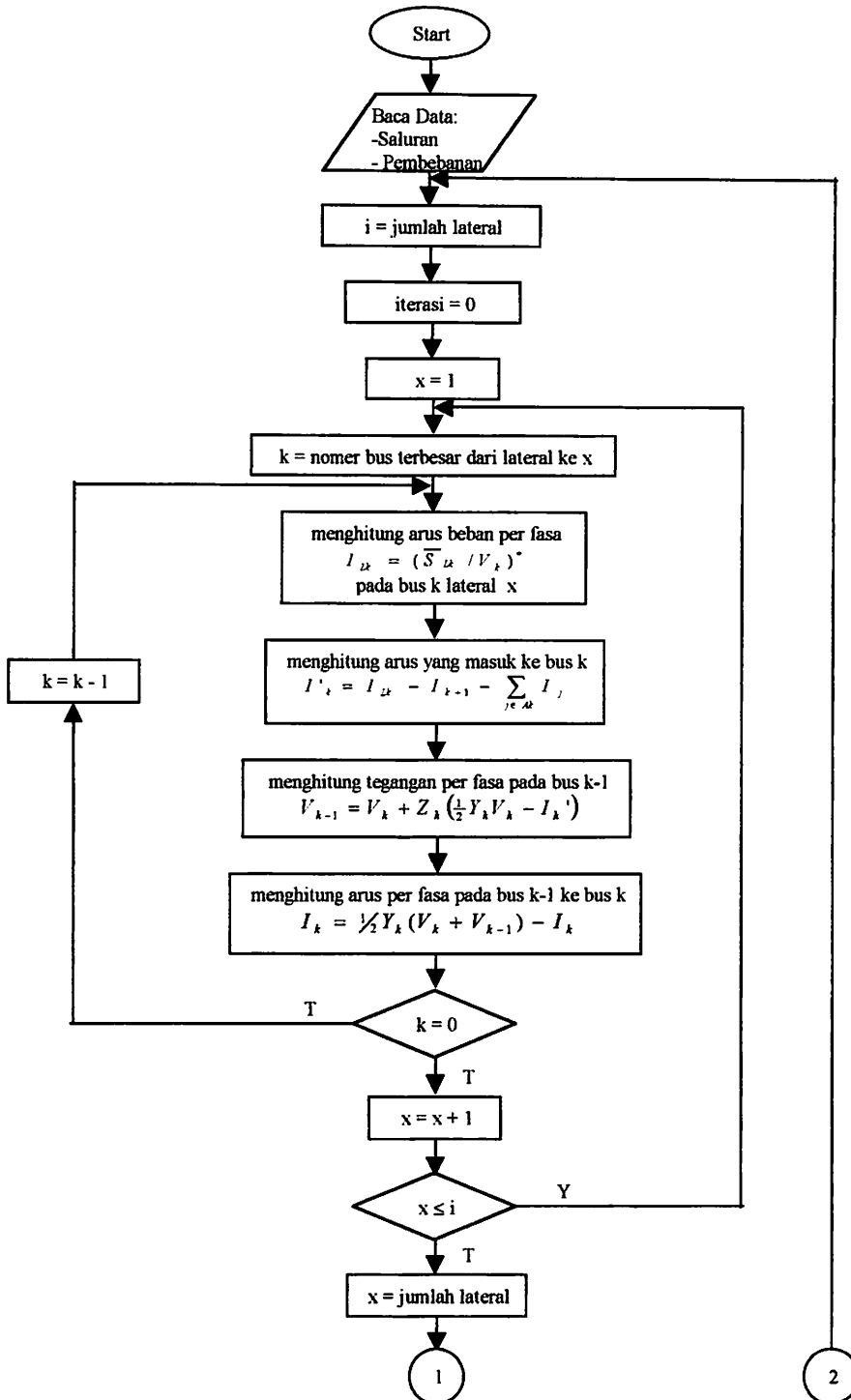
Dari pembahasan diatas dapat disimpulkan bahwa metode *Fast Decoupled* merupakan suatu pendekatan atau penyederhanaan dari metode *Newton Raphson*, karena fakta dalam saluran distribusi bahwa elemen impedansi saluran Z_k dan admitansi *line charging* Y_k sangat kecil bila dibandingkan dengan nilai tegangan, cabang *Jacobian* dapat disederhanakan menjadi matrik identitas sehingga langkah perbaikan (*up date step*) dapat diselesaikan dengan cara sederhana yaitu substitusi terbalik.

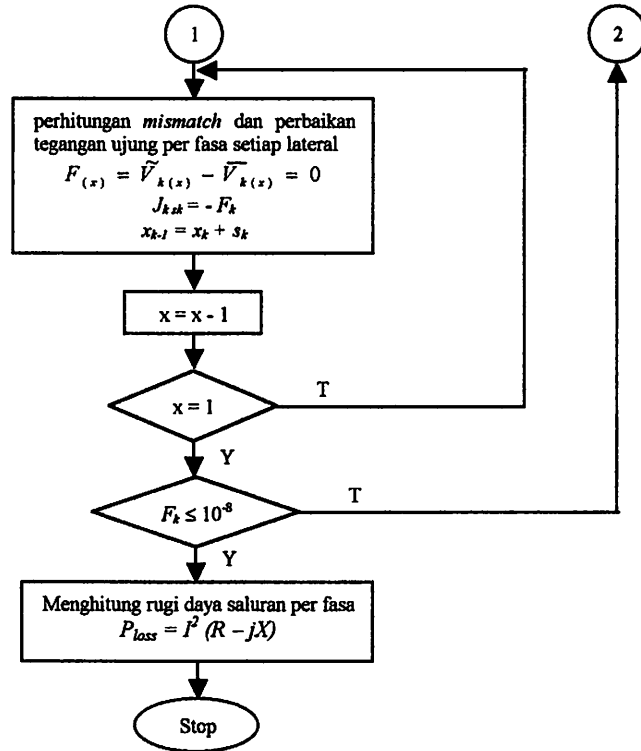
3.1.9.1. Algoritma *Fast Decoupled*

1. Membaca data-data saluran seperti data impedansi dan admitansi *line charging* saluran, data pembebanan, tegangan sumber, tegangan asumsi awal ujung saluran, arus pada ujung lateral, data hubungan antar bus dan antar lateral.
2. Perhitungan dimulai dari bus menurut urutan pembalikan pertama muncul (*reverse-breadth first/RBF*) yaitu dari bus terakhir pada lateral pertama.
3. Menghitung arus injeksi per fasa (I_{ja}, I_{jb}, I_{jc}) pada bus dari suatu lateral dengan persamaan (3.3).
4. Menghitung tegangan per fasa (V_a, V_b, V_c) dari suatu bus ke bus sebelumnya dari suatu lateral dengan persamaan (3.5).
5. Menghitung arus saluran per fasa (I_a, I_b, I_c) dari suatu bus ke bus sebelumnya dari suatu lateral dengan persamaan (3.6).
6. Kembali ke langkah 3 untuk bus berikutnya sampai perhitungan dilakukan pada bus yang merupakan ujung awal dari suatu lateral.

7. Kembali ke langkah 3 untuk lateral berikutnya sampai perhitungan dilakukan pada lateral utama/*main feeder*.
8. Menghitung perbedaan tegangan (*voltage mismatch*) pada setiap lateral dimulai pada lateral utama sampai lateral pertama dengan persamaan (3.14).
9. Menghitung nilai perbaikan (s_k) dan mengoreksi nilai tegangan pada setiap ujung akhir lateral.
10. Jika perbedaan tegangan yang didapatkan belum memenuhi batas yang diijinkan/toleransi maka perhitungan kembali ke langkah 3.
11. Jika perbedaan tegangan sudah memenuhi batas yang diijinkan maka tegangan dianggap konvergen dan menjadi output program.
12. Menghitung rugi-rugi daya saluran per fasa dari total penyulang dan mencetak nilai-nilainya.

3.1.9.2. Flowchart Fast Decoupled





Gambar 3.6. Flowchart Fast Decoupled

3.2. Rekonfigurasi Jaringan^[4]

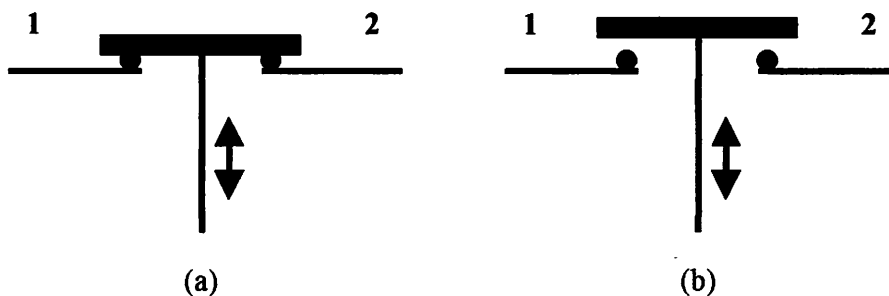
Rekonfigurasi jaringan merupakan proses menata konfigurasi awal dari jaringan (*initial condition*) menjadi suatu konfigurasi optimum akhir (*optimal condition*), sehingga dari konfigurasi yang terakhir ini diperoleh kerugian daya sistem distribusi yang paling kecil (aliran daya yang paling optimum).

Meskipun jaringan distribusi dioperasikan menggunakan sistem radial, namun besarnya sistem tersebut dikombinasikan kedalam beberapa tipe. Tipe yang paling umum adalah model jaringan distribusi radial dengan *tie* dan *switch* pemisah.

3.2.1. Metode Rekonfigurasi Jaringan

Rekonfigurasi jaringan distribusi dapat dilakukan dengan mengatur kondisi dari *switch-switch* ini dalam keadaan *on* atau *off*. *Switch* yang terdapat dalam jaringan distribusi terdapat dua macam:

1. *Normally Closed Switch (NC Switch)* yang dalam keadaan operasi normal, posisi kontakannya selalu tertutup (*on*).
2. *Normally Open Switch (NO Switch)* yang dalam keadaan operasi normal, posisi kontakannya selalu terbuka (*off*).



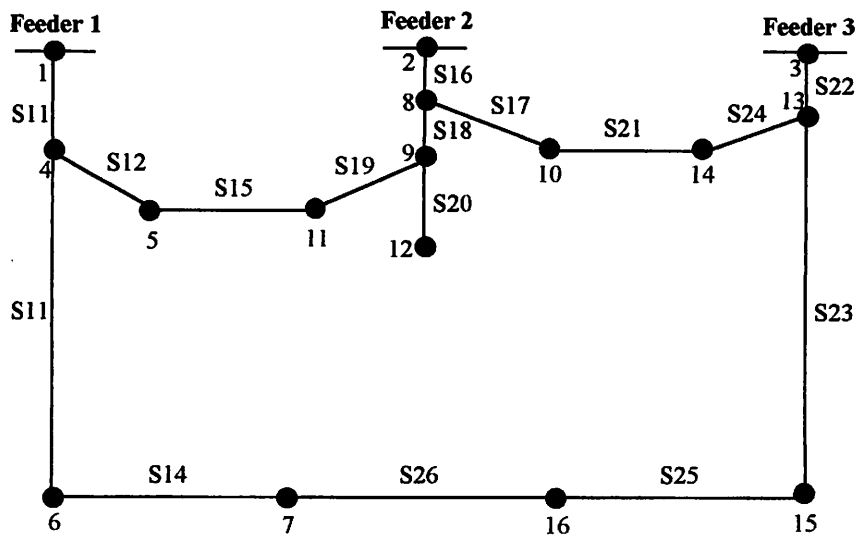
Gambar 3.7. Diagram Skematik Tipe (a) NC Switch dan (b) NO Switch

Jadi jelaslah bahwa sebenarnya rekonfigurasi jaringan dalam sistem distribusi tenaga listrik adalah proses mengganti struktur topologi dari jaringan distribusi dengan mengubah status *open/closed* dari NO/NC Switch sehingga dari proses ini bisa didapatkan aliran daya yang optimal pada sistem distribusi yang menyebabkan kerugian daya jadi minimum.

3.2.2. Metode Rekonfigurasi Menggunakan *Simulated Annealing (SA)*^[4]

Gambar 3.8. menunjukkan contoh jaringan distribusi yang terdiri dari 3 (tiga) *feeder* dengan 13 (tiga belas) *sectionalising switch normally closed* dan 3 (tiga) *tie switch normally open*, yaitu s15, s21, dan s26. Notasi (X,Y) digunakan

untuk menunjukkan operasi pembukaan *switch* Y dan penutupan *switch* X. Contoh, operasi (s12, s15) mentransfer beban 5 dari *feeder* 1 ke *feeder* 2 dengan membuka bagian *switch* s12 dan menutup *tie switch* s15. Dengan demikian, penutupan *switch* dapat selalu mengikuti pembukaan dari sebuah *switch*. Beban pada bus 11 dapat ditransfer ke *feeder* 1 dengan menutup *tie switch* s15 dan membuka bagian *switch* s19. Dengan cara yang sama, beban lainnya dapat ditransfer dari *feeder* ke *feeder* dengan operasi *switch*.



Gambar 3.8. Jaringan Distribusi

3.3. Teori Dasar *Simulated Annealing* (SA)^[4]

3.3.1. Pengenalan Konsep *Simulated Annealing* (SA)^[4]

Konsep *Simulated Annealing* (SA) pertama kali diperkenalkan pada dunia optimasi pada permulaan tahun 1980 oleh Kirkpatrick dan secara bebas dikembangkan oleh Cerny. SA adalah sebuah kombinasi algoritma optimasi

umum-khusus yang kuat berdasarkan metode probabilistik atau kemungkinan yang telah diterapkan dengan baik pada banyak area seperti pada desain sirkuit VSLI, jaringan kerja sistem syaraf, desain kode, dan penempatan kapasitor pada sistem daya.

3.3.2. Kombinasi Optimasi^[4]

Masalah kombinasi optimasi adalah sebuah masalah minimisasi atau maksimisasi yang terkait dengan penemuan solusi terbaik atau optimal dari kumpulan alternatif kemungkinan. Penemuan solusi ini dapat digolongkan secara menyeluruh oleh ruang penelitian dan fungsi harga atau fungsi obyektif.

Ruang penelitian S adalah sebuah kumpulan solusi kemungkinan terbatas atau tak terbatas yang dapat dihitung. Dan fungsi obyektif $f: S \rightarrow \mathbb{R}$ mengelompokkan tiap titik pada ruang penelitian ke dalam garis sesungguhnya, untuk memberikan pengukuran terhadap bagaimana sebuah solusi yang baik itu berhubungan dengan yang lainnya. Pada masalah minimisasi, solusi optimal yang diharapkan adalah $x_{opt} = 1$, untuk:

$$f(x_{opt}) \leq f(x), \text{ untuk semua } x \in S \quad (3.24)$$

Masalah tersebut kemudian dapat dikelompokkan dengan mudah sebagai:

$$\begin{aligned} &\text{minimize } f(x) \\ &x \in S \end{aligned} \quad (3.25)$$

Solusi x_{opt} disebut sebagai global optimum dan nilai obyektif x_{opt} , harga optimal disebut $f_{opt} = f(x_{opt})$. Hal ini dapat dijadikan sebagai lebih dari satu solusi yang memuaskan pada kondisi persamaan (3.9), kumpulan dari solusi optimal secara menyeluruh yang dikenal sebagai S_{opt} .

3.3.3. Analogi Pada Pemanasan Fisik (*Physical Annealing*)^[4]

Nama pemanasan fisik berasal dari sebuah analogi antara pengkombinasian optimasi dan proses fisik pemanasan. Pada pemanasan fisik, zat padat didinginkan dengan sangat perlahan mulai dari temperatur yang sangat tinggi sampai temperatur yang sangat rendah agar dapat mencapai uraian energi internal minimum. Zat padat ini didinginkan secara perlahan supaya keseimbangan panas dicapai pada tiap temperatur. Keseimbangan panas dapat digolongkan oleh distribusi *Boltzman*:

$$P_T = \{X = x\} = \frac{e^{-E_x/k_B T}}{\sum_{\text{all states } i} e^{-E_i/k_B T}} \quad (3.26)$$

dimana X adalah variabel acak yang menunjukkan posisi baru, E_x adalah energi dari posisi x , k_B adalah konstanta *Boltzman*, dan T adalah temperatur.

Evolusi dari posisi zat padat dalam tanur tinggi menuju keseimbangan panas dapat ditiru secara efisien oleh algoritma sederhana berdasarkan teknik *Monte Carlo* yang diusulkan oleh Metropolis tahun 1953. Pada algoritma Metropolis mengambil posisi baru dan menggambarkan posisi tersebut yaitu y dengan mengaplikasikan beberapa gangguan kecil. Kemudian transisi dari posisi x menuju posisi y diterima dengan kemungkinan:

$$P_{\text{accept}}(x, y) = \begin{cases} 1 & , \text{ jika } E_x - E_y \leq 0 \\ e^{-(E_x - E_y)/k_B T} & , \text{ jika } E_x - E_y > 0 \end{cases} \quad (3.27)$$

Jika diterima, y menjadi solusi baru dan prosedurnya diulang. Aturan penerimaan ini dikenal dengan *Metropolis Criterion*.

3.3.4. Algoritma *Simulated Annealing* (SA)^{[4][5]}

SA digolongkan ke dalam kategori algoritma optimasi yang dikenal sebagai metode probabilitas atau kemungkinan, yang sampai sekarang terdiri dari beberapa pengacakan dalam menentukan solusi. Urut-urutan dari solusi yang dijabarkan secara acak menciptakan solusi baru melalui gangguan pada solusi langsung kemudian menerima atau menolak titik baru dengan kemungkinan tertentu yang bergantung pada temperatur dan perubahan dalam fungsi obyektif.

Kemungkinan dari penerimaan perpindahan khusus dikenal dengan penerimaan kemungkinan, yang dituliskan sebagai berikut:

$$P_{accept}(\Delta f, T) = \begin{cases} 1, & \Delta f \leq 0 \\ e^{-\Delta f / T}, & \Delta f > 0 \end{cases} \quad (3.28)$$

atau

$$P_{accept}(\Delta f, T) = \min(1, e^{-\Delta f / T}) \quad (3.29)$$

Persamaan diatas sama dengan persamaan (3.27). Disini Δf adalah perubahan pada fungsi harga yang sesuai dengan perubahan rencana pada posisi dan T adalah artificial parameter kontrol yang mewakili temperatur.

Adapun algoritma *Simulated Annealing* (SA) secara umum adalah:

1. Baca data:

$$X = X_0$$

$$T = T_0$$

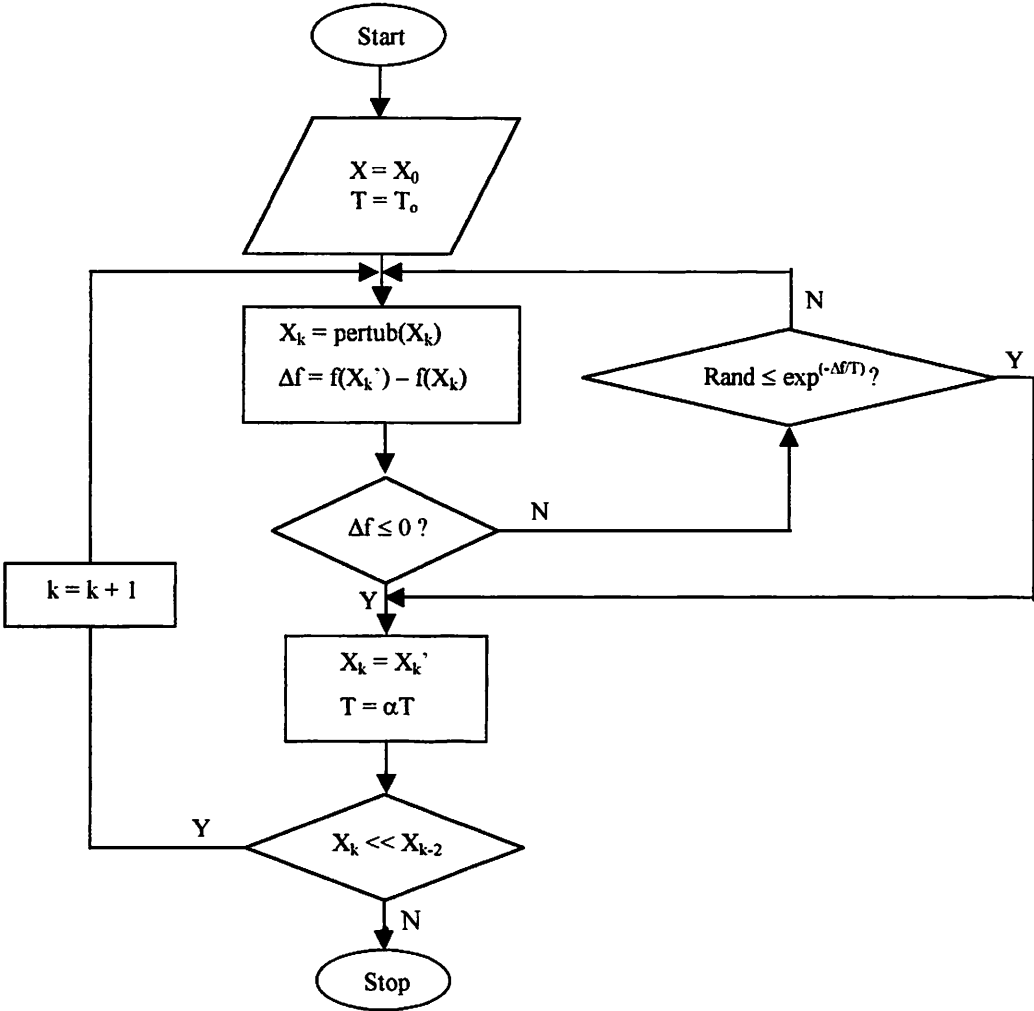
2. Menentukan nilai:

$$X_k' = \text{pertub}(X_k)$$

$$\Delta f = f(X_k') - f(X_k)$$

3. Apakah $\Delta f \leq 0$?
4. Jika ya, maka diperoleh $X_k = X_k'$ dan $T = \alpha T$.
5. Jika tidak, melakukan proses random untuk menentukan nilai $X_k = X_k'$ dan $T = \alpha T$. Tetapi apabila pada proses random masih belum memenuhi $\exp(-\Delta f / T)$ maka harus kembali lagi ke langkah 2.
6. Setelah diperoleh $X_k = X_k'$ dan $T = \alpha T$ maka berlanjut ke langkah berikutnya.
7. Apakah $X_k \ll X_{k-2}$? Jika ya, melakukan proses iterasi. Jika tidak maka proses selesai.

3.3.5. Flowchart Simulated Annealing^[5]



Gambar 3.9. Flowchart Simulated Annealing

BAB IV

APLIKASI PENGGUNAAN METODE REKONFIGURASI JARINGAN

4.1. Program Komputer Metode Rekonfigurasi Jaringan

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 lama jika dikerjakan manual.

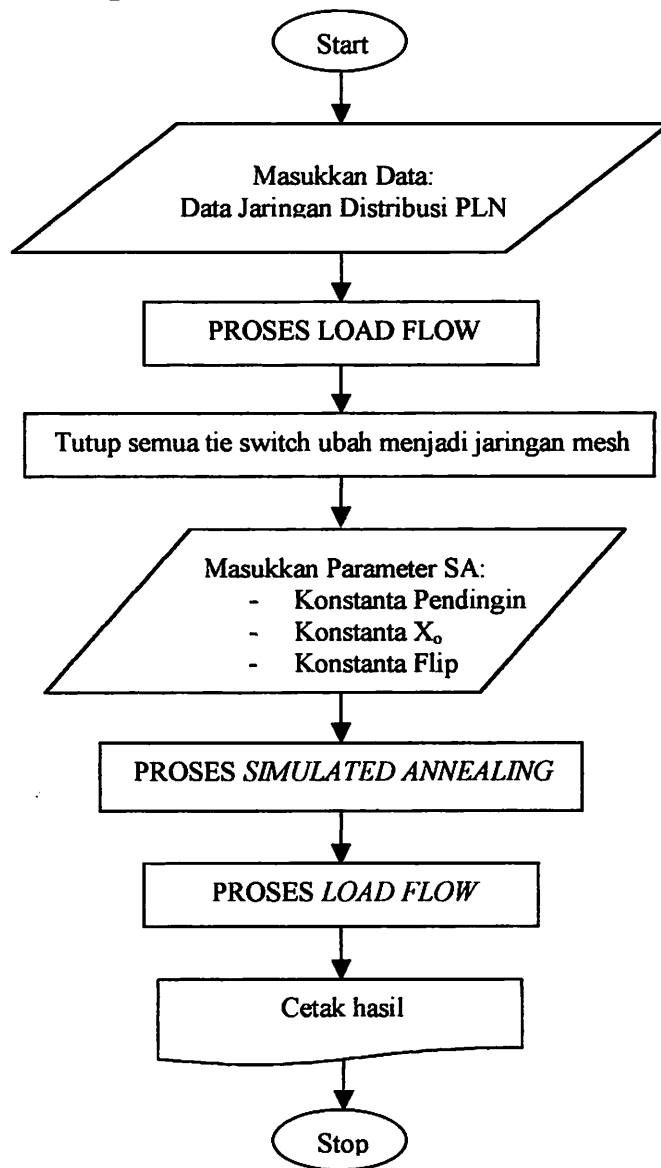
Program komputer ini menggunakan bahasa pemrograman Borland Delphi versi 7.0 merupakan bahasa pemrograman terstruktur yang relatif mudah untuk dipelajari dan mudah penggunaannya.

4.2. Algoritma Program

Urutan langkah-langkah dalam program komputer yang digunakan ini dapat dilihat pada algoritma berikut:

1. Masukkan data jaringan distribusi PLN.
2. Jalankan proses *Load Flow* dengan metode *Fast Decoupled*.
3. Tutup semua *tie switch* ubah menjadi jaringan mesh.
4. Masukkan parameter *Simulated Annealing*.
5. Jalankan proses *Simulated Annealing*.
6. Jalankan proses *Load Flow* dengan metode *Fast Decoupled*.
7. Cetak hasil.
8. Perhitungan selesai.

4.3. Flowchart Program



Gambar 4.1. Flowchart Program

4.4. Data Jaringan

Dalam skripsi ini data yang digunakan adalah penyulang yang keluar dari Trafo II GI Blimbing Malang. Alasan pemilihan data tersebut adalah karena output dari Trafo II GI Blimbing Malang mensupply daerah yang kepadatan bebannya sangat bervariasi.

Data Teknis Trafo II GI Blimbing Malang

Tegangan dasar	: 20 kV
Daya dasar	: 30 MVA
Penghantar SUTM 20 kV	: AAAC 150 mm
Impedansi penghantar	: $0,2162 + j 0,3305$
Cos ϕ	: 0,8

Data beban untuk masing-masing penyulang dapat dilihat pada lampiran.

