

2008
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK ENERGI LISTRIK



Disusun oleh :
VINCENTO SOMADO VILTON TUKAN
NIM. 00.12.149

SKRIPSI

OPTIMASI REKONFIGURASI SISTEM DISTRIBUSI RADIAL
20 KV MENGGUNAKAN METODE MINIMAX GAMES WITH
DEPTH FIRST SEARCH ALGORITHM (MGDFS) PADA
GARDU INDUK RUNGKUT - SURABAYA

**OPTIMASI REKONFIGURASI SISTEM DISTRIBUSI RADIAL
20 KV MENGGUNAKAN METODE *MINIMAX GAMES WITH
DEPTH FIRST SEARCH ALGORITHM* (MGDFS) PADA
GARDU INDUK RUNGKUT - SURABAYA**

SKRIPSI

**Disusun oleh :
VINCENSO SOMADO VILTON TUKAN
NIM. 00.12.149**



**KONSENTRASI TEKNIK ENERGI LISTRIK
JURUSAN TEKNIK ELEKTRO S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2008**

LEMBAR PERSETUJUAN

**OPTIMASI REKONFIGURASI SISTEM DISTRIBUSI RADIAL
20 kV MENGGUNAKAN METODE *MINIMAX GAMES WITH
DEPTH FIRST SEARCH ALGORITHM* (MGDFS) PADA
GARDU INDUK RUNGKUT - SURABAYA**

SKRIPSI

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

Disusun Oleh :

VINCENSO SOMADO VILTON TUKAN
NIM: 00.12.149

Diperiksa dan Disetujui,

Dosen Pembimbing I


Ir. H. Taufik Hidayat, MT
NIP. Y. 101 87000151

Dosen Pembimbing II


Ir. Djojo Priatmono, MT
NIP. Y. 101 8500107



Mengetahui,
Ketua Jurusan Teknik Elektro

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

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



BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

Nama Mahasiswa : VINCENSO SOMADO VILTON TUKAN
N.I.M. : 00.12.149
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Energi Listrik
Judul Skripsi : OPTIMASI REKONFIGURASI SISTEM DISTRIBUSI
RADIAL 20 kV MENGGUNAKAN METODE
MINIMAX GAMES WITH DEPTH FIRST SEARCH
ALGORITHM (MGDFS) PADA GARDU INDUK
RUNGKUT - SURABAYA

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

Hari : Rabu
Tanggal : 19 Maret 2008
Dengan Nilai : 76,25 (B+) *84*



Ketua

Ir. Mochtar Asroni, MSME
NIP. Y. 101 8100 036

Panitia Ujian Skripsi



Sekretaris

Ir. F. Yudi Limpraptono, MT
IP. Y. 103 9500 274

Anggota Penguji

Penguji Pertama

Ir. Yusuf Ismail Nakhoda, MT
NIP. Y. 101 8800 0189

Penguji Kedua

Irrine Budi S. ST, MT
NIP. 132314400

ABSTRAKSI

OPTIMASI REKONFIGURASI SISTEM DISTRIBUSI RADIAL 20 kV MENGGUNAKAN METODE *MINIMAX GAMES WITH DEPTH FIRST SEARCH ALGORITHM* (MGDFS) PADA GARDU INDUK RUNGKUT - SURABAYA

Vincenzo Somado Vilton Tukan
H. Taufik Hidayat
Djojo Priatmono

Dalam sistem distribusi yang efektif, tentunya harus benar-benar memperhatikan dan mengusahakan pelayanan yang baik kepada konsumen dengan cara penyaluran energi listrik yang lebih efektif, mulai dari sistem transmisi tenaga sampai dengan sistem distribusi. Sistem distribusi perlu perhatian khusus sebab kesalahan-kesalahan sistem distribusi ini langsung berpengaruh langsung kepada konsumen. Sesungguhnya 90% dari terputusnya aliran kepada konsumen berasal dari sistem distribusi. Di samping itu sistem distribusi pada umumnya lebih rawan, terutama lokasi beban yang bervariasi, sehingga banyak terjadi rugi-rugi daya dari sistem tersebut. Usaha untuk meminimalisasi rugi-rugi daya dari sistem distribusi dengan memformulasikan permasalahan sebagai persoalan rekonfigurasi jaringan distribusi. Penggunaan metode *Minimax Games With Depth First Search Algorithm* (MGDFS) merupakan penyelesaian yang efisien pada persoalan jaringan konfigurasi secara optimal, dengan penentuan lokasi *open switch* dalam jaringan. Penggunaan program komputer sangat efektif untuk mendukung metode ini dan secara umum memberikan perkiraan yang teliti tentang rugi-rugi daya.

Untuk mempelajari teori-teori yang terkait melalui kajian literatur, pengumpulan data-data berupa data kuantitatif, data kualitatif, serta simulasi dan pembahasan masalah pada gardu induk rungkut surabaya.

Dengan menggunakan data jaringan pada PT. PLN (Persero) Distribusi Jawa Timur, didapat rekonfigurasi dengan melakukan *open switch* dari: bus 13 ke bus 21, bus 14 ke bus 22, bus 17 ke bus 26 dan bus 36 ke bus 37 akan diperoleh reduksi rugi-rugi daya pada saluran, sehingga rugi daya sebelum rekonfigurasi $0.281 + j 0.365$ MVA, setelah rekonfigurasi rugi daya menjadi $0.248 + j 0.337$ MVA, dengan reduksi rugi-rugi dayanya $0.033 + j 0.028$ MVA. Maka didapat prosentasi reduksi rugi-rugi daya sebesar 11.74 %.

Kata Kunci : Sistem Distribusi, Rugi Daya, Rekonfigurasi Jaringan, Aliran Daya, *Minimax Games*.

KATA PENGANTAR

Dengan mengucapkan puji syukur kepada Tuhan Yang Maha Kuasa, atas Karunia dan Rahmat-Nya akhirnya penulis dapat menyelesaikan skripsi yang berjudul **"OPTIMASI REKONFIGURASI SISTEM DISTRIBUSI RADIAL 20 kV MENGGUNAKAN METODE *MINIMAX GAMES WITH DEPTH FIRST SEARCH ALGORITHM* (MGDFS) PADA GARDU INDUK RUNGKUT - SURABAYA"** Skripsi ini bertujuan untuk memenuhi kurikulum akademik yang harus ditempuh oleh setiap Mahasiswa ITN Malang guna mengakhiri pendidikan pada jenjang strata satu (S-1) Jurusan Teknik Elektro konsentrasi Teknik Energi Listrik di Institut Teknologi Nasional Malang.

Dengan segala hormat atas bimbingan, pengarahan dan bantuan yang diberikan, sehingga tersusunnya skripsi ini. Maka penulis menyampaikan terima kasih kepada:

1. Bapak Dr. Ir. Eng. Abraham Lomi, MSEE, selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. Mochtar Asroni, MSME, selaku Dekan Fakultas Teknologi Industri.
3. Bapak Ir. F. Yudi Limpraptono, MT, selaku Ketua Jurusan Teknik Elektro S-1.
4. Bapak Ir. H. Taufik Hidayat, MT, selaku Dosen Pembimbing I.
5. Bapak Ir. Djojo Priatmono, MT, selaku Dosen Pembimbing II.
7. Terima kasih banyak, juga rekan-rekan saya di Jurusan Elektro, Energi Listrik yang saling membantu dan saling mengisi dalam mengerjakan Skripsi ini.

Akhirnya penulis mengharapkan skripsi dapat berguna dan bermanfaat bagi rekan-rekan mahasiswa khususnya pada Jurusan Teknik Elektro, Konsentrasi Teknik Energi Listrik.

Malang, April 2008

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	
LEMBAR PESETUJUAN.....	i
BERITA ACARA UJIAN SKRIPSI.....	ii
ABSTRAKSI	iii
KATA PENGANTAR	iv
DAFTAR ISI.....	vi
DAFTAR GAMBAR	ix
DAFTAR TABEL.....	xi
BAB I. PENDAHULUAN	
1.1. Latar Belakang.....	1
1.2. Rumusan masalah	2
1.3. Tujuan Pembahasan.....	3
1.4. Batasan Masalah	3
1.5. Metodologi Pembahasan.....	3
1.6. Sistematika Penulisan	4
BAB II. LANDASAN TEORI	
2.1. Sistem Distribusi Tenaga Listrik	6
2.2. Struktur Jaringan Radial Distribusi Tenaga Listrik.....	7
2.3. Sistem Jaringan Distribusi Radial.....	7

2.3.1. Struktur Jaringan Pohon.....	8
2.3.2. Sistem Radial dengan <i>Tie</i> dan <i>Switch</i> Pemisah	9
2.3.3. Sistem Radial Dengan Pembagian Phasa Area.....	10
2.3.4. Sistem Radial Dengan Beban Terpusat	10
2.4. Faktor Daya.....	11
2.5. Perbaikan Faktor Daya dan Kapasitas Sistem	13
2.6. Perhitungan Pengaruh Perbaikan Faktor Daya	14
2.7. Sistem Per-Unit.....	16

**BAB III. PENERAPAN METODE MINIMAX GAMES WITH DEPTH
FIRST SEARCH ALGORITHM (MGDFS) PADA
REKONFIGURASI SISTEM DISTRIBUSI RADIAL 20 kV**

3.1. Analisa Aliran Daya Jaringan Radial.....	18
3.1.1. Tujuan.....	18
3.1.2. Metode <i>Newton Raphson</i>	18
3.1.3. Algoritma Perhitungan Aliran Daya Dengan Metode <i>Newton Raphson</i>	22
3.1.4. Diagram Alir Aliran Daya <i>Newton Raphson</i>	24
3.2. Metode <i>Minimax Games Depth First Search</i> (MGDFS).....	25
3.2.1. Metode Pencarian dan Evaluasi.....	26
3.2.2. Pencarian Mendalam Pertama (<i>Depth First Search</i>).....	27
3.2.3. <i>Minimax</i>	29
3.2.4. Permainan Dalam <i>Minimax</i>	31
3.2.5. $\alpha\beta$ - <i>Cutoff</i>	31

3.2.6. Menunggu Hingga Situasi Menjadi Tenang.....	36
3.2.7. Pencarian Sekunder.....	37
3.2.8. <i>Book Moves</i>	37
3.3. Penerapan Metode <i>Minimax Games With Depth First Search Algorithm</i> (<i>MGDFS</i>) Pada Rekonfigurasi Sistem Distribusi Radial 20 kV.....	38

BAB IV. SIMULASI DAN ANALISA SISTEM

4.1. Fitur Aplikasi.....	42
4.2. Analisa Rekonfigurasi Jaringan Distribusi Dengan Metode <i>Minimax</i> <i>Games Depth First Search (MGDFS)</i>	46
4.2.1. Data Perhitungan.....	46
4.2.1.1. Data Saluran.....	50
4.2.1.2. Data Pembebanan.....	50
4.3. Analisa Perhitungan.....	52
4.4. Validasi Program.....	58

BAB V. KESIMPULAN DAN SARAN

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

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar 2-1	Diagram Satu Garis Sistem Tenaga Listrik.....	6
Gambar 2-2	Sistem Jaringan Distribusi Radial.....	8
Gambar 2-3	Sistem Jaringan Distribusi Radial Dengan <i>Tie</i> dan <i>Switch</i> Pemisah.....	9
Gambar 2-4	Sistem Jaringan Distribusi Dengan Phasa Area	10
Gambar 2-5	Sistem Jaringan Distribusi Dengan Beban Terpusat	11
Gambar 2-6	Segi Tiga Daya	11
Gambar 2-7	Segi Tiga Arus.....	12
Gambar 2-8	Perbaikan Faktor Daya	13
Gambar 2-9	Diagram Fasor dan Sudut Daya Beban Pada Jaringan Distribusi.....	14
Gambar 3-1	Diagram Alir Aliran Daya <i>Newton Raphson</i>	24
Gambar 3-2	<i>Depth First Search</i>	27
Gambar 3-3	Satu <i>Ply Search</i>	29
Gambar 3-4	Satu <i>Ply Search</i>	29
Gambar 3-5	Dua <i>Ply Search</i>	30
Gambar 3-6	<i>Backing Up</i> Dalam Dua <i>Ply Search</i>	31
Gambar 3-7	α - <i>Cutoffs</i>	33
Gambar 3-8	$\alpha\beta$ - <i>Cutoffs</i>	33
Gambar 3-9	Hasil $\alpha\beta$ - <i>Cutoffs</i>	34
Gambar 3-10	Contoh Lain $\alpha\beta$ - <i>Cutoffs</i>	35
Gambar 3-11	Hasil $\alpha\beta$ - <i>Cutoffs</i> Untuk Gambar 3-10.....	36

Gambar 3-12 Keadaan Stabil.....	37
Gambar 3-13 <i>Flowchart</i> Penyelesaian Masalah.....	41
Gambar 4-1 Kotak Dialog <i>About</i>	42
Gambar 4-2 <i>Form</i> Utama.....	43
Gambar 4-3 <i>Form</i> Data Bus.....	44
Gambar 4-4 <i>Form</i> Data Saluran.....	44
Gambar 4-5 <i>Form Result Window</i>	45
Gambar 4-6 Konfigurasi Jaringan Radial Dari <i>Single Line Diagram</i> G.I Rungkut Sebelum Rekonfigurasi.....	48
Gambar 4-7 Hasil Perhitungan dan Perbandingan Tegangan Sebelum dan Tegangan Sesudah Konfigurasi.....	55
Gambar 4-8 Konfigurasi Jaringan Radial 20 kV Setelah Rekonfigurasi.....	57
Gambar 4-9 Data Hasil Aliran Daya Pada Program.....	59

DAFTAR TABEL

Tabel 4-1	Data Saluran Sistem 20 kV G.I Rungkut.....	50
Tabel 4-2	Data Penghantar AAAC	50
Tabel 4-3	Data Pembebanan G.I Rungkut.....	51
Tabel 4-4	Data Impedansi Saluran G.I Rungkut.....	53
Tabel 4-5	Aliran Daya Tiap Saluran G.I Rungkut Sebelum Konfigurasi	54
Tabel 4-6	Aliran Daya Tiap Saluran G.I Rungkut Sesudah Konfigurasi	56
Tabel 4-7	Hasil Kombinasi Optimal <i>Open Switch</i>	57
Tabel 4-8	Data Validasi Jurnal	58
Tabel 4-9	Hasil Kombinasi Optimal <i>Open Switch</i> Kompensasi Pada Program	59
Tabel 4-10	Hasil Perbandingan Program.....	60

BAB I

PENDAHULUAN

1.1. Latar Belakang

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

Dalam sistem distribusi yang efektif, tentunya harus benar-benar memperhatikan dan mengusahakan pelayanan yang baik kepada konsumen dengan cara penyaluran energi listrik yang lebih efektif, mulai dari sistem transmisi tenaga sampai dengan sistem distribusi. Sistem distribusi perlu perhatian khusus sebab kesalahan-kesalahan sistem distribusi ini langsung berpengaruh langsung kepada konsumen. Sesungguhnya 90 % dari terputusnya aliran kepada konsumen berasal dari sistem distribusi.

Di samping itu sistem distribusi pada umumnya lebih rawan, terutama lokasi beban yang bervariasi, sehingga banyak terjadi rugi-rugi daya dari sistem tersebut. Usaha untuk meminimalisasi rugi-rugi daya dari sistem distribusi dengan memformulasikan permasalahan sebagai persoalan rekonfigurasi jaringan distribusi.

Penggunaan metode *Minimax Games Depth First Search* (MGDFS) merupakan penyelesaian yang efisien pada persoalan jaringan konfigurasi secara optimal. Metode ini direduksi dengan mengubah status *sectionalizing* tombol, penentuan pemecahan persamaan arus dan persamaan tegangan dari jaringan. Untuk menopang pencarian aliran daya dengan derajat ketelitian yang bermacam-macam, digunakan metode *Minimax Games Depth First Search* (MGDFS). Penggunaan program komputer sangat efektif untuk mendukung metode ini.

1.2. Rumusan Masalah

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

Oleh karena itu pada skripsi ini akan menganalisa hal tersebut dengan suatu alternatif metode *Minimax Games Depth First Search* (MGDFS) untuk mengoptimalkan lokasi *Open Switch* dan *Closed Switch* pada jaringan distribusi.

Dari permasalahan diatas, maka judul skripsi yang diangkat adalah:

**“OPTIMASI REKONFIGURASI SISTEM DISTRIBUSI RADIAL 20 kV
MENGUNAKAN METODE *MINIMAX GAMES WITH DEPTH FIRST
SEARCH ALGORITHM* (MGDFS) PADA GARDU INDUK RUNGKUT”**

1.3. Tujuan Pembahasan

Adapun tujuan penulisan skripsi ini adalah untuk melihat sejauh mana teknik *MINIMAX GAMES DEPTH FIRST SEARCH* (MGDFS) dapat diterapkan pada jaringan distribusi primer dalam menentukan letak *switch Normally Open* (NO) dan *Normally Closed* (NC) yang optimal, sehingga didapat rugi-rugi daya yang minimum.

1.4. Batasan Masalah

Batasan masalah dari skripsi ini adalah:

1. Jaringan yang akan dianalisa adalah jaringan distribusi primer tipe Radial 20 kV di G.I Rungkut - Surabaya.
2. Hanya membahas rugi-rugi daya pada saluran.
3. Tidak membahas secara detail metode aliran daya yang digunakan yaitu metode *Newton Raphson*.
4. Tidak membahas cara pengontrolan *switching* yang dilakukan.
5. Metode rekonfigurasi yang digunakan adalah MGDFS.
6. Implementasi dari analisa menggunakan perangkat lunak *Borland Delphi 7.0*.

1.5. Metodologi Pembahasan

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

1. Studi literatur, yaitu kajian pustaka untuk mempelajari teori-teori yang terkait melalui literatur yang ada, yang berhubungan dengan permasalahan.
 2. Pengambilan data sebagai acuan dalam melengkapi parameter-parameter yang digunakan untuk menganalisis masalah.
-

3. Menganalisis data yang diperoleh dengan mempergunakan metode *Minimax Games Depth First Search* (MGDFS).

1.6. Sistematika Penulisan

Adapun sistematika penulisan pada skripsi ini adalah sebagai berikut:

BAB I : PENDAHULUAN

Pada bab ini berisi Latar Belakang, Rumusan Masalah, Tujuan Pembahasan, Batasan Masalah, Metodologi Penulisan, dan Sistematika Penulisan.

BAB II : LANDASAN TEORI

Di sini akan diuraikan mengenai Sistem Jaringan Distribusi, Faktor Daya, Perbaikan Tegangan dan Sistem Per Unit Dalam Jaringan.

BAB III : PENERAPAN METODE *MINIMAX GAMES WITH DEPTH FIRST SEARCH ALGORITHM* (MGDFS) PADA REKONFIGURASI SISTEM DISTRIBUSI RADIAL 20 kV

Bab ini membahas perhitungan aliran daya sistem distribusi radial 20 kV dengan metode *Minimax Games Depth First Search* (MGDFS) yang meliputi analisa jaringan dengan menentukan tegangan dan sudut fasa tegangan pada tiap node, arus pada tiap saluran, aliran daya, dan rugi-rugi daya.

BAB IV : SIMULASI DAN ANALISA

Bab ini akan dibahas mengenai Analisa Perhitungan dengan menggunakan metode Algoritma Pencarian Grafik, Profil Tegangan dan Aliran Daya Sebelum dan Sesudah Konfigurasi.

BAB V : KESIMPULAN DAN SARAN

Pada bab ini terdapat intisari dari hasil pembahasan, yang berisi Kesimpulan dan Saran yang dapat dipergunakan sebagai bahan pertimbangan untuk pengembangan dan penulisan selanjutnya.

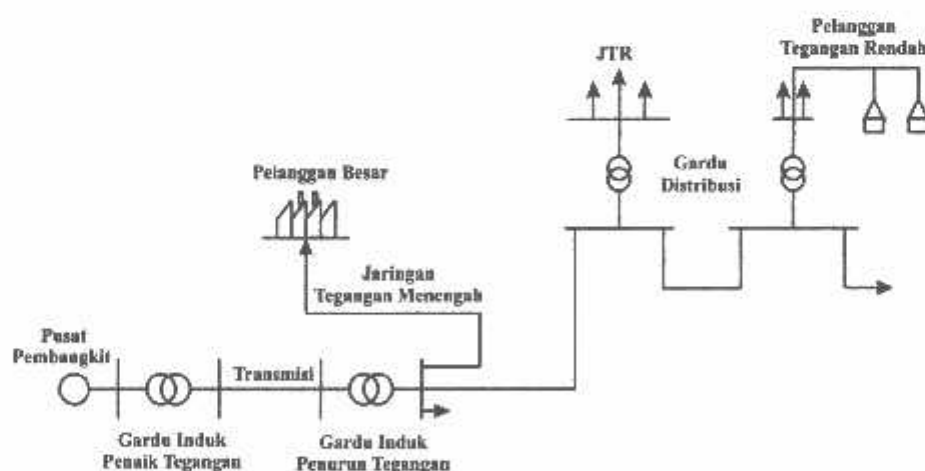
BAB II

LANDASAN TEORI

2.1. Sistem Distribusi Tenaga Listrik^[1]

Di Indonesia, tenaga listrik dibangkitkan di pusat-pusat pembangkit tenaga listrik, seperti PLTA, PLTU, PLTG, PLTGU, PLTP dan PLTD yang kemudian disalurkan melalui saluran transmisi setelah terlebih dahulu dinaikan tegangannya oleh transformator penaik tegangan (*Step Up Transformer*) yang ada pada pusat listrik. Saluran transmisi tegangan tinggi di PLN mempunyai tegangan 150 kV dan 500 kV, selanjutnya tegangan tinggi tersebut diturunkan menjadi tegangan menengah sebesar 20 kV di Gardu Induk, dengan menggunakan transformator penurun tegangan (*Step Down Transformer*), setelah melalui jaringan distribusi, tegangan diturunkan lagi pada gardu-gardu distribusi menjadi tegangan rendah sebesar 380/220 V untuk selanjutnya disalurkan ke pelanggan-pelanggan PLN.

Sebagai gambaran, diagram satu garis sistem tenaga listrik dapat diperjelas seperti pada gambar 2-1 di bawah ini:



Gambar 2-1
Diagram Satu Garis Sistem Penyaluran Energi Listrik Ke Pelanggan^[1]

Jaringan setelah keluar dari Gardu Induk biasa disebut jaringan distribusi.

