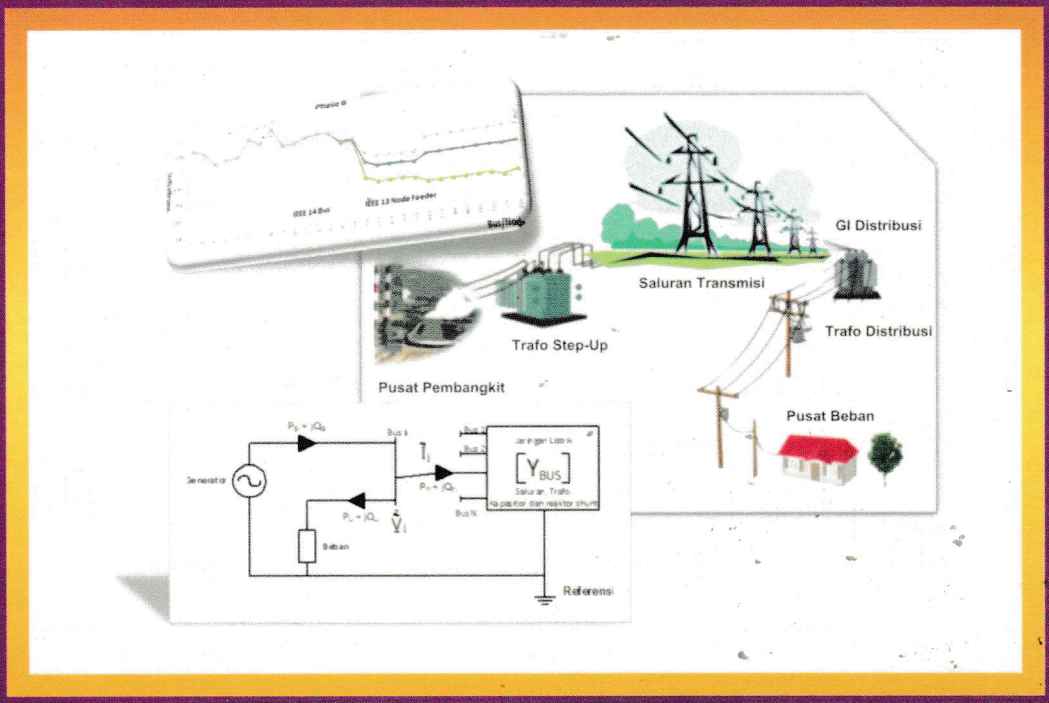
ANDALAS UNIVERSITY Press



KOMPUTASI SISTEM TENAGA

DENGAN PEMOGRAMAN VISUAL C++

Syafii, ST, MT, Ph.D

Komputasi Sistem Tenaga

Dengan Pemograman Visual C++

Penulis : Syafii, ST, MT, Ph.D

Ilustrasi Sampul dan Penata Isi :
Dyans Fahrezionaldo
Safri Y
Dana Prasetyo

Andalas University Press
Jl. Situjuh No. 1, Padang 25129, Telp/Faks. : 0751-27066
email : cebitunand@ymail.com
facebook : AU Press (Andalas University Press)

Anggota :
Asosiasi Penerbit Perguruan Tinggi Indonesia (APPTI)

Cetakan :
I. Padang, 2015

ISBN : 978-602-8821-78-0

PRAKATA

Dengan memanjatkan puji dan syukur kehadirat Allah S.W.T yang telah memberikan karunianya, sehingga buku Komputasi Sistem Tenaga dengan Pemograman Visual C++ ini dapat terselesaikan.

Buku Komputasi Sistem Tenaga ini dapat digunakan sebagai buku pengangan mata kuliah penggunaan komputer dalam sistem tenaga dan buku pendukung mata kuliah analisa sistem tenaga. Mata kuliah penggunaan komputer dalam sistem tenaga merupakan salah satu mata kuliah pilihan yang banyak diajarkan di Jurusan Teknik Elektro khususnya untuk konsentrasi Teknik Tenaga Listrik (TTL). Dengan komputasi persoalan yang rumit dibidang sistem tenaga dapat diselesaikan dengan mudah menggunakan komputer. Permasalahan yang penting dalam komputasi sistem tenaga adalah menemukan algoritma yang akurat dalam memecahkan persoalan dan pembuatan pemograman komputer yang effisien. Dalam buku ini akan disajikan teknik penyelesaian aliran daya dan analisa hubung singkat menggunakan pemograman Visual C++ serta teknik pemogramman berorientasi objek.

Pada kesempatan ini penulis menyampaikan terimakasih kepada Fakultas Teknik Universitas Andalas yang telah memfasilitasi penerbitan buku ini. Ucapan terima kasih juga disampaikan kepada semua pihak yang telah membantu baik langsung maupun tidak langsung penyelesaian penulisan dan penerbitan buku ini. Akhir kata kami berharap semoga buku ini bermanfaat bagi kita sebagai akademisi khususnya dan bagi yang pembaca pada umumnya. Saran dan kritik yang sifatnya membantu kelengkapan dan menuju kesempurnaan untuk edisi selanjutnya dari semua pihak sangat diharapkan.

1 Februari 2015

Syafii, PhD

DAFTAR ISI

KATA PENGANTAR

Dengan memanjatkan puji dan syukur kehadirat Allah S.W.T yang telah memberikan karunianya, sehingga buku Pemograman Komputer dalam Analisa Sistem Tenaga dapat terselesaikan.

Buku Pemograman Komputer dalam Analisa Sistem Tenaga ini dapat digunakan sebagai buku pengangan mata kuliah penggunaan komputer dalam sistem tenaga dan buku pendukung mata kuliah analisa sistem tenaga. Mata kuliah penggunaan komputer dalam sistem tenaga merupakan salah satu kelompok mata kuliah pilihan yang banyak diajarkan di Jurusan Teknik Elektro untuk konsentrasi Teknik Tenaga Listrik (TTL). Dengan komputasi persoalan yang rumit dibidang sistem tenaga dapat dianalisa dengan mudah menggunakan komputer. Permasalahan yang penting dalam komputasi sistem tenaga adalah menemukan algoritma yang akurat dalam memecahkan persoalan dan pembuatan pemograman komputer yang effisien. Dalam buku ini akan disajikan penyelesaian aliran daya, hubung singkat dan teknik pemogramman berorientasi objek yang dilengkapi dengan algoritma dan program komputer berbasis Visual C++. Pada kesempatan ini penulis menyampaikan terimakasih kepada Fakultas Teknik Universitas Andalas yang telah memfasilitasi penerbitan buku ajar ini. Ucapan terima kasih juga disampaikan kepada semua pihak yang telah membantu baik langsung maupun tidak langsung penyelesaian penulisan dan penerbitan buku ini.

Akhir kata kami berharap semoga buku ini bermanfaat bagi kita sebagai akademisi khususnya dan bagi yang pembaca pada umumnya. Saran dan kritik yang sifatnya membantu kelengkapan dan menuju kesempurnaan untuk edisi selanjutnya dari semua pihak sangat diharapkan.

1 Februari 2015

Syafii, PhD

BAB I
PENDAHULUAN

1.1 Komputasi Sistem Tenaga

Penerapan teknologi digital dalam bidang tenaga listrik terus meningkat dalam rangka meningkatkan kehandalan operasi, mengurangi biaya dan meningkatkan efisiensi dengan konsep baru yang dikenal smart grid. Analisis sistem distribusi akan berhadapan dengan sistem grid besar, aplikasi real-time, perulangan aliran daya seperti dalam analisis kontingensi, aliran daya harmonik dan aplikasi interaktif lainnya. Oleh karena itu diperlukan perangkat analisis sistem tenaga listrik yang lebih akurat dan komprehensif untuk menilai benar dan memprediksi status sekarang dan masa depan dari sistem dalam rangka untuk membuat keputusan kontrol yang tepat. Kebutuhan tersebut telah melahirkan minat para peneliti untuk menerapkan teknologi inovatif dalam perhitungan sistem tenaga dan pemodelannya seperti pemrograman paralel, web based programming dan pemograman berorientasi objek.

Analisis aliran daya merupakan jantung dari sebagian besar kegiatan perencanaan sistem dan desain untuk ekspansi masa depan serta dalam menentukan operasi terbaik dari sistem tenaga yang ada. Analisis aliran daya sudah mulai dieksplorasi sejak diperkenalkannya komputer digital. Sebagian besar tantangan dalam algoritma aliran daya telah dapat diselesaikan. Akan tetapi dengan perkembangan teknik analisa sistem tenaga dan teknologi komputasi, isu-isu baru masih perlu peningkatan melalui kajian pendekatan baru dan arah penyelesaian baru. Oleh karena itu, algoritma dan teknologi komputasi sistem tenaga perlu ditingkatkan dan akan selalu ada perkembangan baru dan daerah baru bagi para peneliti untuk mengeksplornya.

Selanjutnya pada bab ini akan dibahas teknik pemograman terakhir yang banyak digunakan dalam membangun program komputasi sistem tenaga listrik yaitu: pemograman berbasis web, pemograman berorientasi objek dan penggunaan parallel processing dalam pemograman.

1.2 Pemograman berbasis Web

Kebutuhan operasional dan komersial industri tenaga memerlukan sistem informasi untuk tidak hanya melakukan fungsi tradisional tetapi juga mendukung

banyak fungsi baru, khususnya untuk memenuhi kebutuhan persaingan dengan deregulasi. Pesatnya perkembangan Internet dan komputasi terdistribusi telah membuka pintu bagi solusi layak dan hemat biaya (Rong-Ceng Leou, 2002). Remote monitoring dan kontrol merupakan salah satu aplikasi yang paling menjanjikan dari internet sebagai media yang tepat untuk menyediakan akses remote. Banyak program aplikasi telah dialihkan kepada platform baru dan analisis aliran daya berbasis Web menjadi salah satu darinya.

Dalam referensi (S. Chen, Jan 2002) menggambarkan aliran daya, paket analisis berbasis Web dengan platform-independen. Tiga tingkatan arsitektur yang mengandung client tier, tingkat menengah, dan data tingkat diadopsi dalam sistem ini. Ini memasok klien pengguna dengan user interface yang independen yang fleksibel dan portabel, dan desain ini dapat berbagi paket ini dengan komputer di Internet. Konsep desain telah diuji di bawah sistem tenaga enam bus dan menunjukkan hasil yang baik. Algoritma dan perangkat lunak masih perlu meningkatkan dalam rangka untuk mempercepat waktu komputasi dan aplikasi user friendly.Setiap komputer dapat melakukan operasi-operasi dasar dalam pemrograman seperti operasi pembacaan data, operasi perbandingan, operasi aritmetika, dan sebagainya. Perkembangan teknologi komputer hanya merubah kecepatan, biaya, atau tingkat ketelitian tidak mengubah operasi-operasi dasar tersebut.

1.3 Pemograman Berorientasi objek

Sebelum diperkenalkan pemrograman berorientasi objek (OOP), perangkat lunak dibangun menggunakan pendekatan pemrograman struktural. Namun, pendekatan ini memiliki banyak kelemahan seperti biaya pembanguna tinggi, produktivitas yang rendah, kualitas perangkat lunak tidak terkendali, dan risiko untuk pindah ke teknologi baru [X. Cai, 2002]. OOP datang untuk menyelesaikan masalah ini di mana menggunakan pengembangan alamiah dari pemrograman terstruktur. Ciri utama dari pemrograman berorientasi obyek adalah pengenalan objek untuk menggantikan fungsi di mana objek memungkinkan kita untuk merangkum data dan fungsi. Hal ini memungkinkan fungsi untuk berperilaku berbeda tergantung pada parameter yang disediakan dan memungkinkan operator untuk membuat perangkat lunak yang lebih efisien [Jacobson, 1997]. Menggunakan kembali objek yang sama sangat mengurangi waktu pengembangan software dan meningkatkan produktivitas. Jacobson mengklasifikasikan objek- objek ke dalam tiga jenis [Jacobson, 1997]:

i. Objek entitas yang mewakili benda-benda di dunia nyata,

ii. Objek kontrol yang dibuat untuk menangani operasi kompleks dalam

iii. Objek antarmuka yang menangani pertukaran data dengan pengguna atau sistem lain.

Aplikasi pertama dari OOP untuk mengembangkan perangkat lunak analisis sistem tenaga dilaporkan oleh (Neyer A. F,1990). Dalam perngembangan tersebut, fitur dasar OOP diperkenalkan untuk membangun program aliran daya. Fokus utama adalah prinsip-prinsip desain OOP dan implementasi praktis untuk sistem tenaga listrik. Model sistem tenaga dan operasi matriks jarang menggunakan OOP dipaparkan dalam (Hakavik B, 1994). Dari hasil tes numerik bahwa penerapan OOP cukup efisien dan dapat menjadi struktur standar rutin perbendaharaan numerik. Akan tetapi fitur OOP yang digunakan untuk mendapatkan fleksibilitas tersebut tidak signifikan meningkatkan waktu eksekusi.

Aplikasi yang komprehensif untuk teknologi objek komponen telah dilaporkan oleh Nor K. M, 2004. Dalam pengembangan tersebut, tool untuk memvisualisasi komponen juga dijelaskan. Tool tersebut dikembang dari integrasi banyak perangkat lunak berbasis komponen dengan memasukkan engine analisis dan komponen interface yang user frienly. Selanjutnya Mamdouh A Akher, 2005, mengembangkan aplikasi aliran daya tiga phasa dengan menggunakan objek komponen aliran daya satu satu phasa sebelumnya berbasis komponen simetris. Penelitian pengembangan aplikasi analisa sistem tenaga berbasis OOP terus berlanjut, terutama dalam menemukan model objek baru komponen sistem tenaga dan teknologi visualisasi yang user frienly.

1.4 Pemograman Parallel

Komputasi sistem tenaga melibatkan perangkat keras dan perangkat lunak komputer. Beberapa tahun terakhir, teknologi komputer berkembang dengan cepat. Performansi mikroprosesor meningkat 52% per tahun sejak 1986 sampai 2002 [Callahan D, 2008]. Sekarang, performansi mikroprosesor meningkat dengan penambahan prosesor pada mesin komputer yang sama yang dikenal dengan sistem *multi-core*. Mesin dengan multi-prosesor sekarang telah menjadi standar semenjak kecepatan prosesor tunggal mendekati stabil atau meningkat sangat lambat dibandingkan dengan perkembagan yang lalu. Oleh karena itu, melihat kecenderungan perkembangan teknologi komputer saat ini, peningkatan performansi akan dapat

dicapai dengan cara menjalankan program pada banyak prosesor secara parallel. Dengan kata lain, pendekatan banyak prosesor akan bermanfaat jika software dapat menjalankan banyak aktifitas pada priode waktu yang sama.

Aplikasi analisa sistem tenaga harus dapat dijalankan serentak untuk mendapatkan manfaat dari perkembangan teknologi ini. Sayangnya, masih sangat sulit untuk mendapatkan algoritma yang benar-benar memberikan keuntungan dari banyak prosesor tersebut. Sebagian besar aplikasi analisa sistem tenaga berjalan menggunakan prosesor *single core*, sehingga tidak terlihat penambahan kecepatan meskipun dijalankan pada mesin multi-core. Sebenarnya program tersebut tetap berjalan dalam mode prosesor tunggal. Oleh karena itu, algoritma perlu diubah untuk mendapatkan manfaat dari perkembangan baru dalam teknologi komputer tersebut.

Beberapa aplikasi menggunakan pemrosesan paralel dalam perhitungan sistem tenaga telah dilakukan, seperti dengan menggunakan prosesor yang saling terinterkoneksi [Zang, 1996] dan klater komputer pribadi (PC) yang terhubung melalui komunikasi Ethernet [Tu F, 2002]. Sistem paralel seperti ini membutuhkan biaya yang mahal untuk menggunakannya dan waktu perhitungan juga tergantung pada kecepatan media komunikasi yang digunakan antara prosesor. Tingginya biaya hardware telah membuat keuntungan dari solusi paralel lebih cepat tidak berharga dan tidak praktis. Alternatif lain untuk mengurangi biaya dan waktu komunikasi adalah dengan menggunakan prosesor multi-core di komputer yang sama yang dikenal sebagai sistem paralel berbasis PC. Dalam waktu dekat semua komputer baru merupakan komputer paralel. Ini juga berarti bahwa biaya hardware adalah sama, apakah dasar PC standar digunakan sebagai sistem pengolahan serial atau sistem pemrosesan paralel.

Kinerja percepatan prosesor multi-core tergantung pada algoritma dan perangkat lunak. Permasalah harus didekomposisi menjadi tugas kecil dalam algoritma pemrograman paralel. Tugas-tugas ini dapat bekerja secara terpisah dari yang lain dan berjalan di bawah beberapa sistem prosesor. Masalah yang tidak bisa diurai menjadi tugas-tugas yang terpisah akan menggunakan teknik paralel loop. Kedua teknik paralel digunakan dalam pengembangan algoritma untuk mempercepat perhitungan analsia sistem tenaga. Teknologi komputasi paralel baru ini dapat memecahkan perhitungan sistem listrik secara lebih efisien.

BAB II
DASAR PEMOGRAMAN C++

2.1 Pendahuluan

Saat ini terdapat banyak bahasa pemrograman seperti bahasa pascal, basic dan bahasa C. Pengguna bahasa pemrograman C dan C++ sangatlah banyak karenakan kemampuan bahasa C yang dianggap bisa dipakai dalam banyak bidang termasuk rekayasa dan terapan termasuk diantaranya pembuatan aplikasi. Kelebihan bahasa pemorgraman C++ adalah kemampuan untuk melakukan pemrograman berorientasi objek (Object Oriented Programming atau OOP). Pada OOP, data dan instruksi dibungkus (encapsulation) menjadi satu. Kesatuan ini disebut kelas (class) dan inisiasi kelas pada saat run-time disebut objek (object). Data di dalam objek hanya dapat diakses oleh instruksi yang ada didalam objek itu saja.

Setiap komputer dapat melakukan operasi-operasi dasar dalam pemrograman seperti operasi pembacaan data, operasi perbandingan, operasi aritmetika, dan sebagainya. Perkembangan teknologi komputer hanya merubah kecepatan, biaya, atau tingkat ketelitian tidak mengubah operasi-operasi dasar tersebut. Bahasa pemograman C++ termasuk dalam katagori bahasa pemograman tingkat menengah yang perlu diterjemahkan terlebih dahulu oleh sebuah translator bahasa (yang disebut kompilator atau compiler) ke dalam bahasa mesin sebelum akhirnya dieksekusi oleh CPU. Setiap instruksi dalam bahasa mesin merupakan kombinasi dari operasi dasar yang bersesuaian, dan menghasilkan efek yang sama pada setiap komputer. Oleh karena itu algoritma merupakan hal penting untuk dapat membuat kode pemograman komputer.

Struktur dasar pemograman C++ adalah:

```
#include
main()
{
Perintah-perintah berdasarkan algoritma pemograman;
return 0;
}
```

Selanjutnya pada bab ini akan dibahas dasar-dasar pemograman C++ yang akan digunakan dalam membuat aplikasi komputasi sistem tenaga.

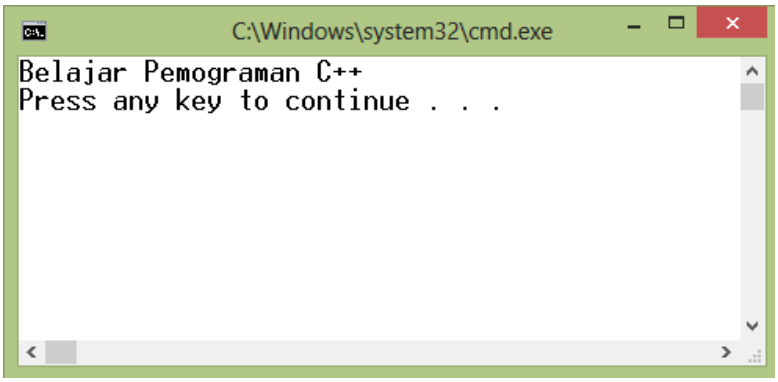
2.2 Perintah Masukan dan Keluaran

Masukan data dari keyboard menggunakan perintah cin>> , sedangkan keluaran ke tampilan dos (*console*) menggunakan perintah cout<<. Library yang digunakan iostream.h ditambahan dengan pernyataan using namespace std;. Iostream adalah salah satu header file yang digunakan untuk fungsi input dan output
yang ada di C++ dan using namespace std adalah perintah yang digunakan untuk mendeklarasikan/memberitahukan kepada compiler C++ bahwa kita akan menggunakan semua fungsi/class/file yang terdapat dalam namespace std.

Perhatikan contoh berikut untuk menampilkan kalimat Belajar Pemograman C++ ke tampilan dos menggunakan pemograman Ms Visual C++ 2010.

```
#include "stdafx.h"
#include <iostream>
using namespace std;

int _tmain(int argc, _TCHAR* argv[])
{
	cout<<"Belajar Pemograman C++"<<endl;
	return 0;
}
```

Apabila program ini dijalankan akan diperoleh hasil berikut:



Gambar 2.1 Tampilan hasil running program

Untuk memperbaiki tampilan numerik gunakan library iomanip.h dengan beberapa sintaks penting berikut:

Contoh:

double f =3.14159

cout << setprecision(5) << f ;

Apabila program dijalankan hasilnya adalah:

3.1416

Selanjutnya mengatur jumlah digit numerik yang ditampilakan dapat dilakukan dengan:

cout << setw(10);

cout << 77 << endl;

Apabila program dijalankan hasilnya adalah:

77

Perintah berikut juga dapat digunakan untuk mencetak hasil tetap dengan lebar 5 digit dan rata kanan.

cout <<fixed <<setw(5) << right;

Untuk data yang banyak, perintah membaca data dari keyboard menjadi tidak efektif dan akan memerlukan waktu yang lama untuk proses input data. Alternatif lain adalah melakukan proses pembacaan data dari text file atau file .txt. Perintah untuk membaca data dari file .txt adalah fin>> dan keluaran ke file .txt menggunakan perintah fout<<. Perhatikan contoh berikut:

Data saluran suatu sistem tenaga sebagai berikut:

```
7 5
1 2 0.02 0.06 0.03
1 3 0.08 0.24 0.025
2 3 0.06 0.18 0.02
2 4 0.06 0.18 0.02
2 5 0.04 0.12 0.015
3 4 0.01 0.03 0.01
4 5 0.08 0.24 0.025
```

Data tersebut disusun mulai dari jumlah saluran dan jumlah bus, selanjutnya di bawah diberikan data parameter saluran sebanyak tujuh saluran dan simpan dalam file data5bus.txt. Maka untuk dapat membaca data tersebut, terlebih dahulu siapkan variabel untuk menampung data tersebut seperti berikut:

```
int i,Nb,Nl,sl[50],el[50];
double rl[50],xl[50],bl[50];
```

dimana:

- Variabel integer Nl dan Nb masing-masing untuk menampung jumlah saluran dan jumlah bus.
- Variabel array satu dimensi dengan tipe data integer untuk data bus kirim dan terima sl[50] dan el[50].
- Variabel array satu dimensi dengan tipe data double untuk reistansi, reaktansi dan suseptansi saluran: rl[50],xl[50],bl[50].

Selanjutnya gunakan perintah fin>> untuk proses pembacaan data seperti kode program berikut:

```
ifstream fin ("data5.txt", ios::in);
fin>>Nl>>Nb;
for(i=1;i<=Nl;i++){
	fin>>sl[i]>>el[i]>>rl[i]>>xl[i]>>bl[i];
```

Sebelumnya tambahkan library fstream.h untuk proses input/output data dari file pada bagian atas program seperti berikut: `#include <fstream>.`

2.3 Operator Aritmatika

Berikut ini merupkan operator penting yang sering digunakan dalam membuat kode program C++:

Operator	Keterangan
+	pertambahan
*	perkalian
%	Sisa pembagian
-	Pengurangan
/	Pembagian

Selain itu pemogramman C++ juga mengenal operator logika berikut:

Operator	Keterangan
&&	Operator Logika AND
||	Operator Logika OR
!	Operator Logika NOT

Operator logika tersebut digunakan untuk menghubungkan dua buah operasi relasi menjadi sebuah ungkapan kondisi. Hasil dari operator logika ini menghasilkan nilai numerik 1 (True) atau 0 (False).

2.4 Perulangan

Statement perulangan yang paling sering digunakan adalah for. Statement for memiliki tiga parameter yang diapit dalam tanda kurung ( ), yaitu nilai awal, akhir perulangan dan penambahan/pengurangan. Contoh perulangan dengan statement for adalah:

```
for(i=1;i<=100;i++){
	Perintah yang diulang;
}
```

Dari contoh tersebut perintah yang akan diulang dilakukan sebanyak 100 kali dimulai dari i = 1 sampai dengan i = 100.

Statement perulangan lain adalah while. Statement while memiliki kondisi yang harus dipenuhi yang berada dalam tanda kurung ( ), jika kondisi bernilai benar akan dilakukan perintah yang diulang, jika tidak, maka keluar dari blok perulangan while .

Contoh perulangan dengan statement while adalah:

```
Iterasi = 1;
while (Iterasi<=100){
       Perintah yang diulang;
       Iterasi=Iterasi+1;
}
```

Dari contoh di atas perintah yang akan diulang dilakukan sebanyak 100 kali dimulai dari Iterasi = 1 sampai dengan Iterasi = 100.

2.5 Perhitungan Waktu Komputasi

Terdapat beberapa perintah dalam C++ untuk mengakses waktu komputer, akan tetapi untuk ketelitian sampai orde millidetik dapat mengunakan perintah berikut:

```
volatile DWORD dwStart;
float T;
dwStart = GetTickCount();
.
.
.
T = (GetTickCount() - dwStart)/1000
cout<<" Waktu Komputasi adalah: "<<T<<" seconds"<<endl;
```

Sebelumnya tambahkan library `#include <windows.h>` pada header program utama. Waktu kumputasi ditampung dalam variabel T dan satuannya adalah second.