Dari lampiran, daya P (kW) dan Q (kVAR) dari bus 2 (dua) penyulang mawar dihitung sebagai berikut:

- Fasa R:

$$P = 502,4 \cdot 0,8 = 427 \text{ kW}$$

$$Q = 502,4 \cdot \sin (\text{arc cos } 0,8) = 264,7 \text{ kVAR}$$

- Fasa S:

$$P = 507,9 \cdot 0,8 = 431,7 \text{ kW}$$

$$Q = 507,9 \cdot \sin (\text{arc cos } 0,8) = 267,5 \text{ kVAR}$$

- Fasa T:

$$P = 504,4 \cdot 0,8 = 428,7 \text{ kW}$$

$$Q = 504,4 \cdot \sin (\text{arc cos } 0,8) = \text{kVAR}$$

Dengan cara perhitungan yang sama, maka beban untuk masing-masing bus akan diperoleh hasil seperti tabel 4.1 berikut:

TABEL 4.1
DATA JARINGAN DISTRIBUSI

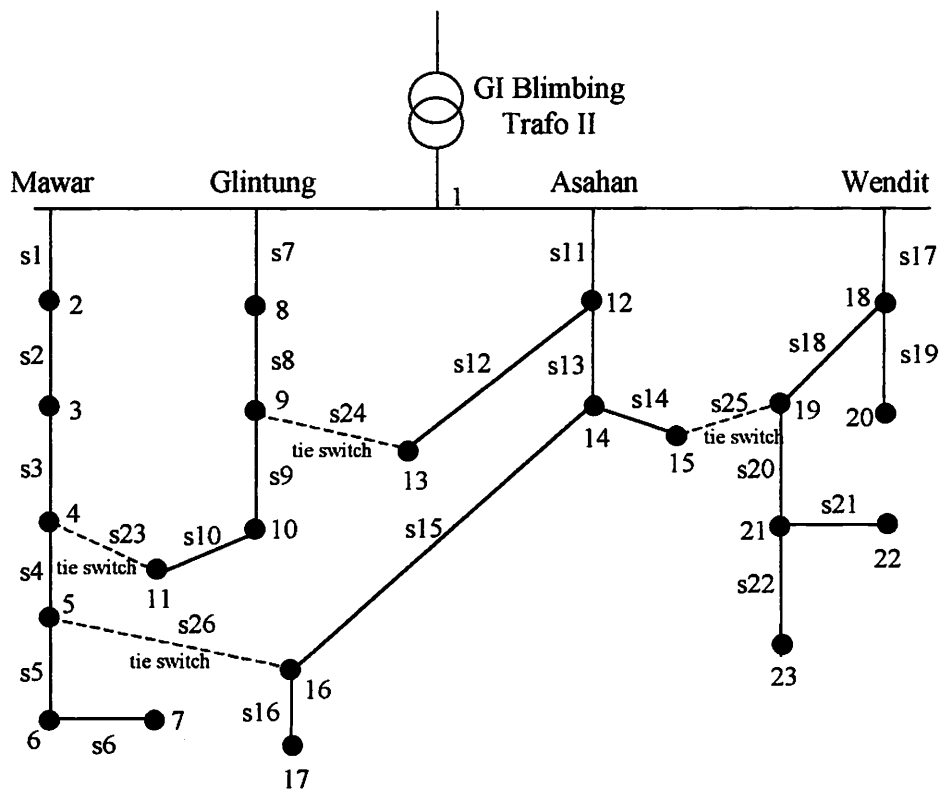
BUS PANGKAL	BUS UJUNG	P BUS UJUNG (kW)			Q BUS UJUNG kVAR)			JARAK (m)	R (Ω)	X (Ω)
		R	S	T	R	S	T			
1	2	427,0	431,7	428,7	264,7	267,5	225,0	5409	1,2142	1,8560
2	3	271,8	298,4	303,1	168,4	184,9	143,2	3014	0,8832	1,3501
3	4	294,5	270,4	295,7	182,5	167,6	155,1	4915	1,0896	1,6657
4	5	188,3	165,5	166,1	116,7	102,5	99,21	3782	0,8425	1,2880
5	6	67,39	55,65	61,34	41,76	34,49	35,5	994	0,1464	0,2238
6	7	172,9	158,3	148,9	107,2	98,12	91,10	2738	0,3628	0,5546
1	8	117,9	100,6	104,8	73,08	62,34	62,11	1479	0,2986	0,4564
8	9	169,6	146,9	151,7	105,1	91,02	89,35	1224	0,2581	0,3946
9	10	143,2	120,7	122,3	88,77	74,82	75,45	1705,5	0,2197	0,3358
10	11	114,6	127,0	138,2	71,03	78,73	60,38	1492	0,3226	0,4931
1	12	156,5	115,2	95,22	97,02	71,41	82,47	2059,4	0,5349	0,8177
12	13	176,5	203,0	208,4	109,4	125,8	93,00	3412,5	0,7444	1,1379
12	14	41,70	39,02	38,90	25,84	24,18	21,97	1645,8	0,3559	0,5440
14	15	4,070	3,101	3,101	2,522	1,922	2,144	1039,3	0,2558	0,3910
14	16	181,5	157,2	165,9	112,5	97,43	95,61	2582	0,5935	0,9072
16	17	0	0	0	0	0	0	131	0,0283	0,0433
1	18	0	0	0	0	0	0	1817	0,3853	0,5890
18	19	155,8	131,3	137,6	96,56	81,38	82,08	2627	0,6553	1,0017
18	20	25,18	38,82	32,50	15,60	23,93	13,26	3008,5	1,6516	2,5247
19	21	254,1	257,9	208,3	157,5	159,8	133,9	529	0,3349	0,5119
21	22	46,72	22,46	58,94	28,95	13,92	24,61	1476	0,3191	0,4878
21	23	267,8	248,5	267,6	165,9	154,0	141,1	4682	1,0138	1,5497
4	11	-	-	-	-	-	-	207	0,0448	0,0648
9	13	-	-	-	-	-	-	173	0,0374	0,0572
15	19	-	-	-	-	-	-	1114	0,2408	0,3682
5	16	-	-	-	-	-	-	262	0,0566	0,0866

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

Output Trafo II GI Blimbing Malang terdiri dari:

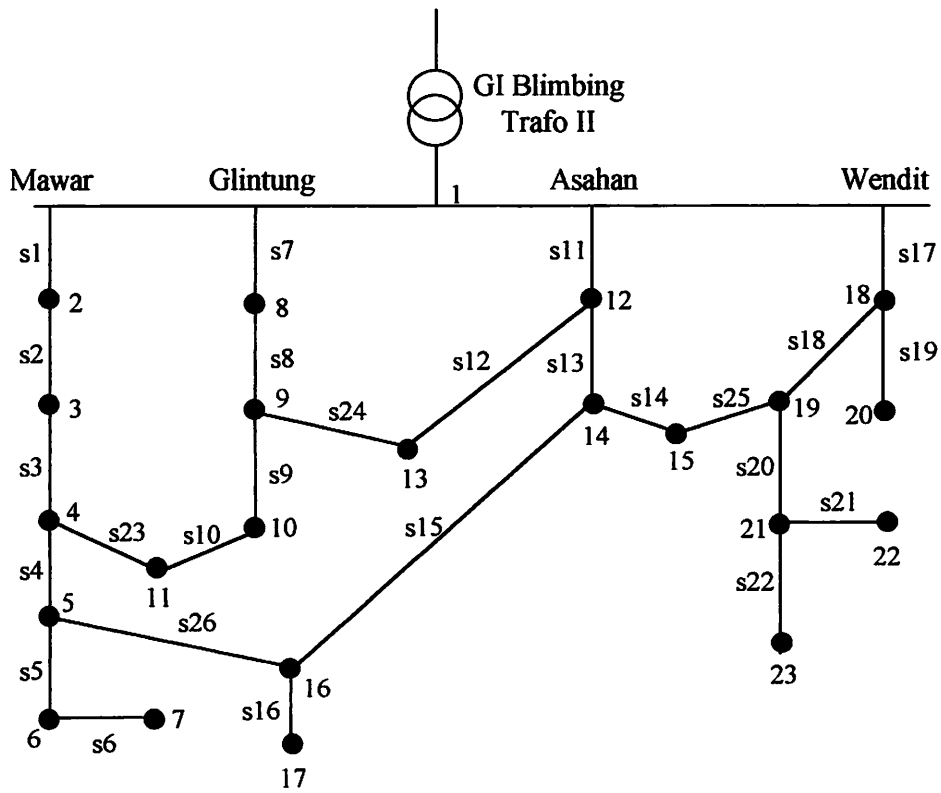
1. Penyulang Mawar (M)
2. Penyulang Glintung (G)
3. Penyulang Asahan (A)
4. Penyulang Wendit (W)

Konfigurasi dari 4 penyulang tersebut dapat dilihat pada gambar 4.1 berikut:



Gambar 4.2. Konfigurasi Jaringan Radial sebelum Rekonfigurasi

Dalam analisa rekonfigurasi semua *tie switch (normally open)* ditutup sehingga jaringan radial seperti gambar 4.1. menjadi jaringan mesh seperti gambar 4.2. Dari penutupan *tie switch* terdapat 4 (empat) mesh yang masing-masing meshnya terdiri dari 8, 5, 6, dan 8 *switch*.

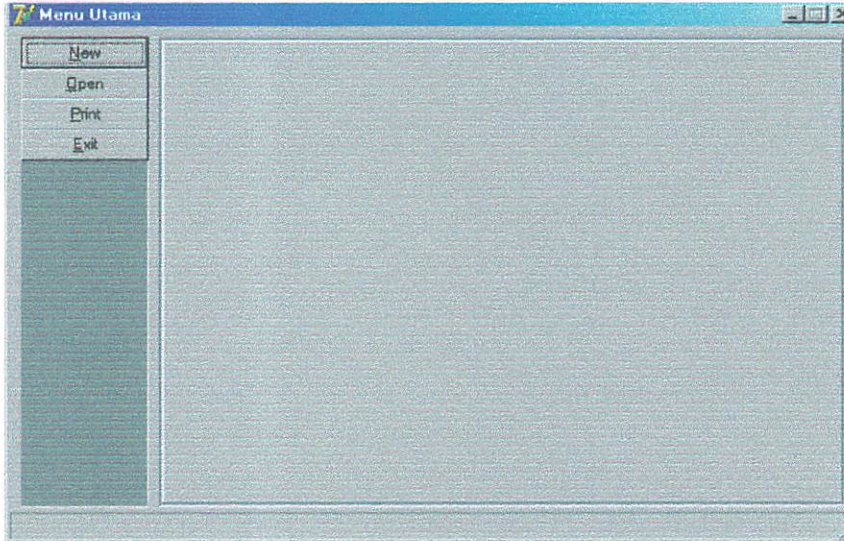


Gambar 4.3. Konfigurasi Jaringan Mesh

4.5. Tampilan Program

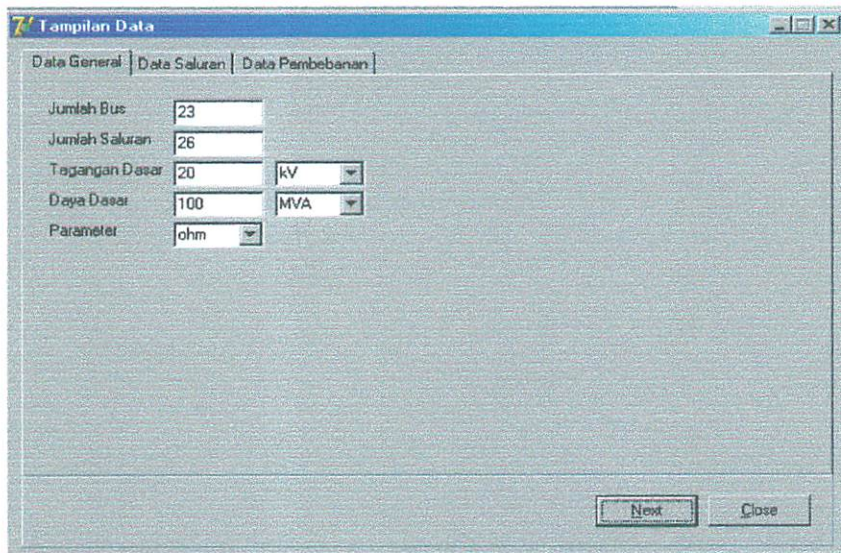
Adapun tahap-tahap dari program ini adalah sebagai berikut:

1. Tampilan Awal



Gambar 4.4. Menu Utama

2. Tampilan Data



Gambar 4.5. Data General

Tampilan Data

Data General | Data Saluran | Data Pembebanan

No	Dari	Ke	R (pu)	X (pu)	Lo (pu)	Tl
1	1	2	1.2142	1.856	0	0
2	2	3	0.8832	1.3501	0	0
3	3	4	1.0896	1.6657	0	0
4	4	5	0.8425	1.288	0	0
5	5	6	0.1464	0.2238	0	0
6	6	7	0.3628	0.5546	0	0
7	1	8	0.2986	0.4564	0	0
8	8	9	0.2581	0.3946	0	0
9	9	10	0.2197	0.3358	0	0
10	10	11	0.3226	0.4931	0	0
11	1	12	0.5349	0.8177	0	0
12	12	13	0.7444	1.1379	0	0
13	12	14	0.3559	0.544	0	0
14	14	15	0.2558	0.391	0	0
15	14	16	0.5935	0.9072	0	0

Next Close

Gambar 4.6. Data Saluran

Tampilan Data

Data General | Data Saluran | Data Pembebanan

No	Abs V (pu)	Sud V (deg)	Pgen (MW)	Qgen (MVAR)	Pload (MW)	Qload (MVAR)	Cap	Typ Bus
1	1	0	0	0	0	0	0	1
2	1	0	0	0	1.4276	1.0707	0	3
3	1	0	0	0	0.7655	0.5741	0	3
4	1	0	0	0	1.4842	1.1132	0	3
5	1	0	0	0	0.8594	0.6446	0	3
6	1	0	0	0	0.1661	0.1245	0	3
7	1	0	0	0	0.619	0.4643	0	3
8	1	0	0	0	0.3342	0.2506	0	3
9	1	0	0	0	0.7353	0.5515	0	3
10	1	0	0	0	0.2871	0.2153	0	3
11	1	0	0	0	0.4388	0.3299	0	3
12	1	0	0	0	1.4908	1.1181	0	3
13	1	0	0	0	1.4538	1.0903	0	3
14	1	0	0	0	0.1872	0.1479	0	3
15	1	0	0	0	0.654	0.4905	0	3

Next Close

Gambar 4.7. Data Pembebanan

3. Hasil Program

Bus	R Sim kVAr	S Sre kV	S Sim kVAr	T Sre kV	T Sim kVAr	R Lue kW	R Lm kV	R Lm kV
1	897,842	1385,965	871,111	1410,227	764,721	46,785	2,117	
2	621,215	949,980	591,839	976,905	528,282	12,793	0,498	
3	449,782	650,561	404,172	672,719	382,321	3,263	0,102	
4	266,408	379,869	236,793	376,732	226,468	0,942	0,023	
5	148,940	214,110	132,603	210,382	126,698	0,828	0,019	
6	107,200	158,300	98,120	148,900	91,100	0,062	-0,172	
7	342,290	496,708	311,049	518,674	291,415	0,521	-1,359	
8	266,899	394,966	245,578	412,617	226,179	3,542	0,319	
9	159,854	247,727	153,615	260,533	135,894	1,008	0,094	
10	71,030	127,000	78,730	138,200	60,380	0,059	0,003	
11	348,184	517,901	321,635	511,936	296,069	3,575	0,088	
12	109,400	203,000	125,800	208,400	93,000	0,290	0,016	
13	141,179	199,440	123,807	208,030	120,007	0,351	0,008	

Gambar 4.8. Sebelum Rekonfigurasi

Bus	R Sim kVAr	S Sre kV	S Sim kVAr	T Sre kV	T Sim kVAr	R Lue kW	R Lm kV	R Lm kV
1	100	0,05	1	3	4			
2	0,95	0,1	2	5	16			
3	0,1	1000	3	12	13			
4	1000	4	4	14	15			
5	4	26						

Gambar 4.9. Simulated Annealing

Gen	Min	Avg	Max
1	0.667	12019.375	18078.250
2	0.833	15520.142	22689.569
3	15495.902	17663.827	22689.569
4	15495.902	17921.778	22689.569
5	15495.902	17907.883	22689.569
6	15495.902	16473.670	22689.569
7	15495.902	17654.336	22689.569
8	15495.902	16230.863	22838.833
9	15495.902	16965.156	22838.833
10	15495.902	17063.051	28005.088
11	15495.902	20815.640	28005.088
12	15495.902	21850.823	28005.088
13	15497.572	24359.472	28071.533
14	15497.572	23115.365	28071.533

Gambar 4.10. Hasil *Simulated Annealing* (SA)

Bus	R Sim kVar	S Sre kW	S Sim kVar	T Sre kW	T Sim kVar	R Lre kW	R Lim kVa
1	434.002	730.482	453.478	732.224	369.188	11.120	0.884
2	168.400	298.400	184.900	303.100	143.200	0.959	0.081
3	449.462	699.464	435.002	726.757	382.465	3.086	0.296
4	374.690	598.232	370.905	621.251	318.652	1.803	0.176
5	109.400	203.000	125.800	208.400	93.000	0.454	0.050
6	159.796	247.780	153.549	260.578	135.823	0.141	-0.356
7	71.030	127.000	78.730	138.200	60.380	0.028	-0.065
8	451.179	650.897	405.415	673.057	383.510	9.993	0.887
9	266.178	379.660	235.585	376.535	226.251	2.687	0.216
10	149.265	214.076	132.894	210.353	126.858	0.236	0.005
11	107.200	158.300	98.120	148.900	91.100	0.337	0.006
12	235.982	311.656	193.564	300.274	200.607	1.183	0.013
13	138.656	196.339	121.884	204.929	117.863	0.338	0.008

Gambar 4.11. Sesudah Rekonfigurasi

4.6. Analisa Perhitungan

Langkah-langkah Analisa Perhitungan:

1. Masukan Awal

Dari data pada tabel 4.1 (data jaringan distribusi) dimasukkan ke dalam software rekonfigurasi, maka hasil perhitungan tersebut dapat dilihat pada tabel 4.2 berikut ini:

TABEL 4.2

HASIL PERHITUNGAN SEBELUM REKONFIGURASI

Bus	Daya Aktif (kW)			Daya Reaktif (kVAR)			Rugi Daya Aktif (kW)			Rugi Daya Reaktif (kVAR)		
	R	S	T	R	S	T	R	S	T	R	S	T
1	1428.59	1385.97	1410.23	897.842	871.111	764.721	46.765	46.839	44.937	2.1170	3.0510	7.096
2	996.934	949.98	976.905	621.215	591.839	528.282	12.793	12.506	12.206	0.4980	0.7430	1.795
3	723.930	650.561	672.719	449.782	404.172	382.321	3.2630	2.9440	2.9490	0.1020	0.1570	0.3210
4	429.078	379.869	376.732	266.408	235.793	226.468	0.9420	0.8310	0.8040	0.0230	0.0380	0.0550
5	240.468	214.110	210.382	148.940	132.603	126.598	0.8280	0.7370	0.7050	0.0190	0.0320	0.0460
6	172.900	158.300	148.900	107.200	98.1200	91.1000	0.0520	0.0560	0.0550	-0.017	-0.150	-0.131
7	546.985	496.708	518.374	342.290	311.049	291.415	0.5210	0.5420	0.6230	-1.136	-1.165	-1.097
8	427.788	394.966	412.617	265.899	245.578	226.179	3.5420	3.3110	3.3090	0.3190	0.3770	0.6390
9	257.823	247.727	260.533	159.854	153.615	135.894	1.0080	0.9960	0.9860	0.0940	0.1150	0.2200
10	114.600	127.000	138.200	71.0700	78.7300	60.3800	0.0590	0.0700	0.0890	0.0030	0.0050	0.0190
11	560.682	517.901	511.936	348.184	321.635	296.069	3.5750	3.3310	3.1640	0.0880	0.1650	0.3110
12	176.500	203.000	208.400	109.400	125.800	93.000	0.2900	0.3650	0.3340	0.0160	0.0250	0.0900
13	227.468	199.440	208.030	141.179	123.807	120.007	0.3510	0.3050	0.3110	0.0080	0.0160	0.0300
14	4.07000	3.10100	3.10100	2.52200	1.99200	2.14400	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
15	181.500	157.200	165.900	112.500	97.4300	95.6100	0.3500	0.2990	0.3100	0.0080	0.0015	0.0300
16	0.0000	0.0000	0.00000	0.00000	0.00000	0.00000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
17	753.382	702.133	708.469	472.674	440.794	402.504	4.2780	4.0390	3.902	0.1130	0.2070	0.4550
18	725.318	660.953	673.323	450.869	410.959	383.565	6.5740	6.0180	5.9160	0.1630	0.3050	0.6360
19	25.1800	38.6200	32.5000	15.6000	23.9300	13.2600	0.0080	0.0150	0.0100	0.0010	0.0010	0.0030
20	569.216	539.385	535.448	353.658	328.959	300.887	0.7170	0.6750	0.6540	0.0180	0.0340	0.0760
21	46.7200	22.4600	58.9400	28.9500	13.9200	24.6100	0.0130	0.0050	0.0160	0.0010	0.0010	0.0050
22	267.800	248.500	267.600	165.900	154.000	141.100	1.4240	1.3380	1.3820	0.0410	0.0740	0.2110
Σ	8876.93	8297.88	8499.24	5531.94	5165.84	4706.11	67.353	65.222	79.130	2.4789	4.0326	10.810

2. Masukan Parameter *Simulated Annealing* (SA)

Masukan parameter SA ini akan diolah oleh software untuk memperoleh hasil yang diinginkan.

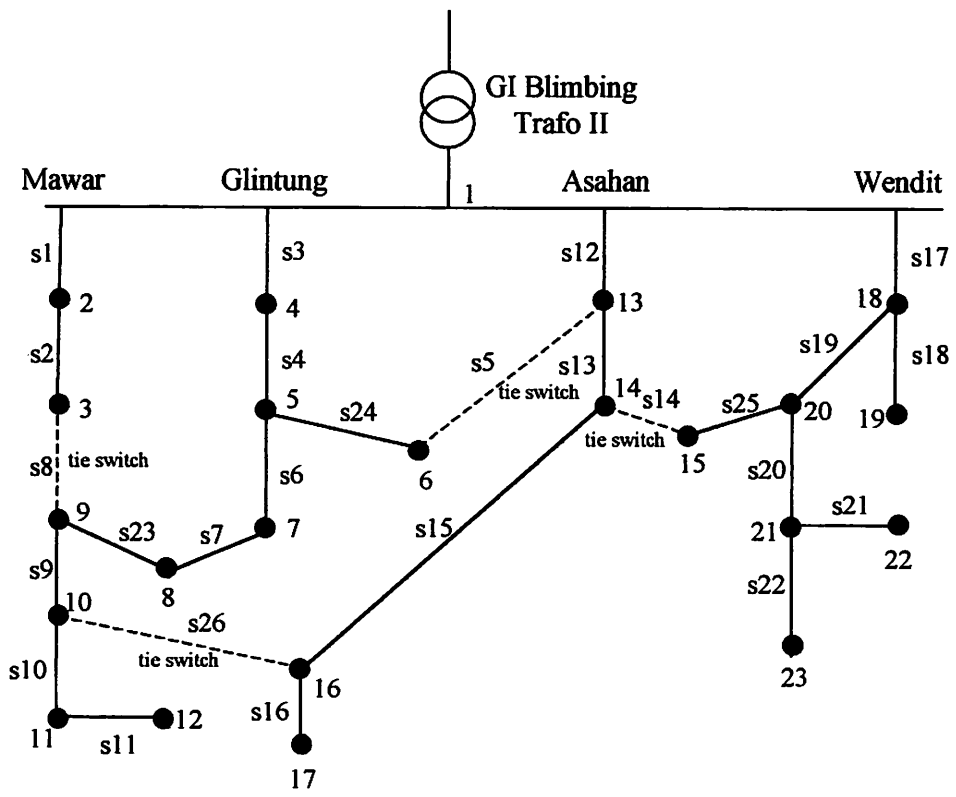
Data masukan parameter SA ini berupa:

- Konstanta Pendingin : 0,05
- Konstanta XO : 0,95
- Konstanta Flip : 0,1

Setelah semua parameter dimasukkan, maka akan diperoleh hasil outputnya, yang dapat dilihat pada gambar 4.9.

Sesuai dengan hasil tersebut, *switch* 3 – 4, *switch* 5 – 16, *switch* 12 – 13, dan *switch* 14 – 15 harus dibuka. Dari gambar 4.2, maka *switch* 3 – 4, *switch* 12 – 13, dan *switch* 14 – 15 yang dibuka, sedangkan *switch* 5 – 16 merupakan *switch normally open*.

Pembukaan *tie switch* 3 – 4, 5 – 16, dan 14 – 15 masing-masing harus diikuti oleh penutupan *switch* 4 – 11, 9 – 13, dan 15 – 20. Sehingga jaringan mesh pada gambar 4.3 menjadi jaringan radial seperti pada gambar 4.12.



Gambar 4.12. Konfigurasi Jaringan Radial Setelah Rekonfigurasi

Hasil aliran daya setelah proses rekonfigurasi dapat dilihat pada tabel 4.3 berikut:

TABEL 4.3**HASIL PERHITUNGAN SETELAH REKONFIGURASI**

Bus	Daya Aktif (kW)			Daya Reaktif (kVAR)			Rugi Daya Aktif (kW)			Rugi Daya Reaktif (kVAR)		
	R	S	T	R	S	T	R	S	T	R	S	T
1	699.137	730.482	732.224	434.002	453.48	369.19	11.20	12.322	11.312	0.884	1.192	2.571
2	271.800	298.400	303.100	168.400	184.900	143.20	0.959	1.138	1.039	0.081	0.113	0.277
3	723.059	699.464	726.757	449.462	435.000	382.47	3.086	3.074	3.002	0.296	0.363	0.67
4	604.509	598.232	621.251	374.690	370.910	318.65	1.803	1.854	1.794	0.176	0.221	0.426
5	176.500	203.000	208.400	109.400	125.800	93.000	0.454	0.573	0.518	0.05	0.072	0.170
6	257.871	247.780	260.578	159.796	153.550	135.82	0.141	0.154	0.187	-0.358	-0.327	-0.30
7	114.600	127.000	138.200	71.0300	78.7300	60.380	0.028	0.035	0.048	-0.065	-0.072	-0.062
8	724.341	650.897	673.057	451.179	405.420	383.51	9.993	8.999	8.999	0.887	1.022	1.534
9	428.835	379.680	376.535	266.178	235.590	226.25	2.687	2.369	2.291	0.216	0.247	0.293
10	240.434	214.076	210.353	149.285	132.890	126.86	0.236	0.209	0.200	0.005	0.01	0.014
11	172.900	158.300	148.900	107.200	98.1200	91.100	0.337	0.31	0.282	0.006	0.013	0.017
12	379.991	311.656	300.274	235.982	193.560	200.61	1.183	0.935	0.904	0.013	0.037	-0.01
13	223.348	196.339	204.929	138.656	121.880	117.86	0.338	0.294	0.300	0.008	0.016	0.030
14	4.07000	3.10100	3.10100	2.52200	1.92200	2.1440	0.000	0.000	0.000	0.000	0.000	0.000
15	181.500	157.200	165.900	112.500	97.4300	95.610	0.349	0.298	0.309	0.008	0.016	0.030
16	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.000	0.000	0.000	0.000	0.000	0.000
17	753.233	701.931	667.051	472.290	440.250	401.90	4.306	4.060	3.699	0.052	0.143	0.250
18	25.1800	38.6200	32.5000	15.6000	23.9300	13.260	0.009	0.018	0.012	0.001	0.001	0.004
19	725.584	661.166	632.657	451.455	41.4730	384.15	5.779	5.283	4.884	0.057	0.179	0.273
20	569.478	529.594	535.727	354.234	329.460	301.47	0.728	0.685	0.664	0.012	0.028	0.070
21	315.121	271.491	327.149	196.167	169.170	166.99	0.623	0.530	0.628	0.013	0.029	0.105
22	267.800	248.500	267.600	165.900	154.000	141.10	1.447	1.360	1.406	0.027	0.061	0.199
Σ	7859.29	7426.89	7538.24	4885.91	4247.50	4155.5	45.686	44.50	42.478	2.369	3.364	6.561

Dari kedua hasil perhitungan (tabel 4.2 dan tabel 4.3) didapat perbandingan rugi-rugi saluran seperti dalam tabel 4.4.

TABEL 4.4
PERBANDINGAN RUGI-RUGI DAYA

Rugi-rugi Sebelum						Rugi-rugi Sesudah						Reduksi Rugi-rugi						Prosentase					
P (kW)			Q (kVAR)			P (kW)			Q (kVAR)			P (kW)			Q (kVAR)			P (%)			Q (%)		
R	S	T	R	S	T	R	S	T	R	S	T	R	S	T	R	S	T	R	S	T	R	S	T
87.353	85.222	79.13	2.4789	4.0325	10.81	45.686	44.5	42.478	2.369	3.364	6.561	41.667	40.722	36.652	0.1099	0.6685	4.249	47.7	47.8	46.3	4.4	16.6	39.3

BAB V

PENUTUP

5.1. Kesimpulan

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

1. Melakukan rekonfigurasi pada penyulang, yaitu mengubah status *switch* 3 – 4, *switch* 5 – 16, dan *switch* 14 – 15 yang semula statusnya *normally close* menjadi *normally open* dan *switch* 4 – 11, *switch* 9 – 13, dan *switch* 15 – 20 yang semula statusnya *normally open* menjadi *normally close*.
2. Daya aktif dan daya reaktif yang diperoleh dengan proses load flow adalah:
 - a. Sebelum rekonfigurasi:

• Fasa R	: 8876,93 kW	5531,94 kVAR
• Fasa S	: 8297,88 kW	5165,84 kVAR
• Fasa T	: 8499,24 kW	4706,11 kVAR
 - b. Setelah rekonfigurasi:

• Fasa R	: 7859,29 kW	4885,91 kVAR
• Fasa S	: 7426,89 kW	4247,50 kVAR
• Fasa T	: 7536,24 kW	4155,50 kVAR
3. Adapun rugi-rugi saluran yang diperoleh adalah:
 - a. Rugi-rugi saluran sebelum rekonfigurasi:

• Fasa R	: 87,353 kW	2,4789 kVAR
• Fasa S	: 85,222 kW	4,0325 kVAR

- Fasa S : 85,222 kW 4,0325 kVAR
 - Fasa T : 79,130 kW 10,810 kVAR
- b. Rugi-rugi saluran setelah rekonfigurasi:
- Fasa R : 45,686 kW 2,369 kVAR
 - Fasa S : 44,500 kW 3,364 kVAR
 - Fasa T : 42,478 kW 6,561 kVAR
- c. Reduksi rugi-rugi saluran:
- Fasa R : 41,667 kW 0,1099 kVAR
 - Fasa S : 40,722 kW 0,6685 kVAR
 - Fasa T : 36,652 kW 4,2490 kVAR

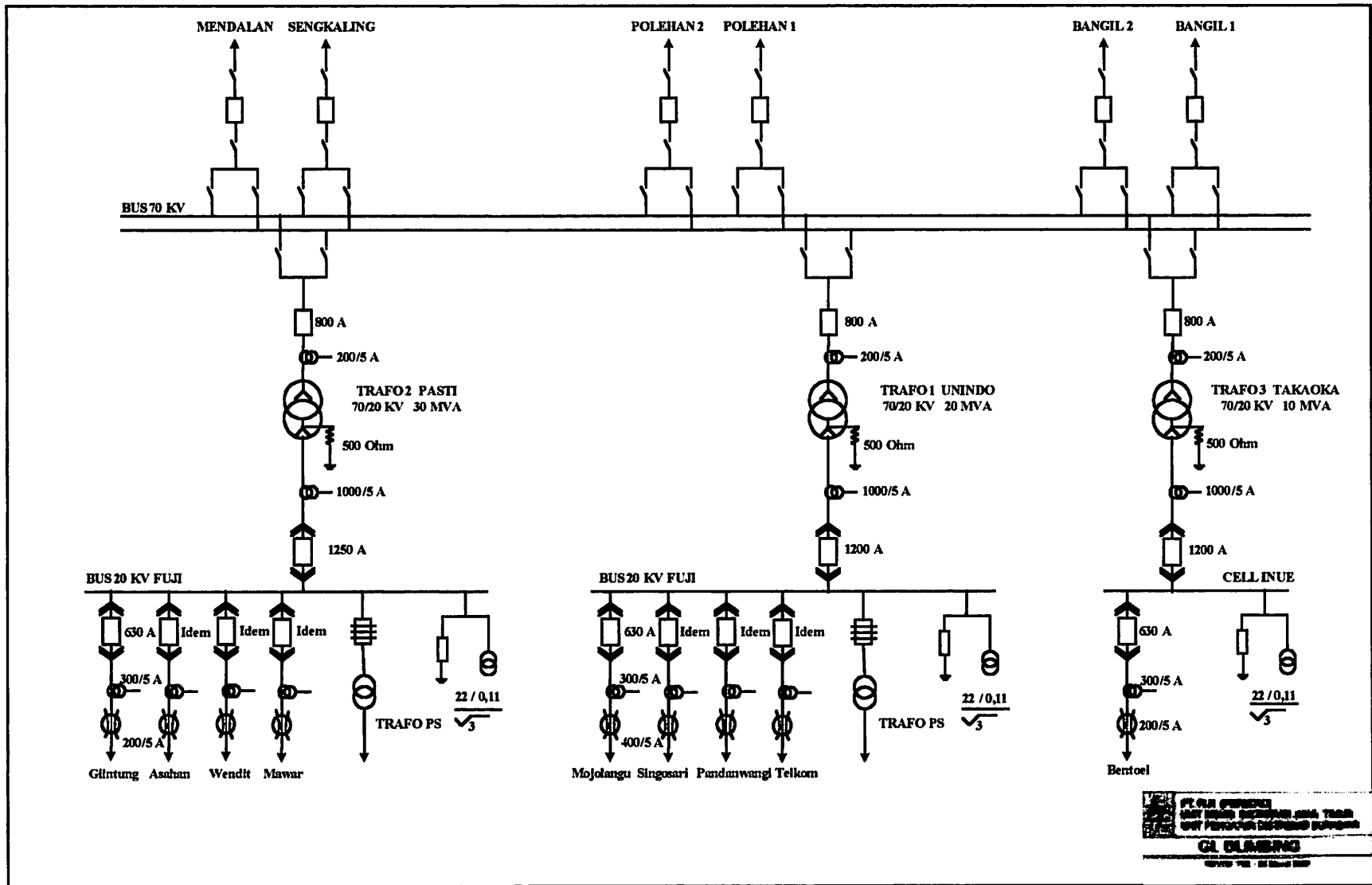
5.2. Saran

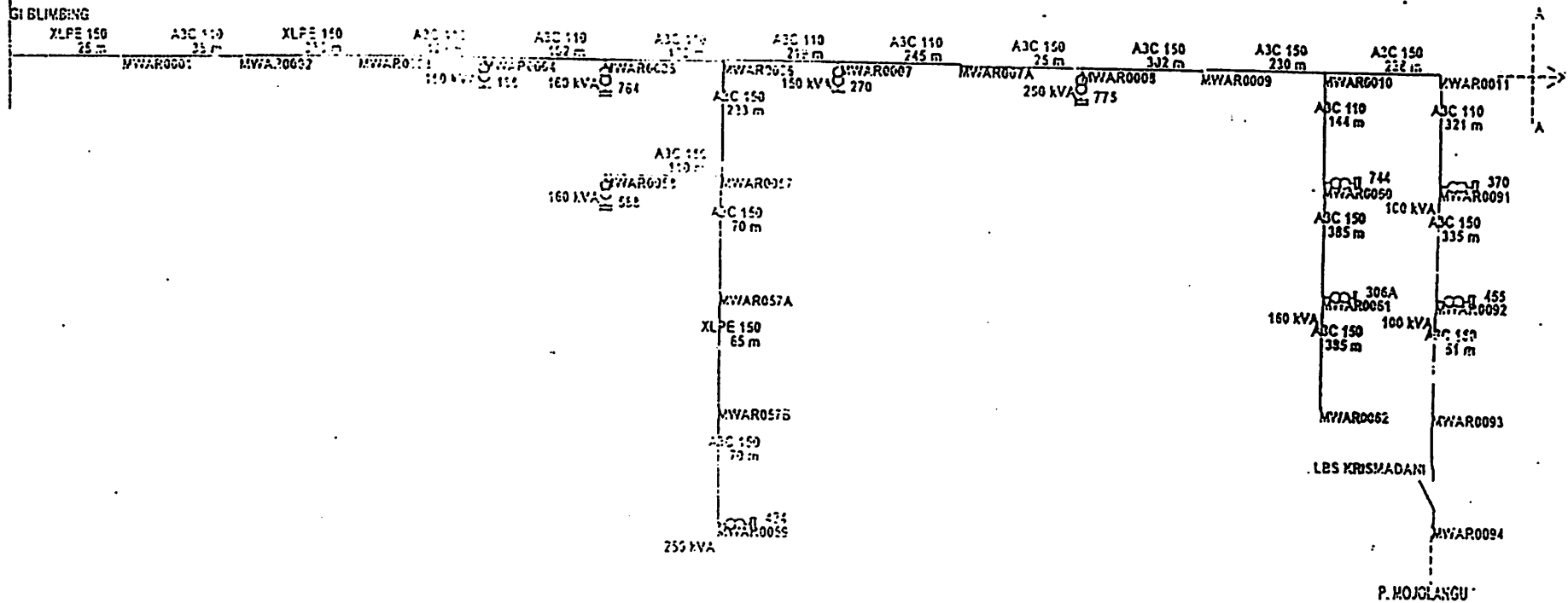
Dengan diperoleh hasil seperti pada kesimpulan, maka rekonfigurasi dengan menggunakan metode *Simulated Annealing* (SA) dapat dijadikan sebagai masukan bagi instansi-instansi terkait, terutama GI Blimbing Malang.