Jarigan distribusi dapat diklasifikasikan menjadi dua bagian sistem, yaitu:

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

Pengklasifikasian sistem distribusi tenaga listrik menjadi dua ini berdasarkan tingkat tegangan distribusinya. Sistem jaringan yang digunakan untuk menyalurkan dan mendistribusikan tenaga listrik tersebut dapat menggunakan sistem satu fasa dengan dua kawat maupun sistem tiga fasa dengan empat kawat.

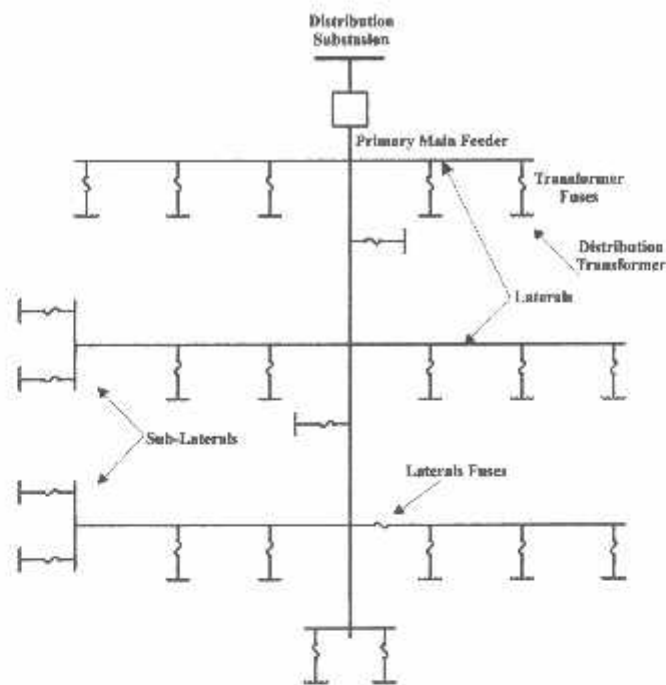
2.2. Struktur Jaringan Distribusi Tenaga Listrik ^[1]

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*.

2.3. Sistem Jaringan Distribusi Radial ^[1]

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



Gambar 2-2
Sistem Jaringan Distribusi Radial ⁽¹⁾

Pelayanan tenaga listrik untuk suatu daerah beban tertentu dilaksanakan dengan memasang transformator disebarkan 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 sebagian atau bahkan keseluruhan beban sistem.

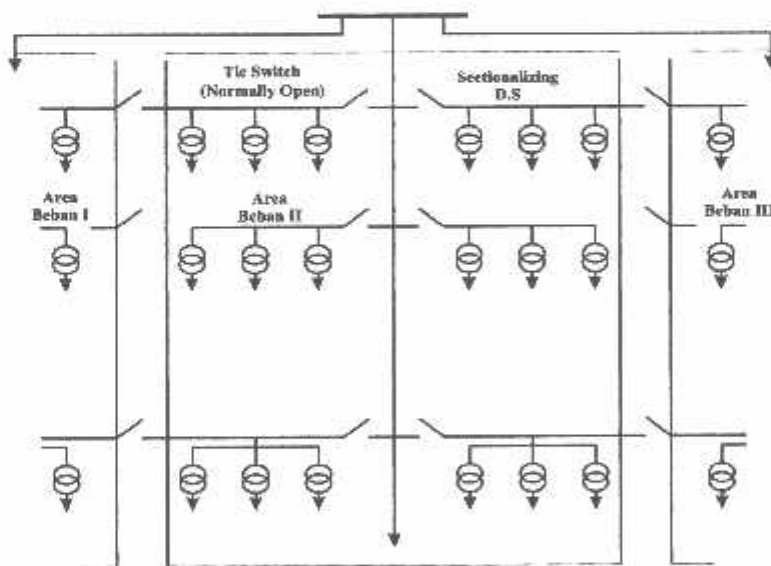
2.3.1. Sistem Radial Pohon

Sistem radial jaringan pohon ini merupakan bentuk yang paling dasar dari sistem jaringan radial. Saluran utama (*main feeder*) ditarik dari suatu gardu induk sesuai dengan kebutuhan kemudian dicabangkan melalui saluran cabang (*lateral*

feeder), selanjutnya dicabangkan lagi melalui saluran anak cabang (*sub-lateral feeder*). Ukuran dari masing-masing saluran tergantung dari kerapatan arus yang ada pada sistem. Dari gambar 2-2, *main feeder* merupakan saluran yang dialiri arus terbesar, selanjutnya arus mengecil pada tiap cabang tergantung dari besarnya beban.

2.3.2. Sistem Radial dengan *Tie* dan *Switch* Pemisah

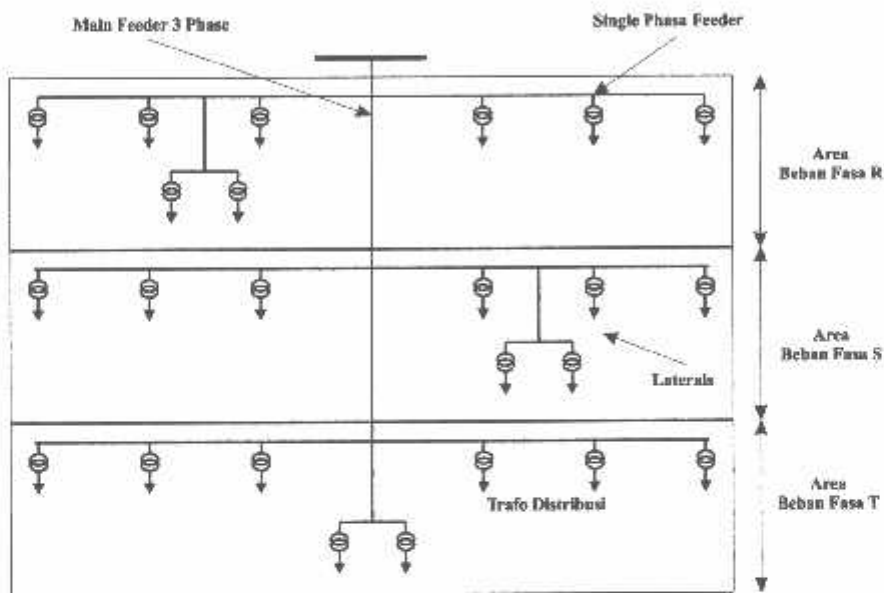
Sistem ini merupakan pengembangan dari sistem radial pohon, untuk meningkatkan keandalan sistem saat terjadinya gangguan maka *feeder* yang terganggu akan dilokalisir sedangkan area yang semula dilayani oleh *feeder* tersebut pelayanannya dialihkan pada *feeder* yang sehat atau yang tidak terganggu.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



Gambar 2-3
Sistem Jaringan Distribusi Radial
Dengan *Tie* dan *Switch* Pemisah ^[1]

2.3.3. Sistem Radial dengan Pembagian Fasa Area

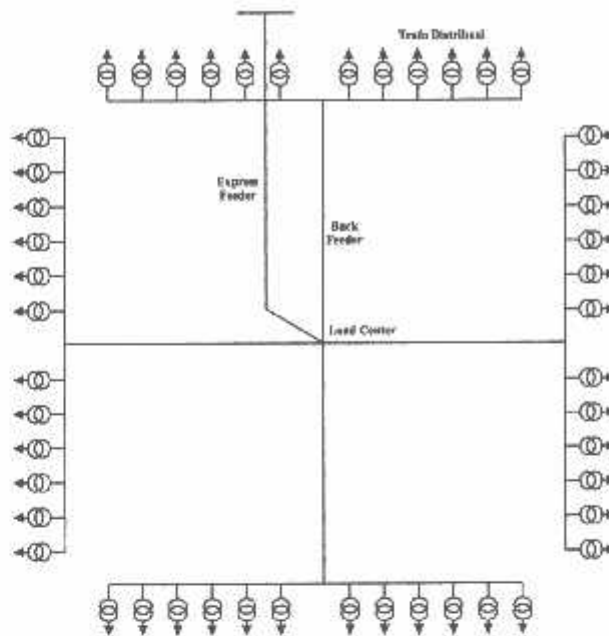
Pada bentuk ini masing-masing fasa dari jaringan bertugas untuk melayani daerah beban yang berlainan. Bentuk ini akan dapat menimbulkan kondisi sistem tiga fasa yang tidak seimbang (simetris), bila digunakan pada daerah beban yang baru dan belum mantap pembagian bebannya. Contoh dari sistem jaringan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



Gambar 2-4
Sistem Jaringan Distribusi Radial Dengan Fasa Area ^[1]

2.3.4. Sistem Radial Dengan Beban Terpusat

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



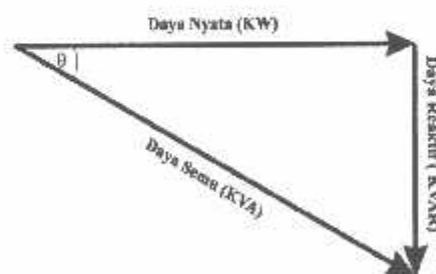
Gambar 2-5
Sistem Jaringan Distribusi Radial Dengan Beban Terpusat ^[1]

2.4. Faktor Daya ^[1]

Faktor daya pada dasarnya didefinisikan sebagai perbandingan antara daya nyata dan daya semu, dan dinyatakan oleh persamaan:

$$\text{Faktor Daya} = \frac{\text{Daya Nyata (kW)}}{\text{Daya Semu (kVA)}} \dots\dots\dots (2.1)$$

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



Gambar 2-6
Segi Tiga Daya ^[1]

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

Dimana:

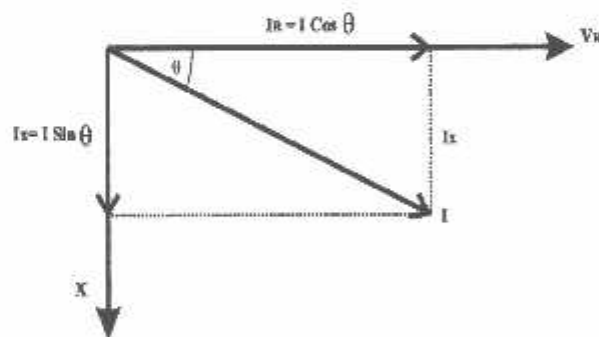
P = Daya Nyata (kW)

S = Daya Semu (kVA)

Q = Daya Reaktif (kVAR)

θ = Sudut Phasa

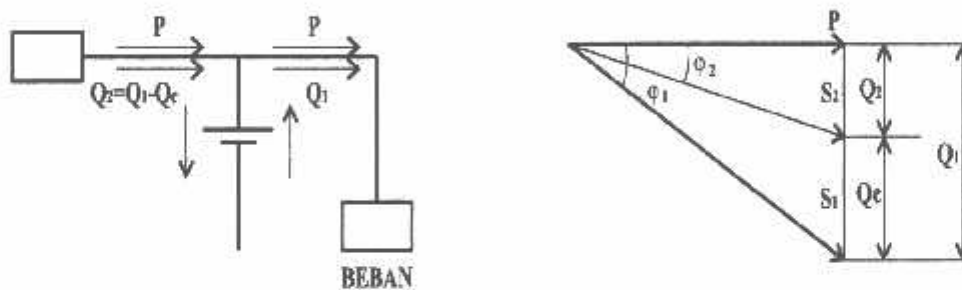
Faktor daya dapat pula dilihat hubungannya dengan arus nyata dan arus total. Arus nyata (I_R) adalah arus yang mengalir pada beban yang diubah menjadi tenaga. Sedangkan arus total (I) adalah arus yang mengalir pada jaringan dan merupakan jumlah vektor antara arus magnetisasi dan arus nyata. Arus magnetisasi (I_x) merupakan arus yang mengalir pada beban dan menimbulkan medan magnet. Arus yang terbaca dalam ampere meter adalah arus total (I). Hubungan ketiga arus tersebut dapat dilihat pada gambar 2-7 sebagai berikut:



Gambar 2-7
Segi Tiga Arus ^[1]

2.5. Perbaikan Faktor Daya dan Kapasitas Sistem ^[1]

Manfaat terbesar yang diperoleh dari perbaikan faktor daya berasal dari pengurangan daya reaktif dalam sistem. Hal ini menghasilkan pengurangan biaya pemakaian daya yang lebih rendah, kenaikan kapasitas sistem, perbaikan tegangan dan pengurangan rugi daya dalam sistem. Satu-satunya cara untuk memperbaiki faktor daya adalah mengurangi daya reaktif pada jaringan. Jika komponen daya reaktif dapat dikurangi, maka total arus akan berkurang, sedang komponen daya aktif tidak berubah, maka faktor daya akan lebih besar sebagai akibat berkurangnya daya reaktif. Faktor daya akan mencapai 100 % jika komponen daya reaktif sama dengan nol (0).



Gambar 2-8
Perbaikan Faktor Daya ^[1]

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

$$\cos \theta_1 = \frac{P}{S_1} \dots \dots \dots (2.2)$$

Persentase pengurangan rugi daya jaringan dapat dihitung menggunakan persamaan berikut:

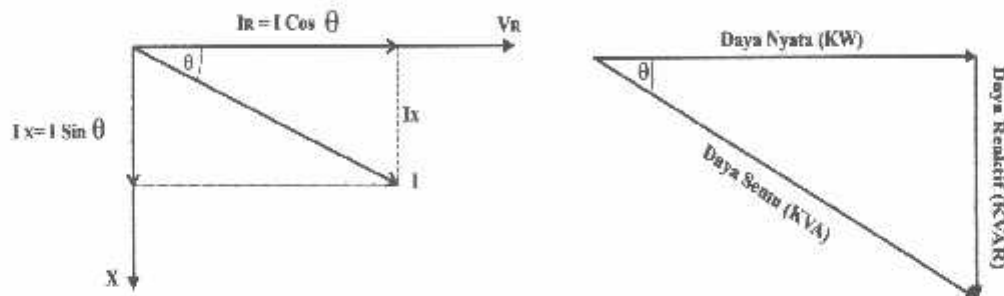
$$\% \text{ Rugi Daya} = 100 \left(\frac{\text{Faktor daya mula} - \text{mula}(\cos \theta_1)}{\text{Faktor daya baru}(\cos \theta_2)} \right) \dots \dots \dots (2.3)$$

% Pengurangan Rugi Daya

$$= 100 \left(1 - \left(\frac{\text{Faktor daya mula - mula} (\cos \theta_1)}{\text{Faktor daya baru} (\cos \theta_2)} \right)^2 \right) \dots \dots \dots (2.4)$$

2.6. Perhitungan Pengaruh Perbaikan Faktor Daya ^[2]

Diagram fasor dari dua komponen arus nyata, arus aktif dan arus reaktif dapat dilihat pada gambar 2-9 berikut:



Gambar 2-9
Diagram Fasor dan Sudut Daya Beban pada
Jaringan Distribusi ^[2]

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

$$\begin{aligned} I = \text{Arus Semu} &= \sqrt{(\text{arus aktif})^2 + (\text{arus reaktif})^2} \\ &= \sqrt{(I \cos \theta)^2 + (I \sin \theta)^2} \end{aligned}$$

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

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

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

$$\text{Faktor Daya} = \frac{\text{Daya aktif}}{\text{Daya semu}} = \frac{KW}{KVA}$$

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

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

$$kW = kVA \cos \theta$$

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

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

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

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

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

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

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

Dengan θ_1 = sudut dari faktor daya mula-mula

θ_2 = sudut dari faktor daya yang telah diperbaiki

$$\begin{aligned} \text{Daya reaktif (kVAR)} &= \text{Daya aktif} \times (\tan \theta_1 - \theta_2) \\ &= (\text{kW}) \times (\tan \theta_1 - \theta_2) \end{aligned}$$

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

$$\text{Daya reaktif (kVAR)} = \text{Daya aktif} \times \Delta \tan = (\text{kW}) \times \Delta \tan.$$

2.7. Sistem Per-Unit ^[3]

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

$$\text{Besaran per-unit} = \frac{\text{Besaran sebenarnya}}{\text{Besaran dasar dengan ukuran yang sama}} \dots\dots\dots(2.5)$$

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

➤ Untuk data 1 fasa:

Arus dasar.

$$I_d = \frac{\text{kVA dasar 1 fasa}}{\text{kVA dasar } L-N} \dots\dots\dots(2.6)$$

Impedansi dasar.

$$\begin{aligned} Z_d &= \frac{(\text{kV dasar } L-N)^2 \times 1000}{\text{kVA dasar 1 fasa}} \\ &= \frac{(\text{kV dasar } L-N)^2}{\text{MVA dasar 1 fasa}} \dots\dots\dots(2.7) \end{aligned}$$

Dalam persamaan di atas nilai-nilai besaran diberikan untuk rangkaian satu fasa. Jadi tegangannya adalah tegangan antara fasa dengan tanah dan daya setiap fasa. Setelah besaran-besaran dasar ditentukan maka besaran-besaran itu dinormalisasikan terhadap besaran dasar. Dengan demikian impedansi per-satuan didefinisikan sebagai berikut:

$$Z = \frac{\text{Im pedansi sebenarnya } Z (\Omega)}{\text{Im pedansi dasar } Z_0} \dots\dots\dots(2.8)$$

BAB III

PENERAPAN METODE *MINIMAX GAMES WITH DEPTII FIRST*

SEARCH ALGORITHM (MGDFS) PADA REKONFIGURASI

SISTEM DISTRIBUSI RADIAL 20 kV

3.1. Analisa Aliran Daya Jaringan Radial ^[3]

Proses analisa aliran daya adalah langkah awal yang dilakukan untuk mengetahui kondisi suatu sistem distribusi radial.

3.1.1. Tujuan

Tujuan analisa aliran daya pada skripsi ini adalah:

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

3.1.2. Metode *Newton Raphson*

Secara matematis persamaan aliran daya *Newton Raphson* dapat diselesaikan dengan menggunakan koordinat rektangular atau koordinat polar. Dalam pembahasan skripsi ini menggunakan bentuk polar. Hubungan antara arus simpul I_p dengan tegangan simpul V_q pada suatu jaringan dengan n simpul dapat dituliskan:

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

Injeksi daya pada simpul p adalah:

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

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

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

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

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

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

Maka persamaan (3.3) dapat ditulis:

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

Dengan memisahkan bagian riil dan bagian imajiner maka diperoleh:

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

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

Dengan menggunakan persamaan (3.5) dan persamaan (3.6) untuk n buah simpul dalam sistem didapat 2n persamaan, sedangkan disetiap simpul ada 4 variabel. Untuk memecahkan persoalan ini, 2n variabel perlu ditentukan terlebih dahulu, sehingga 2n variabel yang lain dapat dicari dengan menggunakan 2n persamaan yang ada. Penentuan 2n variabel ini dilakukan dengan menentukan beberapa macam simpul dalam sistem, yaitu:

a. Simpul PQ (Simpul Beban).

Pada simpul ini jumlah netto daya nyata dan daya reaktif P_p dan Q_p diketahui, sedangkan yang dicari adalah $|V_p|$ dan sudutnya δ . Untuk itu, besarnya beban P_{Bp} ditentukan berdasarkan perkiraan beban, sedangkan daya yang dibangkitkan P_{Gp} dan Q_{Gp} ditentukan besarnya. Selanjutnya $P_p = P_{Bp} - P_{Gp}$ dan $Q_p = Q_{Bp} - Q_{Gp}$.

Simpul beban (P Q) yang murni mempunyai nilai $P_{Gp} = 0$ dan $Q_{Gp} = 0$.

b. Simpul PV atau simpul Generator, atau simpul yang dayanya dapat diatur.

Pada simpul ini nilai P dan $|V|$ diketahui, sedangkan yang dicari adalah nilai Q dan δ .

c. Simpul Referensi (*Slack Bus*).

Bedanya dengan kedua macam simpul yang terdahulu adalah bahwa pada simpul ini, daya nyata maupun daya reaktif tidak ditentukan. Di lain pihak, yang ditentukan adalah besarnya tegangan $|V_i|$ dan sudutnya δ_p yang biasanya ditentukan $= 0$, sehingga merupakan sudut referensi bagi ketegangan dan sistem.

Pada umumnya dalam analisis aliran daya hanya ada sebuah simpul referensi.

Simpul referensi ini diperlukan karena nilai P_p dan Q_p untuk setiap simpul tidak ditentukan terlebih dahulu.

Nilai P dan Q total dai sistem baru dapat dihitung setelah aliran tidak dapat ditentukan terlebih dahulu.

Nilai P dan Q total dari sistem baru dapat dihitung setelah aliran daya antar simpul dihitung, kemudian rugi-rugi pada saluran dihitung. Rugi-rugi pada saluran ini mempunyai nilai daya nyata P_r dan daya reaktif Q_r , hal ini selanjutnya harus diperhitungkan dengan daya nyata dan daya reaktif yang dibangkitkan pada simpul referensi dengan persamaan sebagai berikut:

$$P_1 = \sum P_{Bp} + P_r - \sum P_{Gp} \quad (p \neq 1) \dots\dots\dots(3.7)$$

$$Q_1 = \sum P_{Bp} + P_r - \sum P_{Gp} \quad (p \neq 1) \dots\dots\dots(3.8)$$

Indeks 1 ($p = 1$) adalah indeks bagi simpul referensi.

Berdasarkan uraian diatas untuk sistem yang terdiri dari n buah simpul, 2n variabel telah diketahui, sedangkan 2n variabel yang lain harus dicari. Untuk mencari 2n variabel ini dipakai persamaan (3.5) dan persamaan (3.6) untuk setiap simpul sehingga didapat 2n persamaan yang merupakan syarat untuk mencari 2n variabel tersebut yang merupakan syarat untuk mencari 2n variabel tersebut.

Dalam metode *Newton Rhapson*, variabel-variabel yang harus dicari, dimisalkan dulu nilainya, jadi untuk setiap simpul ada dua variabel yang diketahui dan dua variabel yang dimisalkan, kecuali untuk simpul referensi yang akan dihitung terakhir. Kemudian digunakan persamaan (3.5) dan persamaan (3.6) untuk menghitung nilai P dan nilai Q pada setiap simpul.