2.6 Program C++ untuk Penyelesaian Sistem Persamaan Linear

Dalam perhitungan aliran daya listrik, perilaku sistem dapat dimodelkan dengan persamaan matematika. Biasanya jumlah persamaan matematika yang dihasilkan lebih dari satu. Persamaan tersebut dapat ditulis dalam susunan yang teratur berdasarkan urutan peubah membentuk Sistem Persamaan Linear (SPL) dan selanjutnya diselesaikan secara simultan. Untuk memudahkan dinyatakan dalam bentuk matrik berikut:

$$\boldsymbol{Ax = b} \qquad (2.1)$$

Dimana:

A adalah matriks berukuran n x n dengan elemen matrik a_{ij}

x adalah matriks berukuran n x 1 dengan elemen matrik x_j

b adalah matriks berukuran n x 1 (disebut juga vektor kolom) dengan elemen matrik b_j

Penyelesaian numerik sistem persamaan linear ***Ax = b*** dapat dilakukan dengan dua cara, yaitu metode langsung dan metode tidak langsung. Penyelesaian Sistem Persamaan Linear secara numerik menggunakan metode langsung dapat diklasifikasikan sebagai berikut :

- Metode Eliminasi Gauss, prinsipnya: merupakan operasi eliminasi elemen-elemen pembentuk matriknya sedemikian rupa sehingga dapat terbentuk matriks segitiga atas, dan akhirnya solusinya diselesaikan menggunakan teknik substitusi mundur.
- Dekomposisi LU, prinsipnya: melakukan dekomposisi matriks *A* terlebih dahulu sehingga dapat terbentuk matriks-matrik segitiga atas dan bawah, kemudian secara mudah dapat diselesaikan menggunakan teknik substitusi maju atau mundur.
- Invers Matrik, untuk mencari selesaian SPL digunakan invers matrik

2.6.1 Eliminasi Gauss

Metode Eliminasi Gauss merupakan operasi eliminasi elemen-elemen pembentuk matriknya sedemikian rupa sehingga dapat terbentuk matriks segitiga atas, dan akhirnya solusinya diselesaikan menggunakan teknik substitusi mundur. Penyelessaian SPL menggunakan metode Eliminasi Gauss dapat dilakukan menggunakan operasi baris elementer atau operasi kolom elementer. Perhatikan sistem persamaan linear berikut:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix} \qquad (2.2)$$

- Baris kedua dioperasikan terhadap baris pertama, untuk me-nol-kan a_{21}

$$a_{21} - P \,.\, a_{11} = 0$$

$$P = \frac{a_{21}}{a_{11}} = l_{21}$$

Untuk $j = 2, 3, ..., n$

$$a_{j2} = a_{j2} - l_{21} \,.\, a_{1j}$$

$$b_2 = b_2 - l_{21} \,.\, b_1$$

$$\begin{bmatrix} x & \dots & \dots & x \\ o & x & \dots & x \\ \dots & \dots & \dots & \dots \\ x & \dots & \dots & x \end{bmatrix} \quad \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix}$$

- Baris ke – 3 dioperasikan terhadap baris 1 dan 2
- Me-nol-kan kolom 1 dengan operasi baris 1

  $a_{31} - P \,.\, a_{11} = 0$

  $$P = \frac{a_{31}}{a_{11}} = l_{31}$$

  Untuk $j = 2, 3, ..., n$

  $a_{3j} = a_{3j} - l_{31} \,.\, a_{1j}$

  $b_3 = b_3 - l_{31} \,.\, b_1$
- Me-nol-kan kolom 2 dengan operasi baris 2

  $a_{32} - P \,.\, a_{22} = 0$

  $$P = \frac{a_{32}}{a_{22}} = l_{32}$$

  Untuk $j = 3, ..., n$

  $a_{3j} = a_{3j} - l_{32} \,.\, a_{2j}$

  $b_3 = b_3 - l_{32} \,.\, b_2$

Contoh 2.1:

Selesaikan SPL berikut:

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 2 & 3 \\ -1 & 3 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ -4 \end{bmatrix}$$

Jawab:

Proses yang dilakukan adalah membuat matrik A menjadi matriks segitiga atas:

- l_{21}=2, menolkan a_{21}

  $a_{21} = 2 - 2.1 = 0$

  $a_{22} = 2\text{-}2.2 = -2$

  $a_{23} = 3 - 2.1 = 1$

  $b_2 = 3 - 2.0 = 3$

  diperoleh SPL yang baru berikut:

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & -2 & 1 \\ -1 & 3 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ -4 \end{bmatrix}$$

- l_{31}=-1, menolkan a_{31}

  a_{31} = -1 – (-1).1 = 0

  a_{32} = 3 – (-1).2 = 5

  a_{33} = 0 – (-1).1 = 1

  b_3 = -4-(-1).0=-4

  diperoleh SPL yang baru berikut:

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & -2 & 1 \\ 0 & 5 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ -4 \end{bmatrix}$$

- l_{32}=-2.5, menolkan a_{32}

  a_{32} = 5 – (-2,5).-2 = 0

  a_{33} = 1 – (-2,5).1 = 3,5

  b_3 = -4 – (-2,5).3 = 3,5

  diperoleh SPL yang baru berbentuk segitiga atas berikut:

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & -2 & 1 \\ 0 & 0 & 3{,}5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 3.5 \end{bmatrix}$$

Selanjunya vektor x dapat dihitung menggunakan algoritma substitusi mundur berikut:

$3{,}5x_3 = 3{,}5$

$x_3 = 1$

$-2x_2 + x_3 = 3$

$x_2 = -1$

$x_1 + 2x_2 + x_3 = 0$

$x_1 = 1$

Berdasarkan langkah-langkah di atas dapat disusun algoritma penyelesaian SPL menggunakan operasi baris elementer:

Untuk $k = 1, 2, 3, ..., n-1$

Untuk $i = k+1, k+2, ..., n$

$$l = \frac{a_{ik}}{a_{kk}}$$

Untuk j = k+1, k+2, ..., n

$a_{ij} = a_{ij} - l \,.\, a_{kj}$

$b_i = b_i - l \,.\, b_k$

$a_{ik} = 0$

dan dilanjutkan dengan substitusi mundur berikut:

1. $x_n = \frac{b_n}{a_{nn}}$
2. Untuk $i = n-1, n-2, ..., 1$

$x_i = b_i$

Untuk $j = i+1, i+2, ..., n$

$x_i = x_i - a_{ij} \,.\, x_i$

$$x_i = \frac{x_i}{a_{ii}}$$

Interpretasi dari algoritma di atas ke dalam program komuter adalah sebagai berikut:

Program C++

Masukan:

```
float A[3][3]={{ 1,2,1},
               { 2,2,3},
               {-1,3,0}};
float b[3]={0,3,-4};
```

Langkah-langkah:

```
float x[10],l;
int i,j,k,n=3;

/*eliminasi maju untuk membuat matrik segitiga atas*/

for(k=1;k<=n-1;k++) {
      for(i=0;i<=k-1;i++) {
            l=A[k][i]/A[i][i];
            A[k][i]=0;
            for(j=i+1;j<=n-1;j++)
                  A[k][j]=A[k][j]-l*A[i][j];
            b[k]=b[k]-l*b[i];
      }
}

/*substitusi mundur untuk menghitung penyelesaian nilai x */
x[n-1]=b[n-1]/A[n-1][n-1];
for(i=n-2;i>=0;i--) {
```

```
        x[i]=b[i];
        for(j=i+1;j<=n;j++)
                x[i]=x[i]-A[i][j]*x[j];
        x[i]=x[i]/A[i][i];
}
```

Berikut contoh kasus dimana terdapat elemen diagonal bernilai 0, dalam proses penyelesaian.

Contoh 2.2:

Tentukan vektor x dari SPL berikut:

$$\begin{bmatrix} 2 & -4 & 1 & 3 \\ -1 & 2 & 3 & -2 \\ 3 & -4 & 1 & 2 \\ 1 & 3 & -1 & 4 \end{bmatrix} \bullet \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 6 \\ 3 \\ 5 \\ 10 \end{bmatrix}$$

Jawab:

Langkah pertama adalah menolkan kolom pertama mulai baris ke dua menggunakan faktor pengali l_{21}, l_{31} dan l_{41} diperoleh matrik berikut:

$$\begin{bmatrix} 2 & -4 & 1 & 3 \\ 0 & 0 & 3.5 & -0.5 \\ 0 & 2 & -0.5 & -2.5 \\ 0 & 5 & -1.5 & 2.5 \end{bmatrix} \bullet \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 6 \\ 6 \\ -4 \\ 7 \end{bmatrix}$$

Karena elemen matrik $a_{22} = 0$, maka untuk menghindari pembagian dengan nol dilakukan penggantian baris 2 dengan baris berikutnya yaitu baris 3, diperoleh:

$$\begin{bmatrix} 2 & -4 & 1 & 3 \\ 0 & 2 & -0.5 & -2.5 \\ 0 & 0 & 3.5 & -0.5 \\ 0 & 5 & -1.5 & 2.5 \end{bmatrix} \bullet \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 6 \\ -4 \\ 6 \\ 7 \end{bmatrix}$$

Selanjutnya menolkan a_{42} dengan operasi baris kedua dengan $l_{4.2} = 5/2$ diperoleh:

$$\begin{bmatrix} 2 & -4 & 1 & 3 \\ 0 & 2 & -0.5 & -2.5 \\ 0 & 0 & 3.5 & -0.5 \\ 0 & 0 & -0.25 & 8.75 \end{bmatrix} \bullet \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 6 \\ -4 \\ 6 \\ 17 \end{bmatrix}$$

Selanjutnya menolkan a_{43} dengan operasi baris ketiga dengan $l_{4.3} = -1/14$ diperoleh:

$$\begin{bmatrix} 2 & -4 & 1 & 3 \\ 0 & 2 & -0.5 & -2.5 \\ 0 & 0 & 3.5 & -0.5 \\ 0 & 0 & 0 & 8.714 \end{bmatrix} \bullet \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 6 \\ -4 \\ 6 \\ 17.426 \end{bmatrix}$$

Sehinggan dieroleh SPL akhirnya seperti berikut:

$$\begin{bmatrix} 2 & -4 & 1 & 3 \\ 0 & 2 & -0.5 & -2.5 \\ 0 & 0 & 3.5 & -0.5 \\ 0 & 0 & 0 & 8.714 \end{bmatrix} \bullet \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 6 \\ -4 \\ 6 \\ 17.426 \end{bmatrix}$$

Maka dengan menggunakan substitusi mundur diperoleh:

$$x_4 = \frac{17.426}{8.714} = 2$$

$$x_3 = \frac{6 + 0.5.x_4}{3.5} = 2$$

$$x_2 = \frac{-4 + 2.5.x_4 + 0.5.x_3}{2} = 1$$

$$x_2 = \frac{6 - 3.x_4 - .x_3 + 4.x_2}{2} = 1$$

Dengan demikian diperoleh x = [1, 1, 2, 2]^T

2.6.2 Metode Invers Matrik

Perhatikan sistem persamaan linear berikut:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \bullet \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \qquad (2.3)$$

Persamaan baru berbentuk sama dengan persamaan asal akan tetapi x_2 dan b_2 telah dipertukarkan menjadi:

$$\begin{bmatrix} a'_{11} & a'_{12} \\ a'_{21} & a'_{22} \end{bmatrix} \bullet \begin{bmatrix} x_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ x_2 \end{bmatrix} \qquad (2.4)$$

dimana:

$a'_{11} = a_{11} - a_{12} a_{22}^{-1} a_{21}$
$a'_{12} = a_{12} a_{22}^{-1}$
$a'_{21} = - a_{22}^{-1} a_{21}$
$a'_{22} = a_{22}^{-1}$

Proses yang sama juga dapat dilakukan untuk mempertukarkan x_1 dan b_1.

Algoritma dan program lengkap invers matrik yang diusulkan oleh Shipley and Coleman dapat dilihat pada program berikut:

```
int i,j,k,l;
```

```
for (i=1;i<=n;i++)
{
  a[i][i] = 1.0/a[i][i];
  for (j=1;j<=n;j++)
  {
     if (j!=i)
     {
        a[j][i] = a[j][i] * a[i][i];
        for (k=1;k<=n;k++)
        {
           if (k!=i)
           {
              a[j][k] = a[j][k] - a[j][i]*a[i][k];
              if (j==n)
              {
                 a[i][k] = -a[i][i]* a[i][k];
              }
           }
         }
     }
  }
}
k = n-1;
for (l = 1;l<=k;l++)
{
  a[n][l] = -a[n][n]*a[n][l];
}

for(i=1;i<=n;i++)
      for(j=1;j<=n;j++)
            cout<<" a["<<i<<"]["<<j<<"] = "<<a[i][j]<<endl;
```

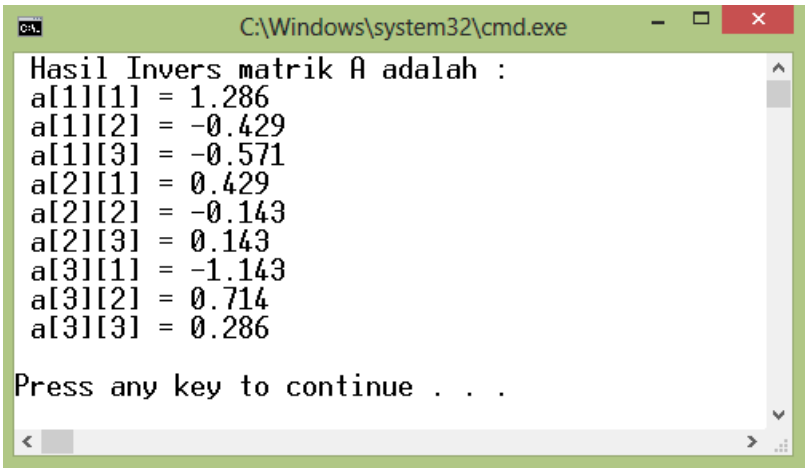
Jika Matrik A adalah:

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 2 & 3 \\ -1 & 3 & 0 \end{bmatrix}$$

dan vektor b adalah:

$$b = \begin{bmatrix} 0 \\ 3 \\ -4 \end{bmatrix}$$

Maka untuk menentukan vektor x, diperlukan invers matrik A menggunaan algoritma invers matrik tersebut. Hasil invers matrik adalah seperti diperlihatkan pada gambar 2.2



Gambar 2.2 Hasil invers matrik

Setelah invers matrik A diperoleh, langkah selanjutnya dalah perkalian matrik untuk memperoleh vektor x, seperti berikut:

$$[x] = [A]^{-1}\,[b] \qquad (2.5)$$

Maka diperoleh:

$$x = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}$$

Pada beberapa aplikasi diperlukan matrik A dinyatakan sebagai bilangan komplek dimana setiap elemen matrik terdiri dari bagian real dan bagian imajiner. Untuk melakukan invers komplek algoritma Shipley and Coleman dapat dikembangkan menjadi sperti berkut:

```
int i,j,k,l;
   double temp1;
   for (i=1;i<=n;i++)
    {
    temp1 = g[i][i]/(g[i][i]*g[i][i]+b[i][i]*b[i][i]);
       b[i][i] = -b[i][i]/(g[i][i]*g[i][i]+b[i][i]*b[i][i]);
       g[i][i] = temp1;
    for (j=1;j<=n;j++)
      {
      if (j!=i)
        {
        temp1 = (g[j][i]*g[i][i])-(b[j][i]*b[i][i]);
      b[j][i] = (g[j][i]*b[i][i])+(b[j][i]*g[i][i]);
      g[j][i] = temp1;
        for (k=1;k<=n;k++)
          {
          if (k!=i)
            {
                    temp1 = g[j][k]-g[j][i]*g[i][k]+b[j][i]*b[i][k];
      b[j][k] = b[j][k]-g[j][i]*b[i][k]-b[j][i]*g[i][k];
      g[j][k] = temp1;
```

```
            if (j==n)
              {
              temp1 = -g[i][i]*g[i][k]+b[i][i]*b[i][k];
        b[i][k] = -b[i][i]*g[i][k]-g[i][i]*b[i][k];
        g[i][k] =temp1;
              }
            }
          }
        }
      }
    }
    k = n-1;
    for (l = 1;l<=k;l++)
      {
      temp1 = -g[n][n]*g[n][l]-b[n][n]*b[n][l];
        b[n][l] = -b[n][n]*g[n][l]-g[n][n]*b[n][l];
        g[n][l] = temp1;
      }
```

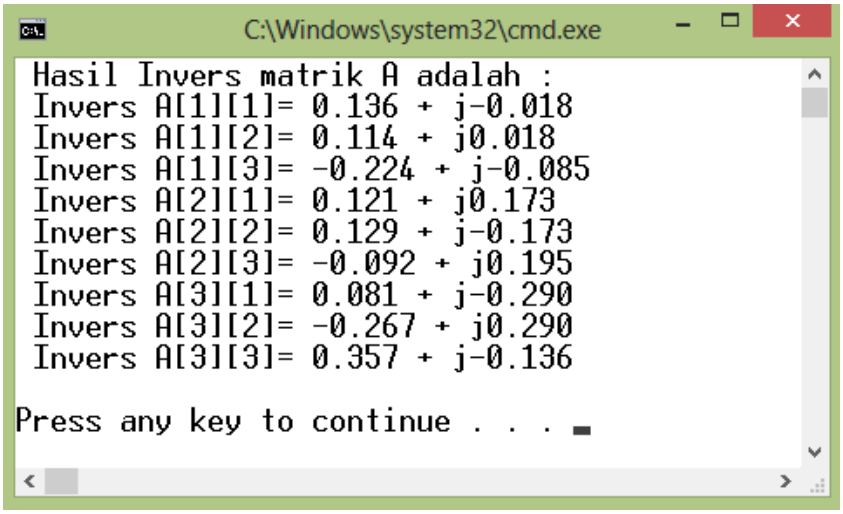
Jika Matrik A adalah:

$$A = \begin{bmatrix} 3+j2 & 2-j & 1+j \\ 2-j & 5-j & 3-2j \\ -1+j & 3-j2 & 2-j \end{bmatrix}$$

dan vektor b adalah:

$$b = \begin{bmatrix} 2+j \\ 3-2j \\ -4+j \end{bmatrix}$$

Maka untuk menentukan vektor x, diperlukan invers komplek dari matrik A menggunaan algoritma invers komplek di atas. Hasil invers matrik A adalah seperti diperlihatkan pada gambar 2.3



Gambar 2.3 Hasil invers matrik komplek

Input data matrik komplek terdiri dari bagian real dan imajiner seperti berikut:

Bagian real:

```
a[1][1] = 3; a[1][2] = 2; a[1][3] = 1;
a[2][1] = 2; a[2][2] = 5; a[2][3] = 3;
a[3][1] =-1; a[3][2] = 3; a[3][3] = 2;
```

Bagian imajiner :

```
b[1][1] = 2; b[1][2] =-1; b[1][3] = 1;
b[2][1] =-1; b[2][2] =-1; b[2][3] =-2;
b[3][1] = 1; b[3][2] =-2; b[3][3] =-1;
```

Kemudian program dicompile dan jalankan untuk memperoleh output invers matrik A. Setelah invers matrik A diperoleh, langkah selanjutnya dalah perkalian matrik untuk memperoleh vektor x, seperti berikut:

$$[x] = [A]^{-1}\,[b]$$

Maka diperoleh:

$$x = \begin{bmatrix} 1.651 + j0.04 \\ 0.283 - j1.18 \\ -1.018 + j1.24 \end{bmatrix}$$

2.6.3 Metode Gauss – Seidel

Metode Gauss – Seidel, merupakan metode penyelesaian SPL secara tidak langsung atau dilakukan secara iterasi. Dalam proses perhitungan iteratif harus ada tebakan awal, skema iterasi dan kriteria penghentian. Proses iterasi Gauss – Seidel dilakukan sampai dicapai suatu nilai yang konvergen sesuai dengan toleransi yang diberikan. Setiap harga x_i yang baru dihasilkan segera dihasilkan segera dipakai pada persamaan berikutnya untuk menentukan harga x_{i+1} yang lainnya.

Untuk menyelesaikan sistem persamaan linier *A x = b* dengan *A* adalah matriks koefisien *n x n, b* vector konstan *n x 1*, dan *x* vektor *n x 1* yang perlu dicari, maka proses yang diulang adalah persamaan umum (2.6) berikut:

$$x_i^{(k+1)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{k+1} - \sum_{j=i+1}^{n} a_{ij} x_j^{k}}{a_{ii}} \qquad (2.6)$$

Contoh 2.3:

Diketahui sistem persamaan linear Ax = b yaitu:

$$10x_1 - x_2 + 2x_3 = 6$$

$-x_1 + 11x_2 - x_3 + 3x_4 = 25$

$2x_1 - x_2 + 10x_3 - x_4 = -11$

$3x_2 - x_3 + 8x_4 = 15$

Nilai awal $(0\ 0\ 0\ 0)^T$

Hitung $[x_1, x_2, x_3, x_4]$ menggunakan Metode Gauss Seidel

Jawab:

$$x_1 = \frac{x_2}{10} - \frac{x_3}{5} + \frac{3}{5}$$

$$x_2 = \frac{x_1}{11} + \frac{x_3}{11} - \frac{3x_4}{11} + \frac{25}{11}$$

$$x_3 = \frac{-x_1}{5} + \frac{x_2}{10} + \frac{x_4}{10} - \frac{11}{10}$$

$$x_4 = \frac{-3x_2}{8} + \frac{x_3}{8} + \frac{15}{8}$$

Iterasi 1,

Hampiran pertama terhadap penyelesaian SPL tersebut adalah

$$x_1 = \frac{3}{5} = 0.6$$

$$x_2 = \frac{0.6}{11} + \frac{25}{11} = 2.3727$$

$$x_3 = \frac{-0.6}{5} + \frac{2.2727}{10} - \frac{11}{10} = -1.1$$

$$x_4 = \frac{-3.2.2727}{8} + \frac{-1.1}{8} + \frac{15}{8} = 1.8750$$

Hasil iterasi pertama ini menjadi nilai awal proses iterasi berikutnya. Proses yang sama dilankutkan untuk iterasi ke 2. Hasil perhitungan dapat dilihat pada tabel berikut:

Iterasi	x1	x2	x3	x4
1	0.6	2.32727	-0.98727	0.878864
2	1.03018	2.03694	-1.01446	0.984341
3	1.00659	2.00356	-1.00253	0.998351
4	1.00086	2.0003	-1.00031	0.99985
5	1.00009	2.00002	-1.00003	0.999988
6	1.00001	2	-1	0.999999
7	1	2	-1	1

Menggunakan selisih < 0.00001, maka setelah iterasi ke-7 diperoleh hampiran penyelesaian $x = (1\ 2\ -1\ \ 1)^T$. Nilai ini sudah sama dengan nilai sebenarnya, *yaitu* $x = (1\ 2\ -1\ 1)^T$.

Berdasarkan langkah-langkah diatas dapat disusun algoritma Metode Gauss Seidel sebagai berikut:

Masukan: a_{ij} ; i.j = 1, 2, …., n

b_i ; i = 1,2,…...,n

x_i ; i = 0

Eps

Langkah-langkah:

1. Untuk i = 1, 2, 3, ..., n.

   $x_i^{baru} = b_i$

   Untuk j = 1,2,3,...,n

   Jika j ≠i,

   $x_i^{baru} = x_i^{baru} - \sum a_{ij} x_j^0$

   $x_i^{baru} = \dfrac{x_i^{baru}}{a_{ii}}$

   Deltax = $|x_i^{baru} - x_i^0|$

   Upadte nilai $x_i^0 = x_i^{baru}$

2. Jika maksimum $|x_i^{k+1} - x_i^k| <$ Eps, Selesai, Print hasil
3. Kembali ke langkah 1

Keluaran: x_i ; untuk i = 1,2,…,n

Contoh 2.4:

Berdasarkan algoritma Gauss Seidel selesaikan SPL berikut:

$$\begin{bmatrix} 3 & 2 & 1 \\ 2 & 4 & 1 \\ 0 & 3 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 4 \\ 5 \\ -2 \end{bmatrix}$$

Nilai awal $[x_1\ x_2\ x_3] = [0\ 0\ 0]$

Jawab :

Iterasi 1

i = 1

x_1^{baru} $= b_1 = 4$

j = 2

$x_1^{baru} = x_1^{baru} - a_{12} \cdot x_2^0 = 4 - 2\,(0) = 4$

j = 3

$x_1^{baru} = x_1^{baru} - a_{13} \cdot x_3^0 = 4 - 1.(0) = 4$

$x_1^{baru} = x_1^{baru} / a_{11} = 4/3$

Deltax $= |x_1^{baru} - x_1^0|$

$= \left|4/3 - 0\right| = 4/3$

Update $x_1^0 = 4/3$

i = 2

x_2^{baru} $= b_2 = 5$

j = 1

$x_2^{baru} = x_2^{baru} - a_{21} \cdot x_1^0 = 5 - 2\,(4/3)$

j = 3

$x_2^{baru} = x_2^{baru} - a_{23} \cdot x_3^0 = 5 - 2\,(4/3) - 1.(0)$

$x_2^{baru} = x_2^{baru} / a_{22} = (5 - 2\,(4/3) - 1.(0))/4$

$= 7/12$

Deltax $= |x_2^{baru} - x_2^0|$

$= \left|7/12 - 0\right| = 7/12$

Update $x_2^0 = 4/3 = 7/12$

i = 3

x_3^{baru} $= b_3 = -2$

j = 1

$x_3^{baru} = x_3^{baru} - a_{31} \cdot x_1^0 = -2\text{-}0(4/3)$

j = 2

$x_3^{baru} = x_3^{baru} - a_{32} \cdot x_2^0 = -2\text{-}0(4/3)\text{-}3(7/12)$

$x_3^{baru} = x_3^{baru} / a_{33} = -2\text{-}0(4/3)\text{-}3(7/12)/5$

$= -45/(12.\ 5)$

$= -9/12 = -3/4$

Deltax $= |x_3^{baru} - x_3^0|$

$= |-3/4 - 0| = -3/4$

Update $x_3^0 = -3/4$

Selanjunya periksa maksimum Deltax apakah sudah < Epsilon, Jika sudah tampilkan hasil perhitungan x_i^{baru} untuk i = 1,2,3, jika tidak lanjutkan proses iterasi 2.

Proses yang sama dilankutkan untuk iterasi ke 2. Hasil perhitungan dapat dilihat pada tabel berikut:

Iterasi	x1	x2	x3
1	1.33333	0.583333	-0.75
2	1.19444	0.840278	-0.90417
3	1.07454	0.938773	-0.96326
4	1.02857	0.97653	-0.98592
5	1.01095	0.991003	-0.9946
6	1.0042	0.996551	-0.99793
7	1.00161	0.998678	-0.99921
8	1.00062	0.999493	-0.9997
9	1.00024	0.999806	-0.99988
10	1.00009	0.999925	-0.99996
11	1.00003	0.999971	-0.99998
12	1.00001	0.999989	-0.99999
13	1.00001	0.999996	-1

Menggunakan selisih < 0.00001, maka setelah iterasi ke-13 diperoleh hampiran penyelesaian $x = (1.00001\ 0.999996\ -1)^T$. Nilai ini sudah mendekati nilai sebenarnya, *yaitu* $x = (1\ 1\ -1)^T$.