DAFTAR PUSTAKA


1. Ir. Hasan Basri, "*Diktat Kuliah Sistem Distribusi*," ISTN Jakarta.
2. John J. Grainger and William D. Stevenson, JR., "*Power System Analysis*," Mc. Graw-Hill International Editions Electrical Engineering Series.
3. Ray D. Zimmerman and Hsiao-Dong Chiang, "*Fast Decoupled Power Flow for Unbalanced Radial Distribution Systems*," IEEE/PES Winter Meeting, 1995.
4. Ray D. Zimmerman, "*Network Reconfiguration For Loss Reduction in Three-Phase Power Distribution Systems*," Cornell University, May 1992.
5. Bradley J. Bucham and Casey Lambert, "*Simulated Annealing Applications*," Mechanical Engineering Department, University of Victoria, 18 November 1999.

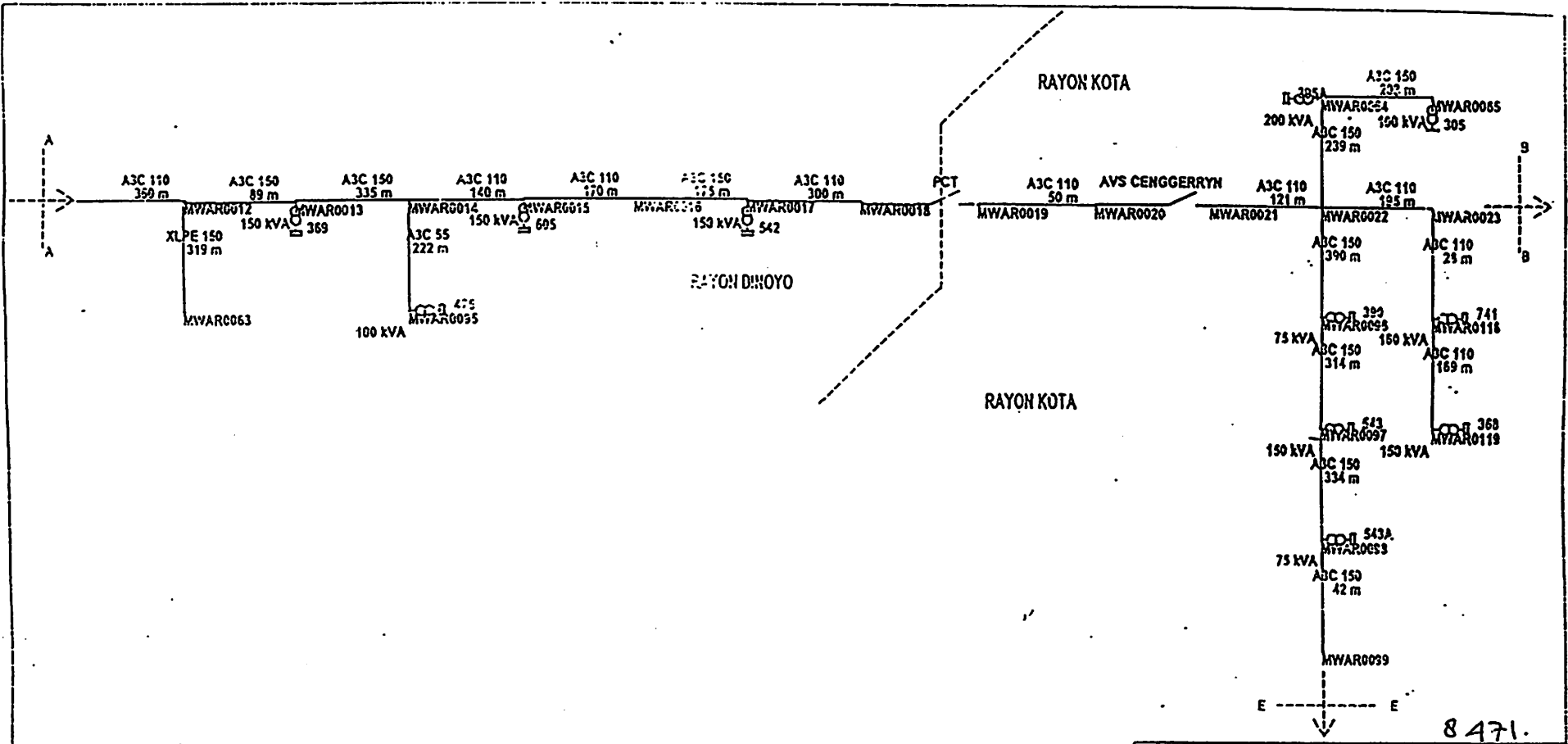
LAMPURAN




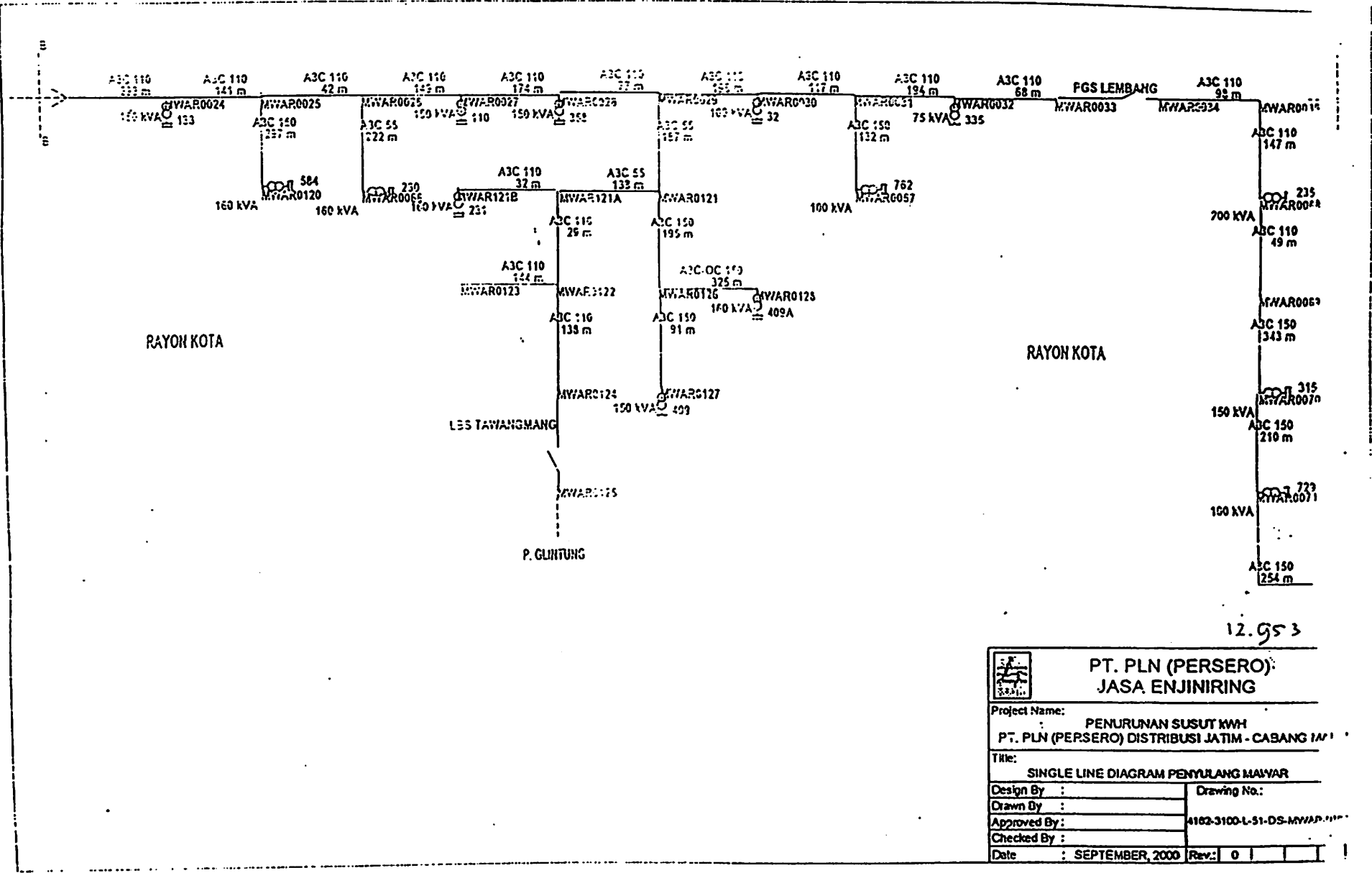


A.276


 PT. PLN (PERSERO) JASA ENJINIRING	
Project Name: PENURUNAN SUSUT KWH PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG MALANG	
Title: SINGLE LINE DIAGRAM PENYULANG MAYAR	
Design By : _____	Drawing No.:
Drawn By : _____	4102-3100-L-51-DS-MWAR-015-01-0
Approved By : _____	
Checked By : _____	
Date : SEPTEMBER, 2000	Rev.: 0

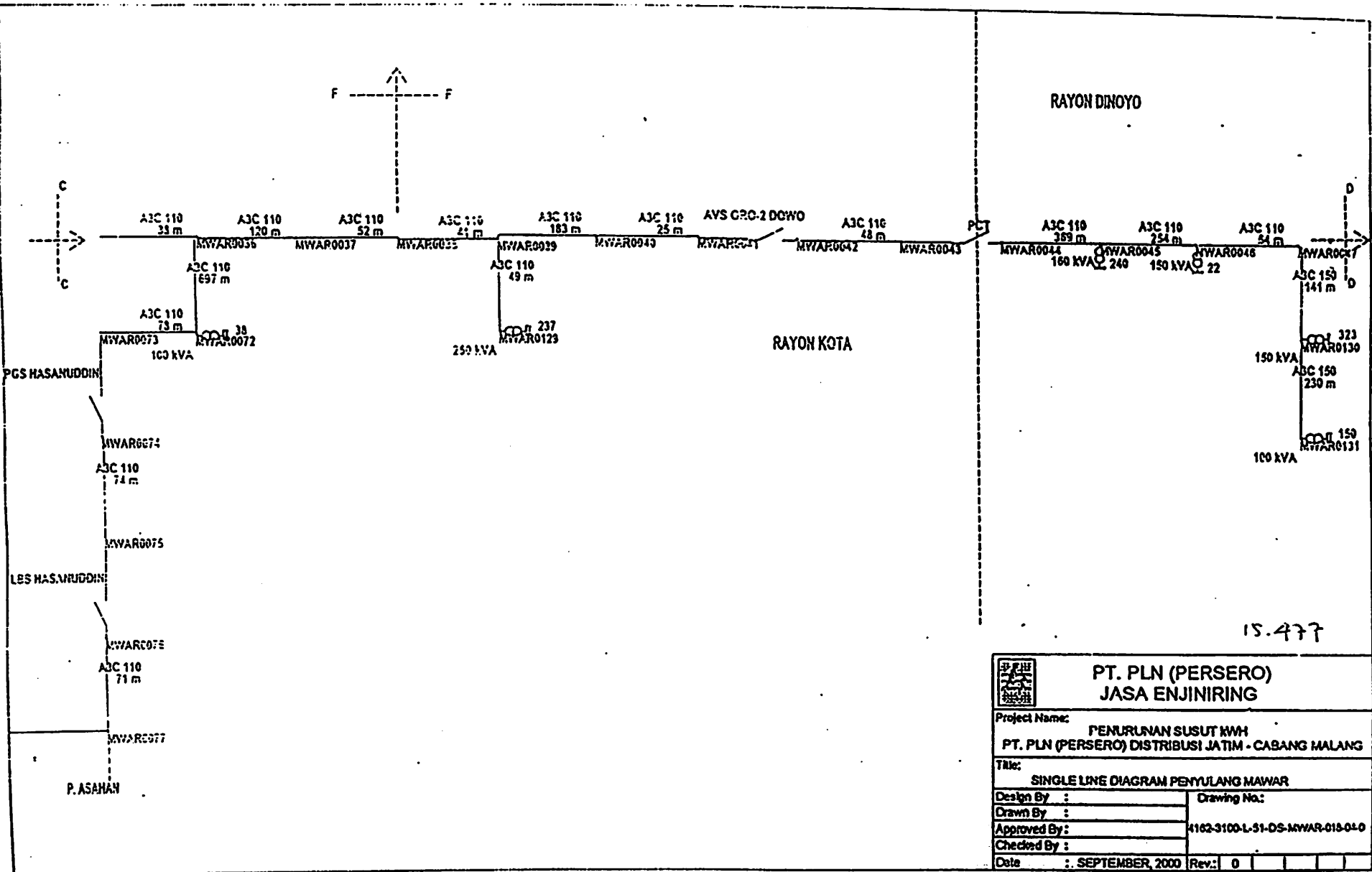



 PT. PLN (PERSERO) JASA ENJINIRING	
Project Name: PENJURUNAN SUSUT KWH PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG MALANG	
Title: SINGLE LINE DIAGRAM PENYULANG MAWAR	
Design By : _____	Drawing No.: _____
Drawn By : _____	4162-3100-L-31-DS-MWAR-015-C2-0
Approved By: _____	
Checked By : _____	
Date : SEPTEMBER, 2000	Rev.: 0

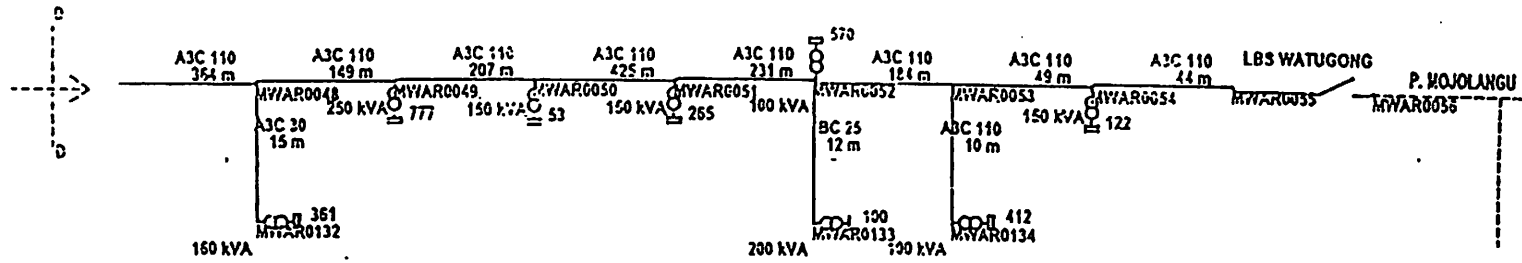


1:2.5


 PT. PLN (PERSERO): JASA ENJINIRING	
Project Name:	
PENURUNAN SUSUT KWH PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG JATIM	
Title:	
SINGLE LINE DIAGRAM PENYULANG MAWAR	
Design By :	Drawing No.:
Drawn By :	4162-3100-L-51-DS-MWAR-111
Approved By :	
Checked By :	
Date : SEPTEMBER, 2000	Rev.: 0

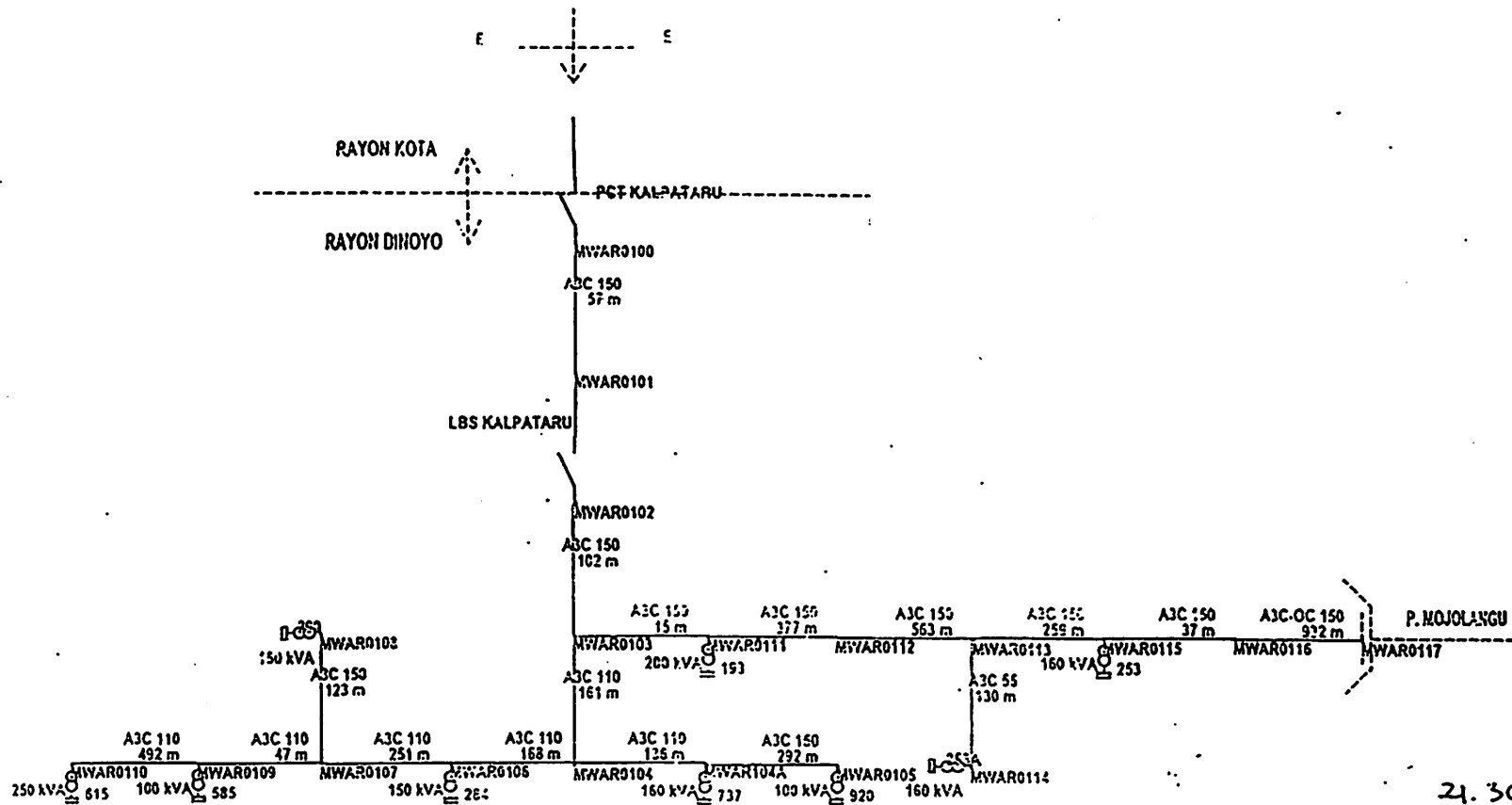


 PT. PLN (PERSERO) JASA ENJINIRING	
Project Name: PENURUNAN SUSUT KWH PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG MALANG	
Title: SINGLE LINE DIAGRAM PENYULANG MAWAR	
Design By :	Drawing No.:
Drawn By :	4162-3100-L-51-DS-MWAR-018-01-0
Approved By :	
Checked By :	
Date : SEPTEMBER, 2000	Rev.: 0



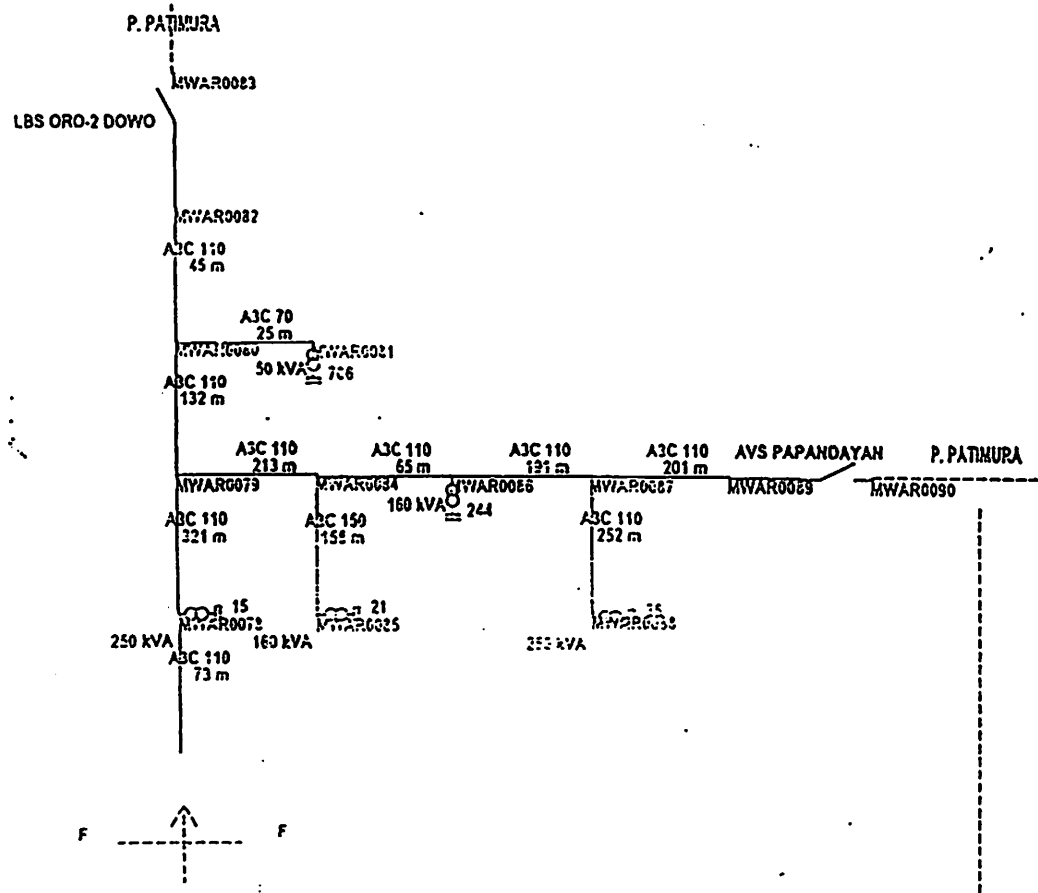
17.167

 PT. PLN (PERSERO) JASA ENJINIRING	
Project Name: PENURUNAN SUSUT KWH PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG MALANG	
Title: SINGLE LINE DIAGRAM PENYULANG MAWAR	
Design By :	Drawing No.:
Drawn By :	4103-3100-L-51-DS-MWAR-01&CC-0
Approved By :	
Checked By :	
Date : SEPTEMBER, 2000	Rev: 0



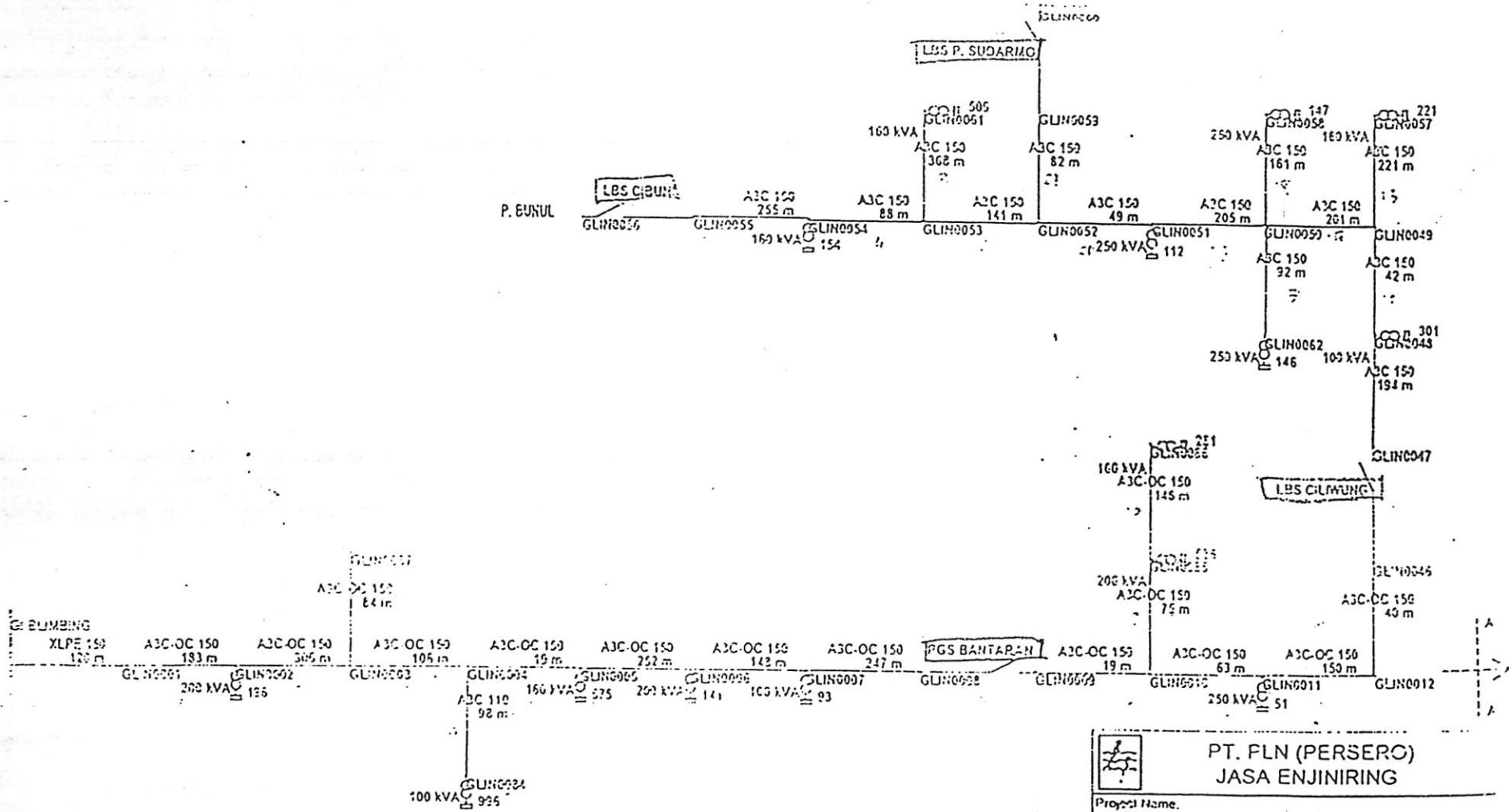
21.309


 PT. PLN (PERSERO) JASA ENJINIRING	
Project Name: PENURUNAN SUSUT KWH PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG MALANG	
Title: SINGLE LINE DIAGRAM PENYULANG MAWAR	
Design By :	Drawing No.:
Drawn By :	4162-3100-L-51-DS-MWAR-DIS-05-0
Approved By :	
Checked By :	
Date : SEPTEMBER, 2000	Rev.: 0