Pada setiap simpul PQ, nilai P dan nilai Q diketahui dan nilai yang diketahui inilah yang dibandingkan dengan nilai hasil perhitungan di atas. Apabila selisih antara nilai yang diketahui dan hasil perhitungan di atas lebih kecil daripada suatu nilai yang dikeendaki, maka nilai variabel yang dimisalkan tersebut diatas dapat dianggap benar. Apabila selisih tersebut lebih besar dari nilai yang

dikhendaki, maka harus dilakukan proses iterasi sampai selisih tersebut mencapai nilai yang dikendaki. Untuk simpul PV yang tidak dibandingkan hanya selisih daya aktif ΔP saja, karena daya reaktif Q yang diketahui, tidak ditentukan, akan merupakan hasil perhitungan. Untuk simpul referensi (*Slack Bus*) dihitung terakhir seperti telah dijelaskan. sedangkan ϵ adalah suatu angka yang ditentukan berdasarkan ketelitian yang diinginkan.

3.1.3. Algoritma Perhitungan Aliran Daya dengan Metode *Newton Rhapson*

Dalam perhitungan aliran daya dengan metode *Newton Rhapson*, langkah-langkahnya adalah sebagai berikut:

A. Kebebasan-kebebasan yang diketahui:

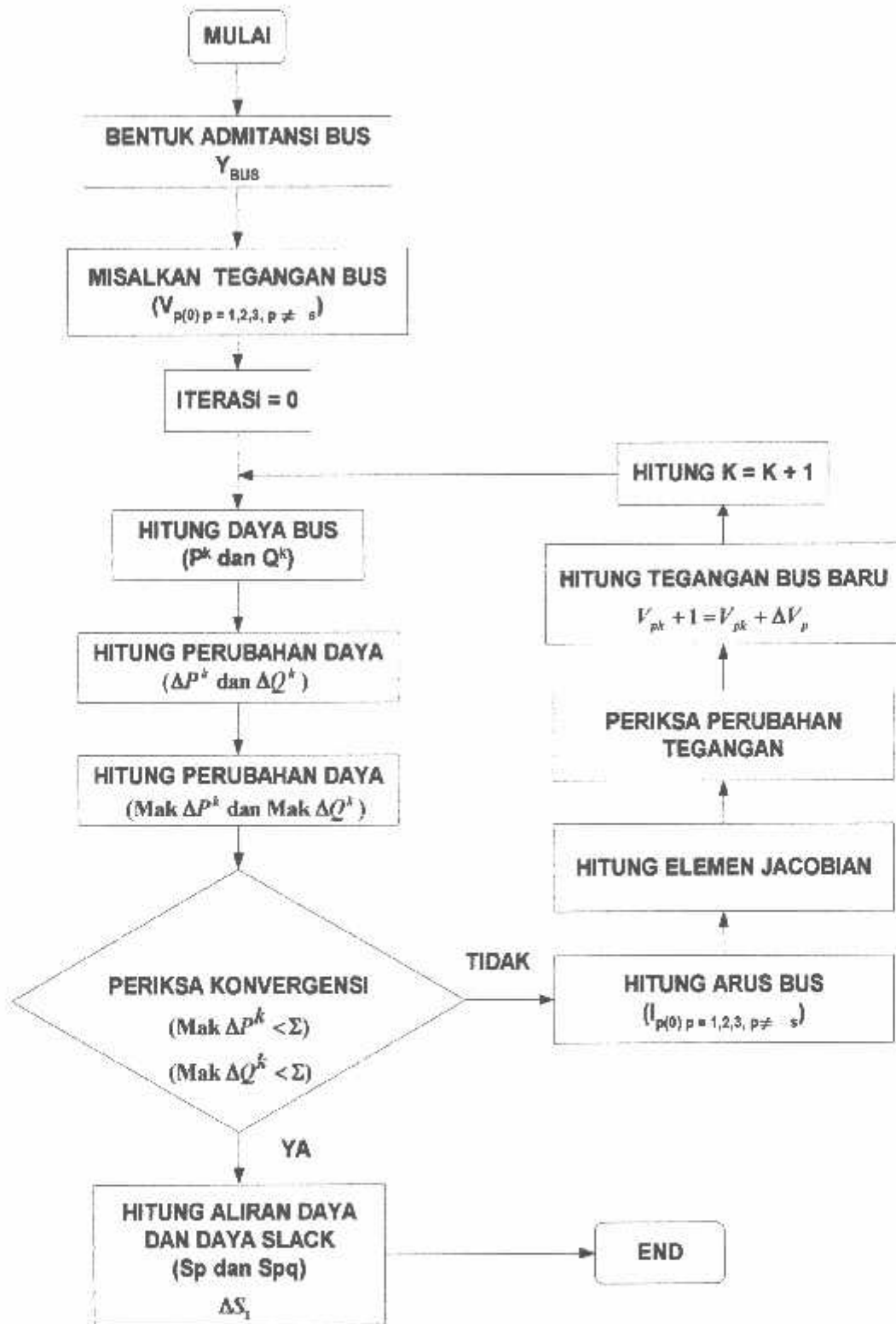
1. Daya nyata P dan daya reaktif Q pada simpul PQ diketahui.
2. Daya nyata P dan besarnya tegangan $|V|$ pada simpul PV diketahui.
3. Besar tegangan $|V|$ dan sudut fasa teangan δ dari simpul referensi (*Slack Bus*) diketahui.

B. Kemudian dicoba nilai tertentu bagi besaran sebagai berikut:

1. Besarnya tegangan $|V|$ beserta sudut fasanya δ pada semua simpul PQ.
2. Besarnya sudut fasa tegangan pada semua simpul PV.

C. Berdasarkan nilai yang dicoba tersebut pada butir B, dilakukan perhitungan dengan menggunakan persamaan (3.5) dan persamaan (3.6) untuk mendapatkan:

1. Nilai P dan Q yang dihitung untuk simpul-simpul PQ.
 2. Nilai P yang dihitung untuk simpul PV.
- D. Nilai P dan Q yang diketahui pada A dikurangi dengan nilai P dan Q yang didapat dari perhitungan pada C disebut nilai residu dari P dan Q. Nilai residu ini harus mendekati nol, atau $< \epsilon$ (nilai yang dikehendaki berdasarkan suatu ketelitian perhitungan yang di inginkan). Apabila nilai residu P dan Q ini belum $< \epsilon$, maka harus dilakukan iterasi.
- E. Proses iterasi dilakukan dengan mengkoreksi nilai yang dicoba bagi $|V|$ dan δ seperti yang disebutkan dalam butir B.
-

3.1.4. Diagram Alir Aliran Daya *Newton Raphson*

Gambar 3-1
Diagram Alir Aliran Daya *Newton Raphson* ^[3]

3.2. Metode *Minimax Games Depth First Search* (MGDFS) ^[4]

Games adalah fasilitas yang sangat menarik dalam komputer. Ide *games* pertama kali dimunculkan oleh *Claude Shannon* (1950) yang menulis paper tentang mekanisme pembuatan program permainan catur. Beberapa tahun kemudian, *Alan Turing* mendeskripsikan program permainan catur namun ia sendiri belum pernah membuat rancangan program. Baru pada awal tahun 1960-an *Arthur Samuel* mencoba untuk membuat program catur tersebut.

Ada beberapa alasan mengapa *games* merupakan domain yang baik untuk di explore, yaitu:

1. Sangat mudah untuk menentukan ukuran kesuksesan dan kegagalanya (menang atau kalah).
2. Tidak membutuhkan terlalu banyak pengetahuan. Permainan dapat diselesaikan dengan melakukan pencarian dari arah *start* sampai posisi menang.
3. Ruang keadaanya mudah direpresentasikan.
4. Operator-operator yang digunakan tidak terlalu banyak.
5. Sebagian besar *game* dapat dimodelkan dengan mudah.
6. Sangat mungkin untuk dibandingkan dengan kemampuan manusia.

Alasan pertama memang masih bisa diterima sampai sekarang. Namun alasan kedua ternyata tidak cocok untuk *games* yang besar (kompleks). Sebagai contoh pada permainan catur:

1. Rata-rata setiap node bercabang sebanyak 35.
 2. Tiap-tiap pemain rata-rata bergerak 50 kali.
 3. Sehingga total dalam satu pohon, akan terdapat posisi sebanyak 35^{100} .
-

3.2.1. Metode Pencarian dan Evaluasi

Ruang keadaan dalam *games* dapat direpresentasikan dengan menggunakan pohon pelacakan. Tiap-tiap node pada pohon tersebut berhubungan dengan keadaan yang mungkin dalam permainan tersebut. Setiap gerakan akan membawa perubahan dari keadaan sekarang (*current state*) ke keadaan selanjutnya (*child state*). Permasalahan yang sangat rumit yang dihadapi adalah menentukan *child state* mana yang terbaik.

Untuk mengefektifkan proses pencarian, maka dapat dilakukan dengan 2 cara, yaitu:

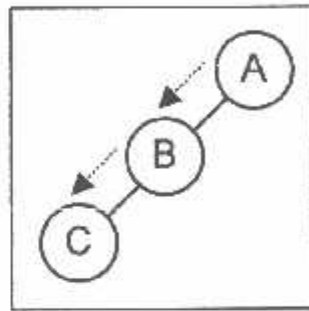
1. Membentuk suatu prosedur sedemikian hingga hanya gerakan-gerakan yang baik saja yang dibangkitkan.
2. Membentuk suatu prosedur pengujian sedemikian hingga gerakan (*path*) yang terbaik yang akan dieksplorasi pertama kali.

Proses evaluasi juga memegang peranan penting. Evaluasi dapat dilakukan secara rekursif atau secara statis. Pada evaluasi rekursif, kita ingin meranking gerakan-gerakan yang mungkin (yang ditunjukkan dengan suatu nilai untuk setiap keadaan). Untuk tiap-tiap gerakan yang mungkin, kita ingin tahu seberapa baik suatu keadaan untuk diikuti (menang atau kalah). Kita dapat mengevaluasi tiap-tiap gerakan dengan mengambil nilai terbaik untuk diikuti. Pada evaluasi fungsi statis, kita mengestimasi seberapa baik suatu node dari pandangan kita. Sebagai contoh:

- Tic-tac-toe: jumlah baris, kolom atau diagonal yang berisi 2 tanda milik kita.
 - Catur: memberikan bobot pada tiap-tiap bidak (raja = 1000, ratu = 10, menteri, kuda, benteng = 5, dst).
-

3.2.2. Pencarian Mendalam Pertama (*Depth First Search*)

Pada *Depth First Search*, proses pencarian akan dilakukan pada semua anaknya sebelum dilakukan pencarian ke node-node yang selevel. Pencarian dimulai dari node akar ke level yang lebih tinggi. Proses ini diulangi terus hingga ditemukannya solusi (gambar 3-2).



Gambar 3-2
Depth First Search ^[4]

➤ Algoritma.

1. Jika keadaan awal merupakan tujuan, keluar (sukses).
2. Jika tidak demikian, kerjakan langkah-langkah berikut ini sampai tercapai keadaan sukses atau gagal:
 - a. Bangkitkan *successor* E dari keadaan awal. Jika tidak ada *successor*, maka akan terjadi kegagalan.
 - b. Panggil *Depth First Search* dengan E sebagai keadaan awal.
 - c. Jika sukses berikan tanda sukses. Namun jika tidak, ulangi langkah ke-2.

➤ Keuntungan.

1. Membutuhkan memori yang relatif kecil, karena hanya node-node pada lintasan yang aktif saja yang disimpan.
2. Secara kebetulan, metode *Depth First Search* akan menemukan solusi tanpa harus menguji lebih banyak lagi dalam ruang keadaan.

➤ Kelemahan.

1. Memungkinkan tidak ditemukannya tujuan yang diharapkan.
2. Hanya akan mendapatkan 1 solusi pada setiap pencarian.

➤ Analisis Ruang dan Waktu.

1. Analisis Ruang.

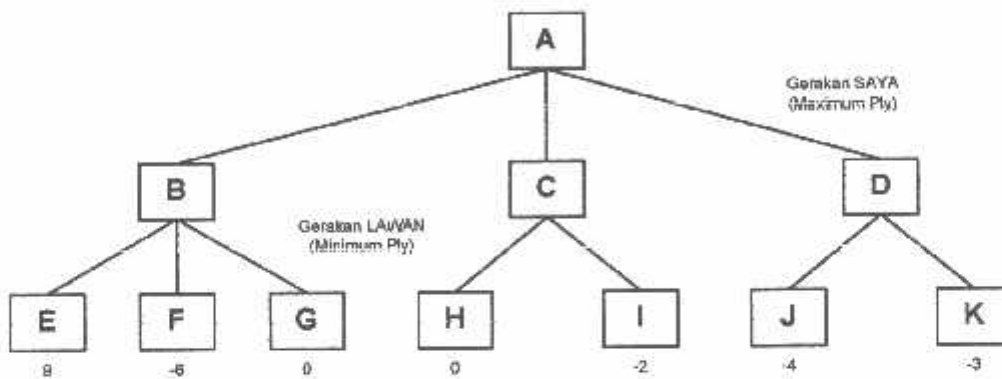
- Setelah berjalan 1 langkah, *stack* akan berisi b node.
- Setelah berjalan 2 langkah, *stack* akan berisi $(b-1) + b$ node.
- Setelah berjalan 3 langkah, *stack* akan berisi $(b-1) + (b-1) + b$ node.
- Setelah berjalan d langkah, *stack* akan berisi $(b-1) \cdot d + 1$ node, mencapai maksimum.

2. Analisis Waktu.

- Pada kasus terbaik, *Depth First Search* akan mencapai tujuan pada kedalaman d pertama, sehingga dibutuhkan pencarian sebanyak $d + 1$ node.
 - Pada kasus terburuk, *Depth First Search* akan mencapai tujuan pada kedalaman d pada node terakhir, sehingga dibutuhkan pencarian sebanyak: $1 + b + b^2 + b^3 + \dots + b^d = (b^{d+1} - 1) / (b - 1)$.
-

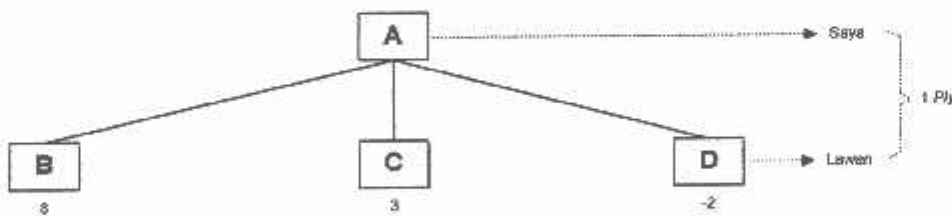
3.2.3. Minimax

Salah satu teknik *games* yang terkenal adalah *minimax*. *Minimax* menggunakan *Depth First Search* dengan kedalaman terbatas. Fungsi evaluasi yang digunakan adalah fungsi evaluasi statis, dengan mengasumsikan bahwa lawan akan membuat langkah terbaik yang mungkin. Gambar 3-3 menunjukkan pohon pelacakan *minimax*.

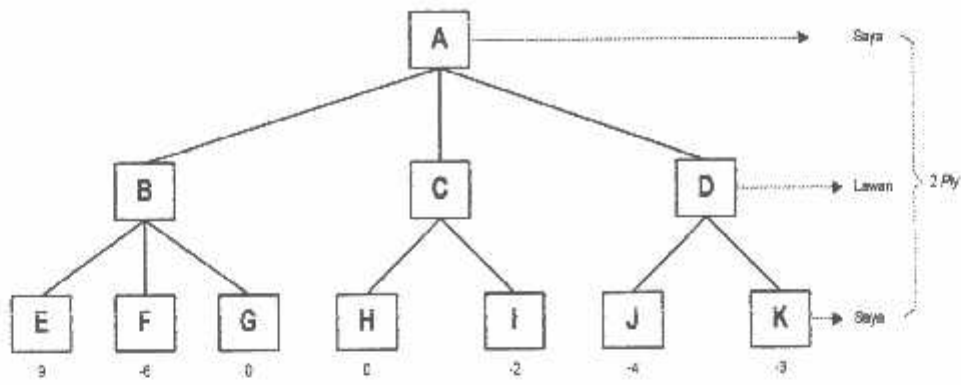


Gambar 3-3
Satu Ply Search^[4]

Pada *minimax* kita mengenal adanya istilah *ply* yaitu gerakan “saya” dan “lawan”. Gambar 3-3 menunjukkan *games* dengan 1 *ply search*, sedangkan gambar 3-5 menunjukkan *games* dengan 2 *ply search*.



Gambar 3-4
Satu Ply Search^[4]



Gambar 3-5
Dua Ply Search ^[4]

Algoritma *Minimax*:

Minimax (StatusSaya, Kedalaman, Pemain)

IF (Kedalaman == Max)

RETURN static (StatusSaya, Pemain)

Bangkitkan successor S [1..n]

IF (Pemain == Saya)

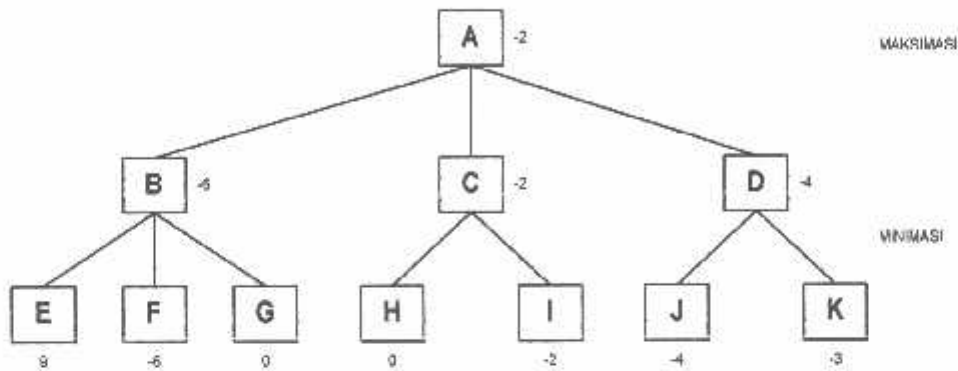
RETURN max of Minimax S[i], kedalaman + 1, Lawan)

ELSE

RETURN min of Minimax (S[i], kedalaman + 1, Saya)

3.2.4. Permainan Dalam *Minimax*

Pada *minimax* ada 2 prosedur yang dijalankan, yaitu: Maksimisasi (dilakukan oleh “saya”), dan Minimisasi (dilakukan oleh “lawan”). Dalam hal ini diperlukan suatu fungsi evaluasi statis yang menyatakan nilai yang mungkin didapat oleh pemain (misalkan nilai tersebut antara -10 sampai +10).



Gambar 3-6
Backing Up Nilai Dalam 2 Ply Search ^[4]

Gambar 3-6 menunjukkan *backing up* nilai yang diperoleh dalam 2 *ply search*. Pada level ke-1 “saya” melakukan maksimisasi, sedangkan pada level ke-2 “lawan” melakukan minimisasi. Lawan memilih nilai yang paling rendah yaitu -6, -2, dan -4. Pada saat “saya” melakukan maksimisasi, saya akan memilih -2 (nilai terbesar di antara -6, -2, dan -4). Sehingga nilai inilah yang nantinya akan didapat.

3.2.5. $\alpha\beta$ -Cutoff

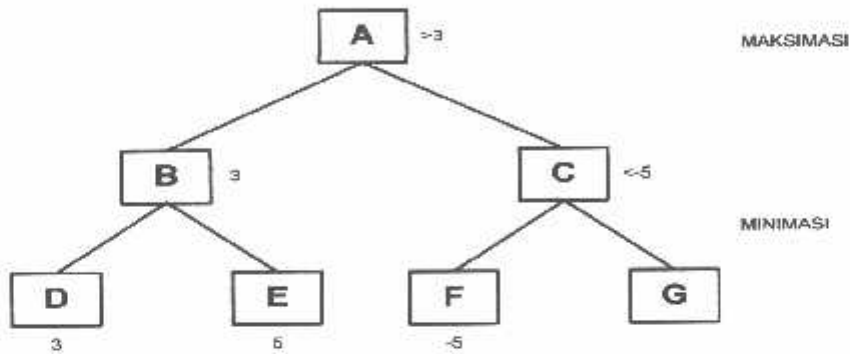
Pada *minimax*, terdapat istilah *pruning*. Kita dapat menggunakan teknik *branch-and-bound* untuk mengurangi jumlah keadaan yang harus diuji untuk menentukan nilai dari suatu pohon pelacakan. Kita dapat menyimpan nilai batas

bawah pada node yang melakukan maksimasi, dan kita tidak perlu menghiraukan cabang-cabang yang tidak akan memperbaiki batas tersebut (lebih tinggi). Demikian pula, kita dapat menyimpan batas atas dari node yang melakukan minimasi, dan kita juga tidak perlu menghiraukan cabang-cabang yang tidak akan memperbaiki batas tersebut (lebih rendah).

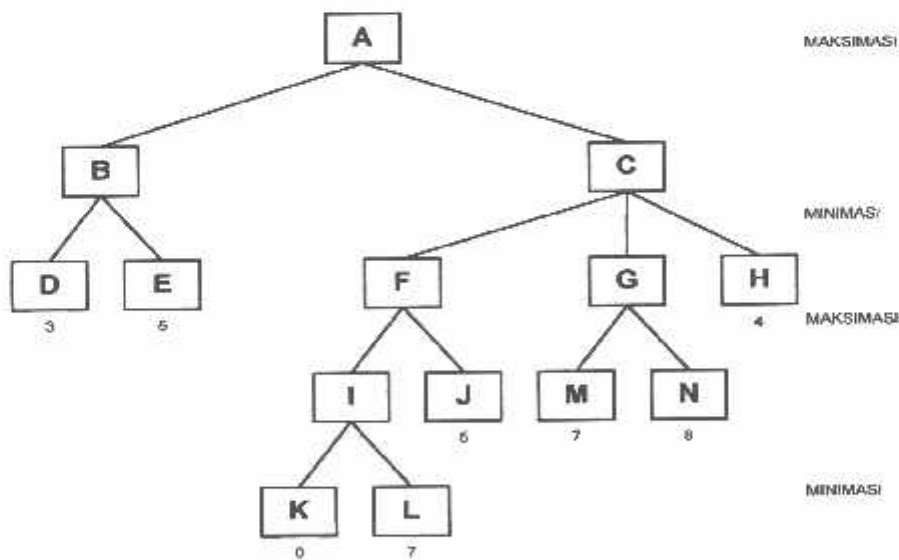
Salah satu metode yang digunakan untuk memodifikasi algoritma *minimax* adalah dengan $\alpha\beta$ -pruning. Variabel alpha (α) digunakan sebagai batas bawah node yang akan melakukan maksimisasi. Sedangkan variabel beta (β) digunakan sebagai batas atas bagi node yang akan melakukan minimisasi. Pada node-node yang melakukan minimasi, evaluasi akan dihentikan jika sudah didapat node anak yang memiliki nilai lebih kecil dibanding dengan batas bawah (α). Sedangkan pada node-node yang melakukan maksimasi, evaluasi akan dihentikan jika sudah didapat node anak yang memiliki nilai lebih besar dibanding dengan batas atas (β). Pada akar pohon pencarian, nilai α diset sama dengan $-\infty$ sedangkan nilai β diset sama dengan $+\infty$.