Program C++

Metode Gauss Seidel

```
S=1.0;
while (abs( S ) > 0.00001){
            for ( i = 0; i < n; i ++ ){
              Xbaru[i] = b[i];
              for ( j = 0; j < n; j ++ ){
                     if ( j != i )
                     Xbaru[i] = Xbaru[i] - a[i][j] * X[j];
              }
              Xbaru[i] = Xbaru[i] / a[i][i];
              DeltaX[i] = Xbaru[i] - X[i];
              X[i] = Xbaru[i]; // update vektor X
            }
            Deltamax=0.0;
            for (i=0;i<n;i++) {
              if (abs(DeltaX[i])>Deltamax)
                     Deltamax=abs(DeltaX[i]);
            }
            S=Deltamax;
}
```

```
Contoh input data:
float a[3][3]={{3,2,1},
               {2,4,1},
                        {0,3,5}};
float b[3]={4,5,-2};
float X[3]={0,0,0};
```

2.7 Aplikasi Windows Form

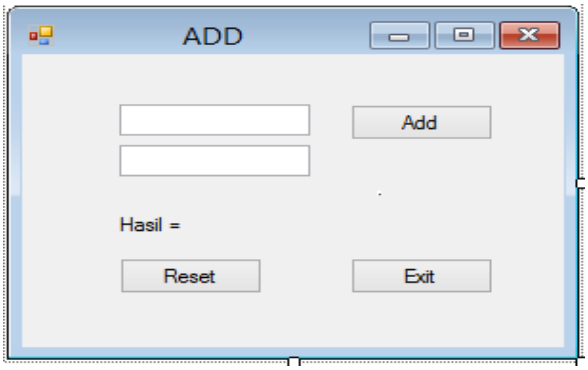
Untuk memperbaiki tampilan supaya lebih menarik dapat menggunakan Visual C++ Windows Form Application. Dalam pemograman bebasis windows form, toolbar-tolbar penting yang tersedia dalam pemograman Visual C++ adalah:

- textbox
- button
- Label

Selanjutnya set properties via Properties window

Contoh:

2.5 Buatlah program C++ untuk melakukan perjumlahan sederhana dengan tampilan berikut:



Gambar 2.4 Tampilan Program Perjumlah

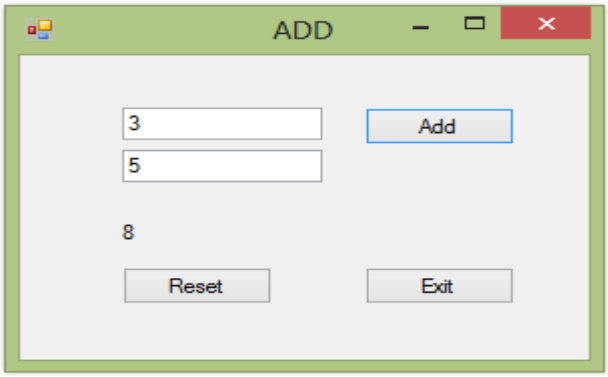
Jawaban:

Klik dan drag tolbar-tolbar berikut 2 textbox, 3 button dan 1 Label kedalam windows form dan atur posisinya seperti tampilan gambar 2.4

Double click Add, maka proses yang dilakukan adalah:

```
double a,b,c;
 a = System::Convert::ToDouble(textBox1->Text);
 b = System::Convert::ToDouble(textBox2->Text);
 c = a + b;
 label1->Text = System::Convert::ToString(c);
```

Hasil running

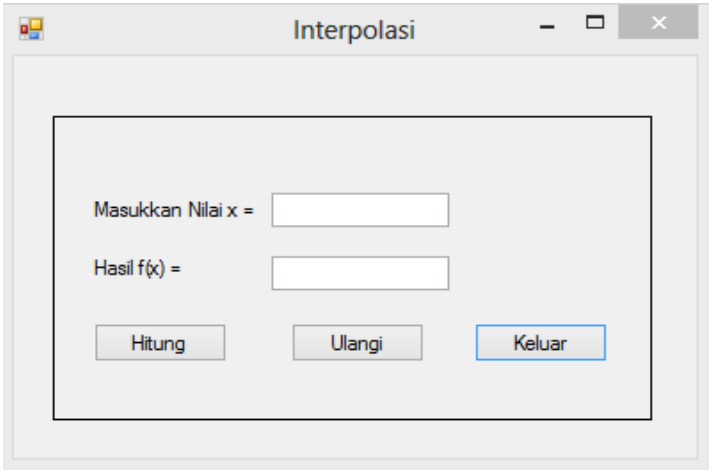


Gambar 2.5 Tampilan hasil running program

Untuk menghapus/ulangi, tambahkan kode program berikut pada editor button Reset:

```
textBox1->Text = " ";
textBox2->Text = " ";
```

Selanjutnya desain form satu sebagai form aplikasi interpolasi dengan tampilan seperti gambar 2.6 berikut:

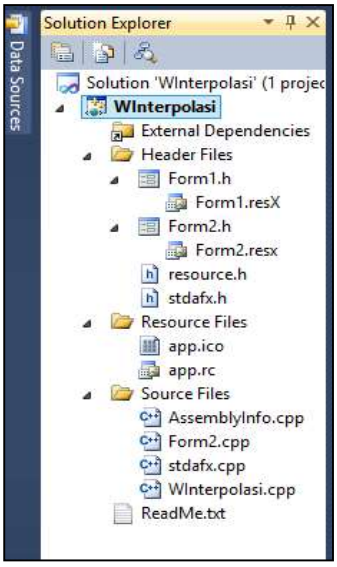


Gambar 2.6 Tampilan Program Interpolasi

Double click button Hitung, dan inputkan program interpolasi berikut:

```
int N=3, i,j;
double
x[10],f[10],PNZ,faktor,z;x[0]=2.0;f[0]=4.0;x[1]=3.0;f[1]=8.0;x[2]=5.0;f[2]
=25.0;
z=System::Convert::ToDouble(textBox1->Text);
PNZ=0.0;
for(i=0;i<N;i++){
   faktor=1;
   for(j=0;j<N;j++){
         if(i!=j){
              faktor=faktor*((z-x[j])/(x[i]-x[j]));
         }
   }
```

```
  PNZ=PNZ+faktor*f[i];
}
textBox2->Text = System::Convert::ToString(PNZ);
```



Gambar 2.7 Solution Explorer untuk aplikasi banyak form

z adalah nilai yang akan dicari interpolasinya diantara x[0] = 2 sampai dengan x[2] = 5, hasilnya disimpan sebagai variabel PNZ.

Aplikasi dengan banyak Wimdows Form dapat dilakukan dengan klik kanan pada nama project, selanjutnya pilih add New Item dan Windows Form. Setelah diberi nama Form2, akan muncul File Form2.h dan Form2.cpp pada Solution Explorer seperti gambar 2.7.

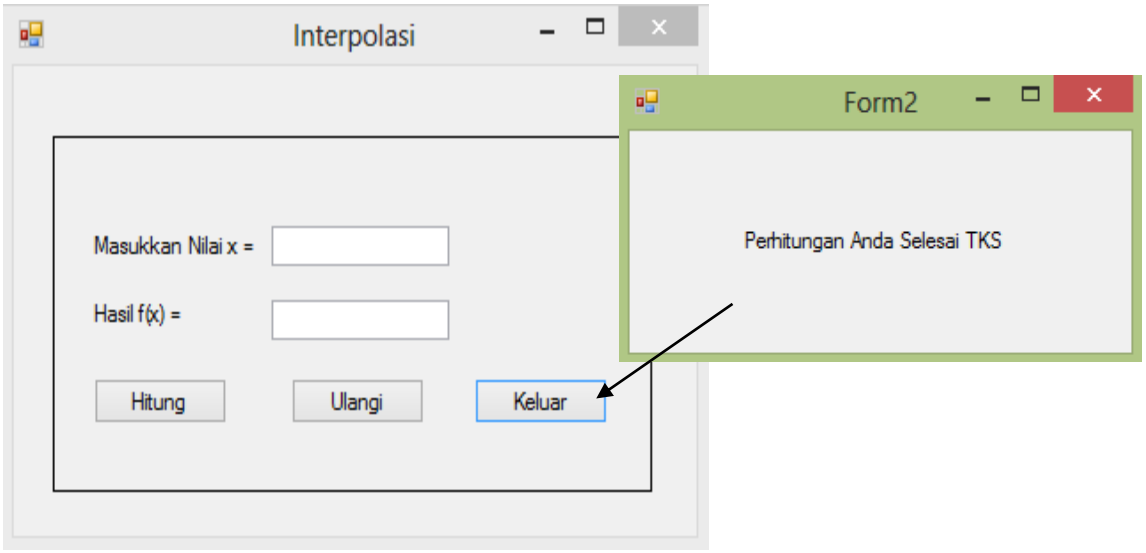
Sebagai contoh form dua akan muncul saat button Keluar ditekan. Kode Program untuk menampilkan form dua adalah:

```
Form2^ form2 = gcnew Form2();
form2->Show();
```

Program ini diketikan dalam `Void button3_Click berikut:`

```
private: System::Void button3_Click(System::Object^  sender,
System::EventArgs^  e) {
       Kode Program untuk menampilkan form dua;
}
```

Sebelumnya tambahkan #include “Form2.h” pada kode editor form1.h. Setelah program dijalankan, hasilnya adalah seperti tampilan berikut:



Gambar 2.8 Tampilan aplikasi banyak form

Untuk keluar dari aplikasi ketika button Keluar ditekan, maka gunakan perintah berikut:

```
Application::Exit();
```

Sehingga program dalam `Void button3_Click` menjadi:

```
private: System::Void button3_Click(System::Object^  sender,
System::EventArgs^  e) {
                    Application::Exit();
                }
```

Soal Latihan:

1. Selesaikan SPL berikut menggunakan:

$$\begin{bmatrix} 2 & 1 & 2 \\ 2 & 3 & -1 \\ 1 & 2 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5 \\ 4 \\ 5 \end{bmatrix}$$

Algoritma pemograman C++ untuk memperoleh sistem segitiga atas dilanjutkan dengan substitusi mundur untuk memperoleh vektor x ?

2. Diketahui SPL dengan elemen matrik seperti berikut:

$$\begin{bmatrix} 1 & 2 & 1 \\ 1 & 2 & 3 \\ -1 & 3 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ -4 \end{bmatrix}$$

Hitung vektor x menggunakan:

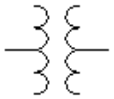
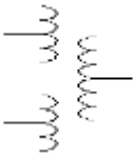
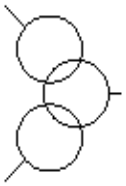
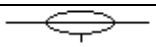
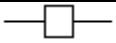
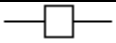
a. Metode langsung invers matrik
b. Metode iteratif Gauss Seidel dengan tebakan awal $x^{(0)} = 1$

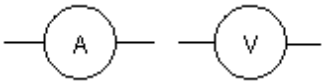
BAB III
REVIEW ANALISA SISTEM TENAGA

3.1. Single Line Diagram

Sistem tenaga listrik terdiri dari sejumlah peralatan yang terhubung membentuk suatu rangkaian. Rangkaian sistem tenaga listrik digambarkan dalam bentuk diagram satu garis. Diagram satu garis terdiri dari sejumlah peralatan yang dinyatakan dalam bentuk simbol-simbol standar. American Nasional Standards Institute (ANSI) dan International Electrotechnical Commision (IEC) telah menerbitkan sejumlah simbol standar untuk mengambarkan diagram sistem kelistrikan. Tabel 3.1 memperlihatkan beberapa simbol yang banyak digunakan:

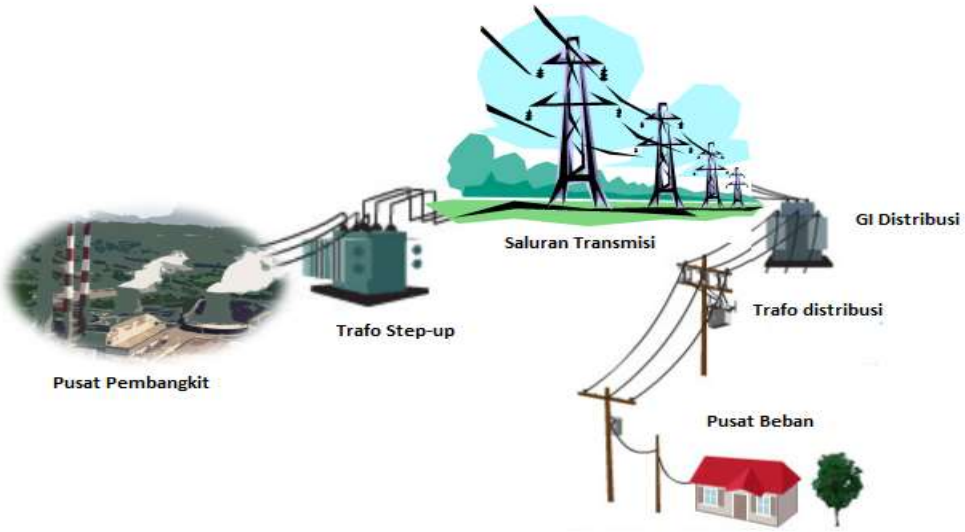
Tabel 3.1 Simbol Peralatan

PERALATAN	SIMBOL	
	ANSI	IEC
Mesin berputar		
Grid PLN		
Trafo Daya Dua Belitan		
Trafo Daya Tiga Belitan		
Fuse		
Trafo Arus		
Pemutus Tenaga HV		
Pemutus Tenaga LV		
Koneksi Delta		
Koneksi Bintang		

Koneksi Bintang dengan Netral ditanahkan		
Trafo Tegangan		
Kapasitor shunt		
Ammeter dan Voltmeter		

Contoh 3.1:

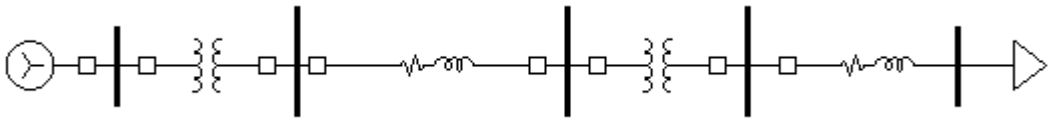
Gambarkan single line diagram sistem tenaga listrik dari pusat pembangkit sampai ke pusat beban berikut:



Gambar 3.1 Komponen fisik sistem tenaga listrik

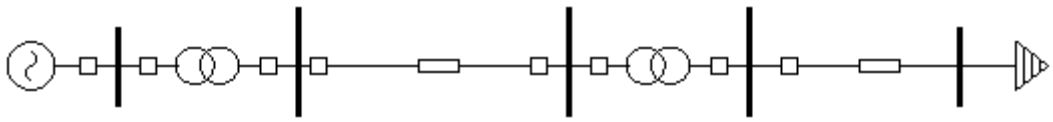
Jawab:

a. Berdasarkan standar ANSI



Gambar 3.2 Single line diagram sistem tenaga listrik berdasarkan standar ANSI

b. Berdasarkan standar IEC



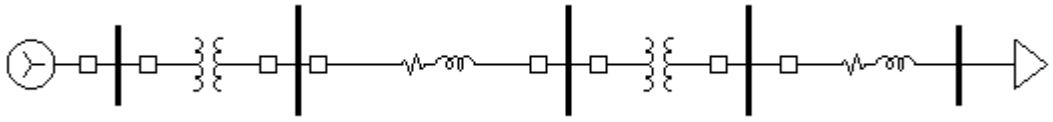
Gambar 3.3 Single line diagram sistem tenaga listrik berdasarkan standar IEC

3.2 Rangakaian Pengganti Reaktansi atau Impedansi

Setiap komponen sistem tenaga listrik dapat dinyatakan menggunakan simbol-simbol seperti tabel 4.1, selanjutnya simbole tersebut memiliki persamaan dalam bentuk komponen rangkaian listrik yang terdiri dari resistor, reaktansi dan sumber AC atau DC. Pernyataan rangkaian sistem tenaga dalam bentuk komponen rangkaian listrik disebut dengan rangkaian pengganti reaktansi atau impedansi.

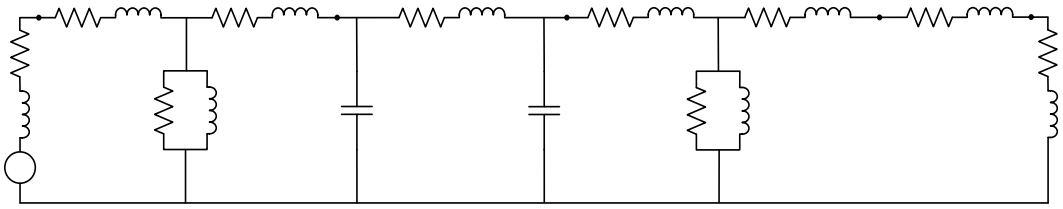
Contoh 3.2:

Nyatakan single line diagram berikut dalam bentuk rangkaian pengganti raktansi?



Gambar 3.4 Single line diagram sistem tenaga listrik radial

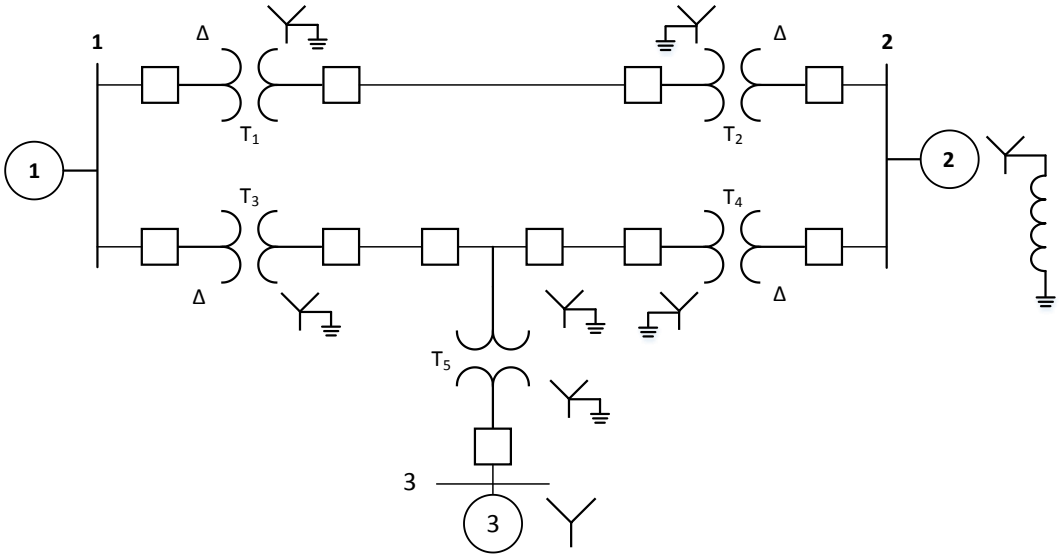
Jawab:



Gambar 3.5 Rangkaian pengganti raktansi sistem radial

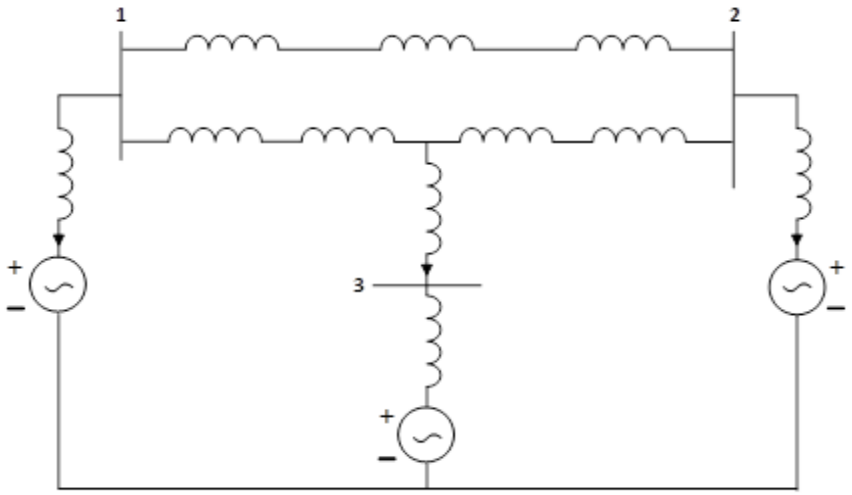
Contoh 3.3:

Nyatakan single line diagram berikut dalam bentuk rangkaian pengganti raktansi?



Gambar 3.6 Single line diagram sistem interkoneksi

Jawab:

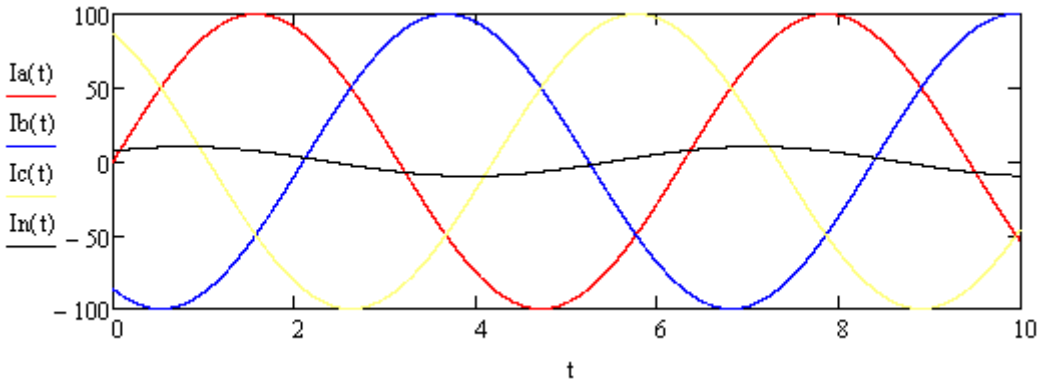


Gambar 3.7 Rangkaian pengganti reaktansi sistem interkoneksi

3.3 Sistem Tiga Phasa

Sifat penting suatu sistem tiga phasa adalah bahwa perjumlahan phasor dari tiga tegangan saluran/phasa dan jumlah phasor arus saluran/phasa sama dengan nol. Jika impedansi beban tiga phasa tidak sama, jumlah phasor dan arus netral I_N tidak nol dan disebut dengan beban tidak seimbang. Ketidak seimbangan beban bisa terjadi ketika ada hubung singkat atau hubung terbuka pada beban.

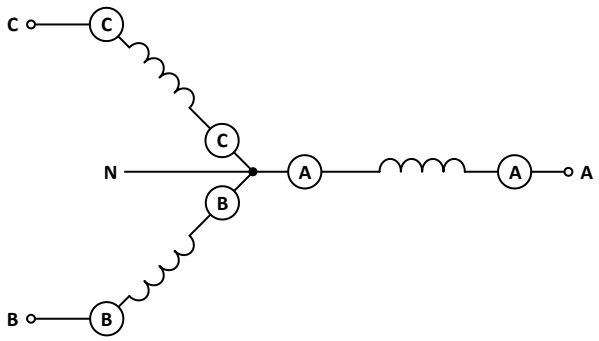
Beriku contoh kurva arus ketiga phasa saat terjadi ketidak seimbangan beban 10 % dan muncul arus I_N yang besar :



Gambar 3.8 Arus ketiga phasa saat terjadi ketidak seimbangan beban 10 %

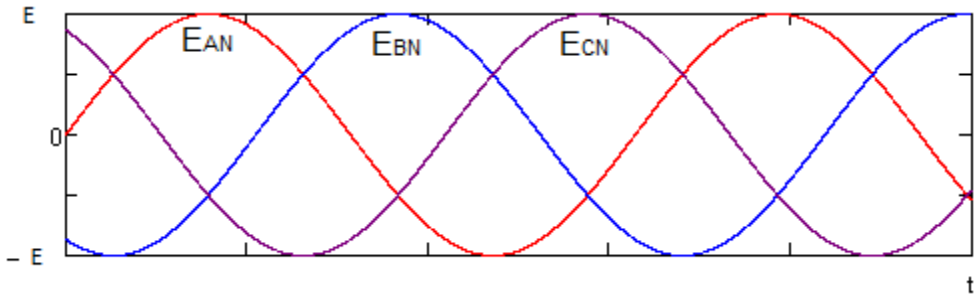
Sistem 3 phasa sangat banyak digunakan dalam industri tenaga listrik untuk menyalurkan daya dari pusat pembangkit ke pusat beban atau untuk pemakaian sendiri. Beban dari sistem 3 phasa banyak dijumpai diindustri-industri yang mengggunakan beban motor listrik. Motor-motor listrik besar dirancang untuk beroperasi dari sumber 3 phasa. Rangkaian 3 phasa sama dengan 3 sumber tegangan AC yang dibangkitkan dari generator AC 3 phasa. Struktur dan kumparan generator 3 phasa hubungan Y dapat dilihat pada gambar 3.9.

Rangkaian ekivalen dari kumparan stator diperlihatkan pada gambar 3. Kumparan tersebut memiliki koneksi bersama yang disebut netral diberi label N.



Gambar 3.9. Rangkaian ekivalennya kumparan stator

Kumparan tersebut memiliki 4 terminal output yaitu A, B, C dan N. Jika tegangan terminal e_{AN} , e_{BN} dan e_{CN} di gambarkan pada sumbu koordinat yang sama diperoleh bentuk kurva tegangan 3 phasa sebagai gambar 3.10.