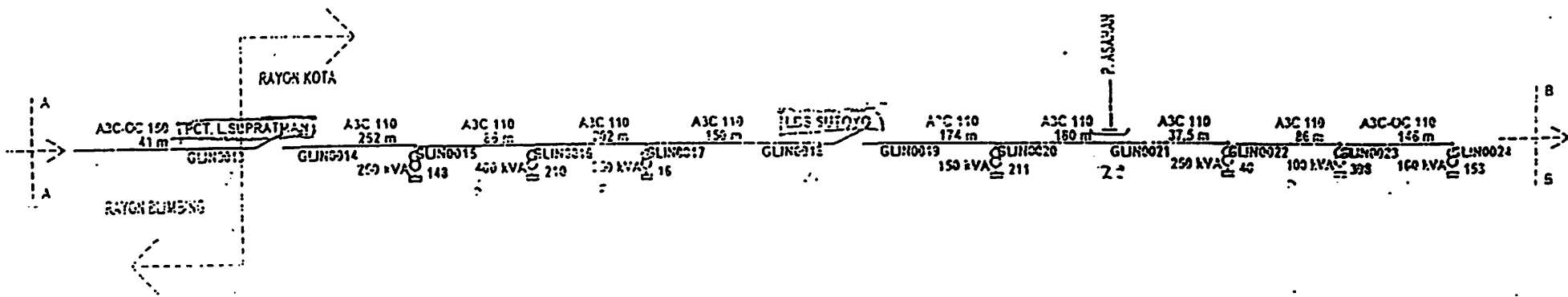



RAYON DINOYO

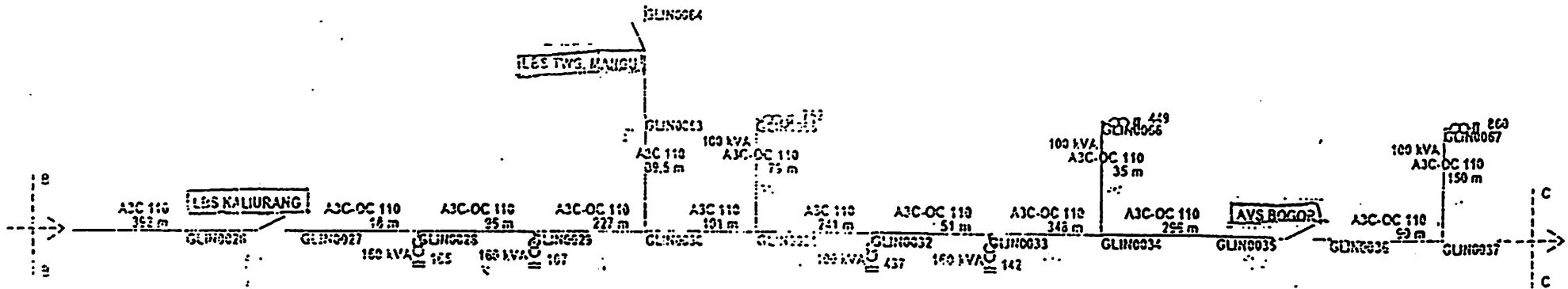
PT. PLN (PERSERO) JASA ENJINIRING	
Project Name: PENURUNAN SUSUT KWH PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG MALANG	
Title: SINGLE LINE DIAGRAM PENYULANG MAWAR 22,985	
Design By :	Drawing No.:
Drawn By :	4162-3150-1-51-05-MWAR-013-07-0
Approved By :	
Checked By :	
Date : SEPTEMBER, 2000	Rev.: 0



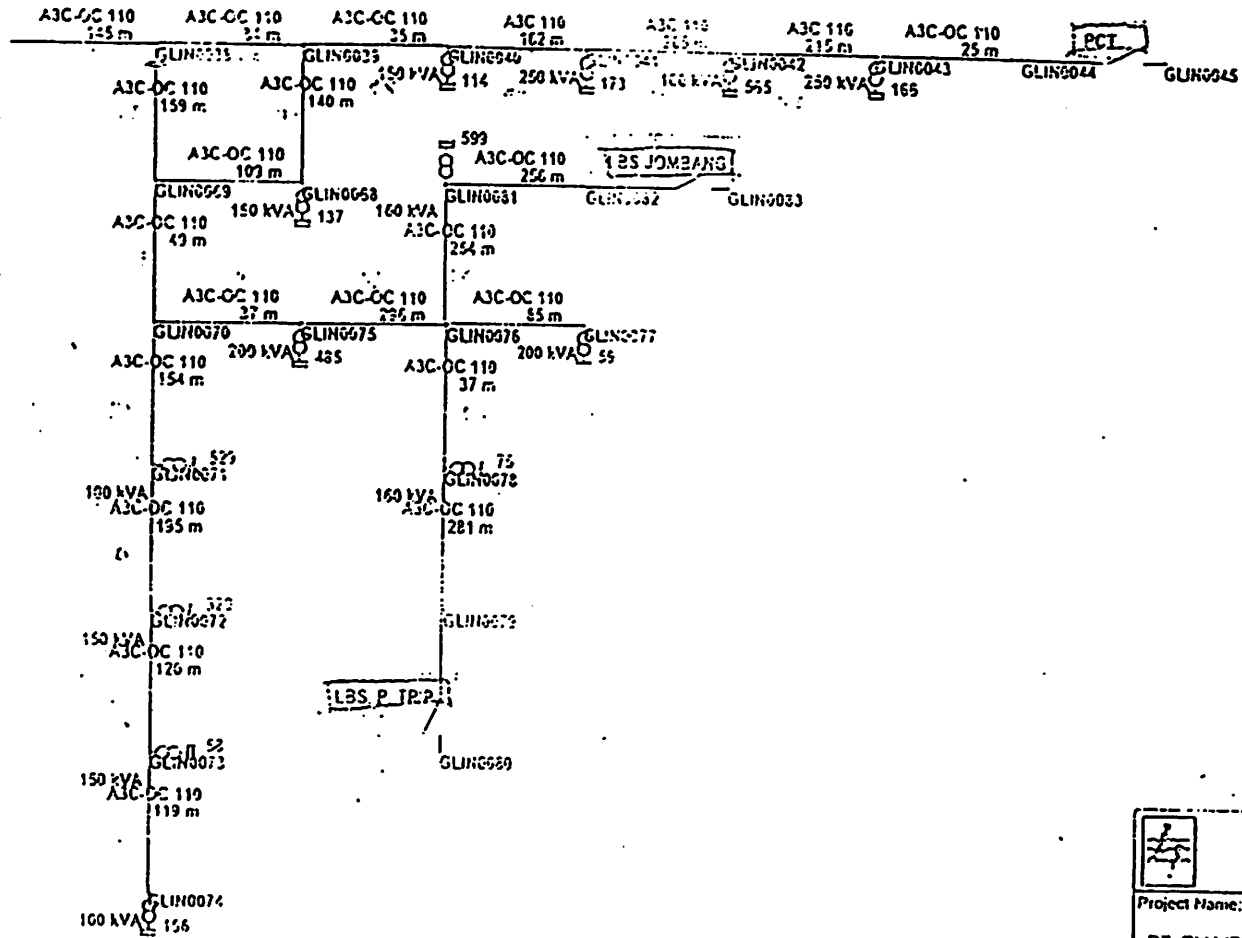
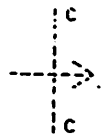
 PT. PLN (PERSERO) JASA ENJINIRING	
Project Name: PENURUNAN SUSUT KWH PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG MALANG	
Title: SINGLE LINE DIAGRAM PENYULANG GUMUNG	
Drawn By: _____	Checked By: _____
Approved By: _____	4162-3100-L-51-DS-GLTU-012/010
Check J By: _____	
Date: SEPTEMBER 2009	Draw: 0




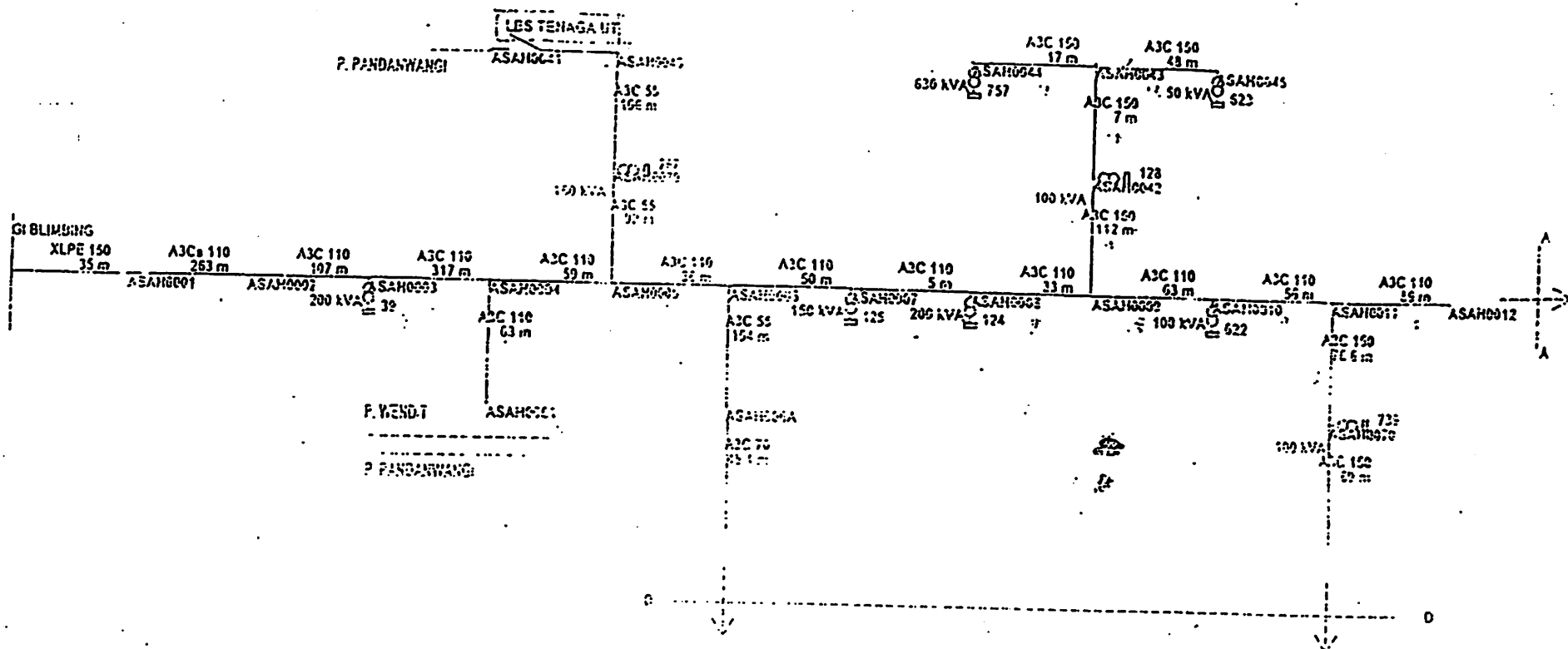
 PT. PLN (PERSERO) JASA ENJINIRING	
Project Name:	
PENURULAN SUSUT KWH PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG MALANG	
Title:	
SINGLE LINE DIAGRAM PENYULANG GUNTING	
Design By	Drawing No.
Drawn By	
Approved By	4112-31031-01-02-CL-016-000
Checked By	
Date	: SEPTEMBER 2000 (Rev.: 0)



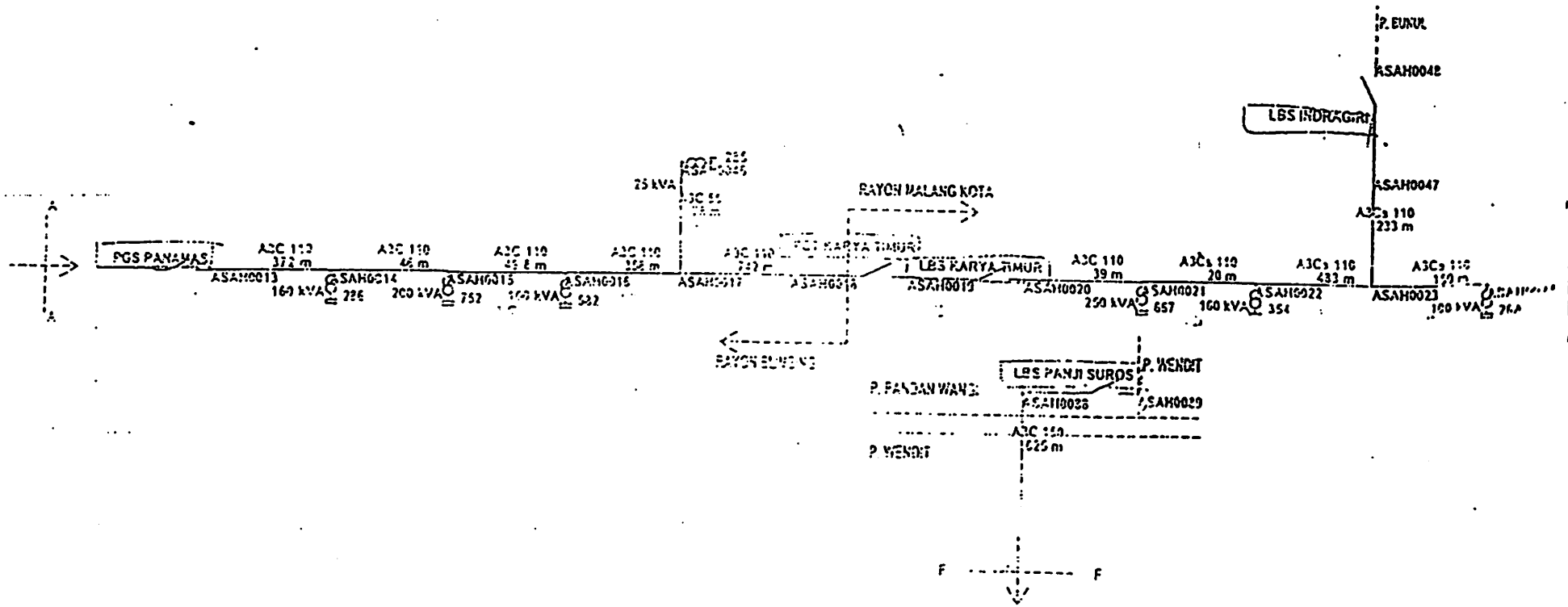
 PT. PLN (PERSERO) JASA ENJINIRING	
Project Name:	
PENUPUNAN SUSUT KWH PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG MALANG	
Title:	
SINGLE LINE DIAGRAM PENYULANG GUNTIF.3	
Design By:	Drawing No.:
Drawn By:	
Approved By:	4162 31001 01 02 011010101
Checked By:	
Date:	SEPTEMBER, 2000
Rev:	0



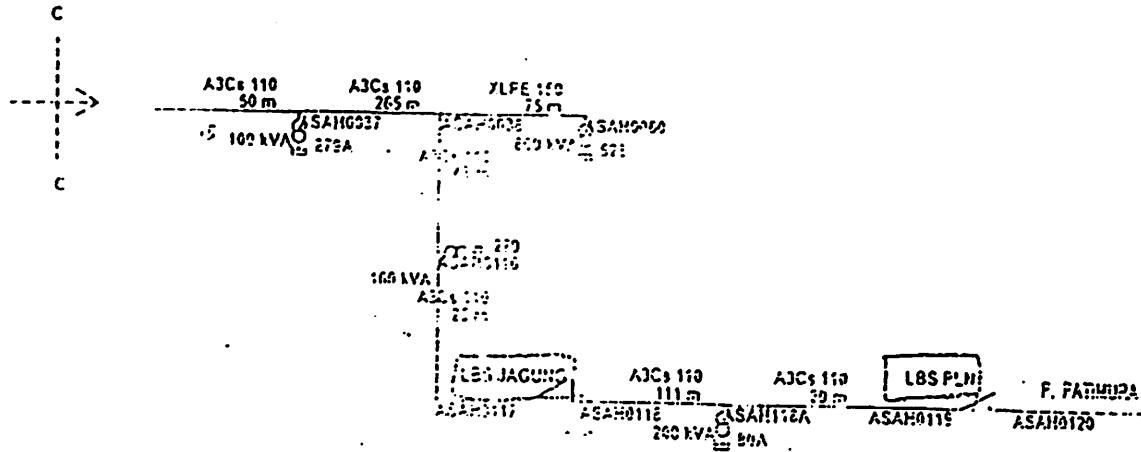
 PT. PLN (PERSERO) JASA ENJINIRING	
Project Name: PENURUMAN SUSUT KVH PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG	
Title: SINGLE LINE DIAGRAM PENYULANG GUNTUR	
Design By :	Drawing No :
Drawn By :	3102 21/7 L 01-03-01
Approved By :	
Checked By :	
Date : SEPTEMBER, 2020	Page : 0



PT. PLN (PERGERO) JASA ENJINIRING	
Project Name: PERUPAHAN SUSUT NYAI PT. PLN (PERGERO) DISTRIKUGI JATIM - CABANG MALAY	
Title: SINGLE LINE DIAGRAM PENYULANG AGAHAN	
Drawn By: _____	Drawing No: _____
Checked By: _____	4162 3079 L-SI-DC-AR-01013 01
Date: 11 Februari 2019	Rev: 0

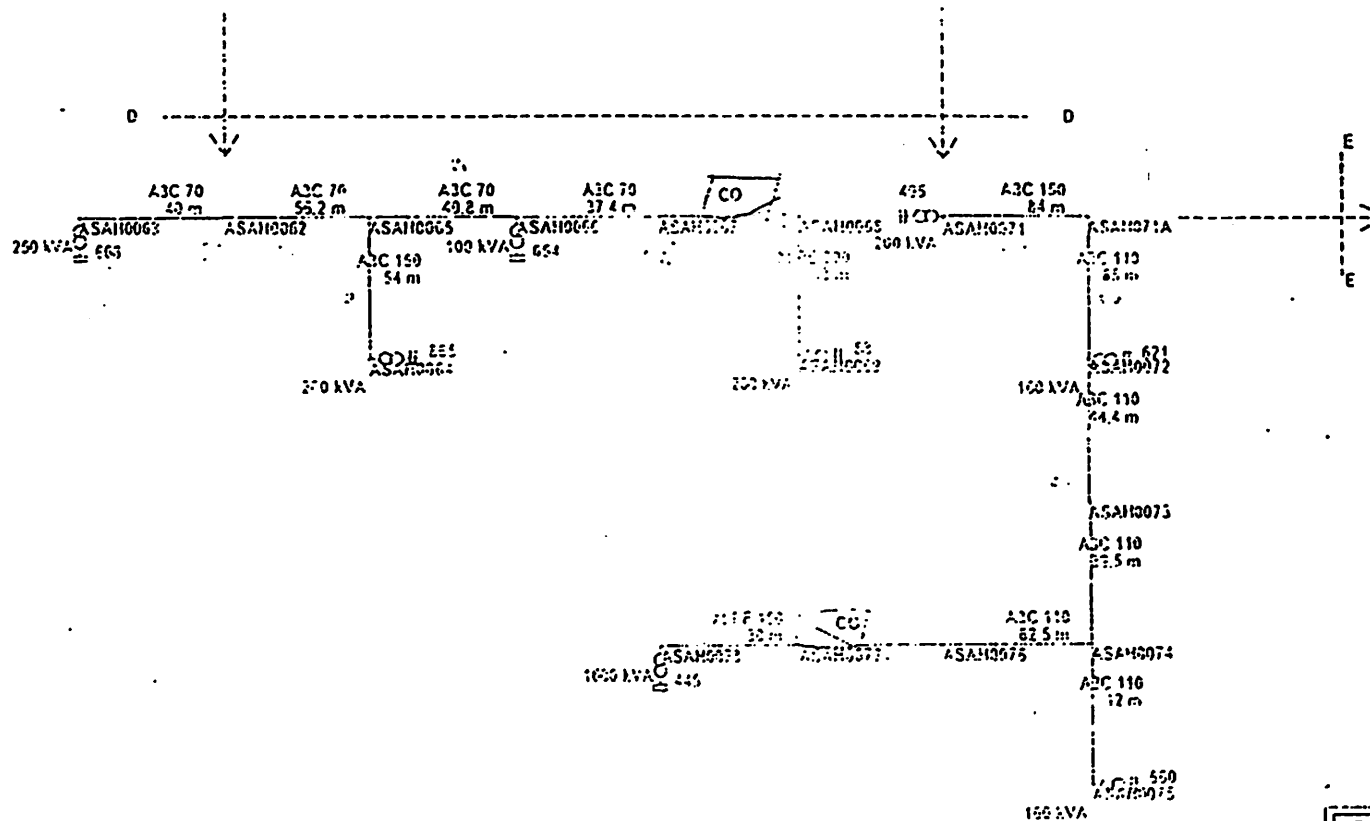



 PT. PLN (PERSERO) JASA ENJINIRING	
Project Name	
PENURUNAN SUSUT VOLTASE PT. PLN (PERSERO) DISTRIBUSI JATIM	
Title	
SINGLE LINE DIAGRAM PENYULUHAN	
Design by	Checked by
Drawn by	Approved by
Approved by	Checked by
Date	SEPTEMBER, 2000 Rev

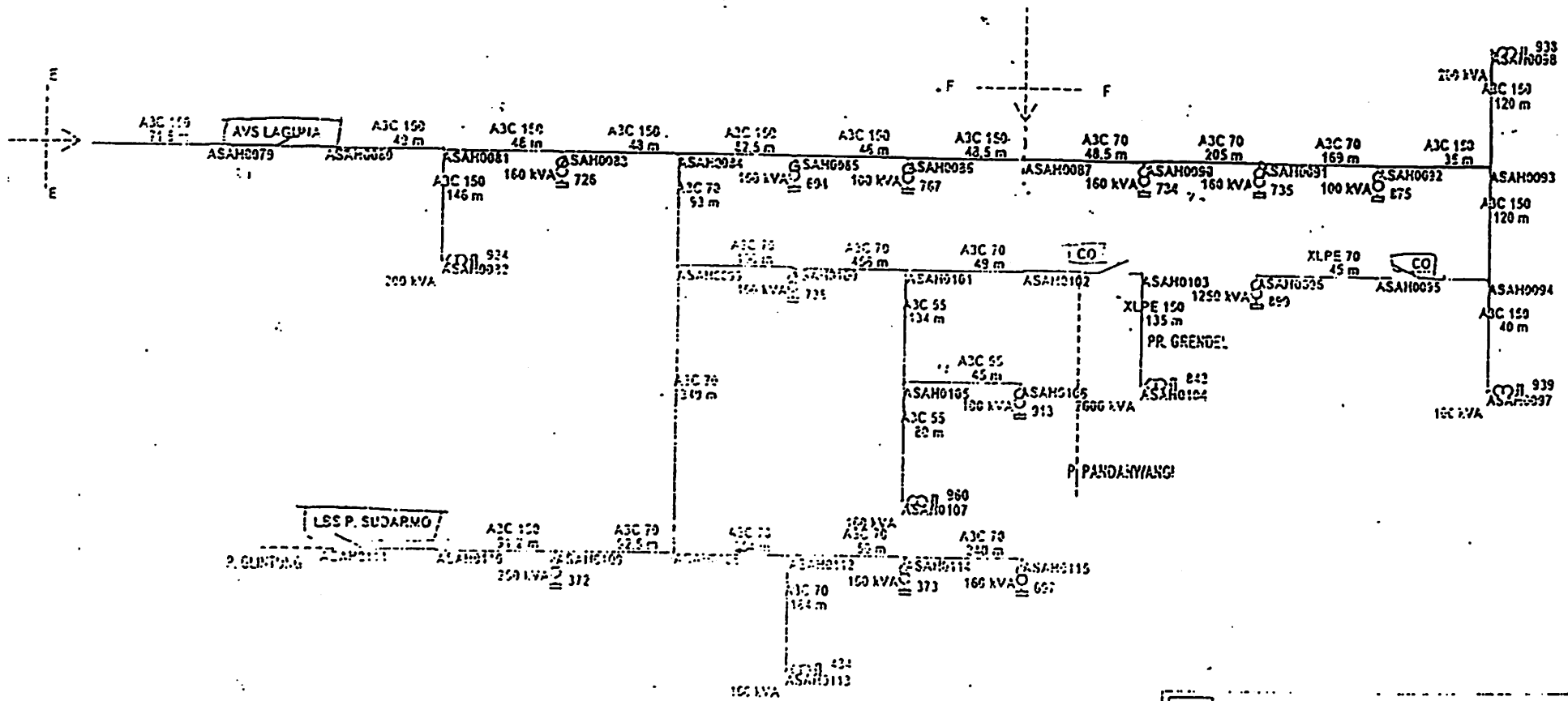


**PT. PLN (PERSERO)
JASA ENJINIRING**

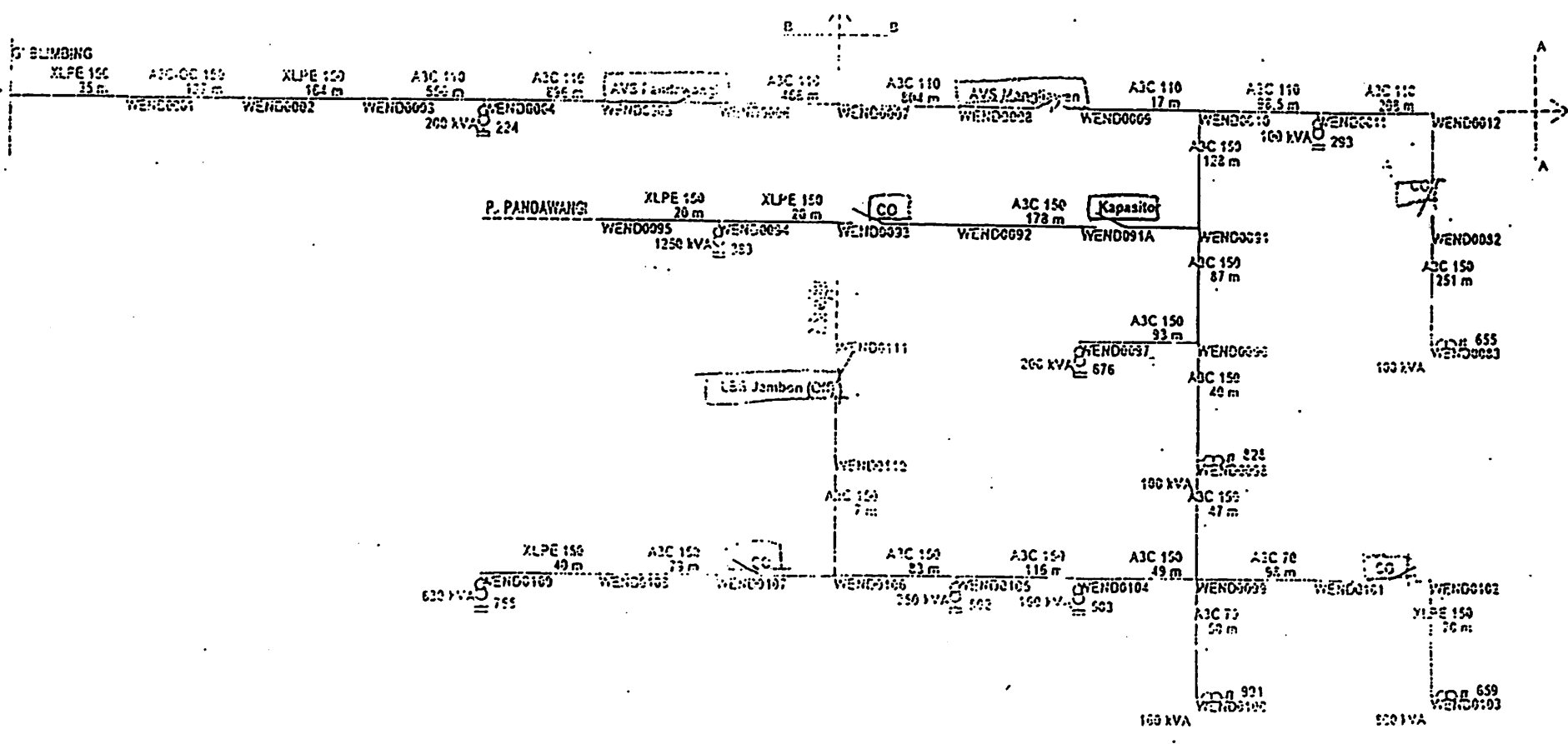
Project Name:	
PENURUNAN SUSUT KWH	
PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG MALANG	
Title:	
SINGLE LINE DIAGRAM PENYULANG ASAHAH	
Design By:	(Signature)
Drawn By:	
Approved By:	4162-3100-L-51-05-ASAH10137
Checked By:	
Date:	SEPTEMBER 2006
Rev:	0