Node-node yang melakukan maksimasi akan memperbaiki nilai α dari nilai anak-anaknya, sedangkan node-node yang melakukan minimasi akan memperbaiki nilai β dari nilai anak-anaknya. Jika $\alpha > \beta$, maka evaluasi dihentikan. Tiap-tiap node yang melalui nilai α dan β akan dievaluasi anak-anaknya. Node-node anak akan memperbaiki nilai α dan β . Node-node yang melakukan minimasi akan memberikan nilai β sesuai dengan nilai node-nya. Sedangkan node-node yang melakukan maksimasi akan memberikan nilai α sesuai dengan nilai nodenya. Sebagai contoh dapat dilihat pada gambar 3-7.

Setelah menguji node F, maka kita tahu bahwa dijaman "lawan" akan mendapatkan nilai -5 atau yang lebih rendah (sebab C akan melakukan minimisasi). Namun kita juga tahu bahwa "saya" akan mendapatkan nilai 3 atau yang lebih besar (sebab A melakukan maksimisasi). Semua gerakan yang bernilai < 3 akan sangat tidak berguna, sehingga langkah yang dipilih adalah melalui B. Dengan demikian kita tidak perlu mengeksplore C lebih lanjut. Dengan kata lain eksplore ke G tidak diperlukan.

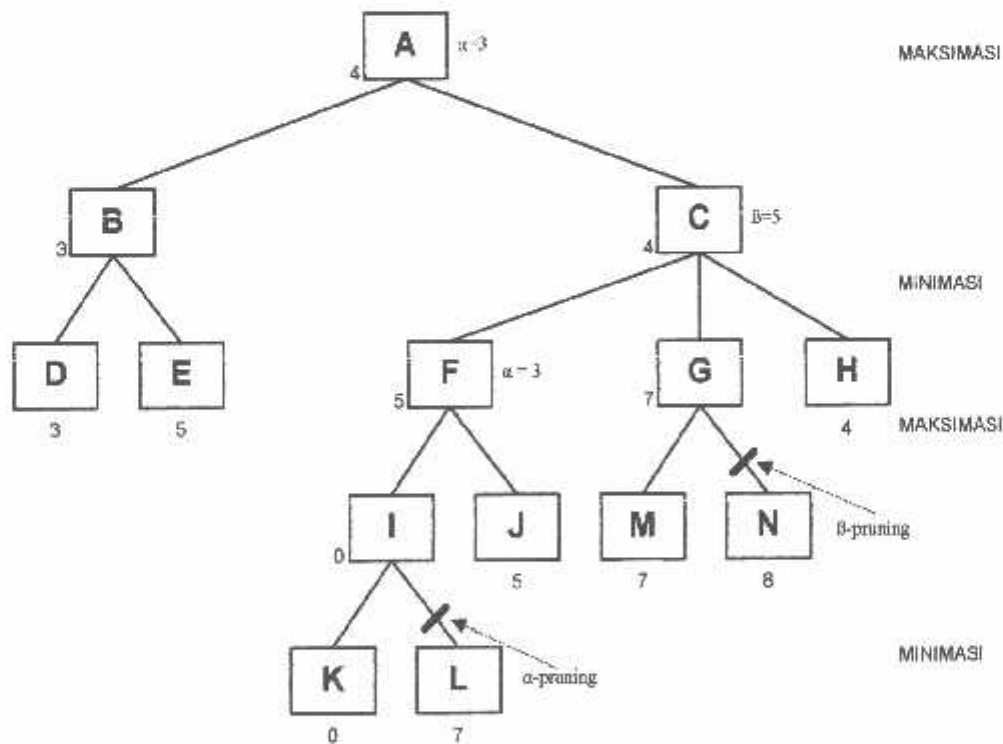


Gambar 3-7
 α -Cutoffs^[4]



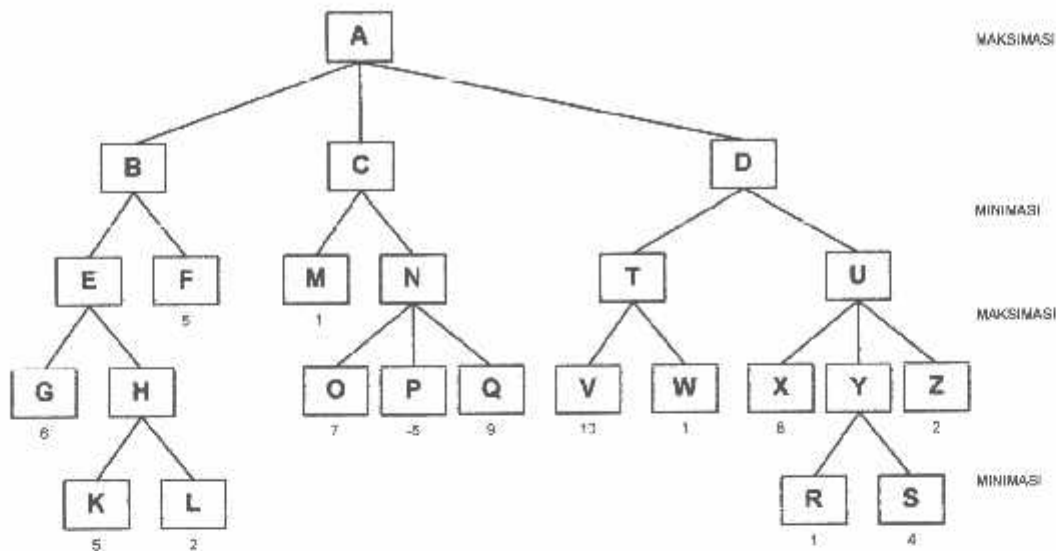
Gambar 3-8
 β -Cutoffs^[4]

Penggunaan *alpha-beta cutoffs* sebagai contoh dapat dilihat pada gambar 3-8. Pada *tree* ini, andaikan kita bergerak pada subtree dengan akar B, kita sampaikan ke F, maka F harus mengeksplorasi dirinya. Dari sana diperoleh nilai I maksimum adalah 0, dengan demikian F minimum akan memperoleh nilai 0. nilai ini jauh lebih kecil dari $\alpha = 3$, sehingga cabangannya ke I yang lainnya tidak dibutuhkan lagi dan harus dipotong (α -pruning) (gambar 3-7). Setelah memotong eksplorasi I, J diuji, dan didapatkan nilai 5. hal ini berarti F minimum akan mendapatkan nilai 5, yang juga berarti C maksimum akan mendapatkan nilai 5 ($\beta = 5$). Hal ini menyebabkan kita harus mengeksplorasi G. dari sana G akan mendapatkan nilai 8. nilai ini jauh lebih besar dari $\beta = 5$. Sehingga tidak diperlukan mengeksplorasi G lebih jauh lagi (β -pruning).



Gambar 3-9
Hasil $\alpha\beta$ -Cutoffs^[4]

Contoh lain terlihat pada gambar 3-10.

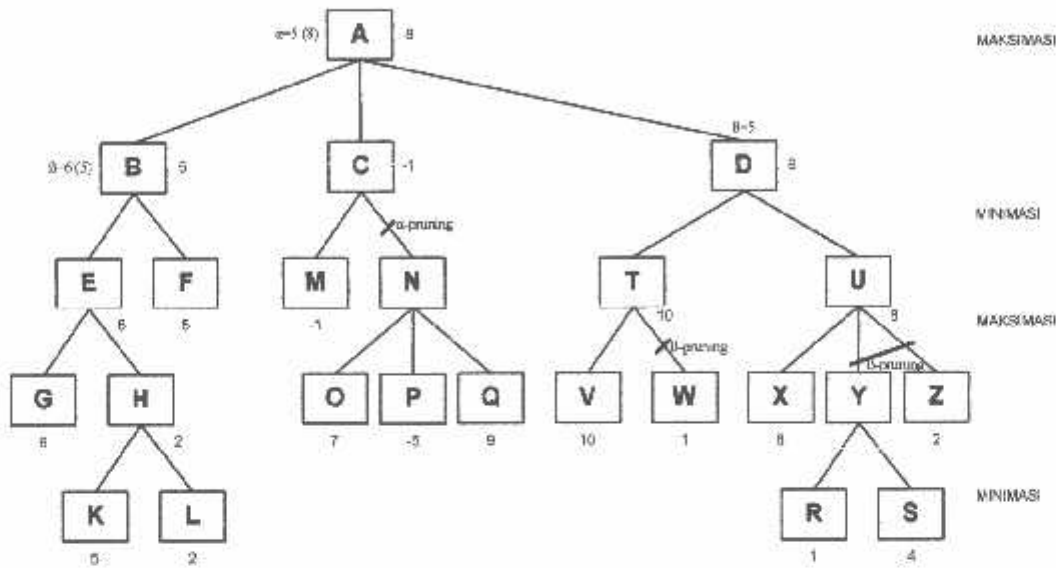


Gambar 3-10
Contoh Lain $\alpha\beta$ -Cutoffs^[4]

Pada gambar 3-10, apabila pelacakan kita mulai pada A, B, kemudian ke subtree dengan akar E, kemudian ke H, maka kita akan mendapatkan nilai $H = 2$ (minimasi), dan nilai $E = 6$ (maksimasi). Dengan demikian, apabila kita melakukan penelusuran melalui node B kemudian dilanjutkan ke node E sebagai akar, maka kita akan mendapatkan nilai $\beta = 6$. namun apabila kita melakukan penelusuran pada node F maka kita akan mendapatkan nilai yang lebih kecil yaitu 5. Sehingga nilai $\beta = 6$ harus diperbaiki menjadi $\beta = 5$. Demikian pula apabila kita naik ke node A, maka kita akan mendapatkan nilai $\alpha = 5$.

Apabila dari A penelusuran diteruskan ke subtree dengan akar C, maka node C akan mendapat 1 atau yang lebih kecil (karena node ini melakukan minimasi), karena $1 < \alpha = 5$, maka akan terjadi α -pruning pada subtree dengan akar N. Apabila penelusuran dilakukan pada subtree dengan akar D kemudian ke node T, maka node D akan mendapatkan nilai $10 > \beta = 5$, maka terjadi β -pruning

pada node W. Apabila penelusuran dilakukan pada subtree dengan akar U, maka D akan mendapatkan nilai 8 atau lebih besar (karena U melakukan maksimasi), karena $8 > \beta = 5$, maka terjadi β -pruning pada node Y dan Z. Hasil penelusuran terlihat pada gambar 3-11.



Gambar 3-11
Hasil $\alpha\beta$ -Cutoffs Untuk Gambar 3-10 [4]

Efektivitas dari $\alpha\beta$ -Cutoffs ini sangat tergantung pada urutan anak-anak yang dikunjungi. Pada kasus terbaik faktor percabangan akan berkurang dari b menjadi \sqrt{b} . Pada kasus rata-rata, faktor percabangan berkurang hingga $b/(\log b)$.

3.2.6. Menunggu Hingga Situasi Menjadi Tenang

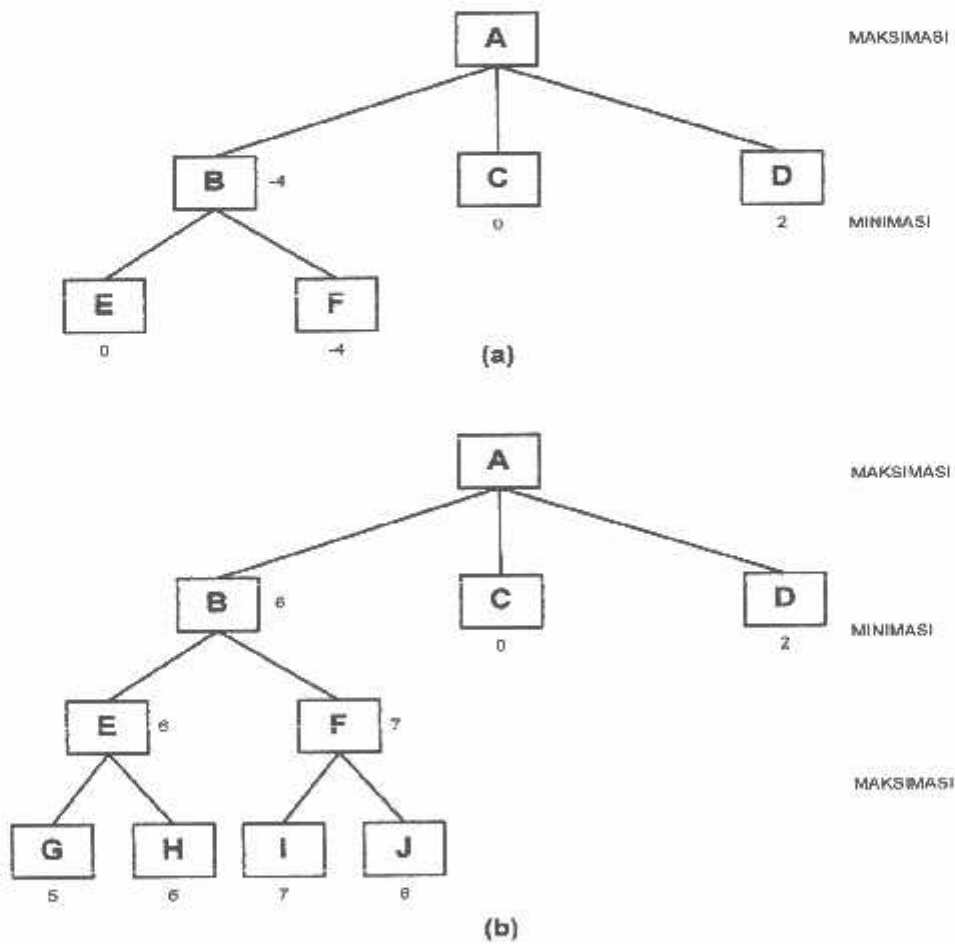
Cara lain untuk membuat gerakan lebih efektif adalah dengan melihat beberapa *ply* ke depan hingga diperoleh keadaan yang stabil. Pada gambar 3-9 terlihat bahwa kita mendapatkan nilai $A = 2$ untuk *games* dengan 2 *ply*. Namun setelah kita explore menjadi 3 *ply*, maka kita akan memperoleh nilai A sebesar 6.

3.2.7. Pencarian Sekunder

Untuk meyakinkan bahwa nilai yang diperoleh sudah optimal, maka dapat dilakukan dengan cara mengekspand beberapa cabang. Proses semacam ini dikenal dengan nama pencarian sekunder.

3.2.8. Book Moves

Cara lain untuk membuat proses pencarian lebih efektif adalah dengan menggunakan bantuan *book moves*. Proses ini dilakukan dengan cara mencari langkah-langkah yang telah terbukti kualitasnya. Proses pencarian hanya dilakukan pada suatu tabel (tidak dalam ruangan).



Gambar 3-12
Keadaan Stabil ^[4]

3.3. Penerapan Metode *Minimax Games With Depth First Search Algorithm* (MGDFS) Pada Rekonfigurasi Sistem Distribusi Radial 20 kV ^[5]

Metode pencarian *Minimax Games With Depth First Search Algorithm* (MGDFS) adalah metode pencarian yang dipelopori oleh metode *primitive games* yang dimunculkan oleh *Claude Shannon* pada tahun 1950. *Minimax Games With Depth First Search Algorithm* (MGDFS) adalah metode games yang menggunakan operator minimax dengan teknik *Depth First Search* yang menggunakan fungsi evaluasi statis yang direpresentasikan dalam suatu pohon faktor suatu ruang keadaan. Fungsi tujuan dari penerapan metode *Minimax Games Depth First Search* (MGDFS) pada permasalahan rekonfigurasi jaringan secara garis besar adalah sebagai berikut:

$$\min C_{(x,t_i)} \dots \dots \dots (3.9)$$

$$F_{(x,t_i)} = 0 \dots \dots \dots (3.10)$$

$$G_{(x,t_i)} = 0 \dots \dots \dots (3.11)$$

Formulasi rumus fungsi tujuan adalah sebagai berikut:

$$C_1(x, t_i) = \sum_{r=1}^{n-1} R_r \frac{P_r^2 + Q_r^2}{V_r^2} [\text{pu}] \dots \dots \dots (3.12)$$

Keterangan:

P = Daya aktif.

Q = Daya reaktif.

V = Tegangan bus.

I = Indeks iterasi untuk setiap bus atau saluran.

R = Resistansi saluran.

Adapun syarat kondisi batasan operasional yang tidak boleh dilanggar adalah sebagai berikut:

$$V_{\min} < V_{\text{bus}(i)} < V_{\max} \dots \dots \dots (3.13)$$

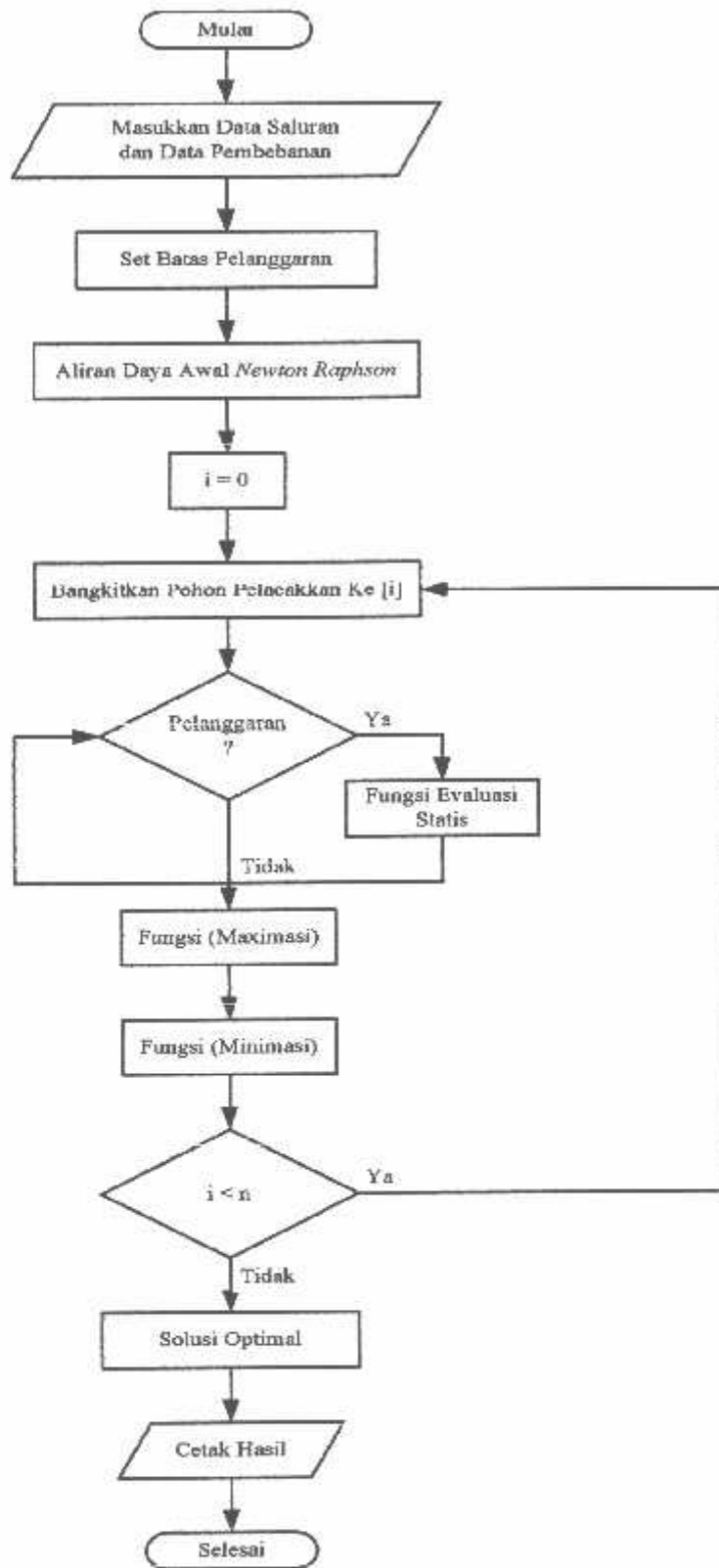
dengan $V_{\min} = 0,95$ pu dan $V_{\max} = 1,05$ pu.

Secara garis besar Konsep metode *Minimax Games With Depth First Search Algorithm* (MGDFS) adalah: Pencarian solusi yang paling optimal namun tetap mempehitungkan faktor-faktor yang dapat menurunkan tingkat solusi.