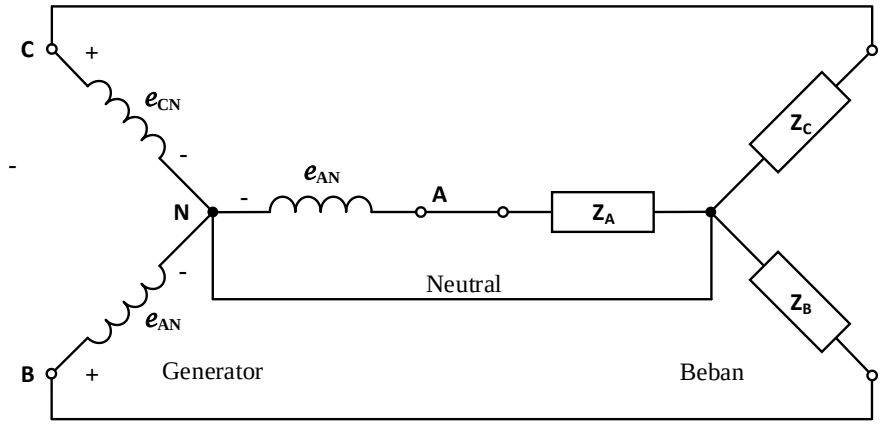


Gambar 3.10. Kurva tegangan kumparan stator

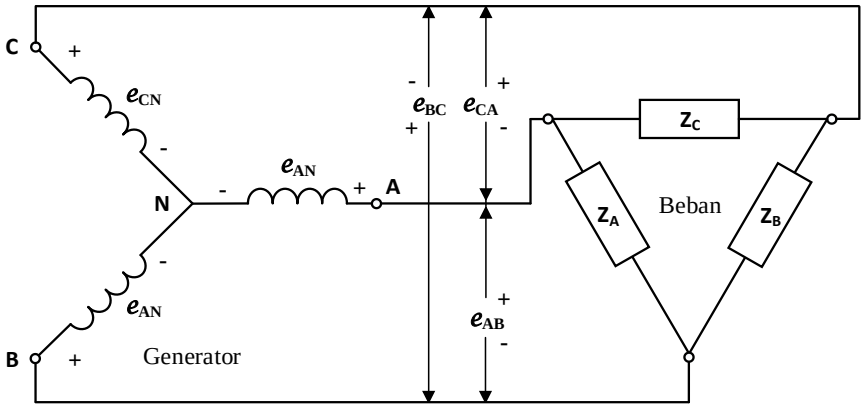
Dari bentuk gelombang output generator, besar tegangan phasa dapat dinyatakan sebagai berikut:

$$e_{AN} = E\ sin\ \omega t = E\ \angle 0^0$$
$$e_{BN} = E\ sin\ (\omega t - 120^0) = E\ \angle -120^0$$
$$e_{CN} = E\ sin\ (\omega t + 120^0) = E\ \angle 120^0$$

Beban yang terhubung pada sistem tiga phasa terdiri dari beban Y dan Δ. Jika terhubung dengan beban Y disebut sistem Y-Y, sedangkan jika terhubung dengan beban Δ disebut system Y-Δ seperti yang ditunjukkan pada gambar 5.



(a).



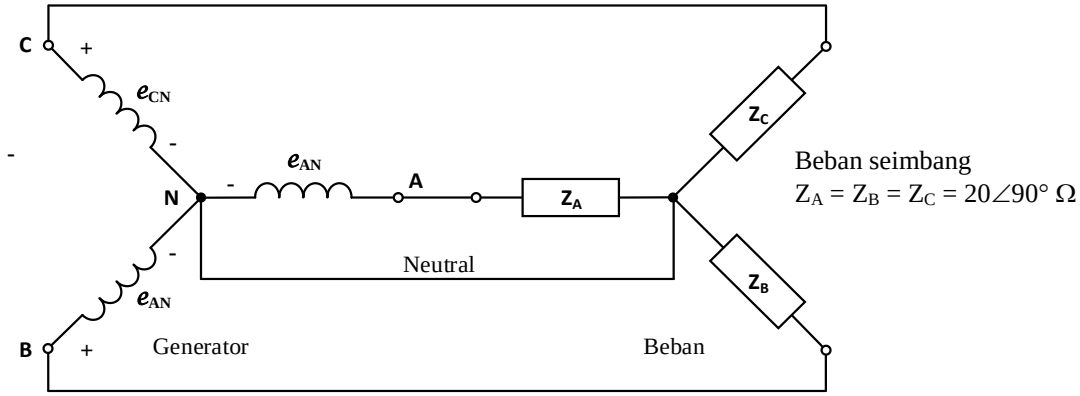
(b).

Gambar 3.11. Sumber terhubung dengan beban a). sistem Y-Y, b). sistem Y- Δ

Untuk sistem Y-Y, jika beban seimbang artinya besar dan sudut phasor impedansi sama, maka tidak ada arus yang lewat melalui kawat netral. Sebaliknya jika beban tidak seimbang yaitu besar dan sudup phasor beban tidak sama/identik, maka pada kawat netral akan mengalir arus yang besarnya tergantung sejauhmana besar ketidakseimbnagn beban tersebut.

Contoh 3.4:

Suatu sistem 3 phasa Y-Y, Tegangan antar phasa 400 V dan frekuensi 50 Hz. Urutan phasa ABC dengan $a_{AN} = 400/\sqrt{3}\ \angle 0°$ V.



Gambar 3.12. Sumber terhubung dengan beban sistem Y-Y

Maka dapat dihitung arus masing-masing phasa sebagai berikut :

$$i_{A\phi} = i_{AL} = \frac{e_{AN}}{Z_A} = \frac{400/\sqrt{3}\angle 0°}{20\angle 90°} = 11{,}55\angle -90° A$$

$$i_{B\phi} = i_{BL} = \frac{e_{BN}}{Z_B} = \frac{400/\sqrt{3}\angle -120°}{20\angle 90°} = 11{,}55\angle -210° A$$

$$i_{C\phi} = i_{CL} = \frac{e_{CN}}{Z_C} = \frac{400/\sqrt{3}\angle 120°}{20\angle 90°} = 11{,}55\angle 30° A$$

Dari hasil perhitungan arus phasa dapat dilihat bahwa besar arus sama satu sama lain berbeda phasa 120° (simetris). Sehingga besar arus i_N adalah :

$$i_N = i_{A\phi} + i_{B\phi} + i_{C\phi} = 0$$

Sekarang jika Z_A berubah menjadi 20∠-90° Ω, maka :

$$i_{A\phi} = i_{AL} = \frac{e_{AN}}{Z_A} = \frac{400/\sqrt{3}\angle 0°}{15\angle 90°} = 15{,}4\angle -90° A$$

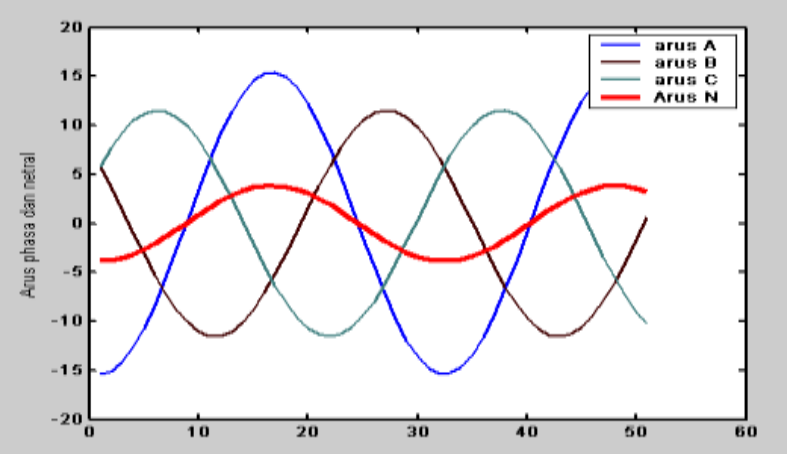
Sehingga

$$i_N = 15{,}4\angle -90° + 11{,}55\angle -210° + 11{,}55\angle 30° = -j15{,}4 - 11{,}55\cos 30 - j11{,}55\sin 30 +$$
$$11{,}55\cos 30 + j11{,}55\sin 30 = (-10+10) + j(-5{,}775 + 5{,}775 + 5{,}775) = j3{,}85$$

Dalam bentuk polar arus netral dapat ditulis : 3,85∠-90° ampere.

Ini merupakan nilai yang sangat besar yang dapat membahayakan sistem.

Bentuk gelombang arus phasa dan netral dapat dilihat pada gambar berikut :



Gambar 3.13. bentuk kurva gelombang arus phasa dan arus netral.

3.4 Sistem Per-Unit

Dalam analisa sistem tenaga listrik, nilai-nilai yang digunakan cukup besar apabila menggunakan besaran aktual yang dapat menyebabkan terjadinya kesulitan atau kesalahan dalam perhitungan. Untuk mengatasi masalah tersebut dapat menggunakan nilai persentase dan satuan per unit. Kedua metode perhitungan tersebut, baik dengan persentase maupun dengan satuan per unit, lebih sederhana dibanding menggunakan langsung nilai-nilai ampere, ohm, dan volt yang sebenarnya. Metode

per unit mempunyai sedikit kelebihan dari metode persentase, karena hasil perkalian dari dua kuantitas (dua nilai) yang dinyatakan dalam per unit sudah langsung diperoleh dalam per unit juga, sedangkan hasil perkalian dari dua kuantitas yang dinyatakan dalam persentase masih harus dibagi dengan 100 untuk mendapatkan hasil dalam persentase.

Definisi satuan per unit untuk suatu kuantitas ialah perbandingan kuantitas tersebut terhadap nilai dasarnya yang dinyatakan dalam desimal. Atau dengan kata lain satuan per unit merupakan sistem penskalaan guna mempermudah kalkulasi atau proses perhitungan dalam menganalisa sebuah sistem jaringan listrik. Besaran-besaran sistem dalam satuan masing-masing, tegangan dalam volt – arus dalam ampere – impedansi dalam ohm, ditransformasikan ke dalam besaran tak berdimensi yaitu per-unit (disingkat pu). Nilai per-unit dari suatu besaran merupakan rasio dari besarantersebut dengan suatu besaran basis. Besaran basis ini berdimensi sama dengan dimensi besaran aslinya sehingga nilai per-unitbesaran itu menjadi tidak berdimensi:

$$\text{Nilai per unit} = = \frac{\text{Nilai Aktual}}{Nilai\ Dasar}$$

Dengan menentukan dua besaran dasar biasanya tegangan dasar dan daya dasar, maka besaran dasar yang lain (arus dan impedansi) dapat ditentukan atau dihitung.

$$\text{Arus dasar, A} = \frac{\text{kVA}_{1\emptyset}\ \text{(dasar)}}{\text{tegangan dasar,kV}_{LN}}$$

$$\text{Impedansi dasar, } \Omega = \frac{\text{tegangan dasar, V}_{LN}}{\text{arus dasar, A}} \text{ atau}$$

$$\text{Impedansi dasar, } \Omega = \frac{(\text{tegangan dasar, kV}_{LN})^2 \text{ x } 1000}{\text{kVA}_{1\emptyset}\ \text{(dasar)}} \text{ atau}$$

$$\text{Impedansi dasar, } \Omega = \frac{(\text{tegangan dasar, kV}_{LN})^2}{\text{MVA}_{1\emptyset}\ \text{(dasar)}}$$

Dimana:

Daya dasar, $kW_{1\phi}$ = $kVA_{1\phi}$ (dasar)

Daya dasar, $MW_{1\phi}$ = $MVA_{1\phi}$ (dasar)

Selanjutnya dapat dihitung:

$$\text{Impedansi per-unit} = \frac{\text{Impedansi aktual, } \Omega}{\text{Impedansi Dasar, } \Omega}$$

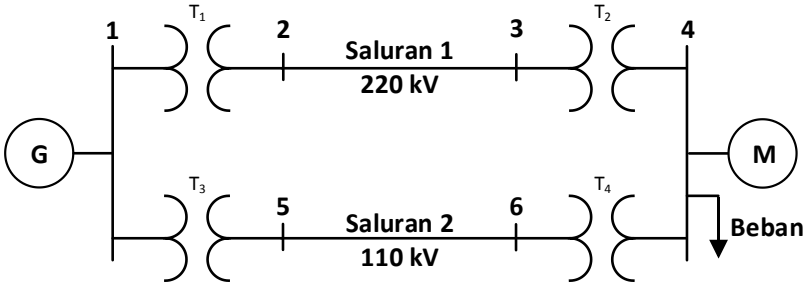
Setelah proses perhitungan selesai, besaran aktual diperoleh dengan cara mengalikan nilai per unit dengan nilai dasarnya.

Selain itu, untuk menentukan nilai per unit dari peralatan yang telah ada nilai pu berdasarkan rating nameplate dapat menggunakan persamaan berikut:

$$\text{Zpu (baru)} = \text{Z pu (lama)}.\left(\frac{Vdasar(lama)}{Vdasar(baru)}\right)^2 \left(\frac{S\ dasar(baru)}{S\ dasar(lama)}\right)$$

Contoh soal 3.5:

Suatu sistem tenaga seperti gambar berikut:



Gambar 3.14. Sistem tenaga contoh soal 3.5.

Dengan data sistem sebagai berikut:

G:	90 MVA	22kV	x = 0.18 per unit
T1:	50 MVA	22/220 kV	x = 0.10 per unit
T2:	40 MVA	220/11 kV	x = 0.06 per unit
T3:	40 MVA	22/110 kV	x = 0.064 per unit
T4:	40 MVA	110/11 kV	x = 0.08 per unit
M:	66.5 MVA	10.45 kV	x = 0.185 per unit

Saluran satu memiliki reaktansi seri 48.4 Ohm dan saluran dua 65.43 Ohm. Beban pada bus empat menyerap daya 57 MVA pada 10.45 kV dengan PF 0.6 lagging.

Gambarkan diagram impedansi dalam per unit termasuk impedansi beban jika digunakan dasar 100 MVA dan 22 kV pada sisi generator.

Jawab:

Base tegangan pada bus 1 adalah 22 kV dan 100 MVA sehingga:

G: $x = 0.18 \left(\frac{100}{90}\right) = 0.2$ pu

T_1 : $x = 0.1 \left(\frac{100}{50}\right) = 0.2$ pu

T_2: $x = 0.06 \left(\frac{100}{40}\right) = 0.15$ pu

T_3: x = 0.64 $\left(\frac{100}{40}\right)$ = 0.16 pu

T_4 : x = 0.08 $\left(\frac{100}{40}\right)$ = 0.2 pu

M: x = 0.185 $\left(\frac{100}{66.5}\right)\left(\frac{10.45}{11}\right)^2$ = 0.25 pu

Untuk saluran 1, $Z_{base} = \frac{(220)^2}{100}$ = 484 Ω , sehingga

x = $\frac{48.4}{484}$ = 0.1 pu

Untuk saluran 2, $Z_{base} = \frac{(110)^2}{100}$ = 121 Ω, sehingga

x = $\frac{65.43}{121}$ = 0.54 pu

Daya komplek beban adalah $S_{L(34)} = 57 \angle 53.13^0$ MVA

Sehingga dapat dihitung impedansi beban

$$\mathbb{Z}_L = \frac{V_{LL}^2}{S_{L\,(34)}^X} = \frac{(10.45)^2}{57 \angle -53.13^2}$$

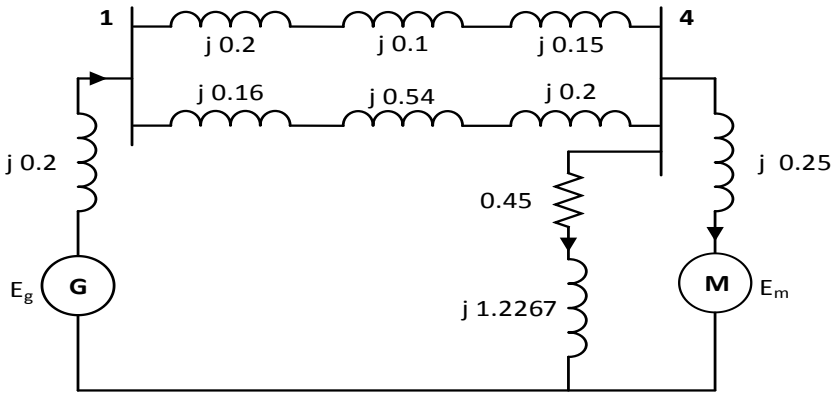
= 1.1493 + j 1.53267 Ω

Dasar Impedansi untuk beban adalah: $11^2/100$ = 1,21 Ω

Sehingga impedansi beban dalam per unit adalah

x = $\frac{1.1495 + j\ 1.53267}{1.21}$ = (0.95 + j 1.2667) pu

Dengan demikian dapat digambarkan rangkan ekivalen dalam nilai per unit sebagai berikut:

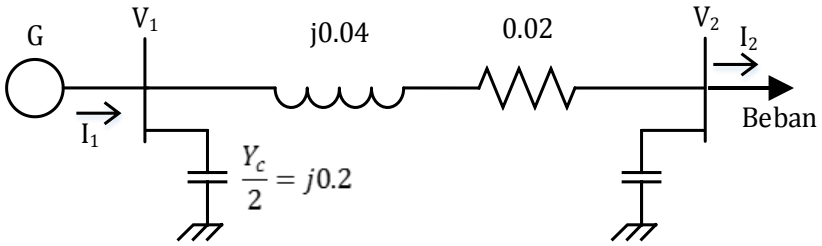


Gambar 3.15. rangkan ekivalen impedansi dalam nilai per unit.

3.5 Perhitungan Matrik Admitansi Bus

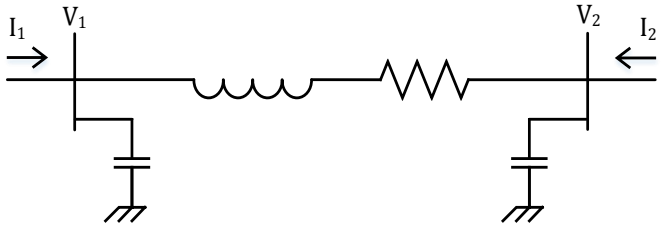
Analisa rangkaian listrik menggunakan analisis nodal menggunakan pendekatan hukum arus Kirchhoff yaitu besar arus masuk dan keluar pada suatu titik simpul sama dengan nol. Dari analisis nodal akan diperoleh sejumlah persamaan tegangan simpul. Matrik admitansi bus diperoleh dari persamaan tegangan simpul tersebut. Perhatikan contoh soal berikut:

Contoh soal 3.6:
Tentukan matrik admitansi bus sistem sederhana berikut:



Gambar 3.16 Sistem sederhana dua bus

Jika besar I_1 = 6A dan I_2 = 5.6A, tentukan nilai tegangan V_1 dan V_2 ?
Jawab :



Gambar 3.17 Rangkaian ekivalen nodal

Dari rangkaian di atas diperoleh persamaan simpul 1:

$$I_1 = \frac{(V_1 - V_2)}{Z_{12}} + V_1.\frac{Y_c}{2}$$

$$= (V_1 - V_2)\, Y_{12} + V_1 \frac{Y_C}{2}$$

$$= \left(Y_{12} + \frac{Y_c}{2}\right) V_1 - Y_{12}\, V_2$$

Untuk simpul 2:

$$I_2 = \frac{(V_2 - V_1)}{Z_{12}} + V_2.\frac{Y_c}{2}$$

$$= (V_2 - V_1)\, Y_{12} + V_2\, \frac{Y_C}{2}$$

$$= -Y_{12}\, V_1 + \left(Y_{12} + \frac{Y_c}{2}\right) V_2$$

Maka dalam bentuk matrik dapat ditulis:

$$\begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} Y_{12} + \frac{Y_c}{2} & -Y_{12} \\ -Y_{12} & Y_{12} + \frac{Y_c}{2} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}$$

Karena $Y_{12} = \frac{1}{z_{12}} = \frac{1}{0.02+j0.04} = 10 - j20$

Maka $\begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} 10 - j19.8 & -10 + j20 \\ -10 + j20 & 10 - j19.8 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}$

Dengan demikian diperoleh:

Ybus= $\begin{bmatrix} 10 - j19.8 & 10 + j20 \\ -10 + j20 & 10 - j19.8 \end{bmatrix}$

Jika besar I_1 = 6A dan I_2 = 5.6A, mana nilai tegangan V_1 dan V_2 adalah:

$$\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} 10 - j19.8 & 10 + j20 \\ -10 + j20 & 10 - j19.8 \end{bmatrix}^{-1} \begin{bmatrix} 6 \\ 5.6 \end{bmatrix}$$

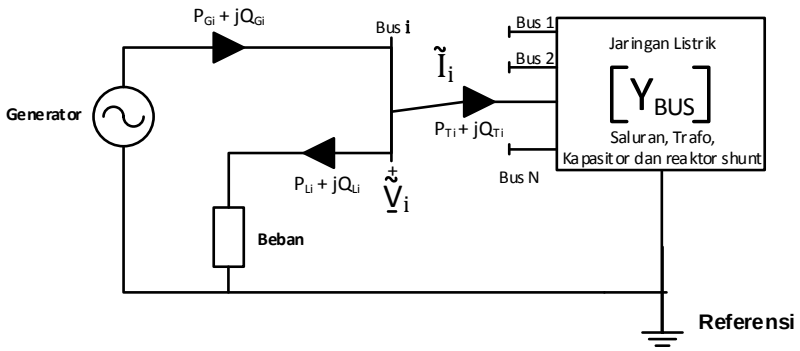
$$\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} 0.058 - j0.884 \\ -0.058 - j1.116 \end{bmatrix}$$

Selain dari itu matrik admitansi bus Y_{bus} juga dapat dihitung dengan menggunakan data input transformator dan saluran menggunakan persamaan berikut:

$$Y_{ii} = \sum_{j=0}^{n} y_{ij} + y_{i0} \qquad j \neq i$$

$$Y_{ij} = Y_{ji} = -y_{ij}$$

Sehingga sistem tenaga listrik dapat dimodelkan sebagai berikut:

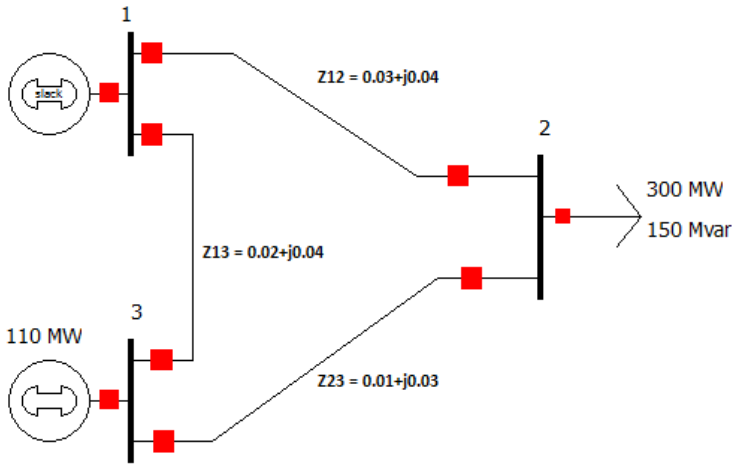


Gambar 3.18 Model sistem tenaga

Dengan mengetahui nilai impedansi saluran dan trafo serta suseptasi saluran dapat dibentuk matrik admitansi bus. Elemen matrik admitansi bus sangat tergantung dari topologi jaringan yang menghubungkan bus-bus dalam sistem kelistrikan. Elemen matrik admitansi yang memiliki nilai jika terdapat paling sedikit satu hubungan dengan bus lain.

Contoh Soal 3.7:

Suatu sistem tenaga listrik tiga bus sebagai berikut:



Gambar 3.19 Sistem 3 bus contoh soal 3.7

Hitung matrik Ybus sistem tiga bus diatas diatas? Bandingkan hasilnya dengan hasil perhitungan powerworld.

Jawaban:

Elemen off diagonal:

$$Y_{21} = Y_{12} = \frac{-1}{z_{12}} = \frac{-1}{0.03+j0.04} = -12 + j16$$

$$Y_{31} = Y_{13} = \frac{-1}{z_{13}} = \frac{-1}{0.02+j0.04} = -10 + j20$$

$$Y_{23} = Y_{32} = \frac{-1}{z_{32}} = \frac{-1}{0.01+j0.03} = -10 + j30$$

Elemen diagonal:

$Y_{11} = y_{12} + y_{13}$ = 12-j16+10-j20 = 22 - j36

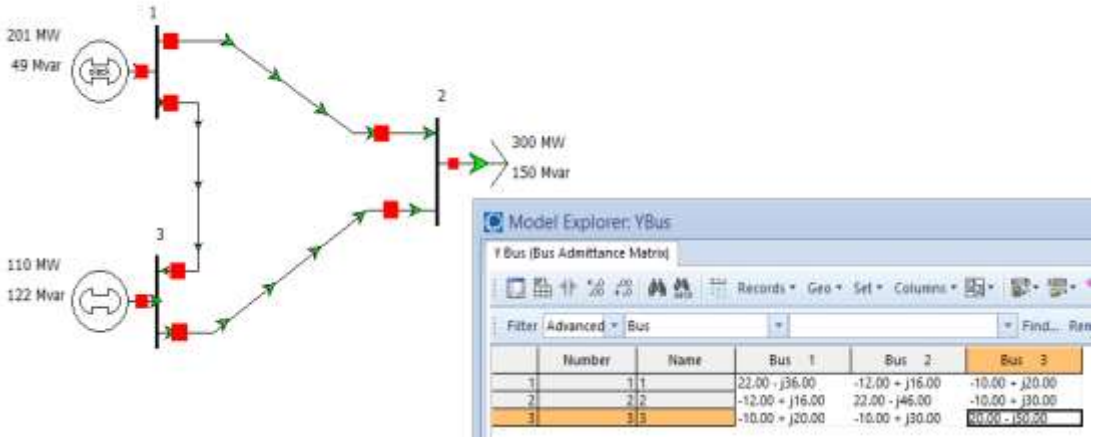
$Y_{22} = y_{21} + y_{23}$ = 12-j16+10-j30 = 22 - j46

$Y_{33} = y_{31} + y_{32}$ = 10-j20+10-j30 = 20 - j50

Sehingga:

$$Y_{bus} = \begin{bmatrix} 22 - \text{j}36 & -12 + j16 & -10 + j20 \\ -12 + j16 & 22 - \text{j}46 & -10 + j30 \\ -10 + j20 & -10 + j30 & 20 - \text{j}50 \end{bmatrix}$$

Jawaban tersebut sudah sama dengan hasil perhitungan software Powerworld berikut:



Gambar 3.20 Tampilan hasil running powerworld

Contoh Soal 3.8:

Suatu sistem tenaga listrik terdiri dari 5 bus dan 5 saluran dengan data impedansi saluran dan trafo seperti tabel berikut:

Bus kirim	Bus terima	Tipe cabang	R (pu)	X (pu)	B (pu)
1	5	Transformer	0,0015	0,02	0
4	2	Line	0,01	0,1	1,72
5	2	Line	0,0045	0,05	0,88
3	4	Transformer	0,00075	0,01	0
5	4	Line	0,01	0,1	0,44

Matrik admitansi bus sistem tenaga tersebut adalah:

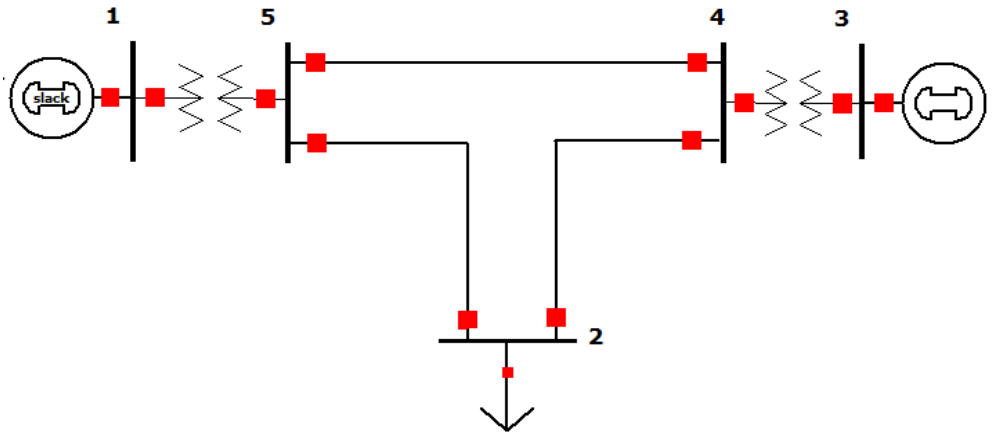
Ybus =

3,73 - j49,72	0	0	0	-3,73 + j49,72
0	Y_{22}	Y_{23}	-0,99 + j9,90	-1,79 + j19,84
0	0	7,46 - j99,44	-7,46 + j99,44	0
0	Y_{42}	-7,46 + j99,44	9,44 - j118,16	-0,99 + j9,90
-3,73 + j49,72	-1,79 + j19,84	0	-0,99 + j9,90	6,50 - j78,80

Hitung elemen matrik Y_{22}, Y_{23} dan Y_{42} yang belum diketahui dari matrik Ybus diatas?