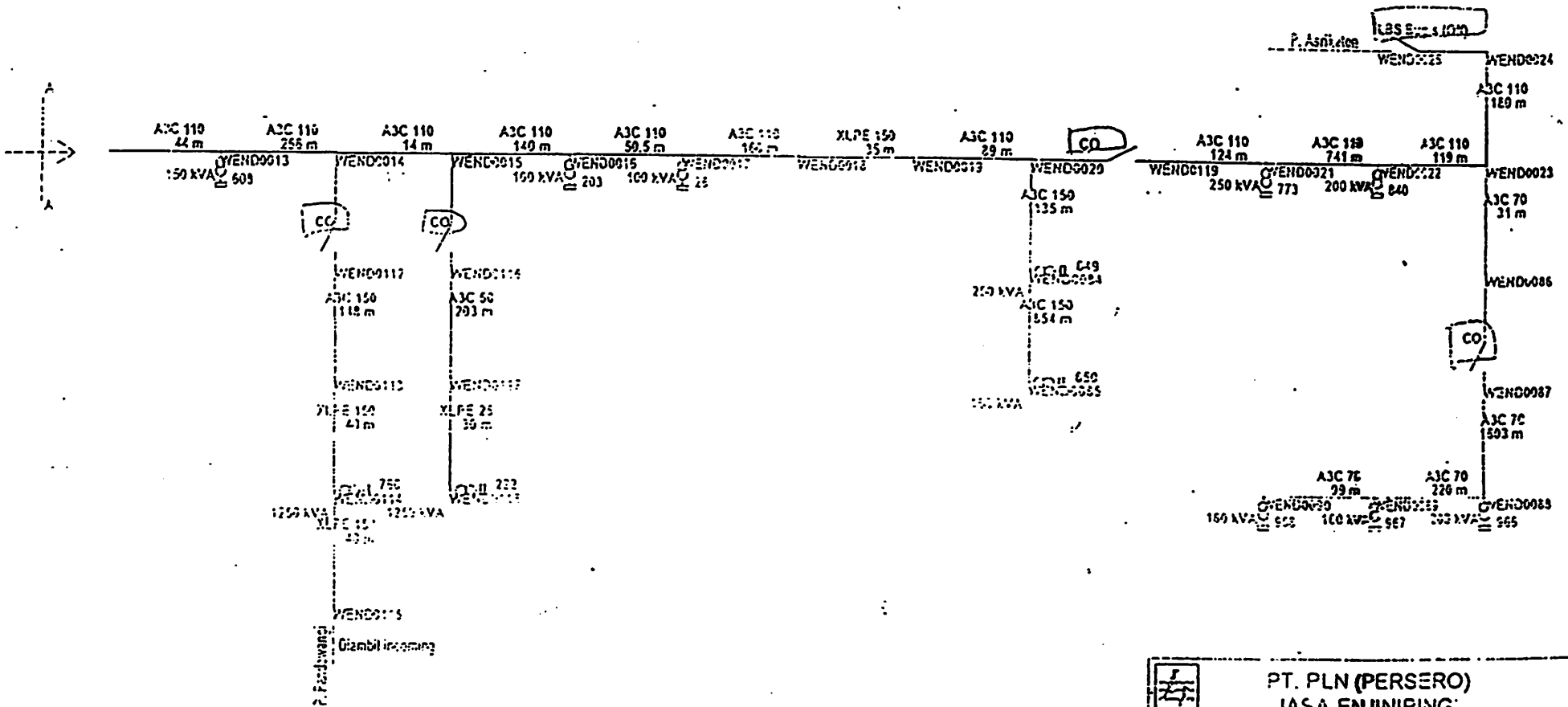
 PT. PLN (PERSERO) JASA ENJINIRING	
Project Name: PENURUNAN SUSUT KWH PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG MALA	
Title: SINGLE LINE DIAGRAM PENYULANG ASAHAN	
Design By	Drawing No
Drawn By	4162-3100-L-51-DS-AS-01013
Approved By	
Checked By	
Date : SEPTEMBER, 2000	Rev: 0



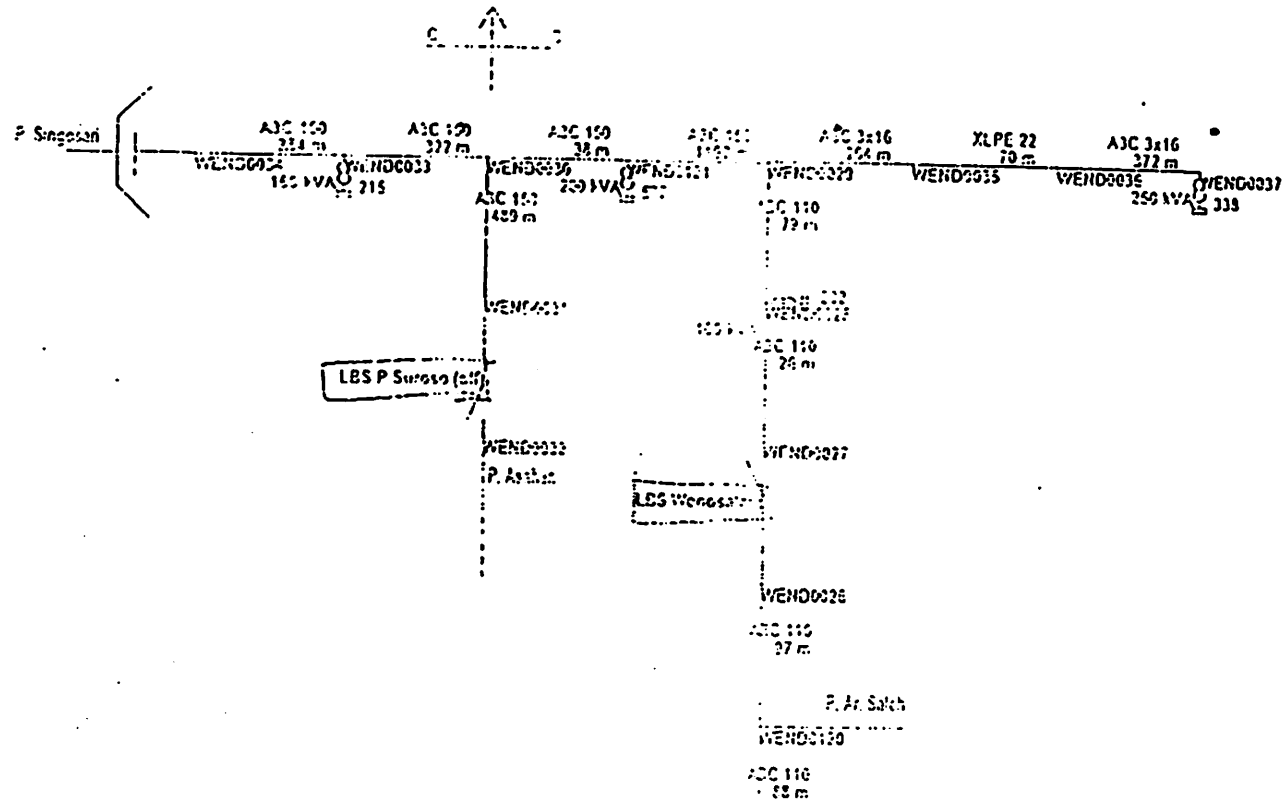
PT. PLN (PERSERO) JASA ENJINIRING	
Project Name: PENURUNAN SUSUT KWH PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG MALANG	
Title: SINGLE LINE DIAGRAM PENYULUHAN ASAMAN	
Designed By:	Drawing No:
Drawn By:	4162-310/1-51-05-AD-01-013 01-0
Approved By:	Checked By:
Date: SEPTEMBER, 2000	Rev: 0



 PT. PLN (PERSERO) JASA ENJINIRING	
Project Name:	
PENURUNAN SUSUT RUMAH PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG MALANG	
Title:	
SINGLE LINE DIAGRAM PEYULANG WENDIT	
Design By	Drawing No
Drawn By	
Approved By	4142-310/1-51-10-5110-0-2-01-0
Checked By	
Date	SEPTEMBER, 2010
Rev	0

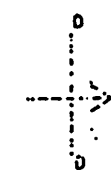
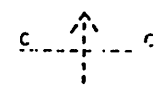
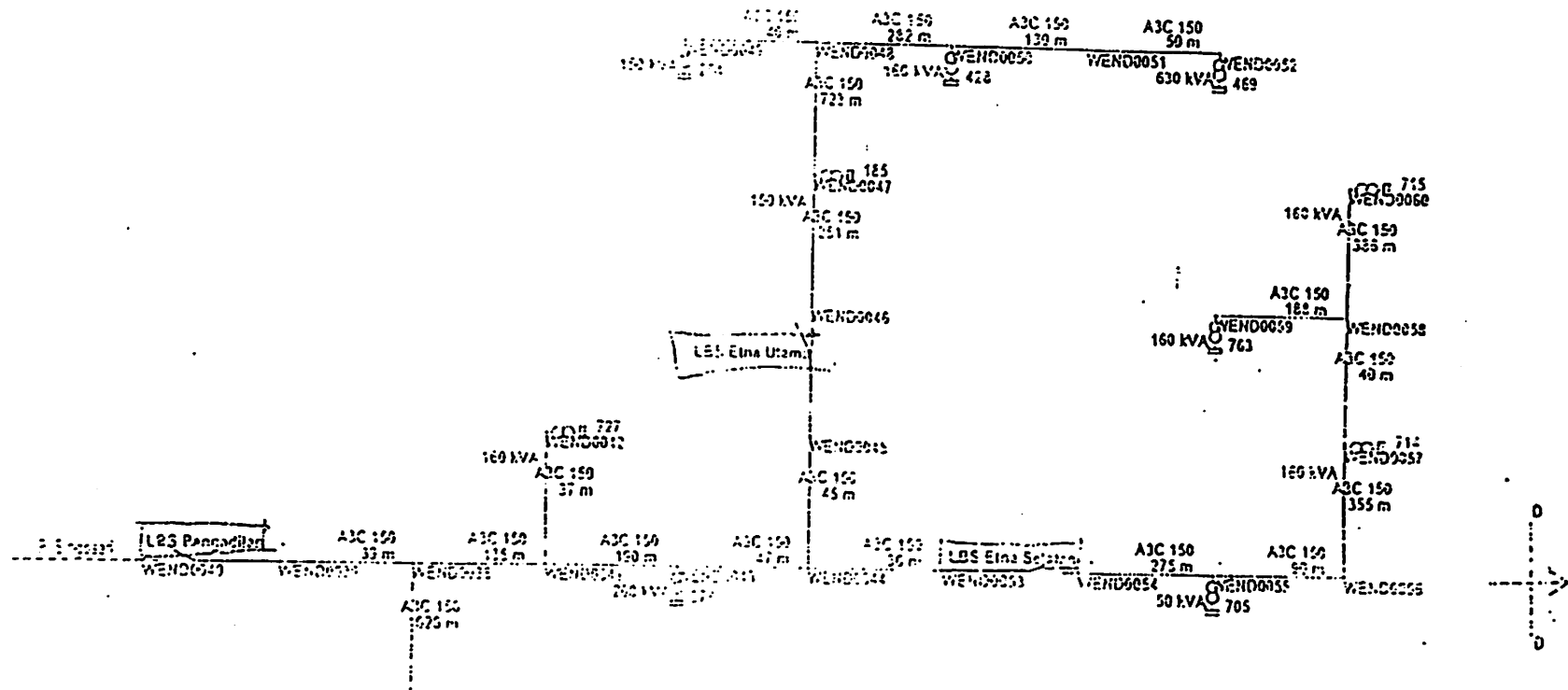



PT. PLN (PERSERO) JASA ENJINIRING	
Project Name: PENURUNAN SUGUT KWH PT. PLN (PERSERO) DISTREKSI JATIM - CABANG MALANG	
Title: SINGLE LINE DIAGRAM PENYULANG WENDIT	
Design By: _____	Drawn By: _____
Checked By: _____	Approved By: _____
Date: SEPTEMBER, 2009	Rev: 0

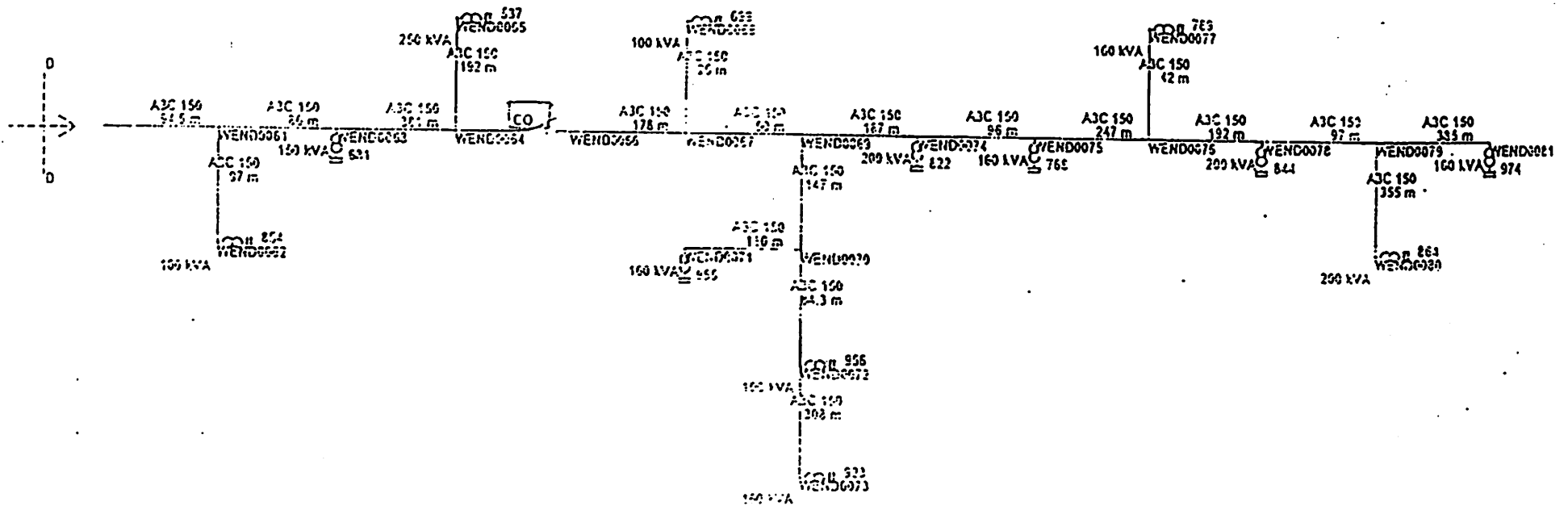



**PT. PLN (PERSERO)
JASA ENJINIRING**

Project Name	
PERIURUNAN SUSUT MWH PT. PLN (PERSERO) DISTRIBUSI JATIM - CANTAP	
Title	
SINGLE LINE DIAGRAM PENYULUANG	
Designed By	Drawing No.
Drawn By	
Approved By	4162-2100-L-5110
Checked By	
Date	SEPTEMBER, 2009 Rev 0



 PT. PLN (PERSERO) JASA ENJINIRING	
Project Name: PENURUNAN SUSUT KWH PT. PLN (PERSERO) DISTRIBUSI JATIM - CABAI # : MAI 117	
Title: SINGLE LINE DIAGRAM PENYULANG V/EI # 41	
Design By	Drawing No
Drawn By
Approved By
Checked By
Date : SEPTEMBER, 2009	Rev: 0



 PT. PLN (PERSERO) JASA ENJINIRING	
Project Name: PENURUHAN SUSUT KWH PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG MALANG	
Title: SINGLE LINE DIAGRAM PENYULANG WENDIT	
Designed By: _____	Drawing No: _____
Drawn By: _____	
Approved By: _____	4120.3100.0.01-05-WEND-012-00.0
Checked By: _____	
Date: SEPTEMBER, 2010	Rev: 0

PENGELOMPOKAN SWITCH

Penyulang	Seksi	Switch	Jumlah beban (kVA)			Jumlah daya aktif (kW)			Jumlah daya reaktif (kVAR)		
			R	S	T	R	S	T	R	S	T
Mawar	1	1	502,4	507,9	504,4	427	431,7	428,7	264,7	267,5	225
	2	2	319,8	351	356,6	271,8	298,4	303,1	168,4	184,9	143,2
	3	3	346,5	318,1	347,9	294,5	270,4	295,7	182,5	167,6	155,1
	4	4	221,6	194,7	195,4	188,3	165,5	166,1	116,7	102,5	99,21
	5	5	79,28	65,47	72,16	67,39	55,65	61,34	41,76	34,49	35,5
	6	6	203,5	186,3	175,2	172,9	158,3	148,9	107,2	98,12	91,1
Glintung	1	7	138,7	118,3	123,3	117,9	100,6	104,8	73,08	62,34	62,11
	2	8	199,5	172,8	178,5	169,6	146,9	151,7	105,1	91,02	89,35
	3	9	168,5	142	143,9	143,2	120,7	122,3	88,77	74,82	75,45
	4	10	134,8	149,5	162,6	114,6	127	138,2	71,03	78,73	60,38
Asahan	1	11	184,2	135,6	112	156,5	115,2	95,22	97,02	71,41	82,47
	2	12	207,7	238,9	245,2	176,5	203	208,4	109,4	125,8	93
	3	13	49,06	45,9	45,77	41,7	39,02	38,9	25,84	24,18	21,97
	4	14	4,788	3,648	3,648	4,07	3,101	3,101	2,522	1,922	2,144
	5	15	213,5	184,9	195,2	181,5	157,2	165,9	112,5	97,43	95,61
	6	16	0	0	0	0	0	0	0	0	0
Wendit	1	17	0	0	0	0	0	0	0	0	0
	2	18	183,3	154,5	161,9	155,8	131,3	137,6	96,56	81,38	82,08
	3	19	29,62	45,43	38,24	25,18	38,62	32,5	15,6	23,93	13,26
	4	20	299	303,4	245,1	254,1	257,9	208,3	157,5	159,8	133,9
	5	21	54,96	26,42	69,34	46,72	22,46	58,94	28,95	13,92	24,61
	6	22	315	292,4	314,9	267,8	248,5	267,6	165,9	154	141,1

	230	160	34	28,49	24,59	54,4	45,58	39,35	139,33	46,24	38,74	33,45	118,4	28,66	24,01	20,73	73,4
	231	160	20,89	23,33	25,91	33,42	37,32	41,45	112,19	28,41	31,72	35,23	95,36	17,61	19,66	21,84	59,1
	305	160	18,59	26,64	23,03	29,75	42,62	36,85	109,22	25,29	36,23	31,32	92,84	15,67	22,45	19,41	57,54
	305A	200	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	335	75	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	358	150	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	368	150	38,09	29,39	26,68	57,14	44,08	40,02	141,24	48,57	37,47	34,02	120,1	30,1	23,22	21,08	74,4
	390	75	45,68	25,64	39,79	34,26	19,23	29,84	83,33	29,12	16,35	25,36	70,83	18,05	10,13	15,72	43,9
	409	150	23,21	22,63	27,3	34,82	33,95	40,95	109,72	29,6	28,86	34,81	93,26	18,34	17,88	21,57	57,8
	409A	160	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	543A	150	23,8	25,84	28,33	35,7	38,76	42,49	116,95	30,35	32,95	36,12	99,41	18,81	20,42	22,38	61,61
	584	160	25,54	24,34	25,95	40,87	38,95	41,52	121,34	34,74	33,11	35,29	103,1	21,53	20,52	21,87	63,92
	741	160	16,31	11,01	22,16	26,09	17,62	35,46	79,17	22,18	14,98	30,14	67,29	13,74	9,282	18,68	41,71
	762	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	15	250	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	21	200	23,96	15,05	19,29	47,91	30,1	38,58	116,59	40,72	25,59	32,79	99,1	25,24	15,86	20,32	61,42
	36	250	20,21	19,35	20,3	50,53	48,38	50,74	149,65	42,95	41,12	43,13	127,2	26,62	25,49	26,73	78,83
	38	100	40,66	26,1	24,72	40,66	26,1	24,72	91,48	34,56	22,19	21,01	77,76	21,42	13,75	13,02	48,19
	235	200	7,52	15,7	12,65	15,04	31,39	25,29	71,72	12,78	26,68	21,5	60,96	7,923	16,54	13,32	37,78
	244	250	6,164	7,724	5,296	15,41	19,31	13,24	47,96	13,1	16,41	11,25	40,77	8,118	10,17	6,975	25,26
	315	150	31,61	119	27,81	47,42	37,62	41,72	126,76	40,31	31,98	35,46	107,7	24,98	19,82	21,98	66,77
	706	50	9,198	3,504	2,19	4,599	1,752	1,095	7,446	3,909	1,489	0,931	6,329	2,423	0,923	0,577	3,922
	729	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	22	150	23,89	29,52	34,13	35,84	44,28	51,19	131,31	30,46	37,64	43,51	111,6	18,88	23,33	26,97	69,17
	240	160	27,15	13,24	13,11	43,44	21,19	20,97	85,6	36,92	18,01	17,82	72,76	22,88	11,16	11,05	45,09
6	53	150	34,49	24,38	27,65	51,74	36,57	41,48	129,79	43,98	31,08	35,26	110,3	27,26	19,26	21,85	68,37
	122	150	28,69	26,29	21,77	43,04	39,43	32,65	115,12	36,58	33,52	27,75	97,85	22,67	20,77	17,2	60,64
	265	150	21,11	13,83	13,08	31,67	20,74	19,62	72,03	26,92	17,63	16,68	61,23	16,68	10,93	10,34	37,94
	361	160	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	412	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	570	100	23	37,32	29,73	23	37,32	29,73	90,05	19,55	31,72	25,27	76,54	12,12	19,66	15,66	47,44
	777	250	21,6	20,88	20,7	54	52,2	51,75	157,95	45,9	44,37	43,99	134,3	28,45	27,5	27,26	83,21

PENYULANG GLINTUNG

Seksi	No. Trafo	Nominal (kVA)	Persen beban (%)			Beban Trafo (kVA)			Total beban (kVA)			Daya aktif (kW)			Total Daya aktif (kW)			Daya reaktif (kVAR)			Total Daya reaktif (kVAR)
			R	S	T	R	S	T	R	S	T	R	S	T	R	S	T	R	S	T	
1	93	100	17,56	13,22	10,94	17,56	13,22	10,94	41,72	14,93	11,24	9,299	35,46	9,25	6,964	5,763	21,98				
	141	200	23,69	24,61	23,46	47,38	49,22	46,92	143,52	40,27	41,84	39,88	122	24,96	25,93	24,72	75,6				
	196	160	36,05	28,04	26,94	57,68	44,86	43,1	145,64	49,03	38,13	36,64	123,8	30,38	23,63	22,7	76,72				
	575	160	10,06	6,9	13,94	16,1	11,04	22,31	49,45	13,69	9,384	18,96	42,03	8,481	5,816	11,75	26,05				
2	16	250	24,99	22,35	20,86	62,48	55,88	52,14	170,5	53,11	47,5	44,32	144,9	32,91	29,44	27,47	89,82				
	51	250	21,76	20,43	21,05	54,41	51,07	52,63	158,11	46,25	43,41	44,74	134,4	28,66	26,9	27,72	83,29				
	148	250	16,77	14,36	12,76	41,92	35,9	31,89	109,71	35,63	30,52	27,11	93,25	22,08	18,91	16,8	57,79				
	210	250	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
3	281	160	25,46	18,71	26,16	40,73	29,93	41,85	112,51	34,62	25,44	35,57	95,63	21,46	15,77	22,05	59,27				
	536	160	30,53	20,99	18,51	48,84	33,59	29,61	112,04	41,51	28,55	25,17	95,23	25,73	17,69	15,6	59,02				
	40	200	13,5	8,775	10,92	27	17,55	21,83	66,38	22,95	14,92	18,56	56,42	14,22	9,245	11,5	34,97				
	153	160	17,78	22,82	16,66	28,45	36,51	26,66	91,62	24,18	31,03	22,66	77,88	14,99	19,23	14,04	48,26				
4	211	150	26,63	26,49	27,09	39,95	39,73	40,63	120,31	33,96	33,77	34,54	102,3	21,04	20,93	21,4	63,38				
	398	150	16,18	9,773	16,79	24,27	14,66	25,19	64,12	20,63	12,46	21,41	54,5	12,79	7,723	13,27	33,78				
	107	100	12,61	17,37	14,76	12,61	17,37	14,76	44,74	10,72	14,76	12,55	38,03	6,643	9,15	7,775	23,57				
	142	160	35,81	34,79	44,9	57,29	55,67	71,84	184,8	48,7	47,32	61,06	157,1	30,18	29,33	37,84	97,35				
	165	160	29,33	30,37	27,99	46,93	48,59	44,79	140,31	39,89	41,3	38,07	119,3	24,72	25,6	23,59	73,91				
	437	100	18,01	27,82	31,24	18,01	27,82	31,24	77,07	15,31	23,65	26,55	65,51	9,487	14,66	16,46	40,6				
	449	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
	762	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				

PENYULANG WENDIT

Seksi	No. Trafo	Nominal (kVA)	Persen beban (%)			Beban Trafo (kVA)			Total beban (kVA)	Daya aktif (kW)			Total Daya aktif (kW)	Daya reaktif (kVAR)			Total Daya reaktif (kVAR)
			R	S	T	R	S	T		R	S	T		R	S	T	
1	225	200	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	215	160	29,98	22,28	25,44	47,96	35,64	40,71	124,31	40,77	30,29	34,6	105,66	25,26	18,77	21,45	65,48
	338	250	23,26	25,4	22,38	58,16	63,49	55,94	177,59	49,44	53,97	47,55	150,95	30,64	33,45	29,47	93,55
	577	200	38,59	27,68	32,63	77,18	55,35	65,25	197,78	65,6	47,05	55,46	168,11	40,66	29,16	34,37	104,2
3	28	100	49,38	49,48	35,86	49,38	49,48	35,86	134,72	41,97	42,06	30,48	114,51	26,01	26,07	18,89	70,97
	203	100	27,6	12,88	13,8	27,6	12,88	13,8	54,28	23,46	10,95	11,73	46,138	14,54	6,785	7,27	28,59
	204	160	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	222	1030	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	293	100	28,35	29,48	27	28,35	29,48	27	84,83	24,1	25,06	22,95	72,106	14,93	15,53	14,22	44,69
	502	250	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	503	160	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	608	150	22,35	22,47	19,88	33,52	33,7	29,82	97,04	28,49	28,65	25,35	82,484	17,66	17,75	15,71	51,12
	649	250	27,74	26,88	27,14	69,34	67,2	67,84	204,38	58,94	57,12	57,66	173,72	36,53	35,4	35,74	107,7
	650	200	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	655	100	19,8	29,03	33,3	19,8	29,03	33,3	82,13	16,83	24,68	28,31	69,811	10,43	15,29	17,54	43,26
	659	800	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	676	200	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	755	630	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	773	250	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	828	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	840	160	22,4	21,08	27,84	35,84	33,73	44,55	114,12	30,46	28,67	37,87	97,002	18,88	17,77	23,47	60,12
966	160	6,581	8,156	8,444	10,53	13,05	13,51	37,09	8,951	11,09	11,48	31,527	5,547	6,875	7,117	19,54	
967	160	0	1,13	0,706	0	1,808	1,13	2,938	0	1,537	0,961	2,4973	0	0,952	0,595	1,548	
968	160	15,39	20,66	14,25	24,62	33,06	22,8	80,48	20,93	28,1	19,38	68,408	12,97	17,42	12,01	42,4	
991	160	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
4	378	160	17,23	27,83	20,91	27,57	44,52	33,45	105,54	23,43	37,84	28,43	89,709	14,52	23,45	17,62	55,6
	727	160	1,283	0,57	2,993	2,052	0,912	4,788	7,752	1,744	0,775	4,07	6,5892	1,081	0,48	2,522	4,084

unit UnitMath;

interface

uses Kompleks;

Procedure

TambahKomplex(A,B,C,D:double;
var Riel,Imag:double);

Procedure

KurangKomplex(A,B,C,D:double;
var Riel,Imag:double);

Procedure KaliKomplex(A,B,C,D:double;
var Riel,Imag:double);

Procedure BagiKomplex(A,B,C,D:double;
var Riel,Imag:double);

Procedure

EllGauss(Baris,Kolom:byte;MPQ:Arr2;
var YE:Arr1);

implementation

Procedure

TambahKomplex(A,B,C,D:double;
var Riel,Imag:double);

begin

Riel:=A+C;

Imag:=B+D;

end;

Procedure

KurangKomplex(A,B,C,D:double;
var Riel,Imag:double);

begin

Riel:=A-C;

Imag:=B-D;

end;

Procedure KaliKomplex(A,B,C,D:double;
var Riel,Imag:double);

begin

Riel:=(A*C-B*D);

Imag:=(A*D+B*C);

end;

Procedure BagiKomplex(A,B,C,D:double;
var Riel,Imag:double);

begin

Riel:=(A*C+B*D)/(sqr(C)+sqr(D));

Imag:=(B*C-A*D)/(sqr(C)+sqr(D));

end;

Procedure

EllGauss(Baris,Kolom:byte;MPQ:Arr2;
var YE:Arr1);

var i,j,k:byte;

DE,AE,Sum,Konst,value:double;

begin

for i:=1 to Baris do

begin

if MPQ[i,i]=0 then

begin

for k:=i+1 to Baris do

begin

if MPQ[k,i] <> 0 then

begin

for j:=1 to Kolom do

begin

Konst:=MPQ[i,j];

value:=MPQ[k,j];

MPQ[i,j]:=value;

MPQ[k,j]:=Konst;

end;

end;

end;

end;

end;

for i:=1 to Baris do

begin

DE:=MPQ[i,i];

if DE=0 then DE:=0.0000001;

for j:=1 to Kolom do

begin

MPQ[i,j]:=MPQ[i,j]/DE;

end;

for k:=i to Baris do

begin

if k <> i then

begin

AE:=MPQ[k,i];

for j:=1 to Kolom do

begin

MPQ[k,j]:=MPQ[k,j]-AE*MPQ[i,j];

end;

end;

end;

end;

for i:=Baris downto 1 do

begin

if i <> Baris then

begin

Sum:=0;

for j:=i+1 to Baris do

begin

Sum:=Sum+MPQ[i,j]*YE[j];

end;

YE[i]:=MPQ[i,Kolom]-Sum;

end

else

begin

```

YE[i]:=MPQ[i,Kolom];
end;
end;
end;
end.