Adapun algoritma dari *Minimax Games With Depth First Search Algorithm* (MGDFS) adalah:

1. Mulai.
 2. Masukkan inputan data saluran dan data pembebanan serta mengasumsikan posisi switch dalam kondisi tertutup.
 3. Set batas pelanggaran.
 4. Aliran Daya *Newton Raphson*
 5. Set $i = 0$.
 6. Membangkitkan pohon pelacakan ke- i .
 7. Jika terjadi pelanggaran maka:
 - a. Jika ya, aplikasikan fungsi evaluasi statis.
 - b. Jika Tidak, lakukan langkah selanjutnya.
 8. Aplikasikan fungsi maksimasi pohon pelacakan ke- i .
 9. Aplikasikan fungsi minimasi.
-

10. Jika $i \leq n$.
 - a. Jika ya, $i = i + 1$, kembali ke langkah 4.
 - b. Jika tidak, lakukan langkah selanjutnya.
 11. Kombinasi optimal telah ditemukan.
 12. Cetak hasil.
 13. Selesai.
-



Gambar 3-13
Flowchart Penyelesaian Masalah ^[5]

BAB IV

SIMULASI DAN ANALISA SISTEM

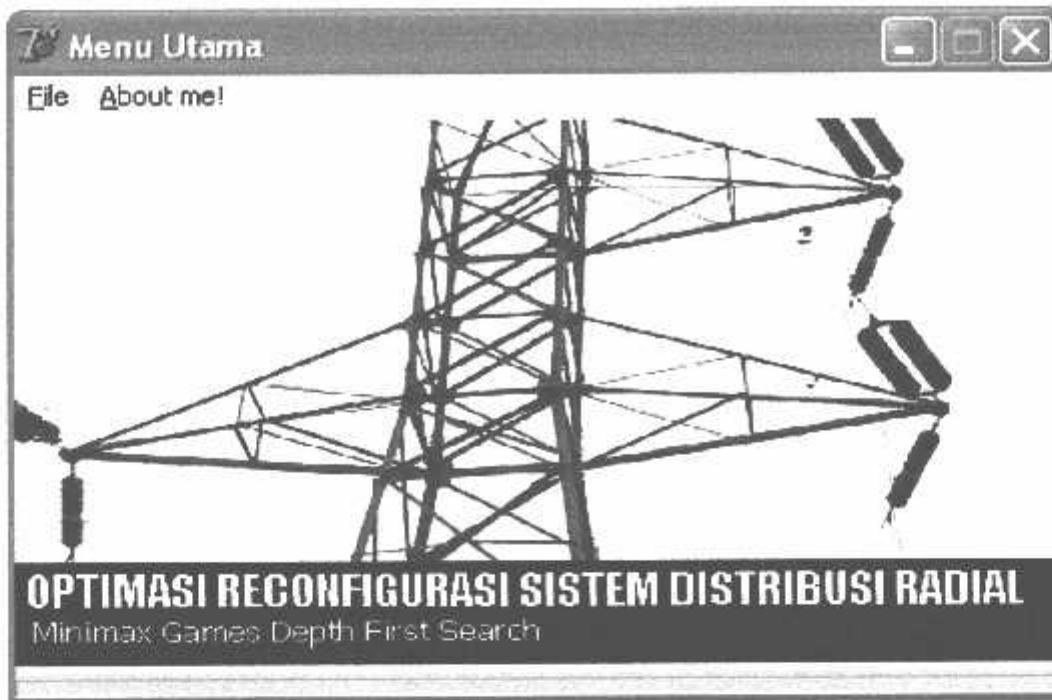
4.1. Fitur Aplikasi

Form utama merupakan tampilan awal program, akan muncul begitu program dijalankan. *Splash screen* dari aplikasi *About* yang berisi identitas dan judul skripsi pembuat dimunculkan pada saat aplikasi pertama kali dieksekusi. *Form* utama akan tampil dengan meng-*clik* tombol OK pada kotak dialog *About*, seperti yang terlihat pada gambar 4-1. Tampilan *about box* seperti gambar berikut:



Gambar 4-1
Kotak Dialog *About*

Begitu *form* utama tampil maka program dapat dijalankan dan *file* yang dipilih dapat dieksekusi. Tampilan *form* utama adalah sebagai berikut:



Gambar 4-2
Form Utama

Untuk membuat file baru untuk dianalisa klik tombol "File" > "Make New", sedangkan untuk membuka file yang sudah tersimpan klik "File" > "Load File", jika pilihan menu utama sudah dipilih maka "Window Data" akan muncul sebagai penyangga data yang akan dianalisa yaitu berupa data batasan operasional, data saluran, dan data pembebanan.

Data Window

General | Data Bus | Data Saluran | Data Generator

Bus	absV (pu)	sudV (deg)	Pg (kW)	Qg (KVAR)	PL (kW)	QL (KVAR)	Cap (pu)	Type Bus
1	1	0	0	0	0	0	0	1
2	1	0	0	0	0.29542	0.22157	0	3
3	1	0	0	0	0.72709	0.54532	0	3
4	1	0	0	0	0.45891	0.33418	0	3
5	1	0	0	0	1.44844	1.08633	0	3
6	1	0	0	0	1.1186	0.83895	0	3
7	1	0	0	0	0	0	0	3
8	1	0	0	0	1.3208	0.9906	0	3
9	1	0	0	0	0	0	0	3
10	1	0	0	0	0.43597	0.32697	0	3
11	1	0	0	0	0	0	0	3
12	1	0	0	0	0	0	0	3
13	1	0	0	0	0	0	0	3
14	1	0	0	0	0	0	0	3
15	1	0	0	0	0	0	0	3

Apply Close

Gambar 4-3
Form Data Bus

Data Window

General | Data Bus | Data Saluran | Data Generator

No	Dari	Ke	R (pu)	X (pu)	Lc (pu)	Tr	Tu	Su (deg)	Kap (VA)
1	1	2	0.1345	0.2057	0	0	0	0	10000
2	2	7	0.0253	0.0318	0	0	0	0	10000
3	2	3	0.0937	0.1433	0	0	0	0	10000
4	3	8	0.2737	0.3438	0	0	0	0	10000
5	3	9	0.0862	0.1082	0	0	0	0	10000
6	3	4	0.2059	0.3148	0	0	0	0	10000
7	4	10	0.2585	0.3247	0	0	0	0	10000
8	4	5	0.473	0.7231	0	0	0	0	10000
9	5	11	0.0506	0.0636	0	0	0	0	10000
10	5	12	0.0202	0.0254	0	0	0	0	10000
11	5	6	0.577	0.8821	0	0	0	0	10000
12	1	13	0.0593	0.0907	0	0	0	0	10000
13	13	21	0.2807	0.3525	0	0	0	0	10000
14	13	14	0.0255	0.0291	0	0	0	0	10000

Apply Close

Gambar 4-4
Form Data Saluran

Dengan data jaringan yang ada maka dilakukan perhitungan untuk mencari aliran daya awal yang merupakan aliran daya dari sistem sebelum rekonfigurasi. Pada tahap ini digunakan metode *Newton Raphson*, hasil dari perhitungan ini akan di bandingkan dengan hasil rekonfigurasi sehingga diketahui besar reduksi rugi daya. Setelah data telah dimuati dengan lengkap klik tombol “*Apply*” untuk melanjutkan perhitungan analisa aliran daya pada *form* selanjutnya, berikut adalah tampilan *form* selanjutnya.

Bus	absV (pu)	sudV (deg)	Pg (MW)	Qg (MVAR)	PL (MW)	QL (MVAR)	Sups (pu)	Type Bus
1	1.00000	0.00000	27.208	20.551	0.000	0.000	0.000	1
2	0.99445	-0.11197	0.000	0.000	0.295	0.222	0.000	3
3	0.99042	-0.18040	0.000	0.000	0.727	0.545	0.000	3
4	0.98652	-0.26210	0.000	0.000	0.459	0.334	0.000	3
5	0.97985	-0.40222	0.000	0.000	1.448	1.086	0.000	3
6	0.97630	-0.47748	0.000	0.000	1.119	0.839	0.000	3
7	0.99452	-0.11406	0.000	0.000	0.000	0.000	0.000	3
8	0.98864	-0.20717	0.000	0.000	1.321	0.991	0.000	3
9	0.98942	-0.18299	0.000	0.000	0.000	0.000	0.000	3
10	0.98596	-0.27050	0.000	0.000	0.436	0.327	0.000	3
11	0.97985	-0.40222	0.000	0.000	0.000	0.000	0.000	3
12	0.97985	-0.40222	0.000	0.000	0.000	0.000	0.000	3

Buttons:

Gambar 4-5
Form Result Window

Untuk melanjutkan dengan menerapkan fungsi *Minimax Games Depth First Search* (MGDFS) yang diimplementasikan pada tombol “*MGDFS Search*”. Untuk transparansi hasil berupa grafik profil tegangan dan solusi lokasi *Open*

Switch serta tabel rugi daya saluran sebelum dan sesudah optimasi dapat dilihat pada *tabsheet "Final Result"*.

4.2. Analisa Rekonfigurasi Jaringan Distribusi Dengan Metode *Minimax Games Depth First Search (MGDFS)*

Dalam memilih suatu metode solusi untuk aplikasi praktis sering sulit, pilihan itu memerlukan analisis yang cermat atas kelebihan-kelebihan dan kekurangan-kekurangan dari metode yang ada. Untuk solusi dari permasalahan diatas maka digunakan suatu alternatif dengan menggunakan metode *Minimax Games Depth First Search (MGDFS)* yang menyediakan cara yang lebih baik dalam menganalisa masalah rekonfigurasi jaringan. Dalam perhitungan analisa rekonfigurasi disini menggunakan program komputer. Pada prinsipnya penggunaan program komputer dapat menganalisa sebuah jaringan distribusi radial dengan jumlah cabang atau jumlah bus yang tidak terbatas, tergantung dari memori yang tersedia pada komputer yang digunakan.

4.2.1. Data Perhitungan

Dalam perhitungan aliran daya data diambil dari GI Rungkut, Surabaya. Sistem distribusi radial G.I Rungkut memakai tegangan distribusi 20 kV. Untuk menyelesaikan perhitungan aliran daya terlebih dahulu ditetapkan *Single Line* diagram yang akan dianalisa. Agar memudahkan perhitungan maka digunakan sistem per-unit (p.u). Transformator IV G.I. Rungkut terdiri dari:

- Tegangan Dasar : 20 kV
- Daya Dasar : 50 MVA
- Penghantar SUTM 20 KV : AAAC 150 mm²
- Impedansi Saluran : 0,2162 + j 0,3305 Ω / km

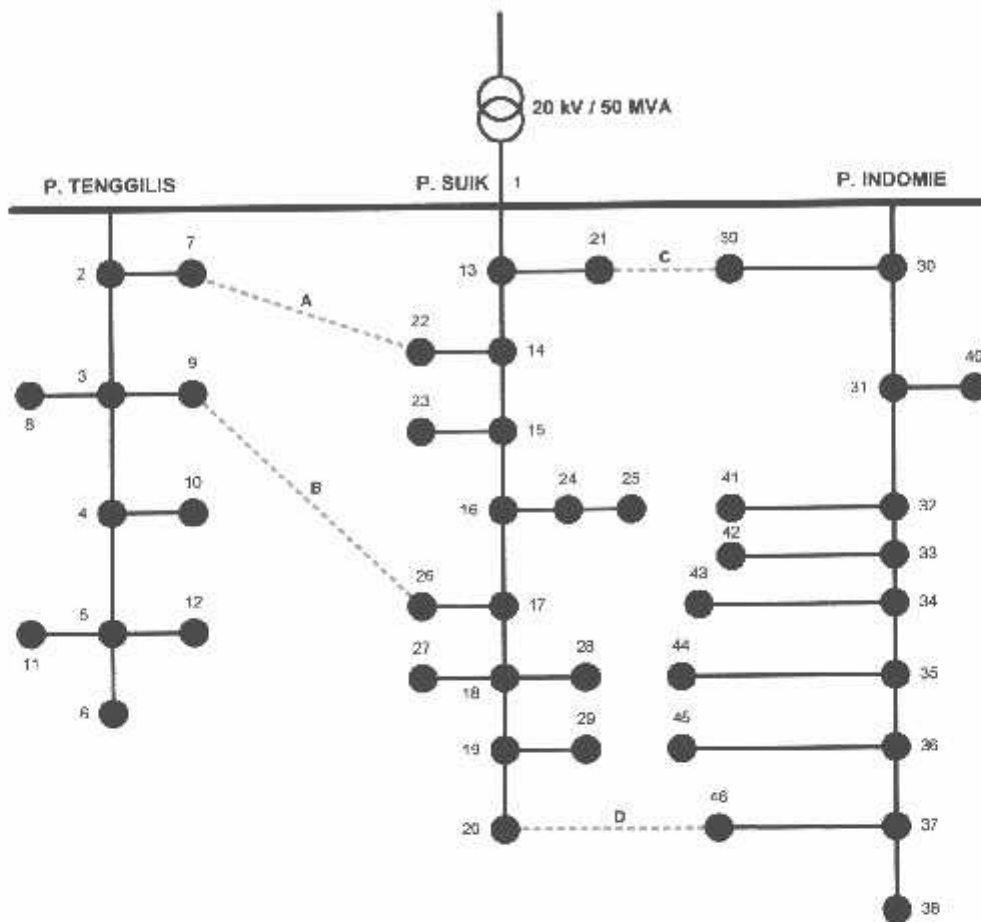
Dari data beban untuk masing-masing penyulang (lampiran) dan data teknis, nilai P (MW), Q (MVAR), R (Ω) dan X (Ω) dari bus 1-2 Penyulang Tenggilis dapat dihitung sebagai berikut:

❖ Untuk bus 1-2:

$$\begin{aligned} R (\Omega) &= R (\text{ohm} / \text{km}) \times \text{Panjang saluran} \\ &= 0,1345 \times 0,6625 = 0,089 \Omega \end{aligned}$$

$$\begin{aligned} X (\Omega) &= X (\text{ohm} / \text{km}) \times \text{Panjang saluran} \\ &= 0,2057 \times 0,6625 = 0,137 \Omega \end{aligned}$$

Dengan cara menghitung yang sama, maka daya R, dan X untuk masing-masing bus akan diperoleh hasil seperti pada tabel 4-3.



Gambar 4-6
Konfigurasi Sistem Distribusi Radial 20 kV dari *Single Line Diagram*
Gardu Induk Rungkut Sebelum Rekonfigurasi

Keterangan Saklar Normal-Open:

A. LBS Kuntisari.

B. AVS Sier Barat.

C. LBS Surya Multi.

D. LBS Henson Makmur.

Sedangkan pada bus-busnya sebagai saklar *Normally Close*.

Dalam skripsi ini data yang digunakan adalah penyulang-penyulang yang keluar dari Trafo IV G.I Rungkut. Kondisi awal jaringan (*Initial Condition*) terlihat pada lampiran *Single Line* diagram. Untuk memudahkan analisis, gambar *Single Line diagram* tersebut diubah ke dalam bentuk gambar konfigurasi jaringan yang telah disusun perseksi seperti yang terlihat pada gambar 4-6. Penyusunan gambar 4-6 dilakukan dengan terlebih dahulu menentukan pembagian seksi tiap penyulang. Pembagian seksi tiap penyulang ditentukan berdasarkan adanya pemisah-pemisah yaitu *Automatic Vacuum Switch (AVS)*, dan *Load Break Switch (LBS)* dan *Pole Mounth SF6 Gas Switch (PGS)* yang terdapat pada lampiran *Single Line diagram* G.I Rungkut untuk *output* dari Trafo IV (penjelasan lebih lanjut dapat dilihat pada lampiran). Atau dengan kata lain bus beban merupakan penyusunan dari trafo-trafo yang ada dalam saluran pada masing-masing penyulang yang dibatasi dengan adanya *switch*.

Alasan dari pemilihan daerah tersebut karena *output* dari Trafo IV G.I Rungkut menyuplai daerah yang kepadatan bebannya bervariasi. Selanjutnya bus-bus yang ada diklasifikasikan, yaitu *Busbar* G.I Rungkut diasumsikan sebagai *Slack Bus*, sedangkan bus-bus yang lain sepanjang saluran radial dipandang sebagai *Load Bus*. Dalam hal ini tidak ada bus generator karena sepanjang saluran tidak terdapat pembangkitan. Dalam analisis rekonfigurasi jaringan semua *Tie Switch (Normally Open)* yang saling interkoneksi pada beberapa bus beban (*Load Bus*) baik dalam satu *feeder* maupun berlainan *feeder* ditutup (*Normally Closed*) sehingga sistem distribusi radial 20 kV seperti pada gambar 4-6.

4.2.1.1. Data Saluran

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

Tabel 4-1
Data Saluran Sistem 20 kV G.I Rungkut

Jenis Konduktor	Penampang Nominal (mm ²)	Resistansi Konduktor (r) (Ω/km)	GMR (mm)	Kuat Hantar Arus (A)
AAAC	35	0,9217	2,4227	170
AAAC	50	0,6452	2,8957	210
AAAC	70	0,4608	3,4262	255
AAAC	120	0,2688	4,6837	365
AAAC	150	0,2162	5,2365	425

Tabel 4-2
Data Penghantar AAAC

Penampang Nominal (mm ²)	Jari-jari (mm)	Urut	GMR (mm)	Impedansi (Ω/km)
35	3,3371	7	2,4227	0,9217 + j 0,3790
50	3,9886	7	2,8957	0,6452 + j 0,3678
70	4,7193	7	3,4262	0,4608 + j 0,3572
120	6,1791	19	4,6837	0,2688 + j 0,3376
150	6,9084	19	5,2365	0,2162 + j 0,3305

Tabel 2 selengkapnya dapat dilihat pada lampiran I

4.2.1.2. Data Pembebanan

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

Tabel 4-3
Data Pembebanan G.I Rungkut

No Bus	Pembebanan		Type Bus
	P (Kw)	Q(Kvar)	
1	0	0	Slack
2	296.519	222.389	Load
3	728.549	546.412	Load
4	476.646	357.485	Load
5	1388.371	1041.279	Load
6	1116.793	837.595	Load
7	0	0	Load
8	1242.884	932.162	Load
9	0	0	Load
10	415.68	311.761	Load
11	0	0	Load
12	0	0	Slack
13	0	0	Load
14	0	0	Load
15	0	0	Load
16	0	0	Load
17	1807.502	1355.626	Load
18	474.852	356.139	Load
19	626.638	469.977	Load
20	0	0	Load
21	2317.167	1737.875	Load
22	0	0	Load
23	1896.623	1422.467	Load
24	810.451	607.838	Load
25	5414.149	4060.612	Load
26	3319.687	2489.84	Load
27	0	0	Load
28	0	0	Load
29	346.74	260.055	Slack
30	103.048	77.286	Load
31	0	0	Load
32	0	0	Load
33	0	0	Load
34	0	0	Load
35	41.715	31.332	Load
36	20.888	15.666	Load
37	27.85	20.887	Load
38	0	0	Load
39	0	0	Load
40	771.46	578.595	Load
41	0	0	Load
42	286.861	215.146	Load
43	0	0	Load
44	3317.001	2487.751	Load
45	2751.634	2063.726	Load
46	955.273	716.457	Load

Bus 1 merupakan *bus slack*, Bus 2 sampai dengan 46 merupakan bus beban

4.3. Analisa Perhitungan

Perhitungan rekonfigurasi ini diawali dengan melakukan studi aliran daya dengan menggunakan metode *Newton Raphson*. Studi aliran daya dilakukan untuk mengetahui harga tegangan dan sudut fasa di tiap-tiap bus. Setelah studi aliran daya dilakukan, maka dapat dilakukan perhitungan aliran daya akhir dan rugi-rugi daya pada saluran.

Untuk memudahkan perhitungan dan analisa pada sistem tenaga, biasanya dipakai harga-harga dalam per-satuan. Harga per-satuan adalah harga yang sebenarnya dibagi dengan harga dasar, dimana harga dasar ini dapat dipilih sembarang. Harga yang dipilih pada studi ini adalah 20 kV dan 50 MVA sebagai harga tegangan dasar dan daya dasar. Mengingat bahwa pada jaringan tidak dilakukan pengukuran faktor daya, maka pada perhitungan ini diambil harga faktor daya sebesar 0.86.

Perhitungan diawali dengan menampilkan data tabel perwakilan *Single Line* diagram dari penyulang untuk mewakili keadaan sistem yang sesungguhnya. Jumlah bus dan jumlah saluran masing-masing adalah:

- Slack bus = 1
- Load bus = 46
- Jumlah saluran = 49

Dengan mengacu pada data saluran G.I Rungkut, maka dapat diperoleh hasil impedansi saluran seperti pada tabel 4-4 dibawah ini. Setelah itu baru kita lakukan perhitungan aliran daya dengan menggunakan metode *Newton Raphson*.

Tabel 4-4
Data Impedansi Saluran G.I Rungkut

Saluran	Dari Node	Ke Node	Panjang (Km)	Impedansi Saluran	
				R (Ω)	X (Ω)
1	1	2	0.6225	0.1345	0.2057
2	2	7	0.0943	0.0253	0.0318
3	2	3	0.4338	0.0937	0.1433
4	3	8	1.0186	0.2737	0.3438
5	3	9	0.3207	0.0862	0.1082
6	3	4	0.9526	0.2059	0.3148
7	4	10	0.962	0.2585	0.3247
8	4	5	2.1881	0.473	0.7231
9	5	11	0.1886	0.0506	0.0636
10	5	12	0.0755	0.0202	0.0254
11	5	6	2.6691	0.577	0.8821
12	1	13	0.2746	0.0593	0.0907
13	13	21	1.0445	0.2807	0.3526
14	13	14	0.1184	0.0255	0.0391
15	14	22	0.0700	0.0188	0.0236
16	14	15	0.1238	0.0267	0.0409
17	15	23	0.4684	0.1259	0.1581
18	15	16	0.0431	0.0931	0.0142
19	16	24	0.1777	0.0477	0.0599
20	24	25	0.9529	0.2561	0.3216
21	16	17	0.4146	0.0896	0.137
22	17	26	0.7645	0.2054	0.258
23	17	18	0.1184	0.0255	0.0391
24	18	27	0.0485	0.013	0.0163
25	18	28	0.0485	0.013	0.0163
26	18	19	0.3661	0.0791	0.1235
27	19	29	0.0538	0.0144	0.0181
28	19	20	0.1615	0.0349	0.0533
29	1	30	0.4502	0.0973	0.1487
30	30	39	0.0048	0.0129	0.0162
31	30	31	0.0718	0.0155	0.0237
32	31	40	0.5029	0.1351	0.1697
33	31	32	0.2347	0.0507	0.0775
34	32	41	0.0239	0.0642	0.0806
35	32	33	0.091	0.0196	0.03
36	33	42	0.0239	0.0642	0.0806
37	33	34	0.0192	0.0415	0.0634
38	34	43	0.0144	0.0387	0.0486
39	34	35	0.1437	0.031	0.0474
40	35	44	0.8573	0.2304	0.2894
41	35	36	0.091	0.0196	0.03
42	36	45	0.5268	0.1416	0.1778
43	36	37	0.1676	0.0362	0.0553
44	37	46	0.7088	0.1905	0.2392
45	37	38	0.0623	0.0134	0.0205
46	7	22	0.1443	0.0387	0.0487
47	9	26	0.3707	0.0996	0.1251
48	21	39	1.0945	0.2942	0.3695
49	20	46	0.7588	0.2039	0.2561

Bus 1 merupakan *bus slack*, Bus 2 sampai dengan 49 merupakan bus beban

Setelah dilakukan analisa aliran daya dengan menggunakan metode *Newton Raphson* maka diperoleh aliran daya tiap saluran seperti pada tabel 4-5.