Jawab:

Topologi jaringan sistem 5 bus tersebut adalah:



Elemen matrik Y_{22}, Y_{23} dan Y_{42} yang belum diketahui dapat dihitung sebagai berikut:

$Y_{23} = Y_{32} = 0$

$Y_{42} = Y_{24} = \frac{-1}{z_{24}} = \frac{-1}{0.01+j0.1} = -0.99 + j9.90$

$Y_{22} = y_{21} + y_{23} + y_{24} + y_{25} + y_{20}$

$= 0 + 0 + (0.99\text{-}j9.90) + (1.79\text{-}j19.84) + j(1.72)/2 + j(0.88)/2$

$= 2{,}78 - j28{,}44$

Sehingga Ybus lengkap menjadi:

3,73 - j49,72	0	0	0	-3,73 + j49,72
0	2,78 - j28,44	0	-0,99 + j9,90	-1,79 + j19,84
0	0	7,46 - j99,44	-7,46 + j99,44	0
0	-0,99 + j9,90	-7,46 + j99,44	9,44 - j118,16	-0,99 + j9,90
-3,73 + j49,72	-1,79 + j19,84	0	-0,99 + j9,90	6,50 - j78,80

Program komputer perhitungan admitansi bus untuk sistem dengan jumlah bus Nb dan jumlah saluran Nl adalah:

1. Proses baca data:

```
fin>>Nl;
for(i=1;i<=Nl;i++){
	fin>>sl[i]>>el[i]>>rl[i]>>xl[i]>>bl[i];
}
```

2. Proses perhitungan matrik admitansi bus:

```
for (l=1;l<=Nl;l++){
	i= sl [l];
	j= el [l];
	aa = rl[l]*rl[l]+xl[l]*xl[l];
	g[i][j] = -rl[l]/aa;
	g[j][i] = g[i][j];
	b[i][j] = xl[l]/aa;
	b[j][i] = b[i][j];
	g[i][i] += rl[l]/aa;
	g[j][j] += rl[l]/aa;
	b[i][i] += -xl[l]/aa + bl[l]/2;
	b[j][j] += -xl[l]/aa + bl[l]/2;
}
```

3. Proses tampilan output:

```
cout<<" Matrik Admitansi Bus:"<<endl;
for(i=1;i<=Nb;i++)
	for(j=1;j<=Nl;j++){
	cout<<" Y["<<i<<"]["<<j<<"]= "<<g[i][j]<<" + j"<<b[i][j]<<endl;
	}
```

Hasil running program untuk data contoh soal 3.8 adalah:

```
Matrik Admitansi Bus:
Y[1][1]= 3.72902 + j-49.7203
Y[1][2]= 0 + j0
Y[1][3]= 0 + j0
Y[1][4]= 0 + j0
Y[1][5]= -3.72902 + j49.7203
Y[2][1]= 0 + j0
Y[2][2]= 2.77564 + j-28.4403
Y[2][3]= 0 + j0
Y[2][4]= -0.990099 + j9.90099
Y[2][5]= -1.78554 + j19.8393
Y[3][1]= 0 + j0
Y[3][2]= 0 + j0
Y[3][3]= 7.45805 + j-99.4407
Y[3][4]= -7.45805 + j99.4407
Y[3][5]= 0 + j0
Y[4][1]= 0 + j0
Y[4][2]= -0.990099 + j9.90099
Y[4][3]= -7.45805 + j99.4407
Y[4][4]= 9.43825 + j-118.163
Y[4][5]= -0.990099 + j9.90099
Y[5][1]= -3.72902 + j49.7203
Y[5][2]= -1.78554 + j19.8393
Y[5][3]= 0 + j0
Y[5][4]= -0.990099 + j9.90099
Y[5][5]= 6.50466 + j-78.8006
```

Hasil yang diperoleh sama dengan perhitungan manual contoh soal 3.8.

Soal Latihan:

1. Suatu sistem tenaga listrik terdiri dari 5 bus dengan dengan data impedansi saluran seperti tabel berikut:

Bus kirim	Bus terima	Tipe cabang	R (pu)	X (pu)	B (pu)
1	5	Transformer	0,02	0,04	0
4	2	Line	0,01	0,03	1,72
5	2	Line	0,01	0,03	0,88
3	4	Transformer	0,0125	0,025	0
5	4	Line	0,02	0,04	0,44

Hitung matrik admitansi bus sistem tenaga tersebut?

2. Suatu sistem tenaga listrik terdiri dari 11 saluran dengan dengan data impedansi saluran seperti tabel berikut:

sl	rl	R	X	Bc
1	2	0.1	0.2	0.02
1	4	0.05	0.2	0.02
1	5	0.08	0.3	0.03
2	3	0.05	0.25	0.03
2	4	0.05	0.1	0.01
2	5	0.1	0.3	0.02
2	6	0.07	0.2	0.025
3	5	0.12	0.26	0.025
3	6	0.02	0.1	0.01
4	5	0.2	0.4	0.04
5	6	0.1	0.3	0.03
7	5	0	0.01	0

Bc adalah setengah suseptasi total saluran

Hitung matrik admitansi bus sistem tenaga tersebut?

BAB IV
ANALISA ALIRAN DAYA

4.1 Pendahuluan

Sistem tenaga listrik berfungsi untuk mendistribusikan energi listrik dari pusat pembangkit ke konsumen. Komponen utama sistem tenaga listrik terdiri dari pembangkit, transformator, saluran transmisi, saluran distribusi dan beban. Pembangkit atau generator berfungsi untuk mengkonversikan energi primer menjadi energi listrik. Energi primer ini dapat berupa air, angin, matahari dan energi yang berasal dari dalam tanah seperti uap panas, gas, batubara atau sumber-sumber lainnya. Selanjutnya energi listrik tersebut disalurkan melalui transformator dan saluran transmisi dan distribusi.

Salah satu analisa yang perlu dilakukan pada tahap perencanaan dan operasi sistem tenaga listrik adalah studi aliran daya. Studi aliran daya bertujuan untuk :

1. Mendapatkan informasi besar dan sudut fase tegangan pada setiap bus serta aliran daya aktif dan daya reaktif setiap saluran.
2. Mengetahui besar daya pembangkitan yang sesuai dalam melayani beban suatu sistem tenaga listrik.
3. Perencanaan dan pengembangan jaringan listrik, dimana studi aliran daya memberikan informasi tentang akibat dari penambahan beban baru, penambahan pembangkitan baru, penambahan gardu induk baru, penambahan saluran transmisi baru, interkoneksi dengan sistem lain, dan sebagainya.
4. Penentuan pembebanan peralatan sistem listrik seperti saluran transmisi dan transformator pada kondisi sekarang maupun masa datang.
5. Memberikan data awal bagi perhitungan gangguan hubung singkat, koordinasi sistem proteksi, studi harmonik dan studi stabilitas sistem tenaga.

Program aliran daya adalah tool dasar untuk menginvestigasi keperluan di atas.

Data yang diperlukan dalam perhitungan aliran daya adalah:

1. Data saluran berupa, resistansi, reaktansi dan suseptansi dan data rating dan impendansi transformator serta setelan tap transformator.
2. Data bus berupa: daya dan tegangan pembangkitan, daya beban, rating kapasitor *shunt* dan lain-lain.

Persyaratan yang harus dipenuhi dalam perhitungan aliran daya adalah:

1. Besar daya pembangkitan harus sama dengan permintaan beban ditambah rugi-rugi.
2. Tegangan bus berada pada rating tegangan yang diizinkan, biasaya ± 5 % atau ± 10 %
3. Pembangkit beroperasi pada batas daya aktif dan reaktif tertentu.
4. Saluran transmisi dan tranfo tidak boleh ada yang overload.

4.2 Klasifikasi Bus

Studi aliran daya digunakan untuk menentukan tegangan, aliran daya aktif dan daya reaktif pada suatu sistem tenaga listrik. Hasil studi aliran daya tersebut dapat digunakan untuk mengevaluasi kinerja sistem tenaga listrik. Evaluasi tersebut dilakukan secara terus-menerus sehingga perhitungan aliran daya diperlukan untuk setiap titik operasi. Dari hasil evaluasi performasi sistem tenaga akan dapat diambil langkah-langkah pengontrolan yang sesuai dengan keadaan sistem sebenarnya. Dengan demikian diperlukan algoritma perhitungan yang cepat dan effisien.

Di dalam studi tenaga, terdapat tiga jenis bus, yaitu:

1. Slack Bus (Swing bus atau bus berayun)

Slack bus sering juga disebut dengan *swing bus* atau bus berayun. Besaran yang diketahui dari bus ini adalah tegangan (V) dan sudut beban (δ). Bus yang memiliki kapasitas pembangkit terbesar biasanya dipilih menjadi slack bus. Bus ini juga merupakan bus referensi dengan sudut tegangan nol. Besaran yang dihitung adalah daya aktif (P) dan reaktif (Q) setelah proses iterasi dan berfungsi untuk menampung rugi-rugi jaringan.

2. P-V Bus (Bus pembangkit)

Bus pembangkit bisa disebut dengan bus kendali tegangan, dimana tegangan pada bus ini dijaga konstan. Besaran yang diketahui adalah daya aktif (P) yang dapat diatur melalui prime mover (penggerak mula) dan magnitut tegangan (|V|) yang dapat diatur melalui arus eksitasi generator sehingga bus ini juga sering disebut P-V bus. Besaran yang dapat dihitung dari bus ini adalah daya reaktif (Q) dan sudut tegangan (δ**)**

3. P-Q Bus (Bus Beban):

Setiap bus yang tidak memiliki generator selain slack bus disebut dengan bus beban. Pada bus ini daya aktif (P) dan daya reaktif (Q) diketahui sehingga sering juga disebut bus PQ. Daya aktif dan reaktif bernilai positif jika arah aliran daya keluar dari bus dan negatif jika masuk ke bus. Besaran yang dihitung pada bus ini adalah tegangan (|V|) dan sudut tegangan (δ).

Secara singkat klasifikasi bus dalam perhitungan aliran daya dapat dilihat pada tabel 4.1:

Tabel 4.1 Klasifikasi Bus Pada Sistem Tenaga

Tipe Bus	Besaran yang diketahui	Besaran akan dihitung	Keterangan
Slack	|V| = 1.0	P_G	Hanya ada satu buah dalam sistem
	$\delta = 0$	Q_G	
Pembangkit (PV)	|V|	δ	Setiap bus yang ada generator
	P_G	Q_G	
Beban (PQ)	P_L	|V|	Bus yang tidak ada generator
	Q_L	δ	

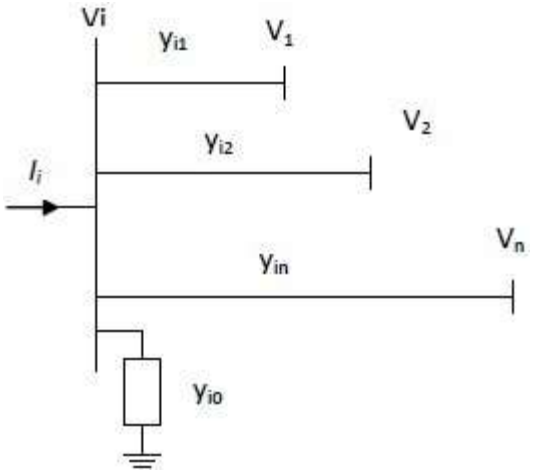
Keterangan :

|V| = Besar Tegangan
δ = Sudut PhasaTegangan
P_G = Daya Aktif Generator
Q_G = Daya Reaktif Generator
P_L = Daya Aktif Beban
Q_L = Daya Reaktif Beban

4.3 Persamaan Umum Aliran Daya

Sistem tenaga listrik terdiri dari sejumlah bus yang terinterkoneksi satu sama lain. Keuntungan sistem interkoneksi, antara lain bisa memperbaiki dan mempertahankan keandalan sistem, biaya operasional menjadi lebih rendah, dan kelebihan daya dapat ditranfer ke pusat beban yang lain.

Suatu sistem multi-bus dapat dapat dinyatakan seperti gambar berikut:



Gambar 4.1 Diagram satu garis sistem tenaga multi bus

Persamaan aliran daya dapat diturunkan dari hukum arus Kircchoff pada bus i, dimana besar arus yang masuk ke bus i adalah:

$$I_i = y_{io} V_i + y_{i1} (V_i - V_1) + y_{i2} (V_i - V_2) + \ldots + y_{in} (V_i - V_n)$$
$$= (y_{io} + y_{i1} + y_{i2} + \ldots + y_{in}) V_i - y_{i1} V_i - y_{i2} V_2 - \ldots - y_{in} V_n \quad (4.1)$$

Persamaan tersebut dapat ditulis kembali menjadi:

$$I_i = \sum_{j=0}^{n} y_{ij} - \sum_{j=1}^{n} y_{ij} V_j \qquad j \neq i \quad (4.2)$$

Daya aktif dan reaktif pada bus *i* adalah

$$P_i + jQ_i = V_i I_i^* \quad (4.3)$$

$$I_i = \frac{P_i + jQ_i}{V_i^*} \quad (4.4)$$

Substitusi persamaan (4.4) ke dalam persamaan *Ii* (4.2) menghasilkan persamaan:

$$\frac{P_i + jQ_i}{V_i^*} = V_i \sum_{j=0}^{n} y_{ij} - \sum_{j=1}^{n} y_{ij} V_j \qquad j \neq i \quad (4.5)$$

Persamaan (4.5) adalah persamaan umum aliran daya dalam bentuk aljabar non linear yang selanjutnya akan diselesaikan secara iteratif. Beberapa metode penyelesaian aliran daya secara iteratif yang umum digunakan adalah metode *Gauss – Seidel, Newton – Raphson,* dan *Fast Decoupled*.

4.4 Metode Gauss Seidel

Dalam studi aliran daya, permasalahan penting adalah penyelesaian sejumlah persamaan nonlinear (4.5) untuk dua variabel yang tidak diketahui pada setiap bus. Metode Gauss Seidel menyelesaikan persamaan (4.5) untuk V_i, dengan urutan iterasi berikut:

$$V_i^{(k+1)} = \frac{\frac{P_i^{sch} - \mathrm{j}\, Q_i^{sch}}{V_i^{*(k)}} + \sum y_{ij} V_j^{(k)}}{\sum y_{ij}} \quad \mathrm{j} \neq \mathrm{i} \tag{4.6}$$

Jika persamaan (4.5) diselesaiakan untuk $P_i^{(k+1)}$ dan $Q_i^{(k+1)}$ diperoleh:

$$P_i^{(k+1)} = \mathcal{R} \left\{ V_i^{*(k)} \left[V_i^{(k)} \sum_{j=0}^{n} y_{ij} - \sum_{j=1}^{n} y_{ij}\ V_j^{(k)} \right] \right\} \quad j \neq i \tag{4.7}$$

$$Q_i^{(k+1)} = -\mathcal{I} \left\{ V_i^{*(k)} \left[V_i^{(k)} \sum_{j=0}^{n} y_{ij} - \sum_{j=1}^{n} y_{ij}\ V_j^{(k)} \right] \right\} \quad j \neq i \tag{4.8}$$

Persamaan aliran daya biasanya dinyatakan dalam elemen matrik admitansi bus, sehingga persamaan (4.6) menjadi:

$$V_i^{(k+1)} = \frac{\frac{P_i^{sch} - \mathrm{j}\, Q_i^{sch}}{V_i^{*(k)}} + \sum_{j \neq i} Y_{ij} V_j^{(k)}}{Y_{ii}} \quad \mathrm{j} \neq \mathrm{I} \tag{4.9}$$

dan

$$P_i^{(k+1)} = \mathcal{R} \left\{ V_i^{*(k)} \left[V_i^{(k)}\, Y_{ii} + \sum_{\substack{j=1 \\ j \neq 1}}^{n} Y_{ij}\ V_j^{(k)} \right] \right\} \quad j \neq i \tag{4.10}$$

$$Q_i^{(k+1)} = -\mathcal{I} \left\{ V_i^{*(k)} \left[V_i^{(k)}\, Y_{ii} + \sum_{\substack{j=1 \\ j \neq 1}}^{n} Y_{ij}\ V_j^{(k)} \right] \right\} \quad j \neq i \tag{4.11}$$

Y_{ii} termasuk admitansi ke tanah dari suseptansi charging saluran dan tap transformator. Karena kedua komponen tegangan slack bus ditentukan, maka terdapat 2(n-1) yang harus diselesaikan secara iteratif.

Dalam proses iterasi nilai awal tegangan diambil: 1 + j0.0. Untuk P-Q bus nilai P_i^{sch} dan Q_i^{sch} adalah diketahui. Mulai dengan tebakan awal tersebut persamaan (4.9) diselesaikan. Untuk P-V bus dimana P_i^{sch} dan $|V_i|$ diketahui, persamaan (4.11) diselesaikan untuk $Q_i^{(k+1)}$ dan menggunakan persamaan (4.9) untuk menyelesaikan $V_i^{(k+1)}$. Oleh karena $|V_i|$ diketahui, hanya bagian imajiner dari tegangan yang perlu dihitung, dengan syarat memenuhi persamaan berikut:

$$\left(e_i^{(k+1)}\right)^2 + \left(f_i^{(k+1)}\right)^2 = |V_i|^2 \tag{4.12}$$

dan

$$e_i^{(k+1)} \sqrt{|V_i|^2 - \left(f_i^{(k+1)}\right)^2} \tag{4.13}$$

Proses iterasi menuju konvergen dapat dipercepat dengan menggunkan faktor percepatan (α) seperti berikut:

$$V_i^{(k+1)} = V_i^{(k)} + \alpha\left(V_{i\,cal}^{(k)} - V_i^{(k)}\right) \tag{4.14}$$

dimana: α = 1.3 s/d 1.7

Proses iterasi akan berhenti sekiranya perbedaan tegangan iterasi telah memenuhi syarat toleransi yang diinginkan berikut:

$$\begin{aligned} \left|e_i^{(k+1)} - e_i^{(k)}\right| &\leq \in \\ \left|f_i^{(k+1)} - f_i^{(k)}\right| &\leq \in \end{aligned} \tag{4.15}$$

nilai ∈ biasanya diambil 0.00001

Setelah perhitungan iterasi konvergen, selanjutnya dapat dihitung nilai daya slack bus menggunakan persamaan (4.10) dan (4.11) di atas.

4.5 Metode Newton Raphson

Metode Newton Raphson dapat diterapkan dalam penyelesaian aliran daya. Persamaan (4.5) dapat diuraikan dan dinyatakan dalam bentuk polar menjadi dua persamaan berikut:

$$P_i = |V_i|^2 G_{ii} + \sum_{\substack{j=1 \\ j \neq 1}}^{N} |V_i V_j Y_{ij}| \cos(\theta_{ij} + \delta_j - \delta_i) \quad (4.16)$$

$$Q_i = |V_i|^2 B_{ii} + \sum_{\substack{j=1 \\ j \neq 1}}^{N} |V_i V_j Y_{ij}| \sin(\theta_{ij} + \delta_j - \delta_i) \quad (4.17)$$

Dimana:

$Y_{ij} = |Y_{ij}| \angle \theta_{ij} = G_{ij} + jB_{ij}$

$V_i = |V_i| \angle \delta_i$

$V_j = |V_j| \angle \delta_j$

Dalam bentuk rektagular dapat ditulis:

$$P_i = \sum_{j=1}^{N} |V_i||V_j|(G_{ij} \cos \delta_{ij} + B_{ij} \sin \delta_{ij}) \quad (4.18)$$

$$Q_i = \sum_{j=1}^{N} |V_i||V_j|(G_{ij} \sin \delta_{ij} - B_{ij} \cos \delta_{ij}) \quad (4.19)$$

Proses iterasi menggunakan metode Newton Raphson adalah:

1. Tentukan tebakan awal δ_i^0 dan $|V_i^0|$ untuk semua bus
2. Menggunakan nilai estimasi langkah satu hitung $P_i^{0,cal}$ dan $Q_i^{0,cal}$ menggunakan persamaan (4.18) dan (4.19).
3. Hitung mismatch:

   $\Delta P_i = P_i^{0,sch} - P_i^{0,cal}$ (4.20)

   $\Delta Q_i = Q_i^{0,sch} - Q_i^{0,cal}$ (4.21)

   Jika $\Delta P_i < Eps$ dan $\Delta Q_i < Eps$, Perhitungan selesai
4. Bentuk matrik Jacobian dan bentuk SPL berikut:

$$\begin{bmatrix} \frac{\partial P_2}{\partial \delta_2} & \cdots & \frac{\partial P_2}{\partial \delta_N} & |V_2|\frac{\partial P_2}{\partial |V_2|} & \cdots & |V_N|\frac{\partial P_2}{\partial |V_N|} \\ \vdots & J_{11} & \vdots & \vdots & J_{12} & \vdots \\ \frac{\partial P_N}{\partial \delta_2} & \cdots & \frac{\partial P_N}{\partial \delta_N} & |V_2|\frac{\partial P_N}{\partial |V_2|} & \cdots & |V_N|\frac{\partial P_N}{\partial |V_N|} \\ \frac{\partial Q_2}{\partial \delta_2} & \cdots & \frac{\partial Q_2}{\partial \delta_N} & |V_2|\frac{\partial Q_2}{\partial |V_2|} & \cdots & |V_N|\frac{\partial Q_2}{\partial |V_N|} \\ \vdots & J_{21} & \vdots & \vdots & J_{22} & \vdots \\ \frac{\partial Q_N}{\partial \delta_2} & \cdots & \frac{\partial Q_N}{\partial \delta_N} & |V_2|\frac{\partial Q_N}{\partial |V_2|} & \cdots & |V_N|\frac{\partial Q_N}{\partial |V_N|} \end{bmatrix} \begin{bmatrix} \Delta\delta_2 \\ \vdots \\ \Delta\delta_N \\ \frac{\Delta|V_2|}{|V_2|} \\ \vdots \\ \frac{\Delta|V_N|}{|V_N|} \end{bmatrix} = \begin{bmatrix} \Delta P_2 \\ \vdots \\ \Delta P_N \\ \Delta Q_2 \\ \vdots \\ \Delta Q_N \end{bmatrix} \quad (4.22)$$

5. Selesaikan persamaan (4.22) untuk mendapatkan perbaikan terhadap nilai awal

   $\Delta\delta_i^{(0)}$ dan $\frac{\Delta|V_i|^{(0)}}{|V_i|^{(0)}}$

6. Update nilai δ_i dan $|V_i|$ menggunakan persamaan berikut:

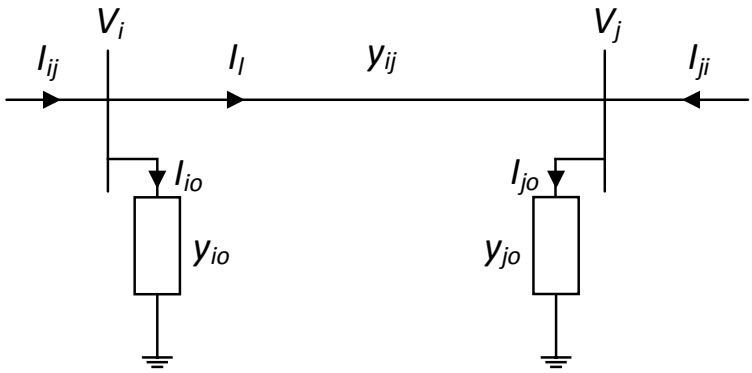
$$\delta_i^{(1)} = \delta_i^{(0)} + \Delta\delta_i^{(0)} \quad (4.23)$$

$$|V_i|^{(1)} = |V_i|^{(0)} + \Delta|V_i|^{(0)} = |V_i|^{(0)}\left(1 + \frac{\Delta|V_i|^{(0)}}{|V_i|^{(0)}}\right) \quad (4.24)$$

7. Gunakan nilai δi^1 dan $|Vi^1|$ sebagai nilai awal untuk iterasi berikutnya dan kembali ke langkah 1.

Proses iterasi metode Newton Raphson akan berhenti jika ΔP_i dan ΔQ_i sudah mendekati nol atau lebih kecil dari toleransi yang diinginkan.