```

```

unit uFD3Phasa;
interface
uses Math,Komplex,SysUtils;

type
  TComplex=record
    re,im:double;
  end;

type
  TSaluran=record
    a,b,c:TComplex;
  end;

type
  CArr1a=Array of TComplex;
  CArr2a=Array of Array of TComplex;
  CArr3a=Array of Array of Array of
TComplex;
  SIArr=Array of TSaluran;

//unit Data
type
  TSaluran1=record

No,Nbhind,LnAHead1,LnAHead2,LnAHea
d3:integer;
  Panjang,GMR,R:double;
  Zs:Array[1..4,1..4] of TComplex;
  end;

  TPhasa=record
    R,S,T:TComplex;
  end;

  TPhasaDouble=record
    R,S,T:double;
  end;

  TBus=record
    No:integer;
    Load:TPhasa;
  end;

  SArr=Array of TSaluran1;
  BArr=Array of TBus;
  PArr=Array of TPhasa;
  PArr2=Array of Array of TPhasa;

procedure HitungFD3Phasa(const
rNbus:integer;
  const
rVbase,rVKonst,rPbase,rPkonst:double;

```

```

const rBus,rBranch: Arr2;
var
rV,rI,rSa,rSL,rSumGen,rSumLoad,rSumLoss: Arr2;
var rIterasi: byte); overload;

procedure HitungFD3Phasa(var
rV,rI,rSa,rSL,rSumGen,rSumLoad,
rSumLoss: Arr2;
var rIterasi: byte); overload;

procedure HitungFD3PhasaSebelum(var
rV,rI,rSa,rSL,rSumGen,rSumLoad,
rSumLoss: Arr2;
var rIterasi: byte);

var N: integer;
terminal: iArr1;
Bus: BArr;
Vr,Ir,Sa,SL: PArr;
Za: SArr;

Pbase, Vbase, KonstP, KonstV, Zbase, Ibase: double;

implementation

function
RealToStr(Num: double; Pecahan: byte): String;
var Hasil: String;
le: byte;
begin
le:=sizeof(Num);
Str(Num:le:Pecahan, Hasil);
Result:=Hasil;
end;

function StrToReal(Huruf: string): double;
var Temp: double;
Code: integer;
begin
val(Huruf, Temp, Code);
Result:=Temp;
end;

function Complex(re, im: double): TComplex;
begin
result.re:=re;
result.im:=im;
end;

function CxRe(z: TComplex): double;
begin
result:=z.re;

```

```

end;

function CxIm(Z: TComplex): double;
begin
result:=z.im;
end;

function CxAbs(Z: TComplex): Double;
begin
result:=sqrt(sqr(Z.re)+sqr(Z.im));
end;

function CxArg(Z: TComplex): Double;
var temp: double;
begin
temp:=0;
result:=0;
if Z.re<>0 then
temp:=ArcTan(Abs(Z.im/Z.re))
else
if Z.im<>0 then
result:=Pi/2*Z.im/abs(Z.im);

if Z.im=0 then
if Z.re>0 then
result:=0
else
result:=Pi;

if Z.re> 0 then
if Z.im> 0 then result:=temp
else if z.im< 0 then
result:=-temp;

if Z.re<0 then
if Z.im>0 then
result:=Pi-temp
else if z.im<0 then
result:=temp-Pi;

if ((Z.re=0) and (Z.im=0)) then result:= 0;
end;

function
PolarToCxRad(r, theta: double): TComplex;
begin
result.re:=r*cos(theta);
result.im:=r*sin(theta);
end;

function
PolarToCxDeg(r, theta: double): TComplex;
begin
theta:=theta*Pi/180;
result.re:=r*cos(theta);

```

```
    result.im:=r*sin(theta);
end;
```

```
function
CxTambah(x,y:TComplex):TComplex;
begin
    result.re:=x.re+y.re;
    result.im:=x.im+y.im;
end;
```

```
function
CxKurang(x,y:TComplex):TComplex;
begin
    result.re:=x.re-y.re;
    result.im:=x.im-y.im;
end;
```

```
function CxKali(x,y:TComplex):TComplex;
begin
    result.re:=x.re*y.re-x.im*y.im;
    result.im:=x.re*y.im+x.im*y.re;
end;
```

```
function CxBagi(x,y:TComplex):TComplex;
begin
```

```
    result.re:=(x.re*y.re+x.im*y.im)/(sqr(y.re)+sqr(y.im));
    result.im:=(x.im*y.re-x.re*y.im)/(sqr(y.re)+sqr(y.im));
end;
```

```
function
CxPow(Z:TComplex;n:Integer):TComplex;
begin
```

```
    result.re:=Power(CxAbs(Z),n)*Cos(n*CxArg(Z));
```

```
    result.im:=Power(CxAbs(Z),n)*Sin(n*CxArg(Z));
end;
```

```
function CxSqrt(Z:TComplex):TComplex;
begin
```

```
    result.re:=Sqrt(CxAbs(Z))*Cos(0.5*CxArg(Z));
```

```
    result.im:=Sqrt(CxAbs(Z))*Sin(0.5*CxArg(Z));
end;
```

```
function CxExp(Z:TComplex):TComplex;
begin
```

```
    result.re:=Exp(Z.re)*Cos(Z.im);
    result.im:=Exp(Z.re)*Sin(Z.im);
end;
```

```
function CxLn(Z:TComplex):TComplex;
begin
    result.re:=Ln(CxAbs(Z));
    result.im:=CxArg(Z);
end;
```

```
function CxCos(Z:TComplex):TComplex;
var
    iZ,miZ,Temp:TComplex;
begin
    iZ.re:=-Z.im;
    iZ.im:=Z.re;
    miZ.re:=Z.im;
    miZ.im:=-Z.re;
    Temp:=
CxTambah(CxExp(iZ),CxExp(miZ));
    result.re:=Temp.re/2;
    result.im:=Temp.im/2;
end;
```

```
function cSin(Z:TComplex):TComplex;
var iZ,miZ,Temp,i:TComplex;
begin
    iZ.re:=-Z.im;
    iZ.im:=Z.re;
    miZ.re:=Z.im;
    miZ.im:=-Z.re;
    i.re:=0;
    i.im:=1;
```

```
Temp:=CxKurang(CxExp(iZ),CxExp(miZ));
    Temp.re:=Temp.re/2;
    Temp.im:=Temp.im/2;
    result:=CxBagi(Temp,i);
end;
```

```
function
CxPangkat(z:TComplex;pangkat:double):TComplex;
var theta,sum:double;
begin
    theta:=arctan(z.re/z.im);

    sum:=exp((pangkat/2)*ln(sqr(z.re)+sqr(z.im)));
    result.re:=sum*cos(pangkat*theta);
    result.im:=sum*sin(pangkat*theta);
end;
```

```
function CxConjg(z:TComplex):TComplex;
begin
```

```

result.re:=z.re;
result.im:=-z.im;
end;

```

```

function
CxNegatif(z:TComplex):TComplex;
begin
result.re:=-z.re;
result.im:=-z.im;
end;

```

```

Procedure CxArray1(N:byte;rea,ima:Arr1;
var Z:CArr1a);

```

```

var i:byte;
begin
for i:=1 to N do
begin
z[i]:=Complex(rea[i],ima[i]);
end;
end;

```

```

Procedure CxArray2(N:byte;rea,ima:Arr2;
var Z:CArr2a);

```

```

var i,j:byte;
begin
for i:=1 to N do
begin
for j:=1 to N do
begin
z[i,j]:=Complex(rea[i,j],ima[i,j]);
end;
end;
end;

```

```

Procedure CxArr2re(N:byte;rea:Arr2;
var z:CArr2a);

```

```

var i,j:byte;
begin
for i:=1 to N do
begin
for j:=1 to N do
begin
z[i,j]:=Complex(rea[i,j],0);
end;
end;
end;

```

```

Procedure CxArr2im(N:byte;ima:Arr2;
var z:CArr2a);

```

```

var i,j:byte;
begin
for i:=1 to N do
begin
for j:=1 to N do
begin

```

```

z[i,j]:=Complex(0,ima[i,j]);
end;
end;
end;

```

```

function
CxToStr(z:TComplex;len:byte):string;

```

```

begin
if z.im<0 then
begin
result:=RealToStr(z.re,len)+' -
j'+RealToStr(Abs(z.im),len);
end
else
begin
result:=RealToStr(z.re,len)+' +
j'+RealToStr(z.im,len);
end;
end;

```

```

//Unit Matrik

```

```

Procedure
TransposeMatrik(N:integer;MA:Arr2;
var MB:Arr2);

```

```

var i,j:integer;
begin
SetLength(MB,N+1,N+1);
for i:=1 to N do
begin
for j:=1 to N do
begin
MB[j,i]:=MA[i,j];
end;
end;
end;

```

```

Procedure InversMatrik(N:byte;MA:Arr2;
var MB:Arr2);

```

```

var Konst,DE,AE:double;
i,j,k:byte;
begin
SetLength(MB,N+1,N+1);
for i:=1 to N do
begin
for j:=1 to N do
begin
MB[i,j]:=MA[i,j];
end;
end;
for i:=1 to N do
begin
if MB[i,i]=0 then
begin
for k:=i+1 to N do
begin

```

```

if MB[k,i] <> 0 then
begin
for j:=1 to N do
begin
Konst:=MB[i,j];
MB[i,j]:=MB[k,j];
MB[k,j]:=Konst;
end;
end;
end;
DE:=MB[i,i];
MB[i,i]:=1;
for j:=1 to N do
begin
MB[i,j]:=MB[i,j]/DE;
end;
for k:=1 to N do
begin
if k <> i then
begin
AE:=MB[k,i];
MB[k,i]:=0;
for j:=1 to N do
begin
MB[k,j]:=MB[k,j]-AE*MB[i,j];
end;
end;
end;
end;
end;
end;

```

```

Procedure
InversMatrikInt(N:byte;MA:iArr2;
var MB:Arr2);
var MAA:Arr2;
i,j:byte;
begin
SetLength(MB,N+1,N+1);
SetLength(MAA,N+1,N+1);
for i:=1 to N do
begin
for j:=1 to N do
begin
MAA[i,j]:=MA[i,j];
end;
end;
InversMatrik(N,MAA,MB);
end;

```

```

Procedure MatrikA(N:integer;Z:CArr2a;
var B:Arr2);
var i,j,k:integer;
A:Arr2;
Zx:double;

```

```

begin
SetLength(A,N+1,N+1);
SetLength(B,N+1,N+1);
for i:=0 to N do
begin
for j:=0 to N do
begin
for k:=i to N do
begin
Zx:=CxIm(Z[j,k]);
if Zx <> 0 then
begin
if k=i then
begin
A[i,k]:=-1;
end
else if j=i then
begin
A[i,k]:=1;
end
else
begin
A[i,k]:=0;
end;
end;
end;
end;
end;
end;
InversMatrik(N,A,B);
for i:=1 to N do
begin
for j:=1 to N do
begin
B[i,j]:=-B[i,j];
end;
end;
end;
end;
Procedure
CxKaliMatrikN1(N:integer;MA:CArr2a;MB
:CArr1a;
var MC:CArr1a);
var i,j:byte;
temp:Tcomplex;
begin
SetLength(MC,N+1);
for i:=1 to N do
begin
MC[i]:=Complex(0,0);
for j:=1 to N do
begin
temp:=CxKali(MA[i,j],MB[j]);
MC[i]:=Cxtambah(MC[i],temp);
end;
end;
end;

```

end;

Procedure

CxKaliMatrikNN(N:byte;MA,MB:CArr2a;
var MC:CArr2a);

var i,j,k:byte;
temp:TComplex;

begin

SetLength(MC,N+1,N+1);

for i:=1 to N do

begin

for j:=1 to N do

begin

MC[i,j]:=Complex(0,0);

for k:=1 to N do

begin

temp:=CxKali(MA[i,k],MB[k,j]);

MC[i,j]:=CxTambah(MC[i,j],temp);

end;

end;

end;

end;

Function

PhTambah(Var1,Var2:TPhasa):TPhasa;

begin

Result.R:=CxTambah(Var1.R,Var2.R);

Result.S:=CxTambah(Var1.S,Var2.S);

Result.T:=CxTambah(Var1.T,Var2.T);

end;

Function

PhKali(Var1,Var2:TPhasa):TPhasa;

begin

Result.R:=CxKali(Var1.R,Var2.R);

Result.S:=CxKali(Var1.S,Var2.S);

Result.T:=CxKali(Var1.T,Var2.T);

end;

Function

PhBagi(Var1,Var2:TPhasa):TPhasa;

begin

Result.R:=CxBagi(Var1.R,Var2.R);

Result.S:=CxBagi(Var1.S,Var2.S);

Result.T:=CxBagi(Var1.T,Var2.T);

end;

Function

PhKurang(Var1,Var2:TPhasa):TPhasa;

begin

Result.R:=CxKurang(Var1.R,Var2.R);

Result.S:=CxKurang(Var1.S,Var2.S);

Result.T:=CxKurang(Var1.T,Var2.T);

end;

Function

PhAbs(Var1:TPhasa):TPhasaDouble;

begin

Result.R:=CxAbs(Var1.R);

Result.S:=CxAbs(Var1.S);

Result.T:=CxAbs(Var1.T);

end;

Function PhConj(Var1:TPhasa):TPhasa;

begin

Result.R:=CxConjg(Var1.R);

Result.S:=CxConjg(Var1.S);

Result.T:=CxConjg(Var1.T);

end;

Procedure

PhKaliMatrikN1(N:integer;MA:Arr2;MB:P
Arr;

var MC:PArr);

var i,j:integer;

MX:CArr2a;

MH,MS:CArr1a;

begin

SetLength(MX,N+1,N+1);

SetLength(MH,N+1);

SetLength(MS,N+1);

SetLength(MC,N+1);

for i:=1 to N do

begin

for j:=1 to N do

begin

MX[i,j]:=Complex(MA[i,j],0);

end;

end;

for i:=1 to N do

begin

MS[i]:=MB[i].R;

end;

CxKaliMatrikN1(N,MX,MS,MH);

for i:=1 to N do

begin

MC[i].R:=MH[i];

end;

for i:=1 to N do

begin

MS[i]:=MB[i].S;

end;

CxKaliMatrikN1(N,MX,MS,MH);

for i:=1 to N do

begin

MC[i].S:=MH[i];

end;

for i:=1 to N do

begin

MS[i]:=MB[i].T;


```

end;
CxKaliMatrikN1(N,MX,MS,MH);
for i:=1 to N do
begin
  MC[i].T:=MH[i];
end;
end;

Procedure
ImpedansiMatrik(N:integer;D:Arr2;
  var Za:SArr);
var i,j,k:integer;
  tmpRe,tmpIm:double;
  temp1:TComplex;
begin
  //modified form of Carson's Equation
  for k:=1 to N do
  begin
    for i:=1 to 4 do
    begin
      for j:=1 to 4 do
      begin
        if i=j then
        begin
          tmpRe:=Za[k].R+0.0953;

tmpIm:=0.12134*(ln(1/Za[k].GMR)+7.934);

Za[k].Zs[i,j]:=Complex(tmpRe,tmpIm);
          end
          else
          begin
            tmpRe:=0.0953;
            tmpIm:=0.12134*(ln(1/D[i,j])+7.934);

Za[k].Zs[i,j]:=Complex(tmpRe,tmpIm);
          end;
        end;
      end;
    end;
  end;
  //This matrix is reduced 3X3 by using Kron
  reduction
  for k:=1 to N do
  begin
    for i:=1 to 3 do
    begin
      for j:=1 to 3 do
      begin
        temp1:=CxKali(Za[k].Zs[i,4],Za[k].Zs[4,j]);
        temp1:=CxBagi(temp1,Za[k].Zs[4,4]);

Za[k].Zs[i,j]:=CxKurang(Za[k].Zs[i,j],temp1
);
          end;
        end;
      end;
    end;
  end;

```

```

end;
end;
end;

Procedure JarakKawat(var D:Arr2);
var ab,bc,an,bn,nn:double;
begin
  ab:=0.9;
  bc:=0.75;
  an:=0.45;
  bn:=0.45;
  nn:=1;
  SetLength(D,5,5);
  D[2,1]:=ab;
  D[1,2]:=D[2,1];
  D[3,1]:=ab+bc;
  D[1,3]:=D[3,1];
  D[4,1]:=sqrt(sqr(an)+sqr(nn));
  D[1,4]:=D[4,1];
  D[3,2]:=bc;
  D[2,3]:=D[3,2];
  D[4,2]:=sqrt(sqr(bn)+sqr(nn));
  D[2,4]:=D[4,2];
  D[4,3]:=sqrt(sqr(bn+bc)+sqr(nn));
  D[3,4]:=D[4,3];
end;

Procedure ImpedansiSaluranKm(N:integer;
  var Za:SArr);
var i,j,k:integer;
  mile,ft:double;
  temp:TComplex;
  D:Arr2;
begin
  mile:=1609;
  ft:=0.3048;
  JarakKawat(D);
  for i:=1 to N do
  begin
    Za[i].GMR:=Za[i].GMR/ft;
    Za[i].R:=Za[i].R*mile;
  end;
  for i:=1 to 4 do
  begin
    for j:=1 to 4 do
    begin
      D[i,j]:=D[i,j]/ft;
    end;
  end;
  ImpedansiMatrik(N,D,Za);
  for k:=1 to N do
  begin
    temp:=Complex(Za[k].Panjang/mile,0);
    for i:=1 to 3 do
    begin

```

```

    for j:=1 to 3 do
    begin
Za[k].Zs[i,j]:=CxKali(Za[k].Zs[i,j],temp);
    end;
    end;
    end;
end;

```

```

//Unit Loadflow FastDecoupled 3 Phasa
Procedure MatrikAm(N:integer;Za:SArr;
    var Am:iArr2);
var i,j,k,ia,ja:integer;
    Zs:iArr2;
begin
    SetLength(Am,N+1,N+1);
    SetLength(Zs,N+1,N+1);
    for i:=1 to N do
    begin
        for j:=1 to N do
        begin
            Am[i,j]:=0;
            Zs[i,j]:=0;
        end;
    end;
    for i:=1 to N do
    begin
        ia:=Za[i].Nbhind;
        ja:=Za[i].No;
        Zs[ia,ja]:=1;
    end;
    for i:=0 to N do
    begin
        for j:=0 to N do
        begin
            for k:=i to N do
            begin
                if Zs[j,k]>0 then
                begin
                    if k=i then
                    begin
                        Am[i,k]:=1;
                    end
                    else if j=i then
                    begin
                        Am[i,k]:=1;
                    end
                    else
                    begin
                        Am[i,k]:=0;
                    end;
                end;
            end;
        end;
    end;
end;
end;
end;

```

```

end;

Procedure
IdentifikasiNode(N:integer;Am:iArr2;
    var Trm,CbTrm:iArr1;
    var Nt,Nc:byte);
var i,j,s,sa,sb:integer;
begin
    SetLength(Trm,N+1);
    SetLength(CbTrm,N+1);
    sa:=1;
    sb:=1;
    for i:=1 to N do
    begin
        s:=0;
        if i<>N then
        begin
            for j:=i+1 to N do
            begin
                if Am[i,j]>0 then
                begin
                    s:=s+1;
                end;
            end;
            if s=0 then
            begin
                Trm[sa]:=i;
                sa:=sa+1;
            end
            else if s>1 then
            begin
                CbTrm[sb]:=i;
                sb:=sb+1;
            end;
        end
        else
        begin
            Trm[sa]:=i;
        end;
        Nt:=sa;
        Nc:=sb-1;
    end;
end;

Procedure
CariTerminal(N,Nt,Node:integer;Am:iArr2;
    Trm:iArr1;
    var endNode:integer);
var j,k,s:integer;
begin
    for j:=Node to N do
    begin
        s:=0;
        for k:=j+1 to N do
        begin

```

```

if Am[j,k]=1 then
begin
s:=s+1;
break;
end;
end;
if s=0 then
begin
for k:=1 to Nt do
begin
if j=Trm[k] then
begin
endNode:=j;
end;
end;
break;
end;
end;
end;

```

```

Procedure
IndexTerminal(Node,Nt:integer;Trm:iArr1;
var idxTerm:integer);
var i:integer;
begin
for i:=1 to Nt do
begin
if Node=Trm[i] then
begin
idxTerm:=Nt-i+1;
end;
end;
end;
end;

```

```

Procedure
MatrikJaqobian(N,Nt,Nc:integer;Am:iArr2;
Trm,CbTrm:iArr1;
var MxJaq:iArr2);
var i,j,ia,ix,jx,ja:integer;
begin
SetLength(MxJaq,Nt+1,Nt+1);
for i:=Nc downto 1 do
begin
ia:=CbTrm[i];
CariTerminal(N,Nt,ia,Am,Trm,ja);
IndexTerminal(ja,Nt,Trm,ix);
for j:=Nc downto ia do
begin
if Am[ia,j]=1 then
begin
CariTerminal(N,Nt,j,Am,Trm,ja);
IndexTerminal(ja,Nt,Trm,jx);
if ix<>jx then
begin
MxJaq[jx,ix]:=-1;

```

```

end;
end;
end;
for i:=1 to Nt do
begin
MxJaq[i,i]:=1;
end;
end;
end;

```

```

Function
DropTegangan(Node:integer;Za:SArr;Ir:PArr):TPhasa;
var i,j:integer;
B,C:CArr1a;
A:CArr2a;
begin
SetLength(A,4,4);
SetLength(B,4);
SetLength(C,4);
for i:=1 to 3 do
begin
for j:=1 to 3 do
begin
A[i,j]:=Za[Node].Zs[i,j];
end;
end;
B[1]:=Ir[Node].R;
B[2]:=Ir[Node].S;
B[3]:=Ir[Node].T;
CxKaliMatrikN1(3,A,B,C);
Result.R:=C[1];
Result.S:=C[2];
Result.T:=C[3];
end;
end;

```

```

Procedure
HitungTegNode(N:integer;Am:iArr2;Za:SArr;Bus:BArr;
var Vr,Ir:PArr);
var i,j:integer;
dV,dI:TPhasa;
begin
for i:=Nc downto 1 do
begin
Vr[i]:=Vr[i];
Ir[i].R:=Complex(0,0);
Ir[i].S:=Complex(0,0);
Ir[i].T:=Complex(0,0);
if i<>N then
begin
for j:=i+1 to N do
begin
if Am[i,j]=1 then
begin

```

```

dV:=DropTegangan(j,Za,Ir);
Vr[j]:=PhTambah(Vr[j],dV);
break;
end;
end;
for j:=i+1 to N do
begin
if Am[i,j]=1 then
begin
Ir[i]:=PhTambah(Ir[i],Ir[j]);
end;
end;
end;
dI:=PhConj(Phbagi(Bus[i].Load,Vr[i]));
Ir[i]:=PhTambah(Ir[i],dI);
end;
end;

```

```

Procedure
HitungMxFs(N:integer;Nt,Nc:byte;Am:iArr
2;Trm,CbTrm:iArr1;
Vr,Ir:PArr;
var MxFs:PArr);
var i,j,ia,ix,jx,ja:integer;
dV,Vs:TPhasa;
begin
SetLength(MxFs,Nt+1);
Vs.R:=Complex(1,0);
Vs.S:=Complex(1,0);
Vs.T:=Complex(1,0);
for i:=Nc downto 1 do
begin
ia:=CbTrm[i];
CariTerminal(N,Nt,ia,Am,Trm,ja);
IndexTerminal(ja,Nt,Trm,ix);
for j:=N downto ia do
begin
if Am[ia,j]=1 then
begin
CariTerminal(N,Nt,j,Am,Trm,ja);
IndexTerminal(ja,Nt,Trm,jx);
if ix<>jx then
begin
dV:=DropTegangan(j,Za,Ir);
dV:=PhTambah(Vr[j],dV);
MxFs[jx]:=PhKurang(dV,Vr[ia]);
end;
end;
end;
end;
dV:=DropTegangan(1,Za,Ir);
dV:=PhTambah(Vr[1],dV);
MxFs[Nt]:=PhKurang(dV,Vs);
for i:=1 to Nt do
begin

```

```

MxFs[i].R:=Complex(CxRe(MxFs[i].R)*-
1,CxIm(MxFs[i].R)*-1);
MxFs[i].S:=Complex(CxRe(MxFs[i].S)*-
1,CxIm(MxFs[i].S)*-1);
MxFs[i].T:=Complex(CxRe(MxFs[i].T)*-
1,CxIm(MxFs[i].T)*-1);
end;
end;

```

```

Procedure
CekKonvergen(Nt:byte;MxFs:PArr;tol:doub
le;
var Konvergen:Boolean);
var i:integer;
err:TPhasaDouble;
begin
Konvergen:=true;
for i:=1 to Nt do
begin
err:=PhAbs(MxFs[i]);
if err.R>tol then
begin
Konvergen:=false;
end
else if err.S>tol then
begin
Konvergen:=false;
end
else if err.T>tol then
begin
Konvergen:=false;
end;
end;
end;
end;

```

```

Procedure
UpdateTeg(Nt:byte;InvJaq:Arr2;MxFs:PArr;
Trm:iArr1;
var Vr:PArr);
var i,ia:byte;
dVr:PArr;
begin
SetLength(dVr,Nt+1);
PhKaliMatrikN1(Nt,InvJaq,MxFs,dVr);
for i:=1 to Nt do
begin
ia:=Trm[Nt-i+1];
Vr[ia]:=PhTambah(Vr[ia],dVr[i]);
end;
end;
end;

```

```

Procedure AliranDaya(N:integer;Vr,Ir:PArr;
var Sa:PArr);
var i:integer;
begin

```

```

SetLength(Sa,N+1);
for i:=1 to N do
begin
  Sa[i]:=PhKali(Vr[i],PhConj(Ir[i]));
end;
end;

```

```

Procedure
RugiSaluran(N:integer;Za:SArr,Ir:PArr;
  var SL:PArr);
var i:integer;
  temp:TPhasa;
begin
  SetLength(SL,N+1);
  for i:=1 to N do
  begin
    temp:=DropTegangan(i,Za,Ir);
    SL[i]:=PhKali(temp,Ir[i]);
  end;
end;

```

```

Procedure
FastDLFlow(N:integer;Za:SArr;Bus:BArr;
  var iterasi:byte;
  var Vr,Ir,Sa,SL:PArr);
var Nc,Nt:byte;
  tol:double;
  Am,MxJaq:iArr2;
  InvJaq:Arr2;
  Trm,CbTrm:iArr1;
  MxFs:PArr;
  Konvergen:boolean;
  Label Mulai;
begin
  SetLength(Ir,N+1);
  tol:=0.000001;
  iterasi:=0;
  MatrikAm(N,Za,Am);

```

```

IdentifikasiNode(N,Am,Trm,CbTrm,Nt,Nc);

```

```

MatrikJacobian(N,Nt,Nc,Am,Trm,CbTrm,MxJaq);
  InversMatrikInt(Nt,MxJaq,InvJaq);
Mulai:
  HitungTegNode(N,Am,Za,Bus,Vr,Ir);

```

```

HitungMxFs(N,Nt,Nc,Am,Trm,CbTrm,Vr,Ir,MxFs);
  CekKonvergen(Nt,MxFs,tol,Konvergen);
  if Konvergen=false then
  begin
    UpdateTeg(Nt,InvJaq,MxFs,Trm,Vr);
    iterasi:=iterasi+1;
    goto Mulai;
  end;

```

```

end;
AliranDaya(N,Vr,Ir,Sa);
RugiSaluran(N,Za,Ir,SL);
end;

```

```

procedure DecodeData(const rNbus:integer;
  const
  rVbase,rVKonst,rPbase,rPkonst:double;
  const rBus,rBranch:Arr2);
var Nterm,i:integer;
begin
  N:=rNbus;
  Vbase:=rVbase;
  KonstV:=rVKonst;
  Pbase:=rPbase;
  KonstP:=rPkonst;

```

```

Zbase:=sqr(Vbase*KonstV)/(Pbase*KonstP);
;

```

```

  Ibase:=Vbase*KonstV/Zbase;
  SetLength(Bus,N+1);
  SetLength(Vr,N+1);
  SetLength(Za,N+1);
  for i:=1 to N do
  begin
    Bus[i].No:=round(rBus[i,1]);

```

```

    Bus[i].Load.R:=complex(rBus[i,2]/Pbase,rBus[i,3]/Pbase);

```

```

    Bus[i].Load.S:=complex(rBus[i,4]/Pbase,rBus[i,5]/Pbase);

```

```

    Bus[i].Load.T:=complex(rBus[i,6]/Pbase,rBus[i,7]/Pbase);
  end;

```

```

  for i:=0 to N do
  begin

```

```

    Vr[i].R:=complex(1,0);
    Vr[i].S:=complex(1,0);
    Vr[i].T:=complex(1,0);
  end;

```

```

  for i:=1 to N do
  begin

```

```

    With Za[i] do
    begin

```

```

      No:=round(rBranch[i,1]);
      Nbhind:=round(rBranch[i,2]);
      LnAHead1:=round(rBranch[i,3]);
      LnAHead2:=round(rBranch[i,4]);
      LnAHead3:=round(rBranch[i,5]);
      Panjang:=rBranch[i,6];
      GMR:=rBranch[i,7]*0.001;
      R:=rBranch[i,8]*0.001;
    end;
  end;

```

```

end;
Nterm:=0;
for i:=1 to N do
begin
  With Za[i] do
  begin
    if (LnAHead1=0) and (LnAHead2=0)
and (LnAHead3=0) then
      begin
        Inc(Nterm);
      end;
    end;
  end;
  SetLength(terminal,Nterm+1);
  Nterm:=0;
  for i:=1 to N do
  begin
    With Za[i] do
    begin
      if (LnAHead1=0) and (LnAHead2=0)
and (LnAHead3=0) then
        begin
          Inc(Nterm);
          terminal[Nterm]:=i;
        end;
      end;
    end;
  end;
end;

procedure DataSkripsi;
var i:integer;
begin
  N:=55;
  Vbase:=20;KonstV:=1000;
  Pbase:=100;KonstP:=1000;

Zbase:=sqr(Vbase*KonstV)/(Pbase*KonstP)
;
  Ibase:=Vbase*KonstV/Zbase;
  SetLength(Bus,N+1);
  SetLength(Vr,N+1);
  SetLength(Za,N+1);
  SetLength(terminal,17+1);
  Bus[1].No:=1;

  Bus[1].Load.R:=complex(46.1648/Pbase,27.
3926/Pbase);

  Bus[1].Load.S:=complex(31.0288/Pbase,18.
4114/Pbase);

  Bus[1].Load.T:=complex(34.2452/Pbase,20.
3199/Pbase);
  Bus[2].No:=2;

```

```

  Bus[2].Load.R:=complex(0.0000/Pbase,
0.0000/Pbase);
  Bus[2].Load.S:=complex(0.0000/Pbase,
0.0000/Pbase);
  Bus[2].Load.T:=complex(0.0000/Pbase,
0.0000/Pbase);
  Bus[3].No:=3;

  Bus[3].Load.R:=complex(10.9736/Pbase,6.5
114/Pbase);

  Bus[3].Load.S:=complex(10.0276/Pbase,5.9
500/Pbase);

  Bus[3].Load.T:=complex(13.2440/Pbase,7.8
585/Pbase);
  Bus[4].No:=4;

  Bus[4].Load.R:=complex(38.2184/Pbase,22.
6775/Pbase);

  Bus[4].Load.S:=complex(43.8944/Pbase,26.
0454/Pbase);

  Bus[4].Load.T:=complex(45.4080/Pbase,26.
9435/Pbase);
  Bus[5].No:=5;

  Bus[5].Load.R:=complex(21.1904/Pbase,12.
5736/Pbase);

  Bus[5].Load.S:=complex(23.8392/Pbase,14.
1454/Pbase);

  Bus[5].Load.T:=complex(26.6772/Pbase,15.
8293/Pbase);
  Bus[6].No:=6;

  Bus[6].Load.R:=complex(0.0000/Pbase,0.00
00/Pbase);

  Bus[6].Load.S:=complex(0.0000/Pbase,0.00
00/Pbase);

  Bus[6].Load.T:=complex(0.0000/Pbase,0.00
00/Pbase);
  Bus[7].No:=7;

  Bus[7].Load.R:=complex(36.7048/Pbase,21.
7793/Pbase);

  Bus[7].Load.S:=complex(39.1644/Pbase,23.
2388/Pbase);