Tabel 4-5
Aliran Daya Tiap Saluran G.I Rungkut Sebelum Konfigurasi

No	Dari	Ke	P (MW)	Q (MVAR)	Dari	Ke	P (MW)	Q (MVAR)
1	1	2	7.594	5.835	2	1	-7.563	-5.788
2	2	7	-0.161	-0.772	7	2	0.161	0.772
3	2	3	7.429	6.339	3	2	-7.406	-6.304
4	3	8	1.323	0.993	8	3	-1.321	-0.991
5	3	9	1.868	2.14	9	3	-1.867	-2.138
6	3	4	3.488	2.626	4	3	-3.478	-2.611
7	4	10	0.436	0.327	10	4	-0.436	-0.327
8	4	5	2.583	1.949	5	4	-2.57	-1.93
9	5	11	0	0	11	5	0	0
10	5	12	0	0	12	5	0	0
11	5	6	1.122	0.843	6	5	-1.119	-0.839
12	1	13	12.181	8.775	13	1	-2.148	-8.724
13	13	21	0.732	0.652	21	13	-0.732	-0.651
14	13	14	11.415	8.071	14	13	-1.403	-8.052
15	14	22	0.161	0.773	22	14	-0.161	-0.773
16	14	15	11.242	7.28	15	14	-11.23	-7.261
17	15	23	0.96	0.721	23	15	-0.96	-0.72
18	15	16	10.27	6.54	16	15	-0.235	-6.535
19	16	24	6.315	4.757	24	16	-6.308	-4.747
20	24	25	5.63	4.239	25	24	-5.598	-4.198
21	16	17	3.919	1.778	17	16	-3.915	-1.772
22	17	26	1.232	0.188	26	17	-1.231	-0.187
23	17	18	1.362	0.593	18	17	-1.362	-0.593
24	18	27	0	0	27	18	0	0
25	18	28	0	0	28	18	0	0
26	18	19	0.693	0.092	19	18	-0.693	-0.091
27	19	29	0.19	0.143	29	19	-0.19	-0.143
28	19	20	-0.026	-0.448	20	19	0.026	0.448
29	1	30	7.432	5.941	30	1	-7.41	-5.907
30	30	39	0.649	0.384	39	30	-0.648	-0.384
31	30	31	6.664	5.45	31	30	-6.661	-5.445
32	31	40	0.927	0.695	40	31	-0.926	-0.695
33	31	32	5.734	4.75	32	31	-5.727	-4.739
34	32	41	0	0	41	32	0	0
35	32	33	5.727	4.739	33	32	-5.724	-4.735
36	33	42	0.071	0.053	42	33	-0.071	-0.053
37	33	34	5.653	4.682	34	33	-5.648	-4.673
38	34	43	0	0	43	34	0	0
39	34	35	5.648	4.673	35	34	-5.643	-4.666

Selengkapnya dapat di lihat pada lampiran 1

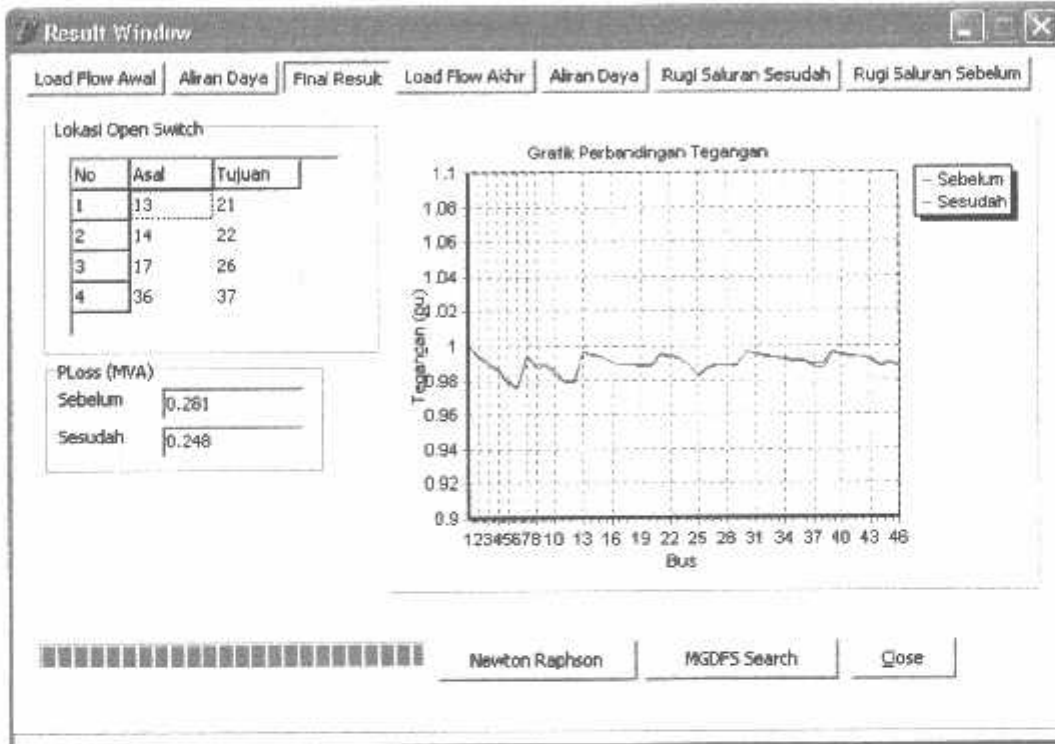
Hasil *Loadflow Newton Raphson*

Jumlah Bus = 46

Jumlah Saluran = 49

Jumlah Iterasi = 2

Analisa dengan menggunakan metode *Minimax Games Depth First Search* (MGDFS), akan diperoleh hasil yaitu lokasi *open switch* yang memiliki imbas positif yang paling besar terhadap kondisi tegangan, Seperti terlihat pada gambar 4-7 tampilan hasil perhitungan dengan metode *Minimax Games Depth First Search* (MGDFS).



Gambar 4-7
Hasil Perhitungan dan Perbandingan Tegangan Sebelum dan Tegangan Sesudah Konfigurasi

Dari gambar diatas dapat dilihat bahwa lokasi *open switch* antara bus 13 dan bus 21, bus 14 dan bus 22, bus ke 17 dan bus 26, bus 36 dan bus 37 memberikan solusi yang paling optimal terlihat dari pengaruhnya terhadap tegangan bus yang dapat dilihat pada grafik. Setelah dilakukan konfigurasi ulang maka diperoleh perbaikan profil tegangan, aliran daya tiap-tiap saluran dan rugi-rugi saluran. Selengkapnya dapat dilihat pada lampiran I.

Tabel 4-6
Aliran Daya Tiap Saluran G.I Rungkut Sesudah Konfigurasi

No	Dari	Ke	P (MW)	Q (MVAR)	Dari	Ke	P (MW)	Q (MVAR)
1	1	2	9.007	6.825	2	1	-8.964	-6.759
2	2	7	0	0	7	2	0	0
3	2	3	8.668	6.538	3	2	-8.64	-6.495
4	3	8	1.323	0.993	8	3	-1.321	-0.991
5	3	9	3.103	2.331	9	3	-3.099	-2.327
6	3	4	3.488	2.626	4	3	-3.478	-2.611
7	4	10	0.436	0.327	10	4	-0.436	-0.327
8	4	5	2.583	1.949	5	4	-2.57	-1.93
9	5	11	0	0	11	5	0	0
10	5	12	0	0	12	5	0	0
11	5	6	1.122	0.843	6	5	-1.119	-0.839
12	1	13	10.962	8.267	13	1	-10.935	-8.224
13	13	21	0	0	21	13	0	0
14	13	14	10.934	8.224	14	13	-10.922	-8.205
15	14	22	0	0	22	14	0	0
16	14	15	10.922	8.205	15	14	-10.91	-8.186
17	15	23	0.96	0.721	23	15	-0.96	-0.72
18	15	16	9.949	7.466	16	15	-9.913	-7.46
19	16	24	6.315	4.757	24	16	-6.308	-4.747
20	24	25	5.63	4.239	25	24	-5.598	-4.198
21	16	17	3.598	2.703	17	16	-3.593	-2.696
22	17	26	0	0	26	17	0	0
23	17	18	2.272	1.705	18	17	-2.271	-1.705
24	18	27	0	0	27	18	0	0
25	18	28	0	0	28	18	0	0
26	18	19	1.603	1.203	19	18	-1.602	-1.202
27	19	29	0.19	0.143	29	19	-0.19	-0.143
28	19	20	0.882	0.662	20	19	-0.882	-0.662
29	1	30	7.247	5.468	30	1	-7.227	-5.438
30	30	39	1.382	1.038	39	30	-1.382	-1.038
31	30	31	5.746	4.326	31	30	-5.744	-4.323
32	31	40	0.927	0.695	40	31	-0.926	-0.695
33	31	32	4.817	3.628	32	31	-4.813	-3.621
34	32	41	0	0	41	32	0	0
35	32	33	4.813	3.621	33	32	-4.811	-3.618
36	33	42	0.071	0.053	42	33	-0.071	-0.053
37	33	34	4.74	3.565	34	33	-4.736	-3.559
38	34	43	0	0	43	34	0	0
39	34	35	4.736	3.559	35	34	-4.733	-3.555
40	35	44	2.587	1.943	44	35	-2.581	-1.935
41	35	36	2.108	1.582	36	35	-2.107	-1.582
42	36	45	2.082	1.563	45	36	-2.08	-1.56
43	36	37	0	0	37	36	0	0
44	37	46	-0.025	-0.019	46	37	0.025	0.019
45	37	38	0	0	38	37	0	0
46	7	22	0	0	22	7	0	0
47	9	26	3.099	2.327	26	9	-3.096	-2.322
48	21	39	-1.38	-1.035	39	21	1.382	1.038
49	20	46	0.882	0.662	46	20	-0.882	-0.661

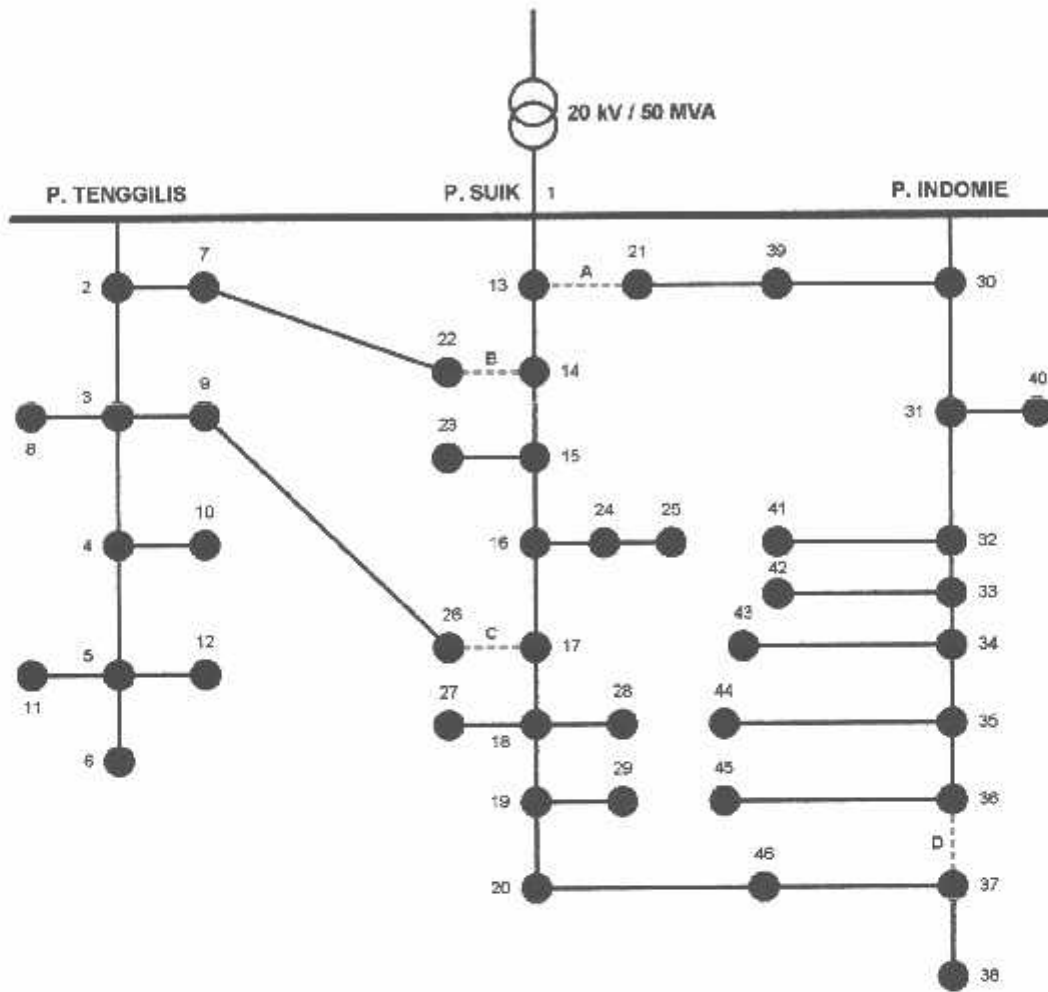
Hasil *Loadflow Newton Raphson*

Jumlah Bus = 46

Jumlah Saluran = 49

Jumlah Iterasi = 3

Setelah melakukan analisa dan perhitungan dengan metode *Minimum Games Depth First Search* (MGDFS), maka diperoleh hasil rekonfigurasi seperti pada gambar 4-8 berikut ini.



Gambar 4-8
Konfigurasi Jaringan Radial Setelah Rekonfigurasi

Tabel 4-7
Hasil Kombinasi Optimal *Open Switch*

No.	Bus Pangkal	Bus Ujung
1.	13	21
2.	14	22
3.	17	26
4.	36	37

Dari gambar 4-7 terlihat bahwa tegangan tiap bus sebelum dilakukan konfigurasi rata-rata sebesar dibawah 0,975 pu, tetapi setelah dilakukan konfigurasi dengan mengkonfigurasi ulang *switch* pada jaringan maka terjadi perbaikan tegangan, rata-rata besar tegangan pada tiap-tiap bus menjadi diatas 0.977 pu.

4.4. Validasi Program

Pada tulisan ini validasi yang diterapkan menggunakan jurnal "*Optimal Radialization Of Primary Distribution Networks With Multicost and Multipoint Alternative Supply*" Session 19, Paper 4. Berikut adalah data analisis yang digunakan.

Tabel 4-8
Data Validasi Jurnal

Bus asal	Bus Tujuan	R (pu)	X (pu)
1	4	0.075	0.1
4	5	0.08	0.11
4	6	0.09	0.18
6	7	0.04	0.04
2	8	0.11	0.11
8	9	0.08	0.11
8	10	0.11	0.11
9	11	0.11	0.11
9	12	0.08	0.11
3	13	0.11	0.11
13	14	0.09	0.12
13	15	0.08	0.11
15	16	0.04	0.04
5	11	0.04	0.04
10	14	0.04	0.04
7	16	0.09	0.12

Hasil optimal lokasi *open switch* pada program antara bus 4 dan bus 6, bus 8 dan bus 10, bus 9 dan bus 11, bus 10 dan bus 14 adalah sebagai berikut:

Tabel 4-9
Hasil Kombinasi Optimal *Open Switch* Pada Program

No	Asal	Tujuan
	4	6
	8	10
	9	11
	10	14

Berikut adalah tampilan hasil aliran daya pada program menggunakan metode *Minimax Games Depth First Search* (MGDFS).

Result Window

Load Flow Awal | Aliran Daya | **Final Result** | Load Flow Akhir | Aliran Daya | Rugi Saluran Sesudah | Rugi Saluran Sebelum

Jumlah Pembangkitan: 3389.381 + j 4111.386 KVA
 Jumlah Pembebanan: 2700.09 + j 1300.00 KVA
 Jumlah Rugi-Rugi: 3125.05 + j 185.00 KVA
 Iterasi: 15
 Waktu Hitung: 0:0:60

Simpan Tabel

No	Dari	Ke	P (KW)	Q (KVAR)	Arus re (A)	Arus Im (A)	Dari	Ke	P (kW)	Q
1	1	4	389.381	411.386	19469.041	20569.297	4	1	412.758	6
2	4	5	1223.679	928.253	-50391.567	-24416.332	5	4	2120.820	36
3	4	6	164.006	252.323	-7670.089	-7844.872	6	4	-19.559	3
4	6	7	16.584	-83.537	-19756.463	-10831.903	7	6	254.161	31
5	2	8	56.022	-33.634	560.014	3364.067	8	2	-38.964	5
6	8	9	-54.855	-34.520	3342.127	-884.066	9	8	67.603	5
7	8	10	272.297	127.345	-14873.381	5989.411	10	8	104.769	3

Newton Raphson | MGDFS Search | Close

Gambar 4-9
Data Hasil Aliran Daya Pada Program

Pada hasil diatas maka total Ploss harus disetarakan dalam satuan MW dengan cara sebagai berikut:

$$\begin{aligned} \text{Ploss (MW)} &= \frac{\text{Ploss (kVA)} \times \cos \varphi}{1000} \\ &= \frac{3125 \text{ kVA} \times 0.8}{1000} \\ &= 2.5 \text{ MW} \end{aligned}$$

Tabel 4-10
Hasil Perbandingan Program

Hasil	Open Switch	Total Ploss (MW)
Jurnal	Bus 4-6, Bus 8-10, Bus 9-11	2.46
Program	Bus 4-6, Bus 8-10, Bus 9-11, Bus 10-14	2.5
Error (%)	-	1.626

Maka didapat hasil persentase errornya sebagai berikut:

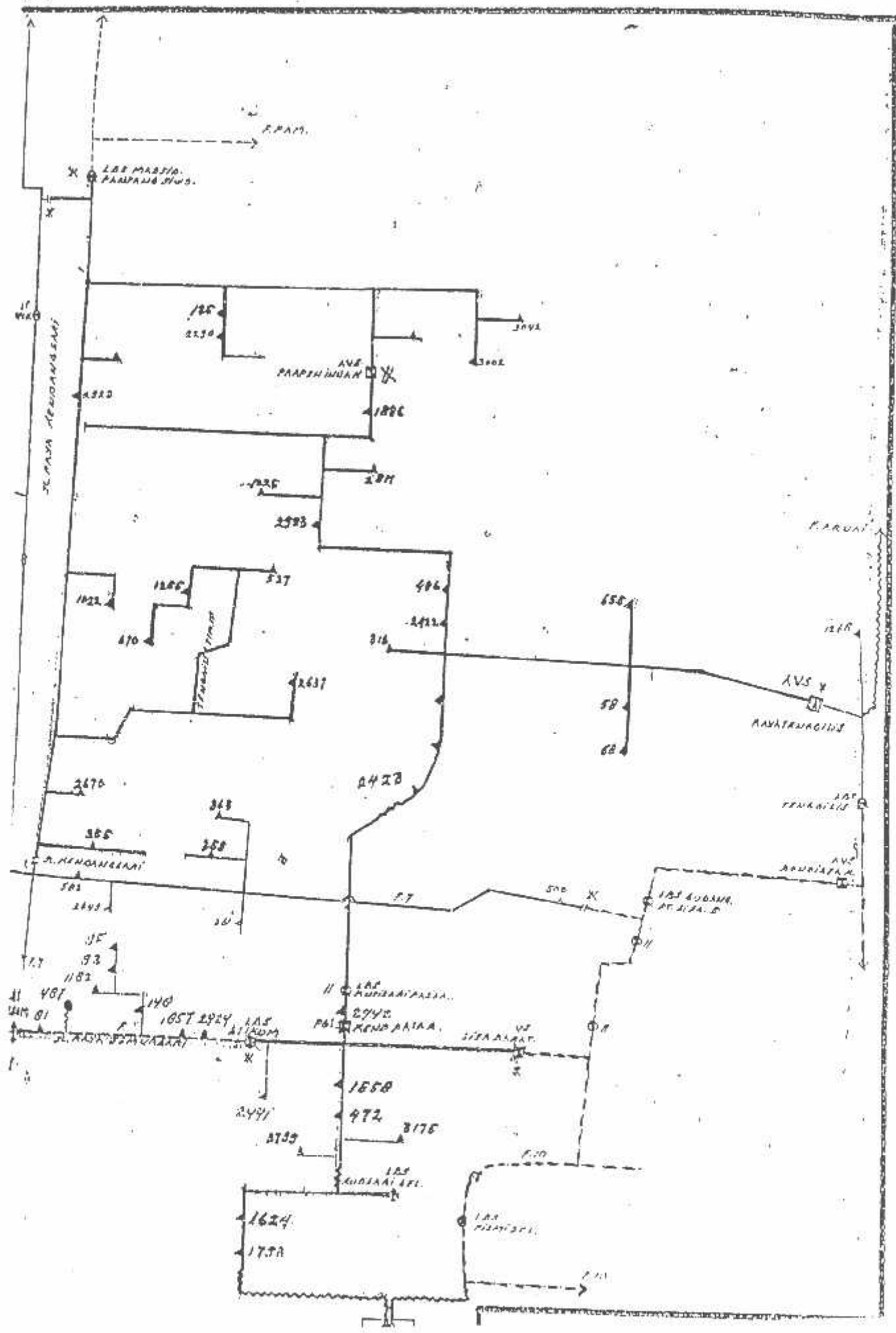
$$\begin{aligned} \% \text{ Error} &= \left(\frac{\text{Hasil Jurnal} - \text{Hasil Program}}{\text{Hasil Jurnal}} \right) \times 100\% \\ &= \left(\frac{2.46 - 2.5}{2.46} \right) \times 100\% \\ &= 1.626 \% \end{aligned}$$