Setelah proses iterasi selesai, selanjutnya menghitung daya aktif dan daya reaktif slack bus. Selanjutnya dapat dihitung aliran daya dan losses setiap jaringan menggunakan persamaan berikut:



Gambar 4.2 Model transmisi untuk perhitungan aliran daya dan losses

Arus dari bus i ke j adalah:

$$I_{ij} = I_i + I_{io} = y_{ij}(V_i - V_j) + Y_{io}V_i$$

Sedangkan arus dari bus j ke i adalah:

$$I_{ji} = -I_i + I_{jo} = y_{ij}(V_j - V_i) + Y_{jo}V_j$$

Daya komplek dari i ke j adalah:

$$S_{ij} = V_i I_{ij}^*$$

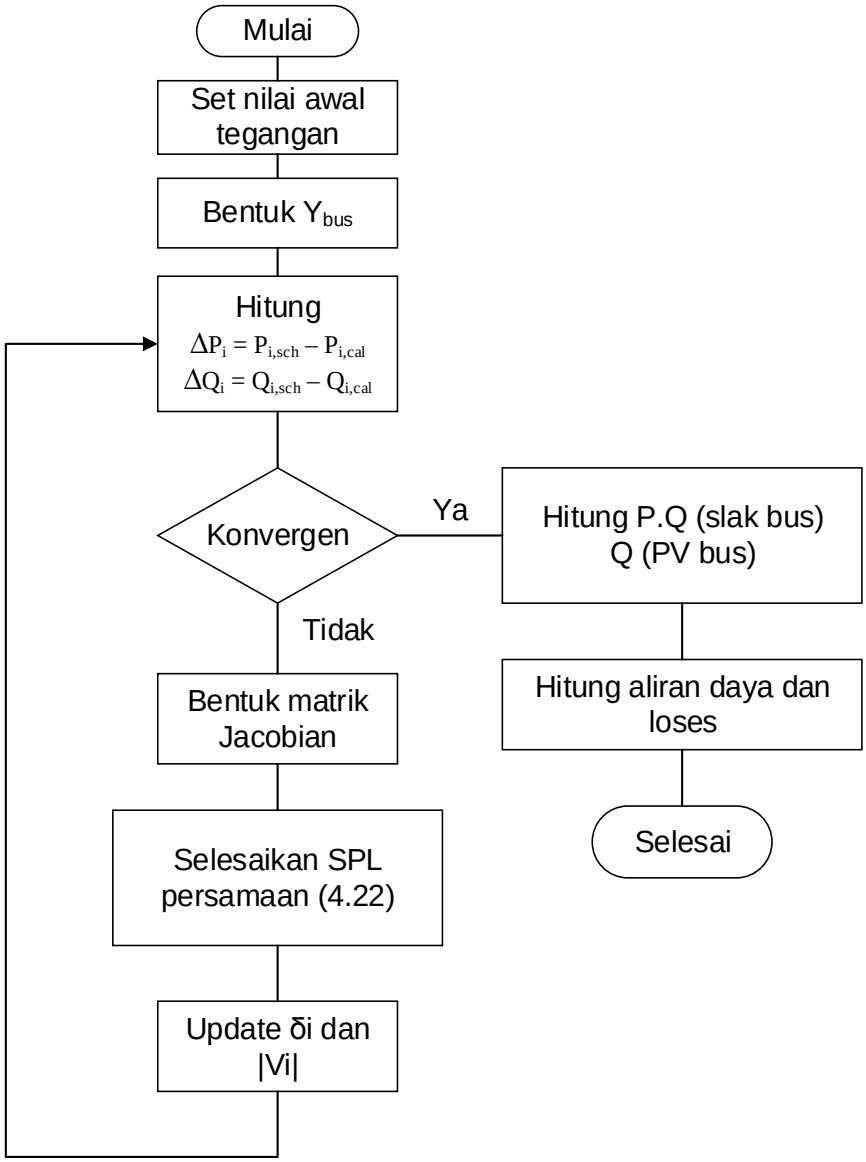
Sedangkan daya komplek dari bus j ke i adalah:

$$S_{ji} = V_j I_{ji}^*$$

Dengan demikian losses saliran ij adalah:

$$S_{losses} = S_{ij} + S_{ji} \quad (4.25)$$

Langkah-langkah perhitungan aliran daya menggunakan Metode Newton Raphson tersebut dapat nyatakan dalam bentuk diagram alir Gambar 4.3 berikut:



Gambar 4.3 Diagram alir Metode Newton Raphson

Metode Newton Raphson memiliki keunggulan dari metode Gauss Seidel karena metode ini lebih kokoh dalam penyelesaian SPNL dan cepat konvergen. Sedangkan kekurangannya adalah banyak membutuhkan memori karena penanganan matrik Jacobian pada setiap iterasi. Ukuran matrik Jacobian adalah:

Orde matrik Jacobian (2n-2-m) x (2n-2-m)

Orde J1 (n-1)x(n-1), J2 (n-1)x(n-1-m), J3 (n-1-m) x (n-1), J4 (n-1-m) x (n-1-m)

Dimana:

Slack bus = 1

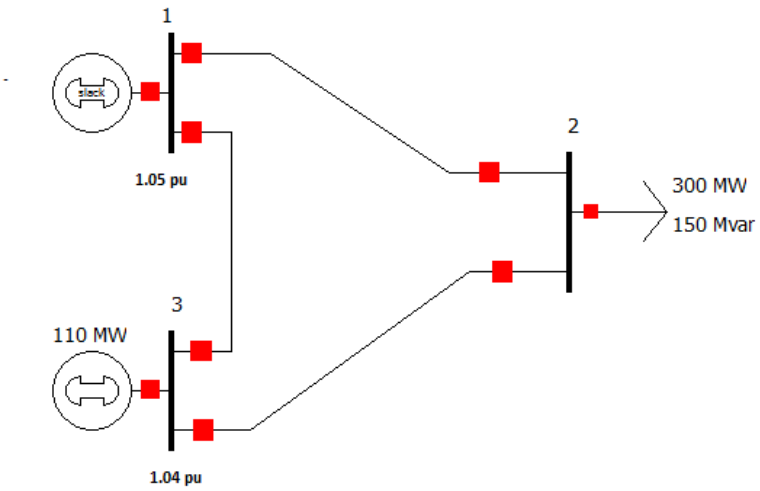
m = jumlah PV bus

n = jumlah bus

Kekurangan tersebut dapat diatasi dengan teknologi penyimpanan komputer atau penerapan metode Fast Decoupled. Penjelasan metode fast decoupled akan dijelaskan pada sub bab berikut.

Contoh 4.1

Suatu sistem tenaga tiga bus seperti contoh 3.1, jika diketahui data beban dan pembangkit sebagai berikut:



Gambar 4.4 Data beban dan pembangkit sistem 3 bus

Gunakan perhitungan manual (Mathcad) menggunakan metode Newton Raphson untuk menentukan nilai tegangan dan injeksi daya bus-bus yang belum diketahui? Bandingkan hasilnya dengan hasil running Powerworld.

Jawab:

Dari penyelesaian soal 3.1 diperoleh matrik admitansi bus sebagai berikut:

$$Y := \begin{pmatrix} 22 - 36i & -12 + 16i & -10 + 20i \\ -12 + 16i & 22 - 46i & -10 + 30i \\ -10 + 20i & -10 + 30i & 20 - 50i \end{pmatrix}$$

Selanjutnya dapat diperoleh:

$G := \mathrm{Re}(Y)$

$B := \mathrm{Im}(Y)$

$$G = \begin{pmatrix} 22 & -12 & -10 \\ -12 & 22 & -10 \\ -10 & -10 & 20 \end{pmatrix}$$

$$B = \begin{pmatrix} -36 & 16 & 20 \\ 16 & -46 & 30 \\ 20 & 30 & -50 \end{pmatrix}$$

Parameter sistem yang diketahui adalah:

Tegangan dan sudut tegangan slack bus atau bus 1 yaitu:

V_0 := 1.05 delta_0 := 0.0

Daya aktif dan daya reaktif beban yaitu:

Psch_1 := −3 Qsch_1 := −1.5

Daya aktif dan tegangan bus pembangkit yaitu:

Psch_2 := 1.1 V_2 := 1.04

Karena indek matrik dimulai dari 0, maka bus satu dalam perhitungan Mathcad diberi indek 0, bus dua berindek 1 dan bus tiga berindek 2.

Variabel state yang perlu dihitung adalah: δ_2 dan δ_3 serta $|V_2|$

Proses iterasi menggunakan metode Newton Raphson dimulai dengan menentukan tebakan awal δ_i^0 dan $|V_i^0|$ untuk semua bus seperti berikut:

delta_1 := 0.0

delta_2 := 0.0

V_1 := 1.0

Selanjutnya menggunakan nilai estimasi awal dihitung $P_i^0{}_{,cal}$ dan $Q_i^0{}_{,cal}$ menggunakan persamaan (4.18) dan (4.19) dalam bentuk rectangular berikut:

$$P_i = \sum_{j=0}^{2} |V_i||V_j|\left(G_{ij}\cos\delta_{ij} + B_{ij}\sin\delta_{ij}\right) \qquad (4.26)$$

$$Q_i = \sum_{j=0}^{2} |V_i||V_j|\left(G_{ij}\sin\delta_{ij} - B_{ij}\cos\delta_{ij}\right) \qquad (4.27)$$

diperoleh:

Untuk bus 2 atau indek 1:

Pcal_1 := V_1·[(G1,0·cos(delta_10) + B1,0·sin(delta_10))·V_0 + (G1,1·cos(delta_11) + B1,1·sin(delta_11))·V_1 + (G1,2·cos(delta_12) + B1,2·sin(delta_12))·V_2]

Pcal_1 = −1

Qcal_1 := V_1·[(G1,0·sin(delta_10) − B1,0·cos(delta_10))·V_0 + (G1,1·sin(delta_11) − B1,1·cos(delta_11))·V_1 + (G1,2·sin(delta_12) − B1,2·cos(delta_12))·V_2]

Qcal_1 = −2

dimana:

delta_10 := delta_1 − delta_0 delta_11 := 0.0 delta_12 := delta_1 − delta_2

Untuk bus 3 atau idek 3:

$$Pcal\_2 := V\_2\cdot\left[\left(G_{2,0}\cdot\cos(delta\_20) + B_{2,0}\cdot\sin(delta\_20)\right)\cdot V\_0 + \left(G_{2,1}\cdot\cos(delta\_21) + B_{2,1}\cdot\sin(delta\_21)\right)\cdot V\_1 + \left(G_{2,2}\cdot\cos(delta\_22) + B_{2,2}\cdot\sin(delta\_22)\right)\cdot V\_2\right]$$

Pcal_2 = 0.312

dimana:

delta_20 := delta_2 − delta_0 delta_21 := delta_2 − delta_1 delta_22 := 0.0

Selanjutnya hitung power mismatch menggunakan persamaan (4.18) dan (4.19):

Untuk bus 2 atau indek 1:

delta_P_1 := Psch_1 − Pcal_1 delta_Q_1 := Qsch_1 − Qcal_1

delta_P_1 = −2 delta_Q_1 = 0.5

Untuk bus 3 atau idek 3:

delta_P_2 := Psch_2 − Pcal_2

delta_P_2 = 0.788

Selanjutnya bentuk matrik Jacobian dalam format rectangular:

$$DP1DDelta1 := -V\_1\cdot\left[\left(G_{1,0}\cdot\sin(delta\_10) - B_{1,0}\cdot\cos(delta\_10)\right)\cdot V\_0 + \left(G_{1,2}\cdot\sin(delta\_12) - B_{1,2}\cdot\cos(delta\_12)\right)\cdot V\_2\right]$$

DP1DDelta1 = 48

$$DP1DDelta2 := V\_1\cdot\left[\left(G_{1,2}\cdot\sin(delta\_12) - B_{1,2}\cdot\cos(delta\_12)\right)\cdot V\_2\right]$$

DP1DDelta2 = −31.2

$$DP1DV1 := 2\cdot V\_1\cdot G_{1,1} + \left(G_{1,0}\cdot\cos(delta\_10) + B_{1,0}\cdot\sin(delta\_10)\right)\cdot V\_0 + \left(G_{1,2}\cdot\cos(delta\_12) + B_{1,2}\cdot\sin(delta\_22)\right)\cdot V\_2$$

DP1DV1 = 21

$$DP2DDelta1 := V\_2\cdot\left[\left(G_{2,1}\cdot\sin(delta\_21) - B_{2,1}\cdot\cos(delta\_21)\right)\cdot V\_1\right]$$

DP2DDelta1 = −31.2

$$DP2DDelta2 := -V\_2\cdot\left[\left(G_{2,0}\cdot\sin(delta\_20) - B_{2,0}\cdot\cos(delta\_20)\right)\cdot V\_0 + \left(G_{2,1}\cdot\sin(delta\_21) - B_{2,1}\cdot\cos(delta\_21)\right)\cdot V\_1\right]$$

DP2DDelta2 = 53.04

$$DP2DV1 := \left(G_{1,2}\cdot\cos(delta\_12) + B_{2,2}\cdot\sin(delta\_22)\right)\cdot V\_2$$

DP2DV1 = −10.4

$$DQ1DDelta1 := V\_1\cdot\left[\left(G_{1,0}\cdot\cos(delta\_10) + B_{1,0}\cdot\sin(delta\_10)\right)\cdot V\_0 + \left(G_{1,2}\cdot\cos(delta\_12) + B_{1,2}\cdot\sin(delta\_12)\right)\cdot V\_2\right]$$

DQ1DDelta1 = −23

$$DQ1DDelta2 := -V\_1 \cdot V\_2 \cdot \left(G_{1,2} \cdot \cos(delta\_12) + B_{1,2} \cdot \sin(delta\_12)\right)$$

$$DQ1DDelta2 = 10.4$$

$$DQ1DV1 := -2 \cdot V\_1 \cdot B_{1,1} + \left(G_{1,0} \cdot \sin(delta\_10) - B_{1,0} \cdot \cos(delta\_10)\right) \cdot V\_0 + \left(G_{1,2} \cdot \sin(delta\_12) - B_{1,2} \cdot \cos(delta\_22)\right) \cdot V\_2$$

$$DQ1DV1 = 44$$

Sehingga diperoleh matrik Jacobian sebagai berikut:

$$J := \begin{pmatrix} DP1DDelta1 & DP1DDelta2 & DP1DV1 \\ DP2DDelta1 & DP2DDelta2 & DP2DV1 \\ DQ1DDelta1 & DQ1DDelta2 & DQ1DV1 \end{pmatrix}$$

$$J = \begin{pmatrix} 48 & -31.2 & 21 \\ -31.2 & 53.04 & -10.4 \\ -23 & 10.4 & 44 \end{pmatrix}$$

Vektor ruas kanan SPL:

$$D\_P\_Q := \begin{pmatrix} delta\_P\_1 \\ delta\_P\_2 \\ delta\_Q\_1 \end{pmatrix} = \begin{pmatrix} -2 \\ 0.788 \\ 0.5 \end{pmatrix}$$

Selanjutnya bentuk sistem persamaan linear berikut:

$$\begin{bmatrix} \frac{\partial P_2}{\partial \delta_2} & \frac{\partial P_2}{\partial \delta_3} & \frac{\partial P_2}{\partial |V_2|} \\ \frac{\partial P_3}{\partial \delta_2} & \frac{\partial P_3}{\partial \delta_3} & \frac{\partial P_3}{\partial |V_2|} \\ \frac{\partial Q_2}{\partial \delta_2} & \frac{\partial Q_2}{\partial \delta_3} & \frac{\partial Q_2}{\partial |V_2|} \end{bmatrix} \begin{bmatrix} \Delta\delta_2 \\ \Delta\delta_3 \\ \Delta|V_2| \end{bmatrix} = \begin{bmatrix} \Delta P_2 \\ \Delta P_3 \\ \Delta Q_2 \end{bmatrix}$$

Variabel state diperoleh melalui proses penyelesaian SPL menggunakan invers matrik jacobian berikut:

$$D\_V\_Delta := J^{-1} \cdot D\_P\_Q = \begin{pmatrix} -0.04695 \\ -0.01467 \\ -0.00971 \end{pmatrix}$$

Selanjutnya update tegangan dan sudut tegangan bus 2:

$$V\_1 := V\_1 + \left(D\_V\_Delta_{2,0}\right) = 0.99029$$

$$delta\_1 := delta\_1 + D\_V\_Delta_{0,0} = -0.04695$$

update sudut tegangan bus 3:

$$delta\_2 := delta\_2 + D\_V\_Delta_{1,0} = -0.01467$$

Selanjutnya nilai tegangan dan sudut tegangan tersebut menjadi nilai awal iterasi berikutnya.

Iterasi ke 2

Selanjutnya menggunakan nilai estimasi awal hasil iterasi 1 dihitung $P_i^{1},_{cal}$ dan $Q_i^{1},_{cal}$ menggunakan persamaan (4.16) dan (4.17) dalam bentuk rectangular berikut:

Untuk bus 2 atau indek 1:

$$Pcal\_1 = V\_1\left[\left(G_{1,0}\cos(delta\_10) + B_{1,0}\sin(delta\_10)\right)\cdot V\_0 + \left(G_{1,1}\cos(delta\_11) + B_{1,1}\sin(delta\_11)\right)\cdot V\_1 + \left(G_{1,2}\cos(delta\_12) + B_{1,2}\sin(delta\_12)\right)\cdot V\_2\right]$$

$$Pcal\_1 = -2.9609$$

$$Qcal\_1 = V\_1\left[\left(G_{1,0}\sin(delta\_10) - B_{1,0}\cos(delta\_10)\right)\cdot V\_0 + \left(G_{1,1}\sin(delta\_11) - B_{1,1}\cos(delta\_11)\right)\cdot V\_1 + \left(G_{1,2}\sin(delta\_12) - B_{1,2}\cos(delta\_12)\right)\cdot V\_2\right]$$

$$Qcal\_1 = -1.47051$$

Untuk bus 3 atau indek 2:

$$Pcal\_2 = V\_2\left[\left(G_{2,0}\cos(delta\_20) + B_{2,0}\sin(delta\_20)\right)\cdot V\_0 + \left(G_{2,1}\cos(delta\_21) + B_{2,1}\sin(delta\_21)\right)\cdot V\_1 + \left(G_{2,2}\cos(delta\_22) + B_{2,2}\sin(delta\_22)\right)\cdot V\_2\right]$$

$$Pcal\_2 = 1.0966$$

Selanjutnya hitung kembali power mismatch menggunakan persamaan (4.18) dan (4.19):

Untuk bus 2 atau indek 1:

$$delta\_P\_1 := Psch\_1 - Pcal\_1 \qquad delta\_Q\_1 := Qsch\_1 - Qcal\_1$$

$$delta\_P\_1 = -0.0391 \qquad delta\_Q\_1 = -0.02949$$

Untuk bus 3 atau idek 3:

$$delta\_P\_2 := Psch\_2 - Pcal\_2$$

$$delta\_P\_2 = 0.0034$$

Selanjutnya bentuk matrik Jacobian dalam format rectangular untuk iterasi ke 2:

$$J := \begin{pmatrix} DP1DDelta1 & DP1DDelta2 & DP1DV1 \\ DP2DDelta1 & DP2DDelta2 & DP2DV1 \\ DQ1DDelta1 & DQ1DDelta2 & DQ1DV1 \end{pmatrix}$$

$$J = \begin{pmatrix} 46.58133 & -30.54844 & 19.8035 \\ -31.21332 & 52.89084 & -10.39458 \\ -24.53565 & 11.29096 & 44.05206 \end{pmatrix}$$

Vektor ruas kanan SPL:

$$D\_P\_Q := \begin{pmatrix} delta\_P\_1 \\ delta\_P\_2 \\ delta\_Q\_1 \end{pmatrix} = \begin{pmatrix} -0.0391 \\ 0.0034 \\ -0.02949 \end{pmatrix}$$

Variabel state diperoleh melalui proses penyelesaian SPL menggunakan invers matrik jacobian berikut:

$$D\_V\_Delta := J^{-1} \cdot D\_P\_Q = \begin{pmatrix} -0.00083 \\ -0.00062 \\ -0.00097 \end{pmatrix}$$

Selanjutnya update tegangan dan sudut tegangan bus 2:

$$V\_1 := V\_1 + \left(D\_V\_Delta_{2,0}\right) = 0.98931$$

$$delta\_1 := delta\_1 + D\_V\_Delta_{0,0} = -0.04778$$

update sudut tegangan bus 3:

$$delta\_2 := delta\_2 + D\_V\_Delta_{1,0} = -0.01528$$

Selanjutnya nilai tegangan dan sudut tegangan tersebut menjadi nilai awal iterasi berikutnya.

Iterasi ke 3

Selanjutnya menggunakan nilai estimasi awal hasil iterasi 1 dihitung $P_i^2{}_{,cal}$ dan $Q_i^2{}_{,cal}$ menggunakan persamaan (4.16) dan (4.17) dalam bentuk rectangular berikut:

Untuk bus 2 atau indek 1:

$$Pcal\_1 := V\_1 \left[\left| G_{1,0} \cos(delta\_10) + B_{1,0} \sin(delta\_10) \right| \cdot V\_0 + \left| G_{1,1} \cos(delta\_11) + B_{1,1} \sin(delta\_11) \right| \cdot V\_1 + \left| G_{1,2} \cos(delta\_12) + B_{1,2} \sin(delta\_12) \right| \cdot V\_2 \right]$$

$$Pcal\_1 = -2.99897$$

$$Qcal\_1 := V\_1 \left[\left| G_{1,0} \sin(delta\_10) - B_{1,0} \cos(delta\_10) \right| \cdot V\_0 + \left| G_{1,1} \sin(delta\_11) - B_{1,1} \cos(delta\_11) \right| \cdot V\_1 + \left| G_{1,2} \sin(delta\_12) - B_{1,2} \cos(delta\_12) \right| \cdot V\_2 \right]$$

$$Qcal\_1 = -1.49998$$

Untuk bus 3 atau indek 2:

$$Pcal\_2 := V\_2 \left[\left| G_{2,0} \cos(delta\_20) + B_{2,0} \sin(delta\_20) \right| \cdot V\_0 + \left| G_{2,1} \cos(delta\_21) + B_{2,1} \sin(delta\_21) \right| \cdot V\_1 + \left| G_{2,2} \cos(delta\_22) + B_{2,2} \sin(delta\_22) \right| \cdot V\_2 \right]$$

$$Pcal\_2 = 1.09902$$

Selanjutnya hitung kembali power mismatch menggunakan persamaan (4.18) dan (4.19):

Untuk bus 2 atau indek 1:

$$delta\_P\_1 := Psch\_1 - Pcal\_1 \qquad delta\_Q\_1 := Qsch\_1 - Qcal\_1$$

$$delta\_P\_1 = -0.00103 \qquad delta\_Q\_1 = -2.08105 \times 10^{-5}$$

Untuk bus 3 atau idek 3:

$$delta\_P\_2 := Psch\_2 - Pcal\_2$$

$$delta\_P\_2 = 0.00098$$

Selanjutnya bentuk matrik Jacobian dalam format rectangular untuk iterasi ke 2:

$$J := \begin{pmatrix} DP1DDelta1 & DP1DDelta2 & DP1DV1 \\ DP2DDelta1 & DP2DDelta2 & DP2DV1 \\ DQ1DDelta1 & DQ1DDelta2 & DQ1DV1 \end{pmatrix}$$

$$J = \begin{pmatrix} 46.52215 & -30.516 & 19.7473 \\ -31.18462 & 52.85519 & -10.39451 \\ -24.53131 & 11.28636 & 43.9758 \end{pmatrix}$$

Vektor ruas kanan SPL:

$$D\_P\_Q := \begin{pmatrix} delta\_P\_1 \\ delta\_P\_2 \\ delta\_Q\_1 \end{pmatrix} = \begin{pmatrix} -0.00103 \\ 0.00098 \\ -2.08105 \times 10^{-5} \end{pmatrix}$$

Variabel state diperoleh melalui proses penyelesaian SPL menggunakan invers matrik jacobian berikut:

$$D\_V\_Delta := J^{-1} \cdot D\_P\_Q = \begin{pmatrix} -1.14787 \times 10^{-5} \\ 1.00012 \times 10^{-5} \\ -9.44329 \times 10^{-6} \end{pmatrix}$$

Selanjutnya update tegangan dan sudut tegangan bus 2:

$$V\_1 := V\_1 + \left(D\_V\_Delta_{2,0}\right) = 0.9893$$

$$delta\_1 := delta\_1 + D\_V\_Delta_{0,0} = -0.04779$$

update sudut tegangan bus 3:

$$delta\_2 := delta\_2 + D\_V\_Delta_{1,0} = -0.01527$$

Selanjutnya nilai tegangan dan sudut tegangan tersebut menjadi nilai awal iterasi berikutnya.