```

Bus[7].Load.T:=complex(37.6508/Pbase,22.3407/Pbase);

Bus[8].No:=8;

Bus[8].Load.R:=complex(0.0000/Pbase,0.0000/Pbase);

Bus[8].Load.S:=complex(0.0000/Pbase,0.0000/Pbase);

Bus[8].Load.T:=complex(0.0000/Pbase,0.0000/Pbase);

Bus[9].No:=9;

Bus[9].Load.R:=complex(50.1380/Pbase,29.7501/Pbase);

Bus[9].Load.S:=complex(48.2460/Pbase,28.6275/Pbase);

Bus[9].Load.T:=complex(51.0840/Pbase,30.3115/Pbase);

Bus[10].No:=10;

Bus[10].Load.R:=complex(122.9800/Pbase,72.9720/Pbase);

Bus[10].Load.S:=complex(157.9820/Pbase,93.7410/Pbase);

Bus[10].Load.T:=complex(132.4400/Pbase,78.5853/Pbase);

Bus[11].No:=11;

Bus[11].Load.R:=complex(35.9480/Pbase,21.3303/Pbase);

Bus[11].Load.S:=complex(34.0560/Pbase,20.2076/Pbase);

Bus[11].Load.T:=complex(35.0020/Pbase,20.7690/Pbase);

Bus[12].No:=12;

Bus[12].Load.R:=complex(24.0284/Pbase,14.2576/Pbase);

Bus[12].Load.S:=complex(24.2176/Pbase,14.3699/Pbase);

Bus[12].Load.T:=complex(17.7848/Pbase,10.5529/Pbase);

Bus[13].No:=13;

Bus[13].Load.R:=complex(31.5964/Pbase,18.7482/Pbase);

Bus[13].Load.S:=complex(32.9208/Pbase,19.5341/Pbase);

Bus[13].Load.T:=complex(39.5428/Pbase,23.4633/Pbase);

Bus[14].No:=14;

Bus[14].Load.R:=complex(10.2168/Pbase,6.0623/Pbase);

Bus[14].Load.S:=complex(13.4332/Pbase,7.9708/Pbase);

Bus[14].Load.T:=complex(13.4332/Pbase,7.9708/Pbase);

Bus[15].No:=15;

Bus[15].Load.R:=complex(18.5416/Pbase,11.0019/Pbase);

Bus[15].Load.S:=complex(19.2984/Pbase,11.4510/Pbase);

Bus[15].Load.T:=complex(17.2172/Pbase,10.2161/Pbase);

Bus[16].No:=16;

Bus[16].Load.R:=complex(29.7044/Pbase,17.6256/Pbase);

Bus[16].Load.S:=complex(43.5160/Pbase,25.8209/Pbase);

Bus[16].Load.T:=complex(32.5424/Pbase,19.3095/Pbase);

Bus[17].No:=17;

Bus[17].Load.R:=complex(4.9192/Pbase,2.9189/Pbase);

Bus[17].Load.S:=complex(14.5684/Pbase,8.6444/Pbase);

Bus[17].Load.T:=complex(18.1632/Pbase,10.7774/Pbase);

Bus[18].No:=18;

Bus[18].Load.R:=complex(0.0000/Pbase,0.0000/Pbase);

```

Bus[29].Load.S:=complex(20.6228/Pbase,1
2.2369/Pbase);
Bus[29].Load.T:=complex(28.1908/Pbase,1
6.7274/Pbase);
Bus[30].No:=30;
Bus[30].Load.R:=complex(0.0000/Pbase,0.0
000/Pbase);//3 5
Bus[30].Load.S:=complex(0.0000/Pbase,0.0
000/Pbase);//3 5
Bus[30].Load.T:=complex(0.0000/Pbase,0.0
000/Pbase);//1 0
Bus[31].No:=31;
Bus[31].Load.R:=complex(0.0000/Pbase,0.0
000/Pbase);//3 8
Bus[31].Load.S:=complex(0.0000/Pbase,0.0
000/Pbase);//9 2
Bus[31].Load.T:=complex(0.0000/Pbase,0.0
000/Pbase);//1 9
Bus[32].No:=32;
Bus[32].Load.R:=complex(26.6772/Pbase,1
5.8293/Pbase);
Bus[32].Load.S:=complex(34.8182/Pbase,2
0.6567/Pbase);
Bus[32].Load.T:=complex(29.1368/Pbase,1
7.2888/Pbase);
Bus[33].No:=33;
Bus[33].Load.R:=complex(0.0000/Pbase,0.0
000/Pbase);
Bus[33].Load.S:=complex(0.0000/Pbase,0.0
000/Pbase);
Bus[33].Load.T:=complex(0.0000/Pbase,0.0
000/Pbase);
Bus[34].No:=34;
Bus[34].Load.R:=complex(34.2452/Pbase,2
0.3199/Pbase);// 2 5
Bus[34].Load.S:=complex(38.4076/Pbase,2
2.7897/Pbase);// 8 7
Bus[34].Load.T:=complex(34.6236/Pbase,2
0.5444/Pbase);// 0 8
Bus[35].No:=35;
Bus[35].Load.R:=complex(9.6492/Pbase,5.7
255/Pbase);
Bus[35].Load.S:=complex(12.2980/Pbase,7.
2972/Pbase);
Bus[35].Load.T:=complex(14.3792/Pbase,8.
5321/Pbase);
Bus[36].No:=36;
Bus[36].Load.R:=complex(33.1100/Pbase,1
9.6463/Pbase);
Bus[36].Load.S:=complex(32.3532/Pbase,1
9.1973/Pbase);
Bus[36].Load.T:=complex(34.4344/Pbase,2
0.4322/Pbase);
Bus[37].No:=37;
Bus[37].Load.R:=complex(32.7316/Pbase,1
9.4218/Pbase);
Bus[37].Load.S:=complex(31.5964/Pbase,1
8.7482/Pbase);
Bus[37].Load.T:=complex(34.0560/Pbase,2
0.2076/Pbase);
Bus[38].No:=38;
Bus[38].Load.R:=complex(27.6232/Pbase,1
6.3906/Pbase);
Bus[38].Load.S:=complex(31.5964/Pbase,1
8.7482/Pbase);
Bus[38].Load.T:=complex(29.3260/Pbase,1
7.4010/Pbase);
Bus[39].No:=39;
Bus[39].Load.R:=complex(7.3788/Pbase,4.3
783/Pbase);
Bus[39].Load.S:=complex(0.7568/Pbase,0.4
491/Pbase);
Bus[39].Load.T:=complex(7.7572/Pbase,4.6
029/Pbase);
Bus[40].No:=40;

```


Bus[18].Load.S:=complex(0.0000/Pbase,0.000/Pbase);

Bus[18].Load.T:=complex(0.0000/Pbase,0.000/Pbase);

Bus[19].No:=19;

Bus[19].Load.R:=complex(8.5140/Pbase,5.0519/Pbase);

Bus[19].Load.S:=complex(9.4600/Pbase,5.6132/Pbase);

Bus[19].Load.T:=complex(9.0816/Pbase,5.3887/Pbase);

Bus[20].No:=20;

Bus[20].Load.R:=complex(5.6760/Pbase,3.3609/Pbase);

Bus[20].Load.S:=complex(11.3520/Pbase,6.7359/Pbase);

Bus[20].Load.T:=complex(9.4600/Pbase,5.6132/Pbase);

Bus[21].No:=21;

Bus[21].Load.R:=complex(0.0000/Pbase,0.0000/Pbase);

Bus[21].Load.S:=complex(0.0000/Pbase,0.0000/Pbase);

Bus[21].Load.T:=complex(0.0000/Pbase,0.0000/Pbase);

Bus[22].No:=22;

Bus[22].Load.R:=complex(0.0000/Pbase,0.000/Pbase);

Bus[22].Load.S:=complex(0.0000/Pbase,0.000/Pbase);

Bus[22].Load.T:=complex(0.0000/Pbase,0.000/Pbase);

Bus[23].No:=23;

Bus[23].Load.R:=complex(0.0000/Pbase,0.000/Pbase);

Bus[23].Load.S:=complex(0.0000/Pbase,0.000/Pbase);

Bus[23].Load.T:=complex(0.0000/Pbase,0.000/Pbase);

Bus[24].No:=24;

Bus[24].Load.R:=complex(0.0000/Pbase,0.000/Pbase);

Bus[24].Load.S:=complex(0.0000/Pbase,0.000/Pbase);

Bus[24].Load.T:=complex(0.0000/Pbase,0.000/Pbase);

Bus[25].No:=25;

Bus[25].Load.R:=complex(17.2172/Pbase,10.2161/Pbase);

Bus[25].Load.S:=complex(11.3520/Pbase,6.7359/Pbase);

Bus[25].Load.T:=complex(11.3520/Pbase,6.7359/Pbase);

Bus[26].No:=26;

Bus[26].Load.R:=complex(37.8400/Pbase,22.4529/Pbase);

Bus[26].Load.S:=complex(39.7320/Pbase,23.5756/Pbase);

Bus[26].Load.T:=complex(46.3540/Pbase,27.5048/Pbase);

Bus[27].No:=27;

Bus[27].Load.R:=complex(12.2980/Pbase,7.2972/Pbase);

Bus[27].Load.S:=complex(3.7840/Pbase,2.2453/Pbase);

Bus[27].Load.T:=complex(10.4060/Pbase,6.1746/Pbase);

Bus[28].No:=28;

Bus[28].Load.R:=complex(43.5160/Pbase,25.8209/Pbase);

Bus[28].Load.S:=complex(45.4080/Pbase,26.9435/Pbase);

Bus[28].Load.T:=complex(40.6780/Pbase,24.1369/Pbase);

Bus[29].No:=29;

Bus[29].Load.R:=complex(14.0008/Pbase,8.3076/Pbase);

Bus[40].Load.R:=complex(8.3248/Pbase,4.9396/Pbase);

Bus[40].Load.S:=complex(11.9196/Pbase,7.0727/Pbase);

Bus[40].Load.T:=complex(10.9736/Pbase,6.5114/Pbase);
Bus[41].No:=41;

Bus[41].Load.R:=complex(6.2436/Pbase,3.7047/Pbase);

Bus[41].Load.S:=complex(15.5144/Pbase,9.2057/Pbase);

Bus[41].Load.T:=complex(14.0008/Pbase,8.3076/Pbase);
Bus[42].No:=42;

Bus[42].Load.R:=complex(17.9740/Pbase,10.6651/Pbase);

Bus[42].Load.S:=complex(18.9200/Pbase,11.2265/Pbase);

Bus[42].Load.T:=complex(18.1632/Pbase,10.7774/Pbase);
Bus[43].No:=43;

Bus[43].Load.R:=complex(0.0000/Pbase,0.0000/Pbase);

Bus[43].Load.S:=complex(0.0000/Pbase,0.0000/Pbase);

Bus[43].Load.T:=complex(0.0000/Pbase,0.0000/Pbase);
Bus[44].No:=44;

Bus[44].Load.R:=complex(3.5948/Pbase,2.1330/Pbase); // 3 6

Bus[44].Load.S:=complex(4.5408/Pbase,2.6944/Pbase); // 2 7

Bus[44].Load.T:=complex(7.7572/Pbase,4.6029/Pbase); // 0 0
Bus[45].No:=45;

Bus[45].Load.R:=complex(17.9740/Pbase,10.6651/Pbase);

Bus[45].Load.S:=complex(22.7040/Pbase,13.4718/Pbase);

Bus[45].Load.T:=complex(15.1360/Pbase,8.9812/Pbase);
Bus[46].No:=46;

Bus[46].Load.R:=complex(3.7840/Pbase,2.2453/Pbase);

Bus[46].Load.S:=complex(12.4872/Pbase,7.4095/Pbase);

Bus[46].Load.T:=complex(13.6224/Pbase,8.0831/Pbase);
Bus[47].No:=47;

Bus[47].Load.R:=complex(17.0280/Pbase,10.1038/Pbase);

Bus[47].Load.S:=complex(10.9736/Pbase,6.5114/Pbase);

Bus[47].Load.T:=complex(15.1360/Pbase,8.9812/Pbase);
Bus[48].No:=48;

Bus[48].Load.R:=complex(3.7840/Pbase,2.2453/Pbase); //5 5

Bus[48].Load.S:=complex(9.4600/Pbase,5.6132/Pbase); //0 5

Bus[48].Load.T:=complex(3.7840/Pbase,2.2403/Pbase); //7 2
Bus[49].No:=49;

Bus[49].Load.R:=complex(0.0000/Pbase,0.0000/Pbase);

Bus[49].Load.S:=complex(0.0000/Pbase,0.0000/Pbase);

Bus[49].Load.T:=complex(0.0000/Pbase,0.0000/Pbase);
Bus[50].No:=50;

Bus[50].Load.R:=complex(2.2704/Pbase,1.3472/Pbase);

Bus[50].Load.S:=complex(2.4596/Pbase,1.4594/Pbase);

```

Bus[50].Load.T:=complex(6.0544/Pbase,3.5
925/Pbase);
  Bus[51].No:=51;

Bus[51].Load.R:=complex(17.9740/Pbase,1
0.6651/Pbase);

Bus[51].Load.S:=complex(22.1364/Pbase,1
3.1350/Pbase);

Bus[51].Load.T:=complex(27.8124/Pbase,1
6.5029/Pbase);
  Bus[52].No:=52;

Bus[52].Load.R:=complex(0.3784/Pbase,0.2
245/Pbase);

Bus[52].Load.S:=complex(1.3244/Pbase,0.7
859/Pbase);

Bus[52].Load.T:=complex(4.7300/Pbase,2.8
066/Pbase);
  Bus[53].No:=53;

Bus[53].Load.R:=complex(6.2436/Pbase,3.7
047/Pbase);

Bus[53].Load.S:=complex(8.8924/Pbase,5.2
764/Pbase);

Bus[53].Load.T:=complex(8.1356/Pbase,4.8
274/Pbase);
  Bus[54].No:=54;

Bus[54].Load.R:=complex(41.8132/Pbase,2
4.8105/Pbase);

Bus[54].Load.S:=complex(35.0020/Pbase,2
0.7690/Pbase);

Bus[54].Load.T:=complex(22.7040/Pbase,1
3.4718/Pbase);
  Bus[55].No:=55;

Bus[55].Load.R:=complex(35.9480/Pbase,2
1.3303/Pbase);

Bus[55].Load.S:=complex(25.7312/Pbase,1
5.2680/Pbase);

Bus[55].Load.T:=complex(31.2180/Pbase,1
8.5237/Pbase);
  for i:=0 to N do
  begin

```

```

Vr[i].R:=complex(1,0);
Vr[i].S:=complex(1,0);
Vr[i].T:=complex(1,0);
end;
With Za[1] do
begin
  No:=1;
  Nbhind:=0;
  LnAHead1:=2;
  LnAHead2:=0;
  LnAHead3:=0;
  Panjang:=303;
  GMR:=5.2365*0.001;
  R:=0.2162*0.001;
end;
With Za[2] do
begin
  No:=2;
  Nbhind:=1;
  LnAHead1:=3;
  LnAHead2:=53;
  LnAHead3:=0;
  Panjang:=412;
  GMR:=5.2365*0.001;
  R:=0.2162*0.001;
end;
With Za[3] do
begin
  No:=3;
  Nbhind:=2;
  LnAHead1:=4;
  LnAHead2:=0;
  LnAHead3:=0;
  Panjang:=19;
  GMR:=5.2365*0.001;
  R:=0.2162*0.001;
end;
With Za[4] do
begin
  No:=4;
  Nbhind:=3;
  LnAHead1:=5;
  LnAHead2:=0;
  LnAHead3:=0;
  Panjang:=252;
  GMR:=5.2365*0.001;
  R:=0.2162*0.001;
end;
With Za[5] do
begin
  No:=5;
  Nbhind:=4;
  LnAHead1:=6;
  LnAHead2:=0;
  LnAHead3:=0;

```

```

Panjang:=148;
GMR:=5.2365*0.001;
R:=0.2162*0.001;
end;
With Za[6] do
begin
No:=6;
Nbhind:=5;
LnAHead1:=7;
LnAHead2:=54;
LnAHead3:=0;
Panjang:=266;
GMR:=5.2365;
R:=0.2162*0.001;
end;
With Za[7] do
begin
No:=7;
Nbhind:=6;
LnAHead1:=8;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=63;
GMR:=5.2365;
R:=0.2162*0.001;
end;
With Za[8] do
begin
No:=8;
Nbhind:=7;
LnAHead1:=9;
LnAHead2:=29;
LnAHead3:=0;
Panjang:=150;
GMR:=5.2365*0.001;
R:=0.2162*0.001;
end;
With Za[9] do
begin
No:=9;
Nbhind:=8;
LnAHead1:=10;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=303;
GMR:=5.2365*0.001;
R:=0.2162*0.001;
end;
With Za[10] do
begin
No:=10;
Nbhind:=9;
LnAHead1:=11;
LnAHead2:=0;
LnAHead3:=0;

```

```

Panjang:=86;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[11] do
begin
No:=11;
Nbhind:=10;
LnAHead1:=12;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=202;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[12] do
begin
No:=12;
Nbhind:=11;
LnAHead1:=13;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=324;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[13] do
begin
No:=13;
Nbhind:=12;
LnAHead1:=14;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=217.5;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[14] do
begin
No:=14;
Nbhind:=13;
LnAHead1:=15;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=86;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[15] do
begin
No:=15;
Nbhind:=14;
LnAHead1:=16;
LnAHead2:=0;
LnAHead3:=0;

```

```
Panjang:=146;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[16] do
begin
No:=16;
Nbhind:=15;
LnAHead1:=17;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=410;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[17] do
begin
No:=17;
Nbhind:=16;
LnAHead1:=18;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=96;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[18] do
begin
No:=18;
Nbhind:=17;
LnAHead1:=19;
LnAHead2:=39;
LnAHead3:=0;
Panjang:=328;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[19] do
begin
No:=19;
Nbhind:=18;
LnAHead1:=20;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=241;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[20] do
begin
No:=20;
Nbhind:=19;
LnAHead1:=21;
LnAHead2:=0;
LnAHead3:=0;
```

```
Panjang:=51;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[21] do
begin
No:=21;
Nbhind:=20;
LnAHead1:=22;
LnAHead2:=40;
LnAHead3:=0;
Panjang:=348;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[22] do
begin
No:=22;
Nbhind:=21;
LnAHead1:=23;
LnAHead2:=41;
LnAHead3:=0;
Panjang:=386;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[23] do
begin
No:=23;
Nbhind:=22;
LnAHead1:=24;
LnAHead2:=43;
LnAHead3:=0;
Panjang:=145;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[24] do
begin
No:=24;
Nbhind:=23;
LnAHead1:=25;
LnAHead2:=42;
LnAHead3:=0;
Panjang:=84;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[25] do
begin
No:=25;
Nbhind:=24;
LnAHead1:=26;
LnAHead2:=0;
LnAHead3:=0;
```

```

Panjang:=35;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[26] do
begin
No:=26;
Nbhind:=25;
LnAHead1:=27;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=162;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[27] do
begin
No:=27;
Nbhind:=26;
LnAHead1:=28;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=265;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[28] do
begin
No:=28;
Nbhind:=27;
LnAHead1:=0;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=215;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[29] do
begin
No:=29;
Nbhind:=8;
LnAHead1:=30;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=234;
GMR:=5.2365*0.001;
R:=0.2162*0.001;
end;
With Za[30] do
begin
No:=30;
Nbhind:=29;
LnAHead1:=31;
LnAHead2:=35;
LnAHead3:=0;

```

```

Panjang:=42;
GMR:=5.2365*0.001;
R:=0.2162*0.001;
end;
With Za[31] do
begin
No:=31;
Nbhind:=30;
LnAHead1:=32;
LnAHead2:=36;
LnAHead3:=38;
Panjang:=201;
GMR:=5.2365*0.001;
R:=0.2162*0.001;
end;
With Za[32] do
begin
No:=32;
Nbhind:=31;
LnAHead1:=33;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=205;
GMR:=5.2365*0.001;
R:=0.2162*0.001;
end;
With Za[33] do
begin
No:=33;
Nbhind:=32;
LnAHead1:=34;
LnAHead2:=37;
LnAHead3:=0;
Panjang:=290;
GMR:=5.2365*0.001;
R:=0.2162*0.001;
end;
With Za[34] do
begin
No:=34;
Nbhind:=33;
LnAHead1:=0;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=88;
GMR:=5.2365*0.001;
R:=0.2162*0.001;
end;
With Za[35] do
begin
No:=35;
Nbhind:=30;
LnAHead1:=0;
LnAHead2:=0;
LnAHead3:=0;

```

```

Panjang:=221;
GMR:=5.2365*0.001;
R:=0.2162*0.001;
end;
With Za[36] do
begin
No:=36;
Nbhind:=31;
LnAHead1:=0;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=161;
GMR:=5.2365*0.001;
R:=0.2162*0.001;
end;
With Za[37] do
begin
No:=37;
Nbhind:=33;
LnAHead1:=0;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=368;
GMR:=5.2365*0.001;
R:=0.2162*0.001;
end;
With Za[38] do
begin
No:=38;
Nbhind:=31;
LnAHead1:=0;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=92;
GMR:=5.2365*0.001;
R:=0.2162*0.001;
end;
With Za[39] do
begin
No:=39;
Nbhind:=18;
LnAHead1:=0;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=79;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[40] do
begin
No:=40;
Nbhind:=21;
LnAHead1:=0;
LnAHead2:=0;
LnAHead3:=0;

```

```

Panjang:=35;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[41] do
begin
No:=41;
Nbhind:=22;
LnAHead1:=0;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=150;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[42] do
begin
No:=42;
Nbhind:=24;
LnAHead1:=0;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=140;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[43] do
begin
No:=43;
Nbhind:=23;
LnAHead1:=44;
LnAHead2:=48;
LnAHead3:=0;
Panjang:=208;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[44] do
begin
No:=44;
Nbhind:=43;
LnAHead1:=45;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=154;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[45] do
begin
No:=45;
Nbhind:=44;
LnAHead1:=46;
LnAHead2:=0;
LnAHead3:=0;

```

```

Panjang:=195;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[46] do
begin
No:=46;
Nbhind:=45;
LnAHead1:=47;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=120;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[47] do
begin
No:=47;
Nbhind:=46;
LnAHead1:=0;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=119;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[48] do
begin
No:=48;
Nbhind:=43;
LnAHead1:=49;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=37;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[49] do
begin
No:=49;
Nbhind:=48;
LnAHead1:=50;
LnAHead2:=51;
LnAHead3:=52;
Panjang:=296;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[50] do
begin
No:=50;
Nbhind:=49;
LnAHead1:=0;
LnAHead2:=0;
LnAHead3:=0;

```

```

Panjang:=85;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[51] do
begin
No:=51;
Nbhind:=49;
LnAHead1:=0;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=37;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[52] do
begin
No:=52;
Nbhind:=49;
LnAHead1:=0;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=259;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[53] do
begin
No:=53;
Nbhind:=2;
LnAHead1:=0;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=98;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[54] do
begin
No:=54;
Nbhind:=6;
LnAHead1:=55;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=75;
GMR:=5.2365*0.001;
R:=0.2162*0.001;
end;
With Za[55] do
begin
No:=55;
Nbhind:=54;
LnAHead1:=0;
LnAHead2:=0;
LnAHead3:=0;

```



```

Panjang:=146;
GMR:=5.2365*0.001;
R:=0.2162*0.001;
end;
terminal[1]:=28;
terminal[2]:=42;
terminal[3]:=52;
terminal[4]:=51;
terminal[5]:=50;
terminal[6]:=47;
terminal[7]:=41;
terminal[8]:=40;
terminal[9]:=39;
terminal[10]:=34;
terminal[11]:=37;
terminal[12]:=38;
terminal[13]:=36;
terminal[14]:=35;
terminal[15]:=55;
terminal[16]:=53;
end;

```

```

procedure EncodeData(var
rV,rI,rSa,rSL,rSumGen,rSumLoad,rSumLoss:Arr2);
var i:integer;
begin
SetLength(rV,N+1,7);
SetLength(rI,N+1,7);
SetLength(rSa,N+1,7);
SetLength(rSL,N+1,7);
SetLength(rSumGen,4,3);
SetLength(rSumLoad,4,3);
SetLength(rSumLoss,4,3);

```

```

rSumGen[1,1]:=Sa[1].R.re;rSumGen[1,2]:=
Sa[1].R.im;

```

```

rSumGen[2,1]:=Sa[1].S.re;rSumGen[2,2]:=S
a[1].S.im;

```

```

rSumGen[3,1]:=Sa[1].T.re;rSumGen[3,2]:=
Sa[1].T.im;

```

```

rSumLoss[1,1]:=0;rSumLoss[1,2]:=0;
rSumLoss[2,1]:=0;rSumLoss[2,2]:=0;
rSumLoss[3,1]:=0;rSumLoss[3,2]:=0;
rSumLoad[1,1]:=0;rSumLoad[1,2]:=0;
rSumLoad[2,1]:=0;rSumLoad[2,2]:=0;
rSumLoad[3,1]:=0;rSumLoad[3,2]:=0;

```

```

for i:=1 to N do

```

```

begin

```

```

rV[i,1]:=Vr[i].R.re;
rV[i,2]:=Vr[i].R.im;
rV[i,3]:=Vr[i].S.re;
rV[i,4]:=Vr[i].S.im;

```

```

rV[i,5]:=Vr[i].T.re;
rV[i,6]:=Vr[i].T.im;
rI[i,1]:=Ir[i].R.re;
rI[i,2]:=Ir[i].R.im;
rI[i,3]:=Ir[i].S.re;
rI[i,4]:=Ir[i].S.im;
rI[i,5]:=Ir[i].T.re;
rI[i,6]:=Ir[i].T.im;
rSa[i,1]:=Sa[i].R.re;
rSa[i,2]:=Sa[i].R.im;
rSa[i,3]:=Sa[i].S.re;
rSa[i,4]:=Sa[i].S.im;
rSa[i,5]:=Sa[i].T.re;
rSa[i,6]:=Sa[i].T.im;
rSL[i,1]:=SL[i].R.re;
rSL[i,2]:=SL[i].R.im;
rSL[i,3]:=SL[i].S.re;
rSL[i,4]:=SL[i].S.im;
rSL[i,5]:=SL[i].T.re;
rSL[i,6]:=SL[i].T.im;
rSumLoss[1,1]:=rSumLoss[1,1]+rSL[i,1];
rSumLoss[1,2]:=rSumLoss[1,2]+rSL[i,2];
rSumLoss[2,1]:=rSumLoss[2,1]+rSL[i,3];
rSumLoss[2,2]:=rSumLoss[2,2]+rSL[i,4];
rSumLoss[3,1]:=rSumLoss[3,1]+rSL[i,5];
rSumLoss[3,2]:=rSumLoss[3,2]+rSL[i,6];

```

```

rSumLoad[1,1]:=rSumLoad[1,1]+Bus[i].Loa
d.R.re;

```

```

rSumLoad[1,2]:=rSumLoad[1,2]+Bus[i].Loa
d.R.im;

```

```

rSumLoad[2,1]:=rSumLoad[2,1]+Bus[i].Loa
d.S.re;

```

```

rSumLoad[2,2]:=rSumLoad[2,2]+Bus[i].Loa
d.S.im;

```

```

rSumLoad[3,1]:=rSumLoad[3,1]+Bus[i].Loa
d.T.re;

```

```

rSumLoad[3,2]:=rSumLoad[3,2]+Bus[i].Loa
d.T.im;

```

```

end;

```

```

end;

```

```

procedure DataSkripsiRecon;

```

```

var i:integer;

```

```

begin

```

```

N:=22;

```

```

Vbase:=20;KonstV:=1000;

```

```

Pbase:=100;KonstP:=1000;

```

```

Zbase:=sqr(Vbase*KonstV)/(Pbase*KonstP)
;
Ibase:=Vbase*KonstV/Zbase;
SetLength(Bus,N+1);
SetLength(Vr,N+1);
SetLength(Za,N+1);
SetLength(terminal,17+1);
Bus[1].No:=1;

Bus[1].Load.R:=complex(427.0/Pbase,264.7
/Pbase);

Bus[1].Load.S:=complex(431.7/Pbase,267.5
/Pbase);

Bus[1].Load.T:=complex(428.7/Pbase,225.0
/Pbase);
Bus[2].No:=2;

Bus[2].Load.R:=complex(271.8/Pbase,168.4
/Pbase);

Bus[2].Load.S:=complex(298.4/Pbase,184.9
/Pbase);

Bus[2].Load.T:=complex(303.1/Pbase,143.2
/Pbase);
Bus[3].No:=3;

Bus[3].Load.R:=complex(117.9/Pbase,73.08
/Pbase);

Bus[3].Load.S:=complex(100.6/Pbase,62.34
/Pbase);

Bus[3].Load.T:=complex(104.8/Pbase,62.11
/Pbase);
Bus[4].No:=4;

Bus[4].Load.R:=complex(169.6/Pbase,105.1
/Pbase);

Bus[4].Load.S:=complex(146.9/Pbase,91.02
/Pbase);

Bus[4].Load.T:=complex(151.7/Pbase,89.35
/Pbase);
Bus[5].No:=5;

Bus[5].Load.R:=complex(176.5/Pbase,109.4
/Pbase);

Bus[5].Load.S:=complex(203.0/Pbase,125.8
/Pbase);

Bus[5].Load.T:=complex(208.4/Pbase,93.0/
Pbase);
Bus[6].No:=6;

Bus[6].Load.R:=complex(143.2/Pbase,88.77
/Pbase);

Bus[6].Load.S:=complex(120.7/Pbase,74.82
/Pbase);

Bus[6].Load.T:=complex(122.3/Pbase,75.45
/Pbase);
Bus[7].No:=7;

Bus[7].Load.R:=complex(114.6/Pbase,71.03
/Pbase);

Bus[7].Load.S:=complex(127.0/Pbase,78.73
/Pbase);

Bus[7].Load.T:=complex(138.2/Pbase,60.38
/Pbase);
Bus[8].No:=8;

Bus[8].Load.R:=complex(294.5/Pbase,182.5
/Pbase);

Bus[8].Load.S:=complex(270.4/Pbase,167.6
/Pbase);

Bus[8].Load.T:=complex(295.7/Pbase,155.1
/Pbase);
Bus[9].No:=9;

Bus[9].Load.R:=complex(188.3/Pbase,116.7
/Pbase);

Bus[9].Load.S:=complex(165.5/Pbase,102.5
/Pbase);

Bus[9].Load.T:=complex(166.1/Pbase,99.21
/Pbase);
Bus[10].No:=10;

Bus[10].Load.R:=complex(67.39/Pbase,41.7
6/Pbase);

Bus[10].Load.S:=complex(55.65/Pbase,34.4
9/Pbase);

Bus[10].Load.T:=complex(61.34/Pbase,35.5
/Pbase);
Bus[11].No:=11;