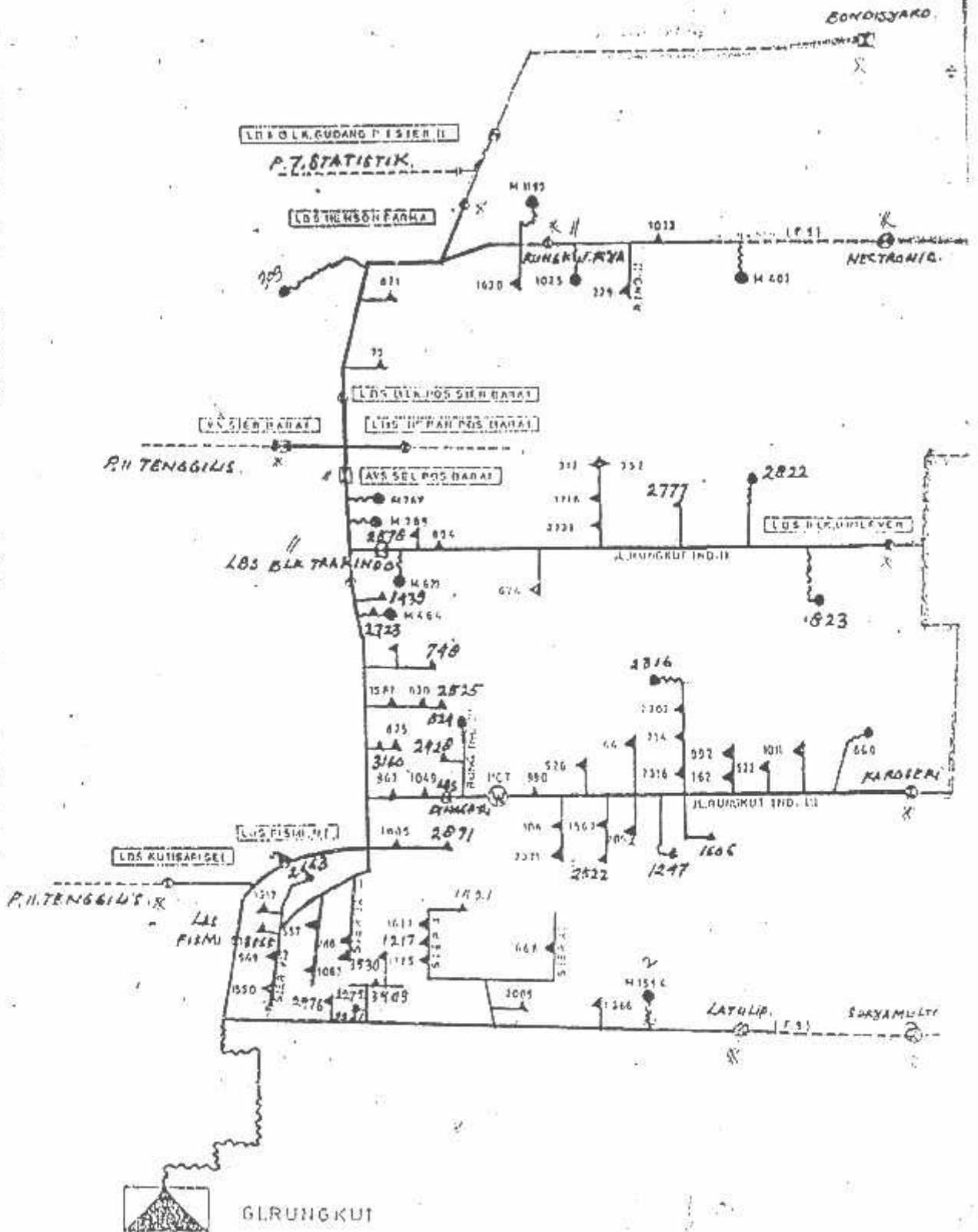
DAFTAR PUSTAKA

- [1] Basri, Hasan. 1996. "*Sistem Distribusi Tenaga Listrik*". Jakarta Selatan: Balai Penerbit dan Humas ISTN Bumi Srengsreng Indah.
 - [2] Stevenson, William D. Jr. 1996. "*Analisis Sistem Tenaga Listrik*". Jakarta: Penerbit Erlangga Edisi Ke Empat.
 - [3] Marsudi, Djiteng. 1990. "*Operasi Sistem Tenaga Listrik*". Jakarta Selatan: Balai Penerbit dan Humas ISTN Bumi Srengsreng Indah.
 - [4] Dewi, Kusuma, Sri. 2005. "*Artificial Intelligence (Teknik dan Aplikasinya)*". Yogyakarta: Graha Ilmu.
 - [5] Rudnick, Hugh. Harnich, Ildefonso. Sanhueza, Raul. 2004. "*Reconfiguration Of Electric Distribution Systems Using Mnimax Games With Depth First Search Algorithm (MGDFS)*". Chile: Revista Facultad De Ingenieria, U. T. A. Vol. 4.
-

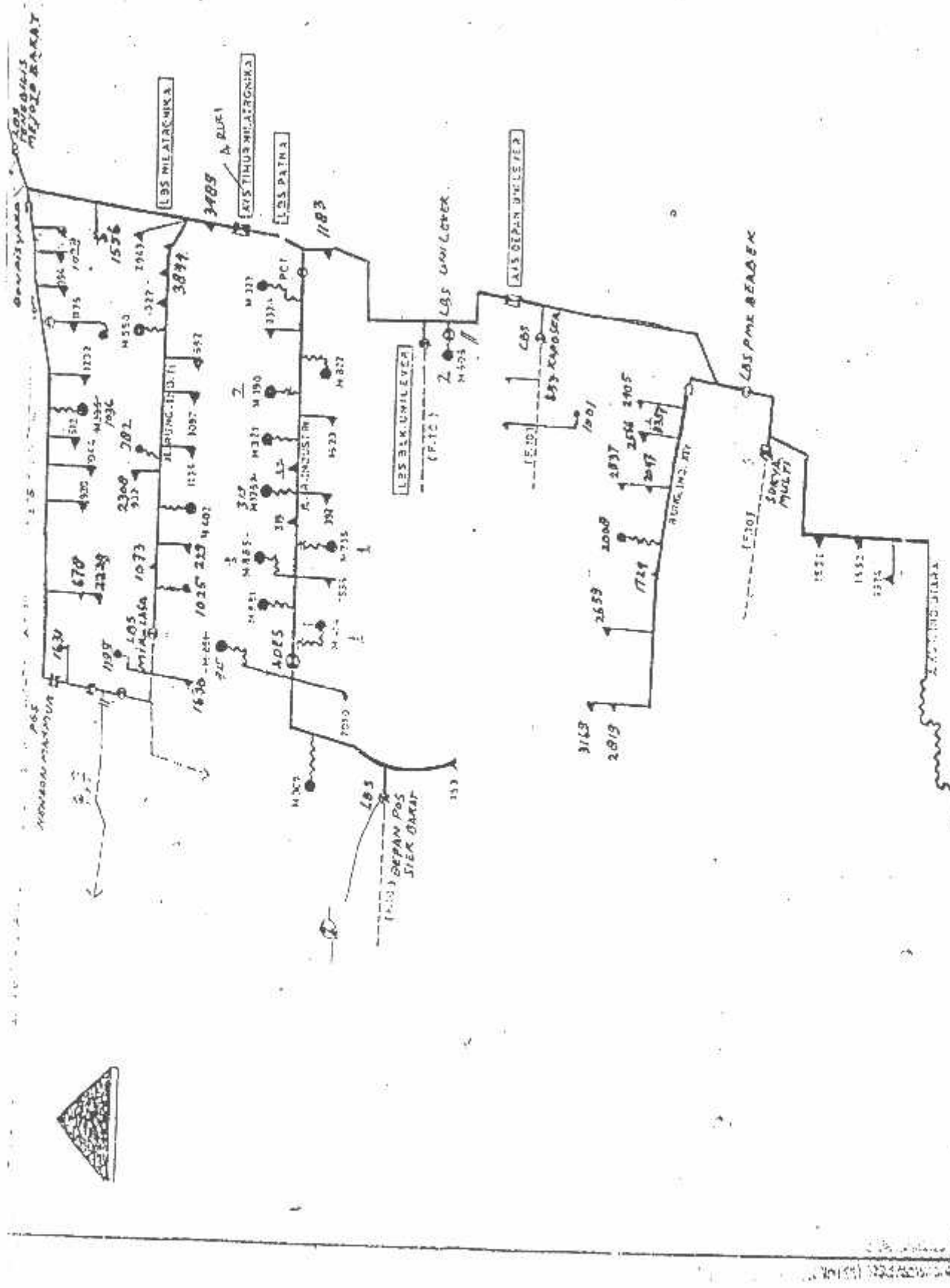
LAMPIRAN

I





SUKA



GIRUNGKUT

Indomie



LAPORAN DATA PEMBEBANAN RATA-RATA G.I. RUNGKUT BULAN DESEMBER 2007

No Bus	Tegangan awal		Tenggilis 2/12/2007		Type Bus
	V(p.u)	0	P (kW)	Q(kVAR)	
1	1	0.00	295.42	221.57	Slack
2	1	0.00	727.09	545.32	Load
3	1	0.00	458.91	334.18	Load
4	1	0.00	1448.44	1086.33	Load
5	1	0.00	1118.6	838.95	Load
6	1	0.00	0	0	Load
7	1	0.00	1320.8	990.6	Load
8	1	0.00	0	0	Load
9	1	0.00	435.97	326.97	Load
10	1	0.00	0	0	Load
11	1	0.00	0	0	Load
No Bus	Tegangan awal		Suik 2/12/2007		Type Bus
	V(p.u)	0	P (kW)	Q(kVAR)	
12	1.00	0.00	0	0	Slack
13	1.00	0.00	0	0	Load
14	1.00	0.00	0	0	Load
15	1.00	0.00	0	0	Load
16	1.00	0.00	1321.09	990.82	Load
17	1.00	0.00	668.94	501.71	Load
18	1.00	0.00	528.99	396.75	Load
19	1.00	0.00	0	0	Load
20	1.00	0.00	1379.86	1034.9	Load
21	1.00	0.00	0	0	Load
22	1.00	0.00	960.03	720.02	Load
23	1.00	0.00	677.34	508	Load
24	1.00	0.00	5597.82	4198.37	Load
25	1.00	0.00	3095.6	2321.7	Load
26	1.00	0.00	0	0	Load
27	1.00	0.00	0	0	Load
28	1.00	0.00	190.33	142.74	Load
No Bus	Tegangan awal		Indomie 2/12/2007		Type Bus
	V(p.u)	0	P (kW)	Q(kVAR)	
29	1.00	0.00	97.96	73.47	Slack
30	1.00	0.00	0	0	Load
31	1.00	0.00	0	0	Load
32	1.00	0.00	0	0	Load
33	1.00	0.00	0	0	Load
34	1.00	0.00	39.19	29.39	Load
35	1.00	0.00	25.19	18.89	Load
36	1.00	0.00	25.19	18.89	Load
37	1.00	0.00	0	0	Load
38	1.00	0.00	0	0	Load
39	1.00	0.00	926.44	694.83	Load
40	1.00	0.00	0	0	Load
41	1.00	0.00	71.09	53.32	Load
42	1.00	0.00	0	0	Load
43	1.00	0.00	2580.6	1935.45	Load
44	1.00	0.00	2079.59	1559.69	Load
45	1.00	0.00	856.47	642.35	Load



Laporan Data Impedansi P. Tenggilis

No. Saluran	Dari Bus	Ke Bus	Jarak (km)	Impedansi (pu)	
				R	X
1	1	2	0,6225	0.1345	0.2057
2	2	7	0,0943	0.0253	0.0318
3	2	3	0,4338	0.0937	0.1433
4	3	8	1,0186	0.2737	0.3438
5	3	9	0,3207	0.0862	0.1082
6	3	4	0,9526	0.2059	0.3148
7	4	10	0,9620	0.2585	0.3247
8	4	5	2,1881	0.473	0.7231
9	5	11	0,1886	0.0506	0.0636
10	5	12	0,0755	0.0202	0.0254
11	5	6	2,6691	0.577	0.8821

Laporan Data Impedansi P. Suik

No. Saluran	Dari Bus	Ke Bus	Jarak (km)	Impedansi (pu)	
				R	X
12	1	13	0,2746	0.0593	0.0907
13	13	21	1,0445	0.2807	0.3526
14	13	14	0,1184	0.0255	0.0391
15	14	22	0,0700	0.0198	0.0236
16	14	15	0,1238	0.0267	0.0409
17	15	23	0,4684	0.1258	0.1581
18	15	16	0,0431	0.0931	0.0142
19	16	24	0,1777	0.0477	0.0599
20	24	25	0,9529	0.2561	0.3216
21	16	17	0,4146	0.0886	0.137
22	17	26	0,7645	0.2054	0.258
23	17	18	0,1184	0.0255	0.0381
24	18	27	0,0485	0.013	0.0163
25	18	28	0,0485	0.013	0.0163
26	18	19	0,3661	0.0791	0.1236
27	19	29	0,0538	0.0144	0.0181
28	19	20	0,1615	0.0349	0.0533

Laporan Data Impedansi P. Incomie

Saluran	Dari Bus	Ke Bus	Jarak (km)	Impedansi (pu)	
				R	X
29	1	30	0,4502	0.0973	0.1487
30	30	39	0,0048	0.0129	0.0162
31	30	31	0,0718	0.0155	0.0237
32	31	40	0,5029	0.1351	0.1697
33	31	32	0,2347	0.0507	0.0775
34	32	41	0,0239	0.0642	0.0806
35	32	33	0,0910	0.0196	0.03
36	33	42	0,0239	0.0642	0.0806
37	33	34	0,0192	0.0415	0.0634
38	34	43	0,0144	0.0387	0.0486
39	34	35	0,1437	0.031	0.0474
40	35	44	0,8573	0.2304	0.2894
41	35	36	0,0910	0.0196	0.03
42	36	45	0,5268	0.1416	0.1778
43	36	37	0,1676	0.0362	0.0553
44	37	46	0,7088	0.1905	0.2392
45	37	38	0,0623	0.0134	0.0205

Data Tie Switch G.I. Rungkut

Saluran	Dari Bus	Ke Bus	Jarak (km)	Impedansi (pu)	
				R	X
47	9	26	0,3707	0.0986	0.1251
48	21	39	1,0945	0.2942	0.3695
49	20	46	0,7588	0.2039	0.2561

**Tegangan dan Sudut Fasa Tegangan Tiap Node
G.I Rungkut Sesudah Konfigurasi**

Bus	absV (pu)	sudV (deg)	Type Bus
1	1	0	Slack
2	0.99346	-0.13477	Load
3	0.98906	-0.22655	Load
4	0.98516	-0.30847	Load
5	0.97849	-0.44897	Load
6	0.97493	-0.52444	Load
7	0.99346	-0.13477	Load
8	0.98729	-0.25339	Load
9	0.98775	-0.24631	Load
10	0.9846	-0.31689	Load
11	0.97849	-0.44897	Load
12	0.97849	-0.44897	Load
13	0.9965	-0.07246	Load
14	0.995	-0.10393	Load
15	0.99342	-0.13692	Load
16	0.99082	-0.05633	Load
17	0.98907	-0.09297	Load
18	0.98876	-0.09961	Load
19	0.98806	-0.11468	Load
20	0.9879	-0.11819	Load
21	0.99414	-0.10945	Load
22	0.99346	-0.13477	Load
23	0.99283	-0.14579	Load
24	0.98934	-0.07845	Load
25	0.98226	-0.18532	Load
26	0.98623	-0.26925	Load
27	0.98876	-0.09961	Load
28	0.98876	-0.09961	Load
29	0.98805	-0.11488	Load
30	0.99621	-0.07844	Load
31	0.99572	-0.08842	Load
32	0.99441	-0.11582	Load
33	0.9939	-0.12646	Load
34	0.99283	-0.14861	Load
35	0.99204	-0.16522	Load
36	0.99181	-0.16991	Load
37	0.98699	-0.13191	Load
38	0.98699	-0.13191	Load
39	0.99612	-0.07974	Load
40	0.99511	-0.09758	Load
41	0.99441	-0.11582	Load
42	0.99387	-0.1268	Load
43	0.99283	-0.14861	Load
44	0.98912	-0.20914	Load
45	0.99037	-0.19162	Load
46	0.98701	-0.13155	Load

Rugi Daya Tiap Saluran G.I Rungkut Sebelum Konfigurasi

Dari	Ke	P	Q
		(MW)	(MVAR)
1	2	15.158	11.623
2	7	-0.321	-1.545
2	3	14.834	12.643
3	8	2.644	1.984
3	9	3.735	4.278
3	4	6.966	5.237
4	10	0.872	0.654
4	5	5.153	3.879
5	11	0.000	0
5	12	-0.000	0
5	6	2.240	1.682
1	13	24.329	17.498
13	21	1.464	1.303
13	14	22.818	16.124
14	22	0.321	1.545
14	15	22.472	14.541
15	23	1.921	1.441
15	16	20.504	13.075
16	24	12.623	9.504
24	25	11.228	8.438
16	17	7.834	3.55
17	26	2.463	0.374
17	18	2.724	1.187
18	27	0.000	0
18	28	0.000	0
18	19	1.386	0.183
19	29	0.381	0.285
19	20	-0.053	-0.896
1	30	14.843	11.848
30	39	1.297	0.768
30	31	13.325	10.895
31	40	1.853	1.39
31	32	11.461	9.489
32	41	-0.000	0
32	33	11.451	9.474
33	42	0.142	0.107
33	34	11.301	9.354
34	43	-0.000	0
34	35	11.291	9.339
35	44	5.167	3.879
35	36	6.034	5.386
36	45	4.162	3.122
36	37	1.818	2.222
37	46	1.767	2.182
37	38	-0.000	0
7	22	-0.321	-1.545
9	26	3.731	4.273
21	39	-1.297	-0.768
20	46	-0.053	-0.896

Rugi Daya Tiap Saluran GI Rungkut Sesudah Konfigurasi

Dari	Ke	P	Q
		(MW)	(MVAR)
1	2	17.971	13.584
2	7	-0.000	0
2	3	17.309	13.033
3	8	2.644	1.984
3	9	6.202	4.657
3	4	6.966	5.237
4	10	0.872	0.654
4	5	5.153	3.879
5	11	0.000	0
5	12	0.000	0
5	6	2.240	1.682
1	13	21.897	16.491
13	21	0.000	0
13	14	21.857	16.429
14	22	0.000	0
14	15	21.832	16.392
15	23	1.921	1.441
15	16	19.862	14.926
16	24	12.623	9.504
24	25	11.228	8.438
16	17	7.191	5.4
17	26	0.000	0
17	18	4.543	3.41
18	27	0.000	0
18	28	0.000	0
18	19	3.204	2.405
19	29	0.381	0.285
19	20	1.765	1.324
1	30	14.473	10.906
30	39	2.764	2.075
30	31	11.491	8.649
31	40	1.853	1.39
31	32	9.630	7.248
32	41	-0.000	0
32	33	9.624	7.239
33	42	0.142	0.107
33	34	9.476	7.123
34	43	-0.000	0
34	35	9.470	7.114
35	44	5.167	3.879
35	36	4.215	3.164
36	45	4.162	3.122
36	37	0.000	0
37	46	-0.050	-0.038
37	38	-0.000	0
7	22	0.000	0
9	26	6.195	4.648
21	39	-2.762	-2.073
20	46	1.764	1.323

Banyaknya Saluran = 49
 Banyaknya Bus = 46
 Banyaknya Iterasi = 3
 Rugi Total Saluran = 0.248 + j 0.337 MVA

```

unit uFitness;

interface

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

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

    TPopulasi=array of TIndividu;

    TFitness=class
    private
        FNbus,FNsal,FNLoop:integer;
        FVb:TBatas;
        FV,FSg,FSL:CxArr1;
        FCap:dArr1;
        FTypBus:iArr1;
        FZ,FTp:CxArr2;
        FLc,FTr:dArr2;
        FSwith:TSwithArr1;
        FBestIndi:TIndividu;
        function getIndividu(const rIndi:TIndividu):TIndividu;
        function CekStopRecomb(const rData:iArr1):boolean;
        function isNumeric(const rStr:Char):boolean;
        function isSameData(const rStr1,rStr2:string):boolean;
        function isSameDataList(var rList:TStringList;
            const rStr:string):boolean;
        function CalcVPinalty(const rV:CxArr1):integer;
        function getChrom:iArr1;
        function CalcFitness(const rData:iArr1):double;overload;
    public
        constructor Create(const rNLoop:integer);
        procedure CreateCombinasi(var rList:TStringList;
            var rCount:integer);
        function CalcFitness(const rData:bArr1):double;overload;
        function GetSwith(const rData:iArr1):TSwithArr1;
        function DecodeStrToData(const rStr:string):iArr1;
        function DecodeDataToStr(const rData:iArr1):string;
        property BestChrom:iArr1 read getChrom;
    end;

var gFit:TFitness;

```

implementation

```
constructor TFitness.Create(const rNLoop:integer);
begin
  inherited Create;
  FNLoop:=rNLoop;
  DecodeCommDataToLFData(gBus, FNbus, FNsal, FV, FSg, FSL, FCap, FTypBus,
  gBranch, FZ, FTp, FLC, FTr);
  FSwith:=InitSwith(FZ);
  FVb.min:=0.95;
  FVb.max:=1.05;
end;
```

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

```
function TFitness.GetSwith(const rData:iArr1):TSwithArr1;
var i,NData:integer;
begin
  NData:=high(rData)+1;
  SetLength(result, NData);
  for i:=0 to NData-1 do
    begin
      if (rData[i]<0) or (rData[i]>=FNsal) then
        begin
          raise Exception.Create('Melebihi Indeks Matrik!');
        end;
      result[i].dari:=FSwith[rData[i]].dari;
      result[i].ke:=FSwith[rData[i]].ke;
    end;
end;
```

```
function TFitness.CekStopRecomb(const rData:iArr1):boolean;
var i:integer;
begin
  result:=true;
  for i:=0 to FNLoop-1 do
    begin
      if rData[i]<>(FNsal-1) then
        begin
```

```
    result:=false;  
    break;  
end;  
end;  
end;
```

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

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

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

```

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

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

function TFitness.CalcVPinality(const rV:CxArr1):integer;
var i:integer;
begin
    result:=0;
    for i:=0 to FNbus-1 do
    begin
        if rV[i].real>1.05 then inc(result);
        if rV[i].real<0.95 then inc(result);
    end;
end;