Iterasi ke 4

Selanjutnya menggunakan nilai estimasi awal hasil iterasi 1 dihitung $P_i^{3}{}_{,cal}$ dan $Q_i^{3}{}_{,cal}$ menggunakan persamaan (4.16) dan (4.17) dalam bentuk rectangular berikut:

Untuk bus 2 atau indek 1:

$$Pcal\_1 = V\_1 \left[\left(G_{1,0} \cos(delta\_10) + B_{1,0} \sin(delta\_10)\right) \cdot V\_0 + \left(G_{1,1} \cos(delta\_11) + B_{1,1} \sin(delta\_11)\right) \cdot V\_1 + \left(G_{1,2} \cos(delta\_12) + B_{1,2} \sin(delta\_12)\right) \cdot V\_2 \right]$$

$$Pcal\_1 = -2.99999$$

$$Qcal\_1 = V\_1 \left[\left(G_{1,0} \sin(delta\_10) - B_{1,0} \cos(delta\_10)\right) \cdot V\_0 + \left(G_{1,1} \sin(delta\_11) - B_{1,1} \cos(delta\_11)\right) \cdot V\_1 + \left(G_{1,2} \sin(delta\_12) - B_{1,2} \cos(delta\_12)\right) \cdot V\_2 \right]$$

$$Qcal\_1 = -1.5$$

Untuk bus 3 atau indek 2:

$$Pcal\_2 = V\_2 \left[\left(G_{2,0} \cos(delta\_20) + B_{2,0} \sin(delta\_20)\right) \cdot V\_0 + \left(G_{2,1} \cos(delta\_21) + B_{2,1} \sin(delta\_21)\right) \cdot V\_1 + \left(G_{2,2} \cos(delta\_22) + B_{2,2} \sin(delta\_22)\right) \cdot V\_2 \right]$$

$$Pcal\_2 = 1.09999$$

Selanjutnya hitung kembali power mismatch menggunakan persamaan (4.18) dan (4.19):

Untuk bus 2 atau indek 1:

$delta\_P\_1 := Psch\_1 - Pcal\_1$ $\quad$ $delta\_Q\_1 := Qsch\_1 - Qcal\_1$

$delta\_P\_1 = -9.58663 \times 10^{-6}$ $\quad$ $delta\_Q\_1 = 1.47137 \times 10^{-7}$

Untuk bus 3 atau idek 3:

$delta\_P\_2 := Psch\_2 - Pcal\_2$

$delta\_P\_2 = 9.57687 \times 10^{-6}$

Power Mismatch ΔP_i dan ΔQ_i sudah sangat kecil atau < dari 10^{-5}, maka proses iterasi dapat dihentikan. Sehingga hasil akhir diambil nilai pada iterasi ke 3 yaitu:

$|V_1| = 0.9893$

$\delta_1 = -0.04779$ rad

$\delta_2 = -0.01527$ rad

Selanjutnya dapat dihitung nilai injeksi daya pada slack bus atau bus 1 sebagai berikut:

$$Pcal\_0 := V\_0\left[\left|G_{0,0}\right|\cdot\cos(delta\_00) + B_{0,0}\cdot\sin(delta\_00)\right]\cdot V\_0 + \left[\left|G_{0,1}\right|\cdot\cos(delta\_01) + B_{0,1}\cdot\sin(delta\_01)\right]\cdot V\_1 + \left[\left|G_{0,2}\right|\cdot\cos(delta\_02) + B_{0,2}\cdot\sin(delta\_02)\right]\cdot V\_2\right]$$

$Pcal\_0 = 2.0128$

$$Qcal\_0 := V\_0\left[\left|G_{0,0}\right|\cdot\sin(delta\_00) - B_{0,0}\cdot\cos(delta\_00)\right]\cdot V\_0 + \left[\left|G_{0,1}\right|\cdot\sin(delta\_01) - B_{0,1}\cdot\cos(delta\_01)\right]\cdot V\_1 + \left[\left|G_{0,2}\right|\cdot\sin(delta\_02) - B_{0,2}\cdot\cos(delta\_02)\right]\cdot V\_2\right]$$

$Qcal\_0 = 0.48893$

dimana:

$delta\_00 := 0$ $\quad$ $delta\_01 := delta\_0 - delta\_1$ $\quad$ $delta\_02 := delta\_0 - delta\_2$

Dan daya reaktif bus pembangkit atau bus 3:

$$Qcal\_2 := V\_2\left[\left|G_{2,0}\right|\cdot\sin(delta\_20) - B_{2,0}\cdot\cos(delta\_20)\right]\cdot V\_0 + \left[\left|G_{2,1}\right|\cdot\sin(delta\_21) - B_{2,1}\cdot\cos(delta\_21)\right]\cdot V\_1 + \left[\left|G_{2,2}\right|\cdot\sin(delta\_22) - B_{2,2}\cdot\cos(delta\_22)\right]\cdot V\_2\right]$$

$Qcal\_2 = 1.22479$

dimana:

$delta\_20 := delta\_2 - delta\_0$ $\quad$ $delta\_21 := delta\_2 - delta\_1$ $\quad$ $delta\_22 := 0.0$

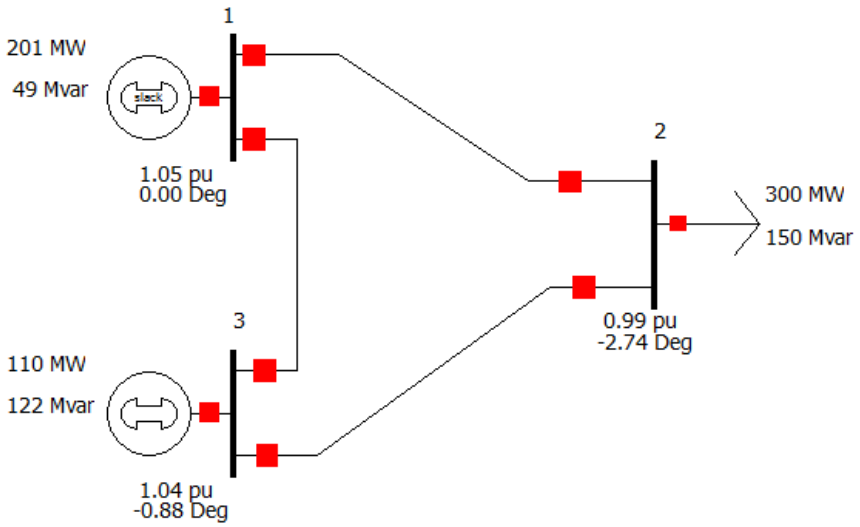
Dengan demikian diperoleh ijeksi daya:

$P_1 = 201.28$ MW

$Q_1 = 48.893$ MVar

$Q_3 = 122.48$ Mvar

Hasil running program powerworld adalah:



Gambar 4.5 Hasil running Powerworld sistem 3 bus

Dibandingkan dengan hasil powerworld Gambar 4.5 dapat disimpulkan bahwa hasilnya sudah mendekati sama.

4.6 Metode Fast Decoupled

Metode *fast decoupled* adalah suatu metode yang diturunkan dari penyederhanaan sekaligus memperbaiki efisiensi perhitungan metode *Newton Raphson* yang memiliki konvergensi yang kuadratis namun banyak memerlukan memori dan waktu.

Dasar penyederhanaan adalah :

Saluran transmisi memiliki rasio X/R yang tinggi sehingga tahanan sistem << rektansi sistem.

Sudut tegangan (δ) cukup kecil sehingga cos δ ≈ 1 dan sin δ ≈ 0

Perubahan daya aktif tidak sensitif terhadap perubahan besar tegangan tetapi lebih sensitif terhadap perubahan sudut tegangan.

Perubahan daya reaktif tidak sensitif terhadap perubahan sudut tegangan tetapi lebih sensitif terhadap perubahan besar tegangan.

Sehingga submatriks J_2 dan J_3 dari matrik jacobian dapat diabaikan dan persamaan Newton – Raphson dapat ditulis sebagai berikut:

$$\begin{bmatrix}\Delta P\\ \Delta Q\end{bmatrix} = \begin{bmatrix}J_1 & 0\\ 0 & J_4\end{bmatrix}\begin{bmatrix}\Delta\delta\\ \frac{\Delta|V|}{|V|}\end{bmatrix} \quad (4.28)$$

Atau

$$[\Delta P] = [J_1][\Delta\delta] \tag{4.29}$$

$$[\Delta Q] = [J_4]\left[\frac{\Delta|V|}{|V|}\right] \tag{4.30}$$

Disini terlihat persamaan dapat diselesaikan secara terurai (*decoupled*) dalam bentuk hubungan P dengan δ dan Q dengan |V|.

Penyederhanaan lebih lanjut adalah penggunaan Jacobian yang sama untuk semua iterasi. Bentuk umum persamaan *fast decoupled* adalah:

$$\left[\frac{\Delta P}{|V|}\right] = [B'][\Delta\delta] \tag{4.31}$$

$$\left[\frac{\Delta Q}{|V|}\right] = [B"][\Delta|V|] \tag{4.32}$$

Dimana elemen-elemen matrik B diperoleh dari pendekatan-pendekatan dan penyederhanaan dalam usaha memperbaiki keakuratan dan kecepatan konvergensi metode fast decoupled. Nilai B' dan B" dapat dihitung menggunakan:

Tabel 4.2 Variasi Metode Fast Decoupled

Metode	B'	B"
BB	$R_{ij} \neq 0$	$R_{ij} \neq 0$
BX	$R_{ij} \neq 0$	$R_{ij} = 0$
XB	$R_{ij} = 0$	$R_{ij} \neq 0$
XX	$R_{ij} = 0$	$R_{ij} = 0$

Dari keempat metode tersebut metode XB yang umumnya digunakan. Sehingga untuk metode XB:

Elemen matrik B' adalah:

Luar diagonal :

$$B_{ij}^{'} = \frac{-1}{x_{ij}}$$

Diagonal:

$$B_{ii}^{'} = \sum_{\substack{j=1 \\ j \neq i}}^{N} \frac{-1}{x_{ij}}$$

Elemen matrik B" merupakan komponen imaginer matrik [-Y_{bus}] yaitu:

Luar diagonal:

$$B_{ij}^{"} = \frac{-x_{ij}}{r_{ij}^2 + x_{ij}^2}$$

Diagonal :

$$B_{ii}^{"} = \sum_{\substack{j=1 \\ j \neq i}}^{N} \left(\frac{-x_{ij}}{r_{ij}^2 + x_{ij}^2} + \frac{y_{ij}^{Shunt}}{2} \right)$$

Y^{shunt} adalah semua admitansi shunt di bus i (termasuk line charging, tap trafo di luar nominal, kapasitor yang terpasang shunt, dan lain – lain).

Ukuran matrik B1→(N - 1) x (N - 1) → semua bus kecuali *slack*

B2→ (N - 1- PV) x (N - 1- PV) → hanya PQ bus saja

Jika menggunakan full matrik, maka ukuran matrik tetap sama tetapi baris dan kolom posisi slack bus dan PV bus bernilai besar misalkan: 10^{10}. Sehingga elemen matrik B' dan B" seperti tabel 4.3 berikut:

Tabel 4.3 Elemen Matrik B' dan B" sistem full matrik

FD/XB		**B1**	**B2**
Mutual		$B_{ij}^{'} = \frac{-1}{x_{ij}}$	$B_{ij}^{"} = \frac{-x_{ij}}{r_{ij}^2 + x_{ij}^2}$
Self	PQ Bus	$B_{ii}^{'} = \sum_{\substack{j=1 \\ j \neq i}}^{N} \frac{-1}{x_{ij}}$	$B_{ii}^{"} = \sum_{\substack{j=1 \\ j \neq i}}^{N} \left(\frac{-x_{ij}}{r_{ij}^2 + x_{ij}^2} + \frac{y_{ij}^{Shunt}}{2} \right)$
	PV		Bilangan besar: 10^{+10}
	Slack	Bilangan besar: 10^{+10}	Bilangan besar: 10^{+10}

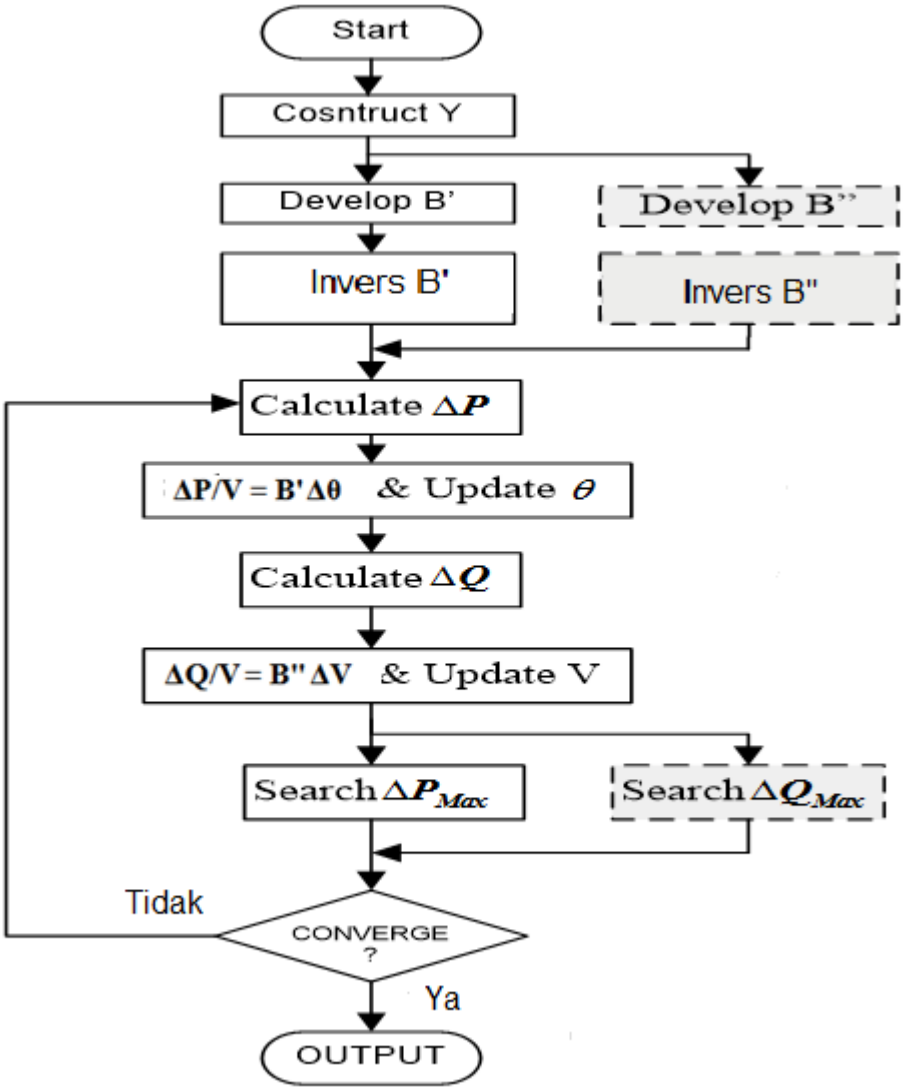
Elemen diagonal yang bertepatan dengan slak bus dan PV bus bernilai besar misalnya diambil $10+^{10}$.

4.6.1. Proses Perhitungan Aliran Daya Metode Fast Decoupled

Perhitungan aliran daya Metode Fast Decoupled menggunakan fungsi invers matrik B' dan B" yang dilakukan sekali sebelum bagian perulangan atau iteratif. Dengan demikian akan sangat menghemat memori dan waktu komputasi. Langkah-langkah perhitungan aliran daya metode Fast Decoupled dapat dituliskan sebagai berikut:

1. Bentuk matrik Admitansi Bus
2. Bentuk matrik B'
3. Invers B'
4. Bentuk Matrik B"
5. Invers B"
6. Proses Iterasi
    a. Hitung P dan Delta P
    b. Hitung delta theta = $-[B1]^{-1}$ (delta_P/|v|)
    c. Update theta
    d. Hitung Q dan Delta Q
    e. Hitung delta V = $-[B2]^{-1}$(delta_Q/|v|)
    f. Update V
    g. Hitung Delta mak
    h. Jika Delta mak < dari mismatch pergi ke langkah 7 jika tidak ulangi langkah 6
7. Keluar dari iterasi
8. Hitung daya PQ Slack bus dan Q PV bus
9. Hitung daya total beban dan pembangkit
10. Print hasil tegangan dan ijeksi daya bus
11. Hitung aliran daya dan losses
12. Print aliran daya dan losses
13. Selesai

Proses Perhitungan aliran daya metode Fast Decoupled dapat digambarkan dalam diagram alir berikut:



Gambar 4.6 Diagram Alir Metode Fast Decoupled

4.6.2. Program Perhitungan Aliran Daya Metode Fast Decoupled

Program Visual C++ untuk perhitungan matrik B' adalah

```
for (l=1;l<=nl;l++){
       i = sl[l];
       j = el[l];
       B1[i][j] = 1.0/xl[l];
       B1[j][i] = B1[i][j];
       B1[i][i] += -1.0/xl[l];
       B1[j][j] += -1.0/xl[l];
}
for (k=1;k<=nb;++k)
if (Type[k]==slack)
B1[k][k]=10e70;
```

Kemudian lakukan invers matrik B' menggunakan fungsi invers matrik sebagaimana yang telah dijelaskan dalam bab 2.

```
//input B1 to invers matrix function
 for (i=1; i<=nb; ++i)
   {
   for (j=1; j<=nb; ++j)
     {
     a[i][j]=B1[i][j];
     }
   }

 //invers B1
 A.invm(a,nb);
 for (i=1;i<=nb;i++)
 {
 for (j=1;j<=nb;j++)
   {
   B1_inv[i][j]= a[i][j];
   }
 }
```

Hasil invers matrik B' ditampung dalam variabel B1_inv[i][j].

Selanjutnya program Visual C++ untuk perhitungan matrik B" adalah:

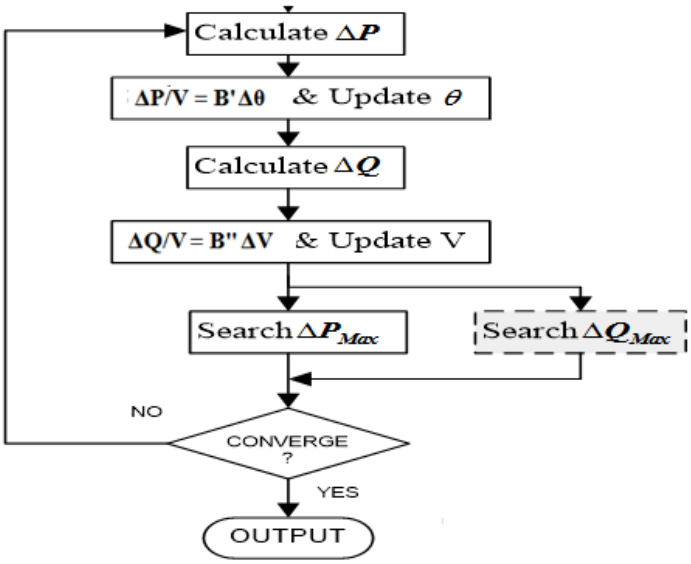
```
      for (i=1;i<=nb;++i)
         {
          for (j=1; j<=nb; ++j)
             {
                B2[i][j]=b[i][j];
              }
         }
       for (k=1;k<=nb;++k)
       if (Type[k]== slack || Type[k]==PV)
         B2[k][k]=10e70 ;
```

Kemudian lakukan invers matrik B" menggunakan fungsi invers matrik sebagaimana yang telah dijelaskan dalam bab 2.

```
//input B2 to invers matrix function
 for (i=1; i<=nb; ++i)
   {
   for (j=1; j<=nb; ++j)
     {
     a[i][j]=B2[i][j];
     }
   }

 //invers B2
 A.invm(a,nb);
 for (i=1;i<=nb;i++)
 {
 for (j=1;j<=nb;j++)
   {
   B2_inv[i][j]=a[i][j];
   }
 }
```

Hasil invers matrik B" ditampung dalam variabel B2_inv[i][j]. Proses invers matrik dilakukan satu kali sebelum proses iterasi.

Selanjutnya Proses iterasi (Perulangan)



Gambar 4.7 Skema Iterasi Metode Fast Decoupled

Kode program proses iterasi metode Fast Decoupled adalah:

```
iter=1;
while (iter<100 && DelMax>=0.001)
{
       Proses ....
       Proses ....
       .
       .
       .
       iter=iter+1;
 }
```

Proses-proses penting dalam iterasi adalah

1. Perhitungan P and delta_P

```
for (i=1; i<=nb; ++i){
   ps[i]=0.0;
   if (Type[i]==PQ||Type[i]==PV){
     psch[i]=pg[i]-pl[i];
     for (k=1; k<=nb; ++k)
     ps[i]+=(g[i][k]*cos(theta[i]-theta[k]) +b[i][k]*sin(theta[i]-
     theta[k]))*v[k]*v[i];
     deltap[i]=psch[i]-ps[i];
     }
   }
```

2. Perhitungan delta theta = $-[B1]^{-1}(delta\_P/|v|)$

```
for (i=1;i<=nb;i++){
  delta_theta[i]=0;
  for (j=1;j<=nb;j++)
  delta_theta[i]=delta_theta[i]-B1_inv[i][j]*deltap[j];
}
//Up date theta
```

```
for (i=1;i<=nb;i++){
     theta[i] = theta[i]+delta_theta[i];
 }
```

3. Perhitungan Q and delta_Q

```
for (i=1; i<=nb; ++i){
     qs[i]=0.0;
     if (Type[i]==PQ){
       qsch[i]=qg[i]-ql[i];
       for (k=1; k<=nb; ++k)
       qs[i]+=(g[i][k]*sin(theta[i]-theta[k])-b[i][k]*cos(theta[i]-
theta[k]))*v[k]*v[i];
       deltaq[i]=qsch[i]-qs[i];
     }
 }
```

4. Perhitungan delta V = $-[B2]^{-1}$(delta_Q/|v|)

```
for (i=1;i<=nb;i++){
  delta_v[i]=0;
  for (j=1;j<=nb;j++){
    delta_v[i]= delta_v[i]-B2_inv[i][j]*deltaq[j];
  }
}
//Up date theta v
for (i=1;i<=nb;i++){
     v[i]=v[i]+ delta_v[i];
}
```

5. Perhitungan Kriteria Penghentian

```
for(i=1;i<=nb;i++){
     if (Type[i]==PV||Type[i]==PQ){
       if (fabs(deltap[i]) >= DelPmax)
       DelPmax = fabs(deltap[i]);
       if (Type[i]==PQ){
         if (fabs(deltaq[i]) >= DelQmax)
         DelQmax = fabs(deltaq[i]);
       }
    }
 }
 if (DelPmax >= DelQmax)
 DelMax=DelPmax;
 else
 DelMax=DelQmax;
```

Perhitungan (P and Q) slack bus dan (Q) PV bus

Setelah proses iterasi selesai, selanjutnya dilakukan per Perhitungan (P and Q) slack bus dan (Q) PV bus menggunakan program berikut:

```
for (i=1; i<=nb; ++i){
     if (Type[i]==slack){
       ps[i]=0.0;
       qs[i]=0.0;
       for (k=1; k<=nb; ++k){
```

```
            ps[i]+=(g[i][k]*cos(theta[i]-theta[k])+b[i][k]*sin(theta[i]-
            theta[k]))*v[k]*v[i];
            qs[i]+=(g[i][k]*sin(theta[i]-theta[k])-b[i][k]*cos(theta[i]-
            theta[k]))*v[k]*v[i];
          }
          pg[i]=ps[i]+pl[i];
          qg[i]=qs[i]+ql[i];
        }
        if (Type[i]==PV){
          qs[i]=0.0;
          for (k=1; k<=nb; ++k){
            qs[i]+=(g[i][k]*sin(theta[i]-theta[k])-b[i][k]*cos(theta[i]-
            theta[k]))*v[k]*v[i];
          }
          qg[i]=qs[i]+ql[i];
        }
}
```

Perhitungan daya aktif dan daya reaktif total pembagkit dan beban:

```
PGtotal=0.0;
 QGtotal=0.0;
 for (i=1;i<=nb;i++){
   PGtotal+=pg[i]*base;
   QGtotal+=qg[i]*base;
 }
 PLtotal=0.0;
 QLtotal=0.0;
 for (i=1;i<=nb;i++){
   PLtotal+=pl[i]*base;
   QLtotal+=ql[i]*base;
}
```

Selanjutnya tampilkan hasil berhitungan aliran daya:

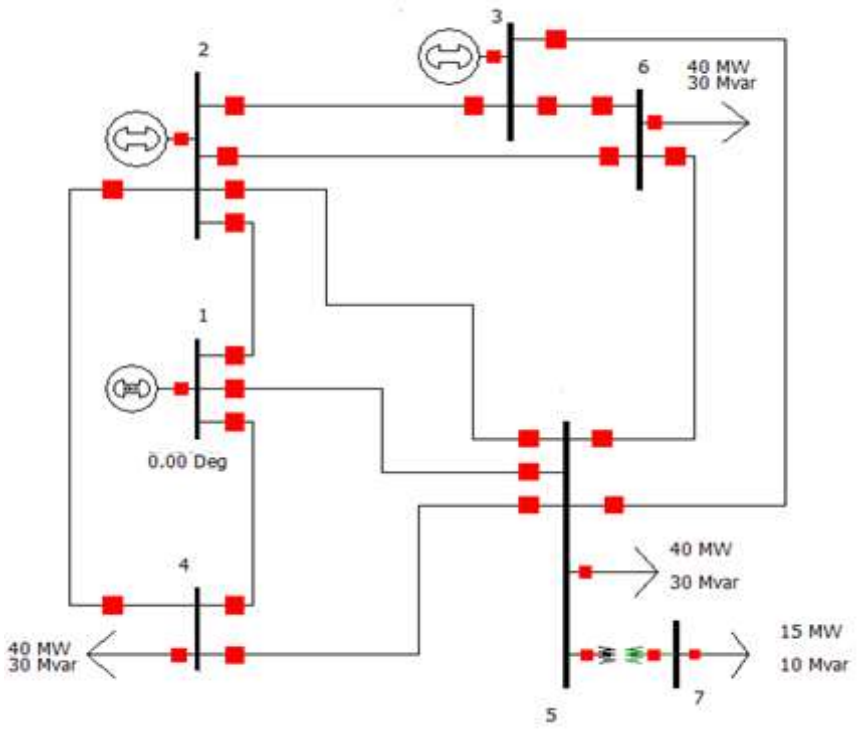
```
fout<<"\n           Load Flow Solution using Fast Decoupled Method      ";
fout<<"\n  ------------------------------------------------------------------  ";
fout<<"\n   Bus.   Volts    Angle     PG(Mw)     QG(Mvar)    PL(Mw)    QL(Mvar)";
fout<<"\n           pu       deg.                                          ";
fout<<"\n  ------------------------------------------------------------------\n";
for (i=1; i<=nb; ++i){
   fout <<fixed <<setw(5) << right <<no[i];
   fout <<fixed <<setw(10) << right << setprecision(4) << v[i];
   fout <<fixed <<setw(10) << right << setprecision(4) << theta[i]*180/3.14;
   fout <<fixed <<setw(11) << right << setprecision(4) << pg[i]*base;
   fout <<fixed <<setw(11) << right << setprecision(4) << qg[i]*base;
   fout <<fixed <<setw(11) << right << setprecision(4) << pl[i]*base;
   fout <<fixed <<setw(11) << right << setprecision(4) << ql[i]*base<<endl;
}
fout<<"  ------------------------------------------------------------------";
fout<<"\n   Total                  ";
fout <<fixed <<setw(5) << right << setprecision(4) <<PGtotal;
fout <<fixed <<setw(11) << right << setprecision(4) <<QGtotal;
fout <<fixed <<setw(11) << right << setprecision(4) <<PLtotal;
fout <<fixed <<setw(11) << right << setprecision(4) <<QLtotal<<endl;
fout<<"  ------------------------------------------------------------------";
fout<<endl<<"   P Losses (MW) = "<<PGtotal-PLtotal<<endl;
```

Selanjutnya dapat dihitung aliran daya listrik masing-masing saluran dan transformator menggunakan program berikut:

```
cout<<"\n  ------------------------------------------------------------------  ";
cout<<"\n                          Line Flow Solution                          ";
cout<<"\n  ------------------------------------------------------------------  ";
cout<<"\n   From    To        Pij        Pji     Qij      Qji    Losses (P) ";
cout<<"\n   Bus.i   Bus.j     (MW)      (MW)    (MVar)   (MVar)      (MW)   ";
cout<<"\n  ------------------------------------------------------------------\n";
for (l=1;l<=nl;l++){
    i = sl[l];
    j = el[l];
    temp = theta[i]-theta[j];
    temp1 = theta[j]-theta[i];
    Pij[i][j]=v[i]*(-v[i]*g[i][j]+(g[i][j]*cos(temp)+
    b[i][j]*sin(temp))*v[j])*base;
    Pij[j][i]=v[j]*(-
    v[j]*g[j][i]+(g[j][i]*cos(temp1)+b[j][i]*sin(temp1))*v[i])*base;
    Qij[i][j]=v[i]*(v[i]*b[i][j]+(g[i][j]*sin(temp)-b[i][j]*cos(temp)*v[j]-
    v[i]*bl[l]))*base;
    Qij[j][i]=v[j]*(v[j]*b[j][i]+(g[j][i]*sin(temp1)-b[j][i]*cos(temp1)*v[i]-
    v[j]*bl[l]))*base;
    fout <<fixed <<setw(5) << right << sl[l];
    fout <<fixed <<setw(8) << right << el[l];
    fout <<fixed <<setw(15) << right << setprecision(4) << Pij[i][j];
    fout <<fixed <<setw(11) << right << setprecision(4) << Pij[j][i];
    fout <<fixed <<setw(11) << right << setprecision(4) << Qij[i][j];
    fout <<fixed <<setw(11) << right << setprecision(4) << Qij[j][i];
    fout <<fixed <<setw(11) << right << setprecision(4) << Pij[i][j]+
    Pij[j][i]<<endl;
}
cout<<"  ------------------------------------------------------------------";
```

Contoh soal 4.2:

Suatu sistem tenaga listrik 7 bus seperti gambar berikut:



Gambar 4.8 Sistem Tenaga Listrik 7 Bus

Data Saluran

sl	rl	R	X	Bc
1	2	0.1	0.2	0.02
1	4	0.05	0.2	0.02
1	5	0.08	0.3	0.03
2	3	0.05	0.25	0.03
2	4	0.05	0.1	0.01
2	5	0.1	0.3	0.02
2	6	0.07	0.2	0.025
3	5	0.12	0.26	0.025
3	6	0.02	0.1	0.01
4	5	0.2	0.4	0.04
5	6	0.1	0.3	0.03
7	5	0	0.01	0

Bc adalah setengah suseptasi total saluran

Data Bus

No Bus	Tegangan (kV)	Tegangan (pu)	Tegangan (kV)	Beban (MW)	Beban (Mvar)	Gen (MW)
1	150	1.05	157.500			
2	150	1.02	153.000			30
3	150	1.02	153.000			30
4	150			40	30	
5	150			40	30	
6	150			40	30	
7	20			15	10	

a. Tentukan matrik admitansi bus sistem tersebut?

b. Lakukan perhitungan aliran daya dan tampilkan hasil perhitungan studi aliran daya yang terdiri dari data bus, aliran daya dan losses.

c. Bandingkan hasil perhitungan soal b tersebut dengan hasil perhitungan menggunakan software powerworld?