```

Bus[11].Load.R:=complex(172.9/Pbase,107.2/Pbase);

Bus[11].Load.S:=complex(158.3/Pbase,98.12/Pbase);

Bus[11].Load.T:=complex(148.9/Pbase,91.1/Pbase);
Bus[12].No:=12;

Bus[12].Load.R:=complex(156.5/Pbase,97.02/Pbase);

Bus[12].Load.S:=complex(115.2/Pbase,71.41/Pbase);

Bus[12].Load.T:=complex(95.22/Pbase,82.47/Pbase);
Bus[13].No:=13;

Bus[13].Load.R:=complex(41.7/Pbase,25.84/Pbase);

Bus[13].Load.S:=complex(39.02/Pbase,24.18/Pbase);

Bus[13].Load.T:=complex(38.9/Pbase,21.97/Pbase);
Bus[14].No:=14;

Bus[14].Load.R:=complex(4.07/Pbase,2.522/Pbase);

Bus[14].Load.S:=complex(3.101/Pbase,1.922/Pbase);

Bus[14].Load.T:=complex(3.101/Pbase,2.144/Pbase);
Bus[15].No:=15;

Bus[15].Load.R:=complex(181.5/Pbase,112.5/Pbase);

Bus[15].Load.S:=complex(157.2/Pbase,97.43/Pbase);

Bus[15].Load.T:=complex(165.9/Pbase,95.61/Pbase);
Bus[16].No:=16;

Bus[16].Load.R:=complex(0.00/Pbase,0.00/Pbase);

Bus[16].Load.S:=complex(0.00/Pbase,0.00/Pbase);

Bus[16].Load.T:=complex(0.00/Pbase,0.00/Pbase);
Bus[17].No:=17;

Bus[17].Load.R:=complex(0.00/Pbase,0.00/Pbase);

Bus[17].Load.S:=complex(0.00/Pbase,0.00/Pbase);

Bus[17].Load.T:=complex(0.00/Pbase,0.00/Pbase);
Bus[18].No:=18;

Bus[18].Load.R:=complex(25.18/Pbase,15.6/Pbase);

Bus[18].Load.S:=complex(38.62/Pbase,23.93/Pbase);

Bus[18].Load.T:=complex(32.5/Pbase,13.26/Pbase);
Bus[19].No:=19;

Bus[19].Load.R:=complex(155.8/Pbase,96.56/Pbase);

Bus[19].Load.S:=complex(131.3/Pbase,81.38/Pbase);

Bus[19].Load.T:=complex(96.56/Pbase,82.08/Pbase);
Bus[20].No:=20;

Bus[20].Load.R:=complex(254.1/Pbase,157.5/Pbase);

Bus[20].Load.S:=complex(257.9/Pbase,159.8/Pbase);

Bus[20].Load.T:=complex(208.3/Pbase,133.9/Pbase);
Bus[21].No:=21;

Bus[21].Load.R:=complex(46.72/Pbase,28.95/Pbase);

Bus[21].Load.S:=complex(22.46/Pbase,13.92/Pbase);

Bus[21].Load.T:=complex(58.94/Pbase,24.61/Pbase);

```

Bus[22].No:=22;
Bus[22].Load.R:=complex(267.8/Pbase,165.9/Pbase);
Bus[22].Load.S:=complex(248.5/Pbase,154.0/Pbase);
Bus[22].Load.T:=complex(267.6/Pbase,141.1/Pbase);
for i:=0 to N do
begin
  Vr[i].R:=complex(1,0);
  Vr[i].S:=complex(1,0);
  Vr[i].T:=complex(1,0);
end;
With Za[1] do
begin
  No:=1;
  Nbhind:=0;
  LnAHead1:=2;
  LnAHead2:=0;
  LnAHead3:=0;
  Panjang:=5409;
  GMR:=5.2365*0.001;
  R:=0.2162*0.001;
end;
With Za[2] do
begin
  No:=2;
  Nbhind:=1;
  LnAHead1:=0;
  LnAHead2:=0;
  LnAHead3:=0;
  Panjang:=3014;
  GMR:=5.2365*0.001;
  R:=0.2162*0.001;
end;
With Za[3] do
begin
  No:=3;
  Nbhind:=0;
  LnAHead1:=4;
  LnAHead2:=0;
  LnAHead3:=0;
  Panjang:=1479;
  GMR:=5.2365*0.001;
  R:=0.2162*0.001;
end;
With Za[4] do
begin
  No:=4;
  Nbhind:=3;
  LnAHead1:=5;
  LnAHead2:=6;
  LnAHead3:=0;
  Panjang:=1224;
  GMR:=5.2365*0.001;
  R:=0.2162*0.001;
end;
With Za[5] do
begin
  No:=5;
  Nbhind:=4;
  LnAHead1:=0;
  LnAHead2:=0;
  LnAHead3:=0;
  Panjang:=3412.5;
  GMR:=5.2365*0.001;
  R:=0.2162*0.001;
end;
With Za[6] do
begin
  No:=6;
  Nbhind:=4;
  LnAHead1:=7;
  LnAHead2:=0;
  LnAHead3:=0;
  Panjang:=1705.5;
  GMR:=5.2365;
  R:=0.2162*0.001;
end;
With Za[7] do
begin
  No:=7;
  Nbhind:=6;
  LnAHead1:=0;
  LnAHead2:=0;
  LnAHead3:=0;
  Panjang:=1492;
  GMR:=5.2365;
  R:=0.2162*0.001;
end;
With Za[8] do
begin
  No:=8;
  Nbhind:=0;
  LnAHead1:=9;
  LnAHead2:=7;
  LnAHead3:=0;
  Panjang:=4915;
  GMR:=5.2365*0.001;
  R:=0.2162*0.001;
end;
With Za[9] do
begin
  No:=9;
  Nbhind:=8;
  LnAHead1:=10;
  LnAHead2:=0;

```

LnAHead3:=0;
Panjang:=3782;
GMR:=5.2365*0.001;
R:=0.2162*0.001;

end;

With Za[10] do

begin

No:=10;
Nbhind:=9;
LnAHead1:=11;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=994;
GMR:=4.6837*0.001;
R:=0.2688*0.001;

end;

With Za[11] do

begin

No:=11;
Nbhind:=10;
LnAHead1:=0;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=2738;
GMR:=4.6837*0.001;
R:=0.2688*0.001;

end;

With Za[12] do

begin

No:=12;
Nbhind:=0;
LnAHead1:=13;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=2059.4;
GMR:=4.6837*0.001;
R:=0.2688*0.001;

end;

With Za[13] do

begin

No:=13;
Nbhind:=12;
LnAHead1:=15;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=1645.8;
GMR:=4.6837*0.001;
R:=0.2688*0.001;

end;

With Za[14] do

begin

No:=14;
Nbhind:=0;
LnAHead1:=0;
LnAHead2:=0;

LnAHead3:=0;
Panjang:=1039.3;
GMR:=4.6837*0.001;
R:=0.2688*0.001;

end;

With Za[15] do

begin

No:=15;
Nbhind:=13;
LnAHead1:=16;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=2582;
GMR:=4.6837*0.001;
R:=0.2688*0.001;

end;

With Za[16] do

begin

No:=16;
Nbhind:=15;
LnAHead1:=0;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=131;
GMR:=4.6837*0.001;
R:=0.2688*0.001;

end;

With Za[17] do

begin

No:=17;
Nbhind:=0;
LnAHead1:=18;
LnAHead2:=19;
LnAHead3:=0;
Panjang:=1817;
GMR:=4.6837*0.001;
R:=0.2688*0.001;

end;

With Za[18] do

begin

No:=18;
Nbhind:=17;
LnAHead1:=0;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=3008.5;
GMR:=4.6837*0.001;
R:=0.2688*0.001;

end;

With Za[19] do

begin

No:=19;
Nbhind:=17;
LnAHead1:=14;
LnAHead2:=0;

```

LnAHead3:=0;
Panjang:=2627;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[20] do
begin
No:=20;
Nbhind:=19;
LnAHead1:=21;
LnAHead2:=22;
LnAHead3:=0;
Panjang:=529;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[21] do
begin
No:=21;
Nbhind:=20;
LnAHead1:=0;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=1476;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[22] do
begin
No:=22;
Nbhind:=21;
LnAHead1:=0;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=4682.8;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
terminal[1]:=3;
terminal[2]:=6;
terminal[3]:=12;
terminal[4]:=15;
terminal[5]:=17;
terminal[6]:=22;
terminal[7]:=23;
end;

//=====
procedure DataSkripsiReconSebelum;
var i:integer;
begin
N:=22;
Vbase:=20;KonstV:=1000;
Pbase:=100;KonstP:=1000;
Zbase:=sqr(Vbase*KonstV)/(Pbase*KonstP)
;
Ibase:=Vbase*KonstV/Zbase;
SetLength(Bus,N+1);
SetLength(Vr,N+1);
SetLength(Za,N+1);
SetLength(terminal,17+1);
Bus[1].No:=1;

Bus[1].Load.R:=complex(427.0/Pbase,264.7
/Pbase);

Bus[1].Load.S:=complex(431.7/Pbase,267.5
/Pbase);

Bus[1].Load.T:=complex(428.7/Pbase,225.0
/Pbase);
Bus[2].No:=2;

Bus[2].Load.R:=complex(271.8/Pbase,168.4
/Pbase);

Bus[2].Load.S:=complex(298.4/Pbase,184.9
/Pbase);

Bus[2].Load.T:=complex(303.1/Pbase,143.2
/Pbase);
Bus[3].No:=3;

Bus[3].Load.R:=complex(294.5/Pbase,182.5
/Pbase);

Bus[3].Load.S:=complex(270.4/Pbase,167.6
/Pbase);

Bus[3].Load.T:=complex(295.7/Pbase,155.1
/Pbase);
Bus[4].No:=4;

Bus[4].Load.R:=complex(188.3/Pbase,116.7
/Pbase);

Bus[4].Load.S:=complex(165.5/Pbase,102.5
/Pbase);

Bus[4].Load.T:=complex(166.1/Pbase,99.21
/Pbase);
Bus[5].No:=5;

Bus[5].Load.R:=complex(67.39/Pbase,41.76
/Pbase);

Bus[5].Load.S:=complex(55.65/Pbase,34.49
/Pbase);

```

Bus[5].Load.T:=complex(61.34/Pbase,35.50/Pbase);

Bus[6].No:=6;

Bus[6].Load.R:=complex(172.9/Pbase,107.2/Pbase);

Bus[6].Load.S:=complex(158.3/Pbase,98.12/Pbase);

Bus[6].Load.T:=complex(148.9/Pbase,91.10/Pbase);

Bus[7].No:=7;

Bus[7].Load.R:=complex(117.9/Pbase,73.08/Pbase);

Bus[7].Load.S:=complex(100.6/Pbase,62.34/Pbase);

Bus[7].Load.T:=complex(104.8/Pbase,62.11/Pbase);

Bus[8].No:=8;

Bus[8].Load.R:=complex(169.6/Pbase,105.1/Pbase);

Bus[8].Load.S:=complex(146.9/Pbase,91.02/Pbase);

Bus[8].Load.T:=complex(151.7/Pbase,89.35/Pbase);

Bus[9].No:=9;

Bus[9].Load.R:=complex(143.2/Pbase,88.77/Pbase);

Bus[9].Load.S:=complex(120.7/Pbase,74.82/Pbase);

Bus[9].Load.T:=complex(122.3/Pbase,75.45/Pbase);

Bus[10].No:=10;

Bus[10].Load.R:=complex(114.6/Pbase,71.03/Pbase);

Bus[10].Load.S:=complex(127.0/Pbase,78.73/Pbase);

Bus[10].Load.T:=complex(138.2/Pbase,60.38/Pbase);

Bus[11].No:=11;

Bus[11].Load.R:=complex(156.5/Pbase,97.02/Pbase);

Bus[11].Load.S:=complex(115.2/Pbase,71.41/Pbase);

Bus[11].Load.T:=complex(95.22/Pbase,82.47/Pbase);

Bus[12].No:=12;

Bus[12].Load.R:=complex(176.5/Pbase,109.4/Pbase);

Bus[12].Load.S:=complex(203.0/Pbase,125.8/Pbase);

Bus[12].Load.T:=complex(208.4/Pbase,93.00/Pbase);

Bus[13].No:=13;

Bus[13].Load.R:=complex(41.70/Pbase,25.84/Pbase);

Bus[13].Load.S:=complex(39.02/Pbase,24.18/Pbase);

Bus[13].Load.T:=complex(38.9/Pbase,21.97/Pbase);

Bus[14].No:=14;

Bus[14].Load.R:=complex(4.07/Pbase,2.522/Pbase);

Bus[14].Load.S:=complex(3.101/Pbase,1.922/Pbase);

Bus[14].Load.T:=complex(3.101/Pbase,2.144/Pbase);

Bus[15].No:=15;

Bus[15].Load.R:=complex(181.5/Pbase,112.5/Pbase);

Bus[15].Load.S:=complex(157.2/Pbase,97.43/Pbase);

Bus[15].Load.T:=complex(165.9/Pbase,95.61/Pbase);

Bus[16].No:=16;

Bus[16].Load.R:=complex(0.00/Pbase,0.00/Pbase);

```

Bus[16].Load.S:=complex(0.00/Pbase,0.00/
Pbase);

Bus[16].Load.T:=complex(0.00/Pbase,0.00/
Pbase);
  Bus[17].No:=17;

Bus[17].Load.R:=complex(0.00/Pbase,0.00/
Pbase);

Bus[17].Load.S:=complex(0.00/Pbase,0.00/
Pbase);

Bus[17].Load.T:=complex(0.00/Pbase,0.00/
Pbase);
  Bus[18].No:=18;

Bus[18].Load.R:=complex(155.8/Pbase,96.5
6/Pbase);

Bus[18].Load.S:=complex(131.3/Pbase,81.3
8/Pbase);

Bus[18].Load.T:=complex(137.6/Pbase,82.0
8/Pbase);
  Bus[19].No:=19;

Bus[19].Load.R:=complex(25.18/Pbase,15.6
0/Pbase);

Bus[19].Load.S:=complex(38.62/Pbase,23.9
3/Pbase);

Bus[19].Load.T:=complex(32.50/Pbase,13.2
6/Pbase);
  Bus[20].No:=20;

Bus[20].Load.R:=complex(254.1/Pbase,157.
5/Pbase);

Bus[20].Load.S:=complex(257.9/Pbase,159.
8/Pbase);

Bus[20].Load.T:=complex(208.3/Pbase,133.
9/Pbase);
  Bus[21].No:=21;
  Bus[21].Load.R:=complex(46.72/Pbase,
28.95/Pbase);

Bus[21].Load.S:=complex(22.46/Pbase,13.9
2/Pbase);

Bus[21].Load.T:=complex(58.94/Pbase,24.6
1/Pbase);

```

```

Bus[22].No:=22;

Bus[22].Load.R:=complex(267.8/Pbase,165.
9/Pbase);

Bus[22].Load.S:=complex(248.5/Pbase,154.
0/Pbase);

Bus[22].Load.T:=complex(267.6/Pbase,141.
1/Pbase);
  for i:=0 to N do
  begin
    Vr[i].R:=complex(1,0);
    Vr[i].S:=complex(1,0);
    Vr[i].T:=complex(1,0);
  end;
  With Za[1] do
  begin
    No:=1;
    Nbhind:=0;
    LnAHead1:=2;
    LnAHead2:=0;
    LnAHead3:=0;
    Panjang:=5409;
    GMR:=5.2365*0.001;
    R:=0.2162*0.001;
  end;
  With Za[2] do
  begin
    No:=2;
    Nbhind:=1;
    LnAHead1:=3;
    LnAHead2:=0;
    LnAHead3:=0;
    Panjang:=3014;
    GMR:=5.2365*0.001;
    R:=0.2162*0.001;
  end;
  With Za[3] do
  begin
    No:=3;
    Nbhind:=2;
    LnAHead1:=4;
    LnAHead2:=0;
    LnAHead3:=0;
    Panjang:=1479;
    GMR:=5.2365*0.001;
    R:=0.2162*0.001;
  end;
  With Za[4] do
  begin
    No:=4;
    Nbhind:=3;
    LnAHead1:=5;
    LnAHead2:=6;

```



```

LnAHead3:=0;
Panjang:=1224;
GMR:=5.2365*0.001;
R:=0.2162*0.001;
end;
With Za[5] do
begin
No:=5;
Nbhind:=4;
LnAHead1:=6;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=3412.5;
GMR:=5.2365*0.001;
R:=0.2162*0.001;
end;
With Za[6] do
begin
No:=6;
Nbhind:=5;
LnAHead1:=0;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=1705.5;
GMR:=5.2365;
R:=0.2162*0.001;
end;
With Za[7] do
begin
No:=7;
Nbhind:=0;
LnAHead1:=8;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=1492;
GMR:=5.2365;
R:=0.2162*0.001;
end;
With Za[8] do
begin
No:=8;
Nbhind:=7;
LnAHead1:=9;
LnAHead2:=7;
LnAHead3:=0;
Panjang:=4915;
GMR:=5.2365*0.001;
R:=0.2162*0.001;
end;
With Za[9] do
begin
No:=9;
Nbhind:=8;
LnAHead1:=10;
LnAHead2:=0;

```

```

LnAHead3:=0;
Panjang:=3782;
GMR:=5.2365*0.001;
R:=0.2162*0.001;
end;
With Za[10] do
begin
No:=10;
Nbhind:=9;
LnAHead1:=0;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=994;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[11] do
begin
No:=11;
Nbhind:=0;
LnAHead1:=12;
LnAHead2:=13;
LnAHead3:=0;
Panjang:=2738;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[12] do
begin
No:=12;
Nbhind:=11;
LnAHead1:=0;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=2059.4;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[13] do
begin
No:=13;
Nbhind:=11;
LnAHead1:=14;
LnAHead2:=15;
LnAHead3:=0;
Panjang:=1645.8;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[14] do
begin
No:=14;
Nbhind:=13;
LnAHead1:=0;
LnAHead2:=0;

```

```

LnAHead3:=0;
Panjang:=1039.3;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[15] do
begin
No:=15;
Nbhind:=13;
LnAHead1:=16;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=2582;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[16] do
begin
No:=16;
Nbhind:=15;
LnAHead1:=0;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=131;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[17] do
begin
No:=17;
Nbhind:=0;
LnAHead1:=18;
LnAHead2:=19;
LnAHead3:=0;
Panjang:=1817;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[18] do
begin
No:=18;
Nbhind:=17;
LnAHead1:=20;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=3008.5;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[19] do
begin
No:=19;
Nbhind:=17;
LnAHead1:=0;
LnAHead2:=0;

```

```

LnAHead3:=0;
Panjang:=2627;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[20] do
begin
No:=20;
Nbhind:=18;
LnAHead1:=21;
LnAHead2:=22;
LnAHead3:=0;
Panjang:=529;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[21] do
begin
No:=21;
Nbhind:=20;
LnAHead1:=0;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=1476;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
With Za[22] do
begin
No:=22;
Nbhind:=20;
LnAHead1:=0;
LnAHead2:=0;
LnAHead3:=0;
Panjang:=4682.8;
GMR:=4.6837*0.001;
R:=0.2688*0.001;
end;
terminal[1]:=7;
terminal[2]:=11;
terminal[3]:=13;
terminal[4]:=15;
terminal[5]:=17;
terminal[6]:=22;
terminal[7]:=23;
end;
//=====

procedure HitungFD3Phasa(const
rNbus:integer;
const
rVbase,rVKonst,rPbase,rPkonst:double;
const rBus,rBranch:Arr2;

```

```

var
rV,rI,rSa,rSL,rSumGen,rSumLoad,rSumLoss:Arr2;
var rIterasi:byte);
var i,j,k:integer;
ib,Pb:TComplex;
begin

```

```

DecodeData(rNbus,rVbase,rVkonst,rPbase,rPkonst,rBus,rBranch);
ImpedansiSaluranKm(N,Za);
Ib:=complex(Ibase,0);
Pb:=complex(Pbase,0);
for i:=1 to N do
begin
With Za[i] do
begin
for j:=1 to 3 do
begin
for k:=1 to 3 do
begin

```

```

Zs[j,k]:=complex(CxRe(Zs[j,k])/Zbase,CxIm(Zs[j,k])/Zbase);
end;
end;
end;
end;

```

```

FastDLFlow(N,Za,Bus,rIterasi,Vr,Ir,Sa,SL);

```

```

EncodeData(rV,rI,rSa,rSL,rSumGen,rSumLoad,rSumLoss);
end;

```

```

procedure HitungFD3Phasa(var
rV,rI,rSa,rSL,rSumGen,rSumLoad,
rSumLoss:Arr2;
var rIterasi:byte);
var i,j,k:integer;
ib,Pb:TComplex;
begin
DataSkripsiRecon;
ImpedansiSaluranKm(N,Za);
Ib:=complex(Ibase,0);
Pb:=complex(Pbase,0);
for i:=1 to N do
begin
With Za[i] do
begin
for j:=1 to 3 do
begin
for k:=1 to 3 do
begin

```

```

Zs[j,k]:=complex(CxRe(Zs[j,k])/Zbase,CxIm(Zs[j,k])/Zbase);
end;
end;
end;
end;

```

```

FastDLFlow(N,Za,Bus,rIterasi,Vr,Ir,Sa,SL);

```

```

EncodeData(rV,rI,rSa,rSL,rSumGen,rSumLoad,rSumLoss);
end;

```

```

procedure HitungFD3PhasaSebelum(var
rV,rI,rSa,rSL,rSumGen,rSumLoad,
rSumLoss:Arr2;
var rIterasi:byte);
var i,j,k:integer;
ib,Pb:TComplex;
begin

```

```

DataSkripsiReconSebelum;
ImpedansiSaluranKm(N,Za);
Ib:=complex(Ibase,0);
Pb:=complex(Pbase,0);
for i:=1 to N do
begin
With Za[i] do
begin
for j:=1 to 3 do
begin
for k:=1 to 3 do
begin

```

```

Zs[j,k]:=complex(CxRe(Zs[j,k])/Zbase,CxIm(Zs[j,k])/Zbase);
end;
end;
end;
end;

```

```

FastDLFlow(N,Za,Bus,rIterasi,Vr,Ir,Sa,SL);

```

```

EncodeData(rV,rI,rSa,rSL,rSumGen,rSumLoad,rSumLoss);
end;

```

```

end.

```

```

unit Recursive;

interface

uses Kompleks;

function TesJaringan(const
dJalur:Arr2):boolean;

implementation

var bus:array of boolean;
    Njar:integer;
    findLoop:boolean;

function TesJaringan(const
dJalur:Arr2):boolean;
var i:integer;

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:=1 to Njar do
    begin
      if dJalur[ke,i]=1 then
        begin
          if i <> dari then
            begin
              doRecursive(ke,i);
              if findLoop=true then
                begin
                  exit;
                end;
            end;
          end;
        end;
      end;
    end;
  end;
end;

begin
  Njar:=high(dJalur);
  SetLength(bus,Njar+1);
  for i:=1 to Njar do

```

```

begin
  bus[i]:=false;
end;
  findLoop:=false;
  result:=true;
  doRecursive(0,1);
  if findLoop=true then
    begin
      result:=false;
      exit;
    end;
  for i:=1 to Njar do
    begin
      if bus[i]=false then
        begin
          result:=false;
          exit;
        end;
      end;
    end;
  end;
end.

```

program Recon3PhasaSA;

uses

Forms,
uAbout in 'uAbout.pas' {frmAbout},
Fitness in 'Fitness.pas',
Fuzzy in 'Fuzzy.pas',
GeneticAlgorithm in
'GeneticAlgorithm.pas',
Hasil in 'Hasil.pas' {frmHasil},
IGA in 'IGA.pas',
Input in 'Input.pas' {frmInput},
input2 in 'input2.pas',
Komplex in 'Komplex.pas',
LFNRPolar in 'LFNRPolar.pas',
Matrik in 'Matrik.pas',
ParamGA in 'ParamGA.pas'
{frmParamGA},
Recursive in 'Recursive.pas',
Trapmf in 'Trapmf.pas',
Triamf in 'Triamf.pas',
TypDatGA in 'TypDatGA.pas',
TypeData in 'TypeData.pas',
uFD3Phasa in 'uFD3Phasa.pas',
UnitMath in 'UnitMath.pas',
Utama in 'Utama.pas' {frmUtama},
uUtils in 'uUtils.pas';

{SR *.res}

begin

Application.Initialize;
Application.CreateForm(TfrmUtama,
frmUtama);
Application.CreateForm(TfrmHasil,
frmHasil);
Application.CreateForm(TfrmInput,
frmInput);
Application.CreateForm(TfrmParamGA,
frmParamGA);
Application.CreateForm(TfrmAbout,
frmAbout);
Application.Run;
end.



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

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

(PERSERO) MALANG
 BANK NIAGA MALANG

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

Malang, 23 Okt. 2004

Nomor : ITN-919/I.SKP /2/04
 Lampiran : satu lembar
 Perihal : **BIMBINGAN SKRIPSI**

Kepada : Yth. Sdr. **Ir. H. TAUFIK HIDAYAT, MT**
 Dosen Institut Teknologi Nasional
 di -
 Malang

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

Nama : DYAH AYU ANGGRAENI
 Nim : 9912027
 Fakultas : Teknologi Industri
 Jurusan : Teknik Elektro
 Konsentrasi : T. Energi Listrik (S-1)

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

13 Oktober 2004 s/d 13 April 2005

Adapun tugas tersebut merupakan salah satu syarat untuk memperoleh gelar Sarjana Teknik, Jurusan Teknik Elektro. Demikian atas perhatian serta kerjasamanya kami ucapkan terima kasih



Ketua
 Jurusan Teknik Elektro

 Ir. I Made Wartana, MT
 Nip. 131 991 182

Tindakan :

1. Mahasiswa yang bersangkutan
2. Arsip.

Form. S-4a

FORMULIR BIMBINGAN SKRIPSI

Nama : Dyah Ayu Anggraeni
 Nim : 99.12.027
 Masa Bimbingan : 13 Oktober 2004 s/d 13 April 2005
 Judul Skripsi : Analisis Rekonfigurasi Jaringan Untuk Mereduksi Rugi-
 Rugi Sistem Distribusi Tiga Fasa Dengan Menggunakan
 Metode *Simulated Annealing* di GI Blimbing-Malang.

No.	Tanggal	Uraian	Parap Pembimbing
1.	22 04 04	Lanjutan bab IV (Analisa)	Ah
2.	5 11 04	Tambahkan flowchart program	Ah
3.	20 12 04	Tambahkan flow chart load flow nya	Ah
4.	20 1 05	Lanjutan Bab III.	Ah -
5.	15 02 05	Semua tampilan direvisi kembali	Ah
6.	12 03 05	Tujuan harus sinkron dg kesimpulan - penyesuaian - teori penyesuaian tambahkan rugi daya	Ah
7.	15 03 05	Kesimpulan direrun sesuaikan dari Bab IV (munculkan angka-angka) - Tujuan sesuaikan dg kesimpulan nya	Ah
8.	24 03 05	Ace mengikuti uraian.	Ah
9.			
10.			

Malang, 24 - Maret 2005
 Dosen Pembimbing,



Ir. Taufik Hidayat, MT



LEMBAR BIMBINGAN SKRIPSI

1. Nama : DYAH AYU ANGGRAENI
2. NIM : 99.12.027
3. Jurusan : Teknik Elektro S-1
4. Konsentrasi : Teknik Energi Listrik
5. Judul Skripsi : ANALISIS REKONFIGURASI
JARINGAN UNTUK MEREDUKSI
RUGI-RUGI SISTEM DISTRIBUSI
TIGA FASA DENGAN
MENGUNAKAN METODE
SIMULATED ANNEALING DI GI
BLIMBING MALANG.
6. Tanggal Mengajukan Skripsi : 25 Juni 2004
7. Tanggal Menyelesaikan Skripsi : 30 Maret 2005
8. Dosen Pembimbing : Ir. Taufik Hidayat, MT.
9. Telah Dievaluasikan Dengan Nilai : 80,0 (Delapan Puluh Koma Nol) $\frac{8}{10}$

Mengetahui,
Ketua Jurusan Teknik Elektro S-1

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

Diperiksa dan Disetujui,
Dosen Pembimbing

(Ir. Taufik Hidayat, MT.)
NIP. P. 108 700 151



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

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

Nama Mahasiswa : DYAH AYU ANGGRAENI
N.I.M. : 99.12.027
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Energi Listrik
Judul Skripsi : ANALISIS REKONFIGURASI JARINGAN
UNTUK MEREDUKSI RUGI-RUGI SISTEM
DISTRIBUSI TIGA FASA DENGAN
MENGUNAKAN METODE *SIMULATED
ANNEALING* DI GI BLIMBING MALANG

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

Hari : Rabu
Tanggal : 30 Maret 2005
Dengan Nilai : 75,50 (Tujuh Puluh Lima Koma Lima Puluh) *hm*



Ketua Majelis Penguji

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

Panitia Ujian Skripsi

Sekretaris Majelis Penguji

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

Anggota Penguji

Penguji Pertama

(Ir. Abdul Hamid, MT.)
NIP. Y. 101 880 0188

Penguji Kedua

(Ir. Djojo Priatmono, MT.)
NIP. Y. 101 850 0107