```

```

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

function TFitness.CalcFitness(const rData:bArr1):double;
var Vpin:integer;
    V,Sg:CxArr1;
    Zb,Alir,Arus:CxArr2;
    TesRadial:boolean;
begin
    Zb:=BuatZBaru(FZ,rData);
    TesRadial:=TesJaringan(Zb);
    if TesRadial=true then
    begin
        V:=CopyMatrix(FV);
        Sg:=CopyMatrix(FSg);
        NewtonRaphson(gParamLF, V,Sg,FSL,FCap,FTypBus,
        Zb,FTp,Alir,Arus,FLc,FTr);
        Vpin:=CalcVPinalty(FV);
        result:=gParamLF.SumLoss.real+1000*Vpin;
    end
    else
    begin
        result:=(high(FV)+5)*1000;
    end;
end;

procedure TFitness.CreateCombinasi(var rList:TStringList;
    var rCount:integer);
var i:integer;
    CekRadial,CekCombinasi:boolean;
    str:string;

```

```

sw:TSwithArr1;
Zb:CxArr2;
Data:iArr1;
CekIndi:TIndividu;
begin
SetLength(Data,FNLoop);
for i:=0 to FNLoop-1 do
begin
Data[i]:=0;
end;
rCount:=0;
rList.Clear;
repeat
GenCombinasi(FNsal,Data,CekCombinasi);
if CekCombinasi=true then
begin
sw:=GetSwith(Data);
Zb:=BuatZBaru(FZ,sw);
CekRadial:=TesJaringan(Zb);
if (CekCombinasi=true) and (CekRadial=true) then
begin
str:=DecodeDataToStr(Data);
if isSameDataList(rList,str)=false then
begin
if rCount=0 then
begin
FBestIndi.chrom:=CopyMatrix(Data);
FBestIndi.fitness:=CalcFitness(Data);
end
else
begin
CekIndi.chrom:=CopyMatrix(Data);
CekIndi.fitness:=CalcFitness(Data);
if CekIndi.fitness<FBestIndi.fitness then
begin
FBestIndi:=getIndividu(CekIndi);
end;
end;
rList.Add(str);
inc(rCount);
end;
end;
end;
frmHasil.pbGen.StepBy(1);
until CekStopRecomb(Data)=true;
end;

function TFitness.getChrom:iArr1;

```

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

end.
```

```

unit uGenetic;

interface

uses uUtils,uFitness,uRandom,uHasil;

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

  TPopBin1=array of TIndiBin1;

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

  TGenRecon=class(TGenetic)
  private
    FNloop,FNFault,FNTabu:integer;
    FMin1,FAvg1,FMax1,FSumFitness:double;
    FParent,FChild,FTabuList:TPopBin1;
    FBestIndi:TIndiBin1;
    function getIndividu(const rIndi:TIndiBin1):TIndiBin1;
    function getBestChrom:bArr1;
    procedure RepairChrom(var rChrom:bArr1);
  end;

```

```

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

var gGAREcon:TGenRecon;

implementation

{ TGenetic }

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

```

```

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

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

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

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

{ TGenRecon }

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

```

```
FNTabu:=rNTabu;  
FNloop:=rNloop;  
FNfault:=rNFault;  
end;
```

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

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

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

```

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

```

```

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

```

```

function TGenRecon.isSameChrom(const rChrom1,rChrom2:bArr1):boolean;
var i:integer;
begin
result:=true;
for i:=0 to Length-1 do
begin
if rChrom1[i]<>rChrom2[i] then
begin
result:=false;
break;
end;
end;

```

```
end;  
end;
```

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

```
procedure TGenRecon.Statistik;  
var i:integer;  
begin  
  FMin1:=FParent[0].fitness;  
  FMax1:=FParent[0].fitness;  
  FSumFitness:=FParent[0].fitness;  
  for i:=1 to PopSize-1 do  
    begin  
      if FMin1>FParent[i].fitness then  
        begin  
          FMin1:=FParent[i].fitness;  
        end;  
      if FMax1<FParent[i].fitness then  
        begin  
          FMax1:=FParent[i].fitness;  
        end;  
    end;  
  end;
```

```

    end;
    FSumFitness:=FSumFitness+FParent[i].fitness;
end;
FAvg1:=FSumFitness/PopSize;
end;

```

```

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

```

```

function TGenRecon.Mutasi(const rAllele:boolean):boolean;
begin
    if FRandom.NextBoolean(PMutasi)=true then
        begin
            result:=not rAllele;
        end
    else
        begin
            result:=rAllele;
        end;
    end;
end;

```

```

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

```

```

end;
end
else
begin
for i:=0 to Length-1 do
begin
rChild1[i]:=Mutasi(rParent1[i]);
rChild2[i]:=Mutasi(rParent2[i]);
end;
end;
end;
end;

```

```

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

```

```

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

```

```

end;
for i:=0 to PopSize-1 do
begin
  FParent[i]:=getIndividu(tmpPop[i]);
end;
end;

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

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

procedure TGenRecon.DecodeIndiToData(const rIndi:TIndiBin1;
    var rData:iArr1;
    var rFitness:double);
var i,sa:integer;

```

```
begin
  SetLength(rData, FNloop);
  sa:=0;
  for i:=0 to Length-1 do
  begin
    if rIndi.chrom[i]=true then
    begin
      rData[sa]:=i;
      inc(sa);
    end;
  end;
  rFitness:=rIndi.fitness;
end;

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

end.
```

LAMPIRAN

II

Malang, 22 April 2006

Perihal : Permohonan dispensasi dalam pengajuan skripsi

Lampiran :

Kepada:

Yth. Ketua Jurusan Teknik Elektro

Di tempat.

Dengan hormat,

Sehubungan dengan adanya kendala dalam pengajuan skripsi saya, yang mengharuskan seluruh nilai mata kuliah dicantumkan, maka dengan ini saya yang bertandatangan di bawah ini:

Nama : Vincenzo Somado Vilton Tukan

Nim : 00.12.149

Jurusan : Teknik Elektro S-1


Kosentrasi : Teknik Energi Listrik

Memohon agar bapak berkenan mengizinkan saya untuk mengajukan skripsi dimana ada beberapa kegiatan mata kuliah yang merupakan prasyarat pengajuan skripsi yang sedang saya ikuti. Apabila nilai tersebut belum juga keluar, maka saya bersedia untuk menunda mengikuti yudisium.

Demikian surat permohonan ini atas perhatian dan kebijakan bapak, saya ucapkan banyak terima kasih.

Mengetahui

Ketua Jurusan Teknik Elektro


(Ir. F. Yudi Limpraptono, MT)
NIP. 7: 1039500274

Hormat saya,

Permohonan

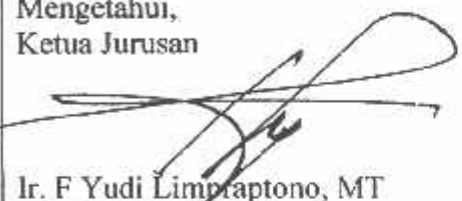


(Vincenzo Somado V. T.)
NIM: 00.12.149

Dasar Kurikulum Energi IT → diprogramkan sesuai dengan surat/2006



**LEMBAR PENGAJUAN JUDUL SKRIPSI
 JURUSAN TEKNIK ELEKTRO S-1**

Konsentrasi : Teknik Energi Listrik/~~Teknik Elektronika~~ *)

1.	Nama Mahasiswa : Vincenzo Somado Vilton Tukan	Nim : 00.12.149
2.	Waktu Pengajuan :	Tanggal : Bulan : 01 Tahun : 2007
Spesifikasi Judul (berilah tanda silang)		
3.	<input checked="" type="checkbox"/> a. Sistem Tenaga Elektrik <input type="checkbox"/> b. Energi & Konversi Energi <input type="checkbox"/> c. Tegangan Tinggi & Pengukuran <input type="checkbox"/> d. Sistem Kendali Industri	<input type="checkbox"/> e. Elektronika & Komponen <input type="checkbox"/> f. Elektronika Digital & Komputer <input type="checkbox"/> g. Elektronika Komunikasi <input type="checkbox"/> h. Lainnya
4.	Konsultasikan judul sesuai materi bidang ilmu kepada Dosen **): <i>Ir. Taufik Hidayat, MT</i>	Mengetahui, Ketua Jurusan  Ir. F Yudi Limpraptono, MT Nip. Y.1039500274
5.	Judul yang diajukan mahasiswa	OPTIMASI REKONFIGURASI SISTEM DISTRIBUSI RADIAL MENGGUNAKAN METODE MINIMAX GAMES WITH DEPTH FIRST SEARCH ALGORITHM (MGDFS)
6.	Perubahan Judul yang disetujui Dosen sesuai materi bidang ilmu
7.	Catatan :	
8.	Persetujuan Judul Skripsi yang dikonsultasikan kepada Kelompok Dosen materi bidang ilmu	Disetujui,.....,2007 Dosen  <i>Ir. H. TAUFIK HIDAYAT, MT</i>

Perhatian :

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



INSTITUT TEKNOLOGI NASIONAL MALANG
JL. RAYA KARANGLO KM 02
MALANG

Lampiran : 1 (satu) berkas
Pembimbing Skripsi

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

Yang bertandatangan di bawah ini:

Nama : Vincenzo Somado Vilton Tukan
Nim : 00. 12. 149
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Energi Listrik

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

**OPTIMASI REKONFIGURASI SISTEM DISTRIBUSI RADIAL
MENGUNAKAN METODE MINIMAX GAMES WITH DEPTH FIRST
SEARCH ALGORITHM (MGDFS)**

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

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

Malang, Februari 2007

Hormat saya,

Ketua
Jurusan Teknik Elektro S-1

Ir. F. Yudi Limpraptono, MT
Nip. Y.1039500274 *Yudi*


Vincenzo Somado Vilton T.
Nim: 00. 12. 149

*) coret yang tidak perlu

Form. S-3a



INSTITUT TEKNOLOGI NASIONAL MALANG
JL. RAYA KARANGLO KM 02
MALANG

Lampiran : 1 (satu) berkas
Pembimbing Skripsi

Kepada : Yth. Bpk Ir. Djojo Priatmono, MT
Dosen Institut Teknologi Nasional
MALANG

Yang bertandatangan di bawah ini:

Nama : Vincenzo Somado Vilton Tukan
Nim : 00. 12. 149
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Energi Listrik

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

**OPTIMASI REKONFIGURASI SISTEM DISTRIBUSI RADIAL
MENGUNAKAN METODE MINIMAX GAMES WITH DEPTH FIRST
SEARCH ALGORITHM (MGDFS)**

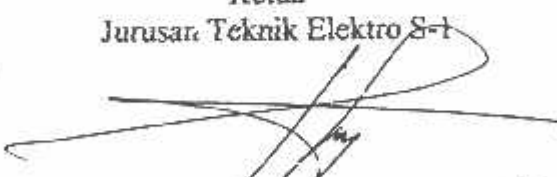
Adapun tugas tersebut sebagai salah satu syarat untuk menempuh Ujian Akhir Sarjana Teknik.


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

Malang, Februari 2007

Hormat saya,

Ketua
Jurusan Teknik Elektro S-1


Ir. F. Yudi Limpraptono, MT *BY*
Nip. Y.1039500274


Vincenzo Somado Vilton T.
Nim: 00. 12. 149

*) coret yang tidak perlu

Form. S-3a



PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa :

Nama : Vincenzo Somado Vilton Tukan

Nim : 00.12.149

Semester : 14 (Empat Belas)

Jurusan : Teknik Elektro S-1

Konsentrasi : Energi Listrik

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

**OPTIMASI REKONFIGURASI SISTEM DISTRIBUSI RADIAL
MENGUNAKAN METODE MINIMAX GAMES WITH DEPTH FIRST
SEARCH ALGORITHM (MGDFS)**

Demikian surat Pernyataan ini kami buat agar dapat digunakan seperlunya.

Malang, Februari, 2007

Kami yang Membuat pernyataan,

Ir. H. Taufik Hidayat, MT
NIP. P. 101 8700 015

Catatan:

Setelah disetujui agar formulir ini
Diserahkan mahasiswa/i yang bersangkutan
Kepada Jurusan untuk diproses lebih lanjut.

*) Coret yang tidak perlu

Form.S-3b

PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa :

Nama : Vincenzo Somado Vilton Tukan

Nim : 00.12.149

Semester : 14 (Empat Belas)

Jurusan : Teknik Elektro S-1

Kosentrasi : Energi Listrik

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

**OPTIMASI REKONFIGURASI SISTEM DISTRIBUSI RADIAL
MENGUNAKAN METODE MINIMAX GAMES WITH DEPTH FIRST
SEARCH ALGORITHM (MGDFS)**

Demikian surat pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Malang, Februari 2007

Kami yang membuat pernyataan,



Ir. Djojo Priatmono, MT
NIP. 101 8500 107

Catatan :

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



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

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

PERSEROJ MALANG
< 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, 14 Februari 2008

No/nor : ITN-124/1.TA/2/2008
Lampiran : -
Perihal : **BIMBINGAN SKRIPSI**

Kepada : Yth. Sdr. **Ir.H TAUFIK HIDAYAT, MT ***)

Dosen Pembimbing
Jurusan Teknik Elektro S-1
di
Malang

Dengan Hormat,
Sesuai dengan permohonan dan persetujuan dalam proposal Skripsi untuk Mahasiswa:

Nama : VINSENDO SOMADO VILTON T
Nim : 0012149
Fakultas : Teknologi Industri
Jurusan : Teknik Elektro S-1
Kosentrasi : **Teknik Energi Listrik S-1**

Maka dengan ini pembimbingan Skripsi tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu **6 (enam) bulan**, terhitung mulai tanggal

21 Agustus 2007 s/d 21 Februari 2008.

Adapun tugas tersebut merupakan salah satu syarat untuk memperoleh gelar Sarjana Teknik, Jurusan Elektro apabila lewat dari batas waktu tsb. Maka, Skripsinya akan digugurkan.

Demikian atas perhatian serta kerjasamanya saya ucapkan terima kasih



Letua Jurusan
Teknik Elektro S-1

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

Tembusan Kepada Yth:

1. *) Perpanjangan
2. Mahasiswa Yang Bersangkutan
3. Arsip

Form S4a



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

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

PERSERU MALANG
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, 14 Februari 2008

Nomor : ITN-125/LTA/2/2008
Lampiran : -
Perihal : **BIMBINGAN SKRIPSI**

Kepada : Yth. Sdr. Ir. **DJOJO PRIATMONO, MT ***)

Dosen Pembimbing
Jurusan Teknik Elektro S-1
di
Malang

Dengan Hormat,
Sesuai dengan permohonan dan persetujuan dalam proposal Skripsi untuk Mahasiswa:

Nama : VINSENDO SOMADO VILTON T
Nim : 0012149
Fakultas : Teknologi Industri
Jurusan : Teknik Elektro S-1
Konsentrasi : **Teknik Energi Listrik S-1**

Maka dengan ini pembimbingan Skripsi tersebut kami serahkan sepenuhnya kepada Saudara/l selama masa waktu **6 (enam) bulan**, terhitung mulai tanggal

21 Agustus 2007 s/d 21 Februari 2008

Adapun tugas tersebut merupakan salah satu syarat untuk memperoleh gelar Sarjana Teknik, Jurusan Elektro apabila lewat dari batas waktu tsb. Maka, Skripsinya akan digugurkan.

Demikian atas perhatian serta kerjasamanya yang baik kami ucapkan terima kasih



Ketua Jurusan
Teknik Elektro S-1

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

Tembusan Kepada Yth:

- *) Perpanjangan
- Mahasiswa Yang Bersangkutan
- Arsip

Form S4a



FORMULIR BIMBINGAN SKRIPSI

Nama : Vincenzo Somado Vilton Tukan
Nim : 00.12.149
Masa Bimbingan : 21 Agustus 2007 s/d 21 Februari 2008
Judul Skripsi : OPTIMASI REKONFIGURASI SISTEM DISTRIBUSI RADIAL
MENGUNAKAN METODE MINIMAX GAMES WITH DEPTH
FIRST SEARCH ALGORITHM (MGDFS)
DI G. L. RUNGKUT - SURABAYA

No.	Tanggal	Uraian	Paraf Pembimbing
1.	16/02 08	- Beri penjelasan tentang penyertan feeder.	
2.	19/02 08	- Single line diagram busbar G. I Rungkut - data beban - proses state data.	
3.	21/02 08	- buat tabel hand schedule - dari data beban ke model konfigurasi.	
4.	21/02 08	- lampirkan proses konfigurasi dan "tabelan" yang sudah	
5.		- "file" mengolah" semua data	
6.			
7.			
8.			
9.			
10.			

Malang,
Dosen Pembimbing I,

Ir. H. Taufik Hidayat, MT
NIP. Y. 101 8700 015



FORMULIR BIMBINGAN SKRIPSI

Nama : Vincenzo Somado Vilton Tukan
Nim : 00.12.149
Masa Bimbingan : 21 Agustus 2007 s/d 21 Februari 2008
Judul Skripsi : OPTIMASI REKONFIGURASI SISTEM DISTRIBUSI RADIAL
MENGGUNAKAN METODE MINIMAX GAMES WITH DEPTH
FIRST SEARCH ALGORITHM (MGDFS)
DI G. I. RUNGKUT - SURABAYA

No.	Tanggal	Uraian	Paraf Pembimbing
1.	14/02, 08	- Beri penjelasan tentang pengertian Feeder.	
2.	19/02, 08	- Single Line Diagram lengkap G.I Rungkut - Data lapangan	
3.			
4.			
5.			
6.			
7.			
8.			
9.			
10.			

Malang,
Dosen Pembimbing II,

Ir. Djojo Priatmono, MT
NIP. Y. 101 8500 107

Form.S-4b



INSTITUT TEKNOLOGI NASIONAL
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

Formulir Perbaikan Ujian Skripsi

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

NAMA

Vincenzo Somado Vilton Tukan

NIM

0012149

Perbaikan meliputi

Uji validasi program (test program) ?
dengan metode, data dan hasil yang
mendekati sama (referensi dari jurnal).

Malang,



Formulir Perbaikan Ujian Skripsi

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

NAMA

Vincenzo S

NIM

00.12.149

Perbaikan meliputi

- abstrak disempurnakan
- Daftar pustaka di perbaiki.
- perbaikan kelengkapan data higen
- hasil analisa aliran daya setelah konfigurasi pemadatan daya konfigurasi jaringan yg ada! km ada beberapa menjadi nol
- Revisikan tabel & gbr konfigurasi apa kompresasi!

Malang,

19 Desember 2008

(Wane Kus S, ST, MT



PERSETUJUAN PERBAIKAN SKRIPSI

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

Hari : Rabu
Tanggal : 19 Maret 2008

Telah dilakukan perbaikan skripsi oleh:

1. Nama : Vincenzo Somado Vilton Tukan
2. NIM : 00.12.149
3. Jurusan : Teknik Elektro
4. Konsentrasi : Teknik Energi Listrik S-1
5. Judul Skripsi : OPTIMASI REKONFIGURASI SISTEM DISTRIBUSI RADIAL 20 kV MENGGUNAKAN METODE *MINIMAX GAMES WITH DEPTH FIRST SEARCH ALGORITHM* (MGDFS) PADA GARDU INDUK RUNGKUT - SURABAYA

Perbaikan meliputi:

No	Materi Perbaikan	Ket
1.	Uji validasi program (test program), Dengan metode, data dan hasil yang mendekati sama (referensi dari jurnal).	

Diperiksa/Disetujui
Dosen Penguji

Ir. Yusuf Ismail Nakhoda, MT
Nip. Y. 101 8800 189

Dosen Pembimbing I

Ir. H. Taufik Hidayat, MT
NIP. Y. 101 8700 015

Dosen Pembimbing II

Ir. Djoko Priatmono, MT
NIP. Y. 101 8500 107



PERSETUJUAN PERBAIKAN SKRIPSI

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

Hari : Rabu
Tanggal : 19 Maret 2008

Telah dilakukan perbaikan skripsi oleh:

1. Nama : Vincenzo Somado Vilton Tukan
2. NIM : 00.12.149
3. Jurusan : Teknik Elektro
4. Konsentrasi : Teknik Energi Listrik S-1
5. Judul Skripsi : OPTIMASI REKONFIGURASI SISTEM DISTRIBUSI RADIAL 20 kV MENGGUNAKAN METODE MINIMAX GAMES WITH DEPTH FIRST SEARCH ALGORITHM (MGDFS) PADA GARDU INDUK RUNGKUT - SURABAYA

Perbaikan meliputi:

No	Materi Perbaikan	Ket
1.	Abstrak disempurnakan.	
2.	Daftar pustaka.	
3.	Sesuaikan kesimpulan dengan tujuan.	
4.	Hasil analisa aliran daya setelah konfigurasi sesuaikan dengan konfigurasi jaringan yang ada, karena ada beberapa menjadi nol.	
5.	Kata kompensasi diganti dengan kata konfigurasi pada penulisan tabel dan gambar.	

Diperiksa/Disetujui
Dosen Penguji

Irrine Budi S. ST, MT
NIP. 132314400

Dosen Pembimbing I

Ir. H. Taufik Hidayat, MT
NIP. Y. 101 8700 015

Dosen Pembimbing II

Ir. Djojo Priatmono, MT
NIP. Y. 101 8500 107

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Dari hasil analisis berdasarkan hasil program rekonfigurasi pada sistem jaringan distribusi radial 20 kV dengan menggunakan metode *Minimax Games Depth First Search* (MGDFS), maka penentuan letak kombinasi *switch Normally Open* (NO) adalah dari bus 13 ke bus 21, bus 14 ke bus 22, bus 17 ke bus 26, dan bus 36 ke bus 37. Total rugi-rugi daya turun dari $0.281 + j 0.365$ MVA menjadi $0.248 + j 0.337$ MVA, sehingga reduksi rugi-rugi dayanya $0.033 + j 0.028$ MVA maka didapat prosentasi reduksi rugi-rugi daya sebesar 11.74 %.

5.2. Saran

Penggunaan metode *Minimax Games Depth First Search* (MGDFS) untuk menyelesaikan permasalahan rekonfigurasi jaringan masih perlu dikembangkan lagi terhadap sistem jaringan yang lebih luas dan penggunaan waktu komputasi yang lebih cepat sehingga nantinya dapat digunakan lebih efektif dan efisien dalam pemecahan permasalahan rekonfigurasi jaringan.