Jawab:

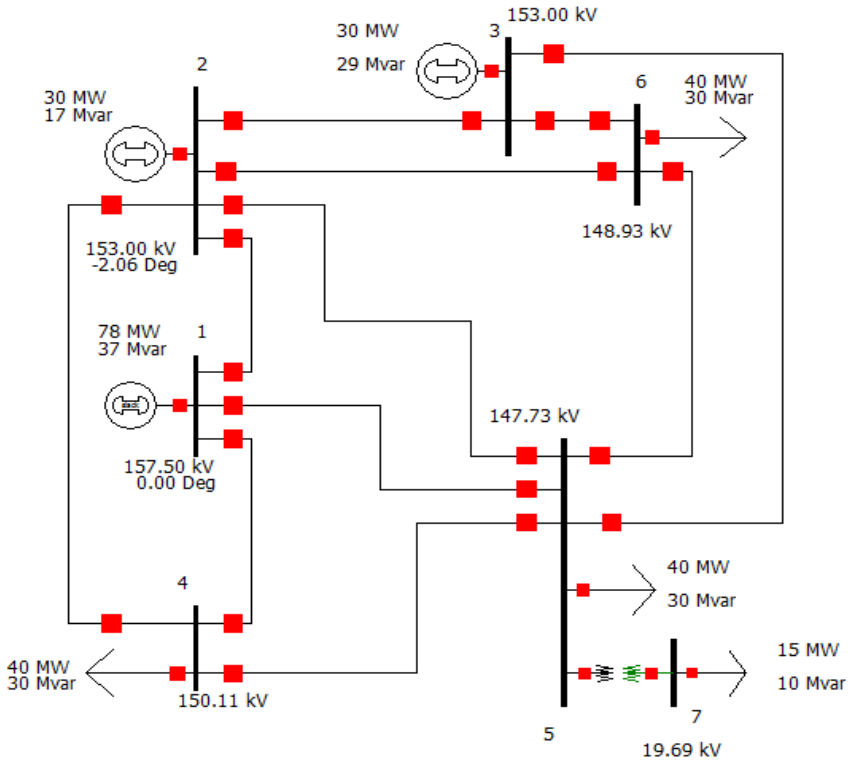
a. Matrik admitansi bus sistem tersebut adalah:

Name	Bus 1	Bus 2	Bus 3	Bus 4	Bus 5	Bus 6	Bus 7
1	4.01 - j11.78	-2.00 + j4.00		-1.18 + j4.71	-0.83 + j3.11		
2	-2.00 + j4.00	9.33 - j23.25	-0.77 + j3.85	-4.00 + j8.00	-1.00 + j3.00	-1.56 + j4.45	
3		-0.77 + j3.85	4.16 - j16.60		-1.46 + j3.17	-1.92 + j9.62	
4	-1.18 + j4.71	-4.00 + j8.00		6.18 - j14.67	-1.00 + j2.00		
5	-0.83 + j3.11	-1.00 + j3.00	-1.46 + j3.17	-1.00 + j2.00	5.29 - j114.21	-1.00 + j3.00	-0.00 + j100
6		-1.56 + j4.45	-1.92 + j9.62		-1.00 + j3.00	4.48 - j17.04	
7					-0.00 + j100.00		0.00 - j100

Sedangkan hasil running program aliran daya menggunakan program Visual C++ adalah:

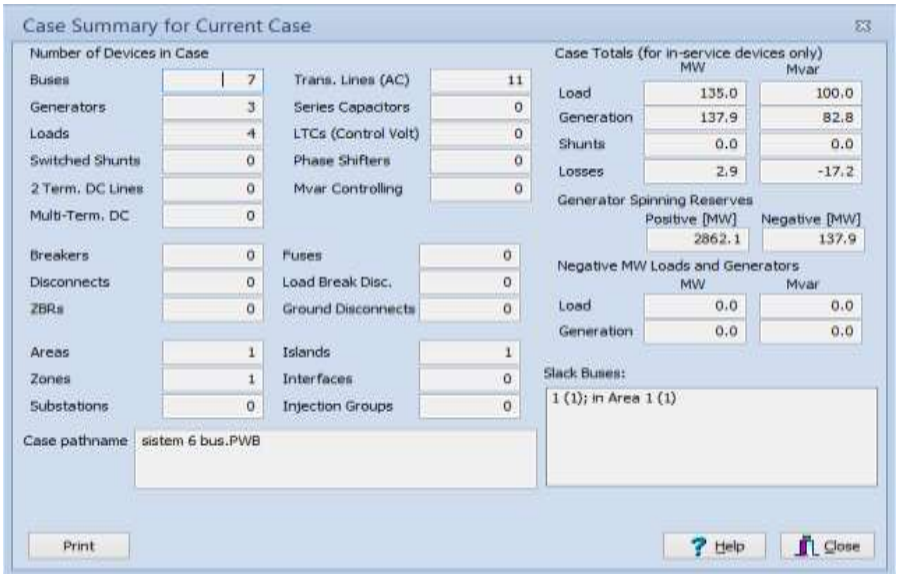
```
  ============================================================

   Iteration  |   Mismatch   |  P Mismatch   |  Q Mismatch
  ------------------------------------------------------------
       1          0.330358       0.330358       0.100380
       2          0.034839       0.034839       0.012133
       3          0.003218       0.003218       0.000967
       4          0.000413       0.000413       0.000087
  ------------------------------------------------------------

---------------------------------------------------------------------
           Load Flow Solution using Fast Decoupled Method
  ---------------------------------------------------------------------
   Bus.    Volts     Angle      PG(Mw)      QG(Mvar)     PL(Mw)     QL(Mvar)
            pu        deg.
  ---------------------------------------------------------------------
    1     1.0500     0.0000     77.8938     36.8467      0.0000      0.0000
    2     1.0200    -2.0564     30.0000     16.5556      0.0000      0.0000
    3     1.0200    -2.6935     30.0000     29.4445      0.0000      0.0000
    4     1.0007    -2.6752      0.0000      0.0000     40.0000     30.0000
    5     0.9848    -3.7332      0.0000      0.0000     40.0000     30.0000
    6     0.9929    -3.7556      0.0000      0.0000     40.0000     30.0000
    7     0.9847    -3.7420      0.0000      0.0000     15.0000     10.0000
  ---------------------------------------------------------------------
   Total                     137.8938     82.8467    135.0000    100.0000
  ---------------------------------------------------------------------
   P Losses (MW) = 2.8938

  ---------------------------------------------------------------------------
                            Line Flow Solution
  ---------------------------------------------------------------------------
   From     To          Pij         Pji         Qij          Qji     Losses (P)
   Bus.i    Bus.j      (MW)        (MW)       (MVar)       (MVar)     (MW)
  ---------------------------------------------------------------------------
    1        2         21.8023    -21.3467     3.1390     -6.7288      0.4556
    1        4         29.2896    -28.7346    16.9226    -19.1813      0.5549
    1        5         26.8020    -26.0858    12.9952    -16.8788      0.7161
    2        3          4.4519     -4.4420    -3.9684     -2.2245      0.0099
    2        4         16.7057    -16.5094    10.3386    -12.0710      0.1963
    2        5         12.4427    -12.2332     5.8232     -9.3180      0.2095
    2        6         17.7511    -17.4977     5.2050     -9.6719      0.2534
    3        5         11.0486    -10.8194     6.1147    -10.7371      0.2292
    3        6         23.3939    -23.1857    22.0887    -23.1701      0.2083
    4        5          5.2430     -5.1842    -2.6419     -5.1552      0.0588
    5        6         -0.6771      0.6836    -5.3237     -0.5238      0.0065
    5        7         15.0002    -15.0002    10.0034    -10.0000      0.0000
  ---------------------------------------------------------------------
```

c. Hasil perhitungan menggunakan software powerworld adalah:



Gambar 4.9 Hasil Running Powerworld Sistem 7 bus

Ringkasan hasil running powerworld untuk perhitungan aliran daya adalah:

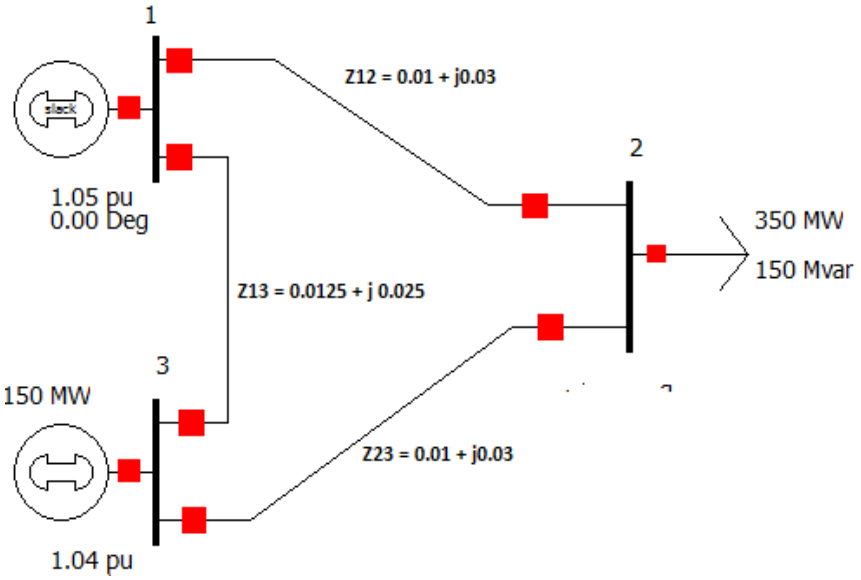


Gambar 4.10 Case Summary Hasil Running Powerworld Sistem 7 bus

Hasil yang diperoleh hampir sama dengan hasil running software powerlworld. Terjadinya perbedaan disebabkan metode yang digunakan pada software powerworld adalah metode Newton Raphson.

Soal Latihan:

1. Sebutkan tiga jenis besaran listrik yang diperoleh dari hasil perhitungan aliran daya?
2. Hitung aliran daya sistem berikut menggunakan metode Newton Raphson:



a. Hitung matrik admitansi bus?

b. Hitung matrik Jacobian untuk iterasi pertama?

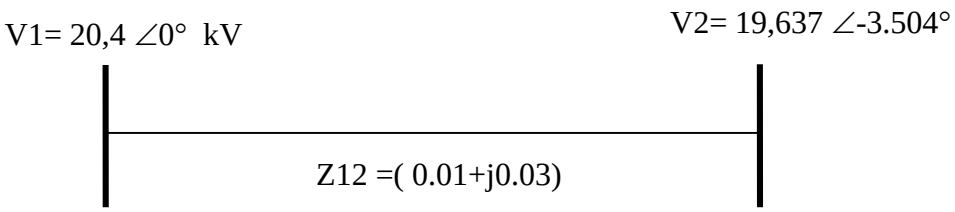
c. Tentukan nilai tegangan bus setelah interasi pertama?

Diketahui invers matrik Jacobian adalah:

$$J^{-1} = \begin{pmatrix} 0.01929 & 0.00983 & -0.00465 \\ 0.00992 & 0.01976 & 0.00028 \\ 0.00524 & 0 & 0.01571 \end{pmatrix}$$

Asumsi nilai awal magnitude tegangan = 1 dan sudut tegangan = 0.

3. Sebuah saluran transmisi menghubungkan bus 1 dengan bus 2 dengan data seperti berikut :



Hitung:

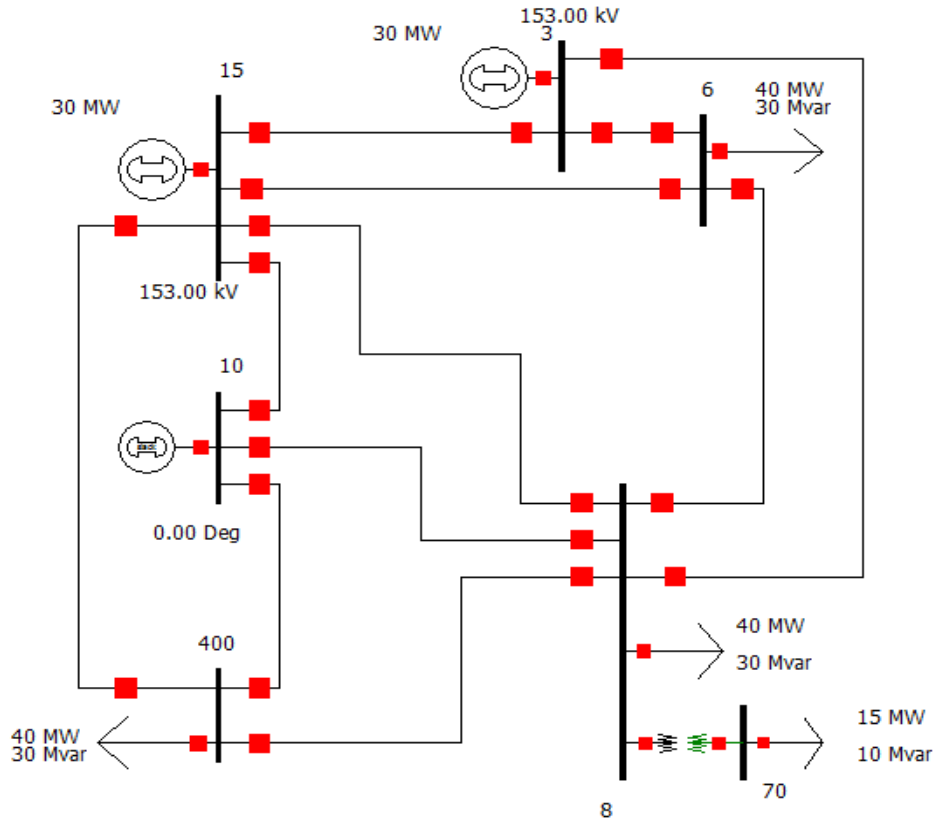
a. Aliran daya dari bus 1 ke bus 2?
b. Aliran daya dari bus 2 ke bus 1?
c. Rugi-rugi saluran?

BAB V
PENGATURAN ALIRAN DAYA

Pengaturan aliran dilakukan untuk memperoleh perbaikan proses perhitungan aliran daya seperti kecepatan hitung, keakuratan hasil dan perbaikan performasi sistem tenaga. Diantara pengaturan aliran daya sering dilakukan adalah: penambahan kapaistor bank dan pengaturan tap trafo untuk memperbaiki performasi sistem tenaga. Penggunaan teknik alokasi memori dinamis dan penggunaan rutin SuperLU dalam proses komputasi aliran daya akan mempercepat proses perhitungan.

5.1 Urutan No Bus

Adakalanya no bus tidak beraturan, sedangkan program aliran daya hanya mengenal input data dengan no bus yang berurutan, maka jika program dirunning akan terjadi error. Sebagai contoh, jika data sistem tujuh bus dengan penomoran bus yang tidak beraturan seperti berikut:



Gambar 5.1 Sistem 7 bus dengan no bus yang tidak beraturan

Maka input data sistem tersebut adalah sebagai berikut:

```
12 7
10 15 0 0.10 0.20 0.02   0
10 400 0 0.05 0.20 0.02   0
10 8 0 0.08 0.30 0.03   0
15 3 0 0.05 0.25 0.03   0
15 400 0 0.05 0.10 0.01   0
15 8 0 0.10 0.30 0.02   0
15 6 0 0.07 0.20 0.025   0
3  8 0 0.12 0.26 0.025   0
3  6 0 0.02 0.10 0.01   0
400 8 0 0.20 0.40 0.04   0
8  6 0 0.10 0.30 0.03   0
8  70 0 0.00 0.001 0.00   0

10  1  0.00   0.0    0.0    0.0  1.05    0.0 0
15  2  0.00   0.00   0.3    0.0  1.02    0.0 0
3   2  0.00   0.00   0.3    0.0  1.02    0.0 0
400 0  0.40   0.30   0.0    0.0  1.00    0.0 0
8   0  0.40   0.30   0.0    0.0  1.00    0.0 0
6   0  0.40   0.30   0.0    0.0  1.00    0.0 0
70  0  0.15   0.10   0.0    0.0  1.00    0.0 0
```

Karena penomoran bus menjadi tidak beraturan, maka program aliran daya sebelumnya tidak dapat dijalankan. Untuk mengatasi hal tersebut maka harus ditambahkan kode program pengaturan bus sebagai berikut di awal program:

```
//Rearrangement bus name
 for (i=1;i<=nb;i++)
   {
   new_bus[i]=0;
   }
 for (i=1;i<=nb;i++)
   {
   old_bus[i]=no[i];
   new_bus[old_bus[i]]=i;
   }
 for (i=1;i<=nb;i++)
   {
   no[i]=new_bus[no[i]];
   }
 for (i=1;i<=nl;i++)
   {
   sl[i]=new_bus[sl[i]];
   el[i]=new_bus[el[i]];
   }
```

Dan diakhir program ditambahkan kode program untuk mengembalikan kenomor aktual berikut:

```
for (i=1;i<=nb;i++)
   {
   no[i]=old_bus[no[i]];
   }
```

Dan untuk data saluran tambahkan:

```
for (l=1;l<=nl;l++)
  {
  .
  .
  sl[l]=old_bus[sl[l]];
  el[l]=old_bus[el[l]];
  .
  .
  .
  }
```

Sehingga tampilan hasil program menjadi:

```
  ---------------------------------------------------------------------
   Bus.    Volts     Angle      PG(Mw)      QG(Mvar)     PL(Mw)     QL(Mvar)
           pu        deg.
  ---------------------------------------------------------------------
   10     1.0500     0.0000     77.8938     36.8467      0.0000      0.0000
   15     1.0200    -2.0564     30.0000     16.5556      0.0000      0.0000
    3     1.0200    -2.6935     30.0000     29.4445      0.0000      0.0000
  400     1.0007    -2.6752      0.0000      0.0000     40.0000     30.0000
    8     0.9848    -3.7332      0.0000      0.0000     40.0000     30.0000
    6     0.9929    -3.7556      0.0000      0.0000     40.0000     30.0000
   70     0.9847    -3.7420      0.0000      0.0000     15.0000     10.0000
  ---------------------------------------------------------------------
   Total                     137.8938     82.8467    135.0000    100.0000
  ---------------------------------------------------------------------
   P Losses (MW) = 2.8938
```

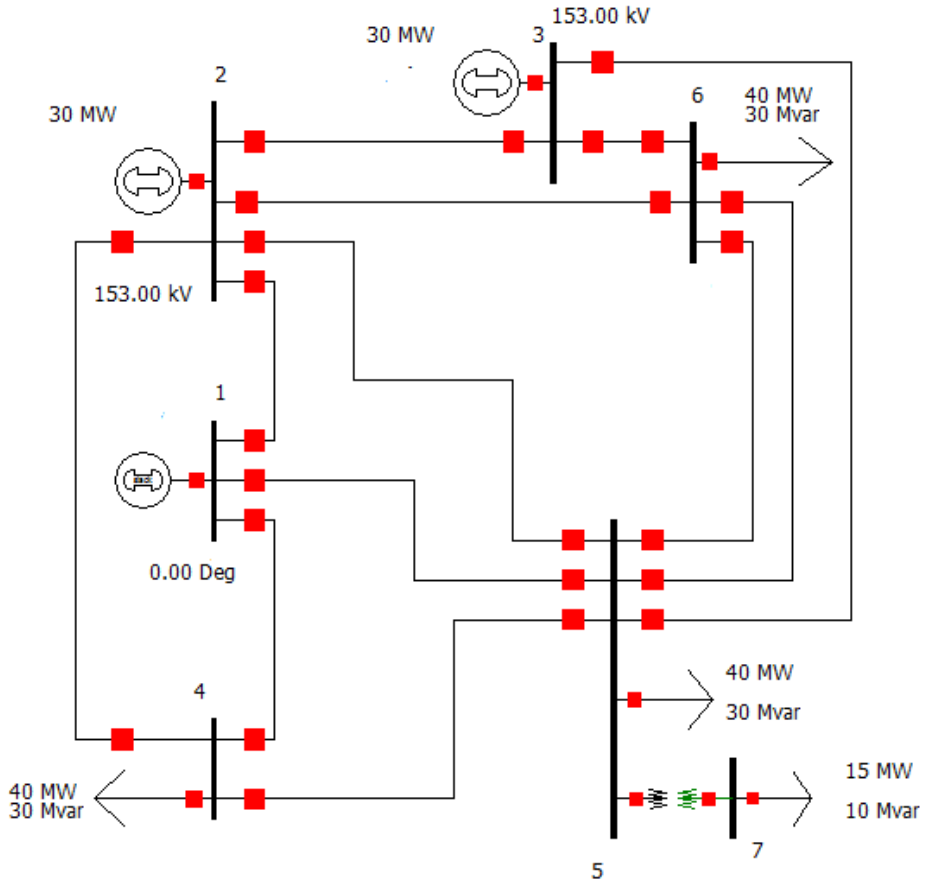
Sedangkan untuk data saluran:

```
-------------------------------------------------------------------------
   From     To          Pij         Pji         Qij          Qji
   Bus.i    Bus.j      (MW)        (MW)       (MVar)       (MVar)
  -------------------------------------------------------------------------
   10       15         21.8023    -21.3467     3.1390     -6.7288
   10      400         29.2896    -28.7346    16.9226    -19.1813
   10        8         26.8020    -26.0858    12.9952    -16.8788
   15        3          4.4519     -4.4420    -3.9684     -2.2245
   15      400         16.7057    -16.5094    10.3386    -12.0710
   15        8         12.4427    -12.2332     5.8232     -9.3180
   15        6         17.7511    -17.4977     5.2050     -9.6719
    3        8         11.0486    -10.8194     6.1147    -10.7371
    3        6         23.3939    -23.1857    22.0887    -23.1701
  400        8          5.2430     -5.1842    -2.6419     -5.1552
    8        6         -0.6771      0.6836    -5.3237     -0.5238
    8       70         15.0002    -15.0002    10.0034    -10.0000
  -------------------------------------------------------------------------
```

Beberapa data standar IEEE seperti Data IEEE 300 bus menggunakan penomoran bus yang tidak beraturan. Dengan menggunakan tambahan kode program tersebut, persoalan no urut bus dapat diselesaikan.

5.2 Model Saluran ganda

Program aliran daya hanya mengenal input data dengan tipe saluran tunggal, maka jika ada saluran ganda (double line) program akan error. Sebagai contoh, jika data sistem tujuh bus dengan saluran ganda seperti berikut:



Gambar 5.2 Sistem 7 bus dengan saluran ganda dari bus 5 ke bus 6.

Maka input data sistem tersebut adalah sebagai berikut:

```
13 7
1 2 0 0.10 0.20 0.02   0
1 4 0 0.05 0.20 0.02   0
1 5 0 0.08 0.30 0.03   0
2 3 0 0.05 0.25 0.03   0
2 4 0 0.05 0.10 0.01   0
2 5 0 0.10 0.30 0.02   0
2 6 0 0.07 0.20 0.025   0
3 5 0 0.12 0.26 0.025   0
3 6 0 0.02 0.10 0.01   0
4 5 0 0.20 0.40 0.04   0
5 6 0 0.10 0.30 0.03   0
5 6 0 0.10 0.30 0.03   0
5 7 0 0.00 0.001 0.00   0
```

Karena ada tambahan saluran dengan no bus yang sama yaitu dari bus 5 ke bus 6, maka program aliran daya sebelumnya tidak dapat dijalankan. Untuk mengatasi hal tersebut maka harus ditambahkan kode program yang mendukung saluran ganda tersebut di awal program sebagai berikut:

```
// Support double line
 for (k=1; k<=nl; ++k)
   dl[k]=1;

 for (k=1;k<=nl;++k)
   for (l=k+1;l<=nl;++l)
     {
     if(sl[k]==sl[l] && el[k]==el[l])
       {
       dl[l]=2;
       xl[k]=xl[k]/2;
       rl[k]=rl[k]/2;
       }
     else if  (sl[k]==el[l] && el[k]==sl[l])
       {
       dl[l]=2;
       xl[k]=xl[k]/2;
       rl[k]=rl[k]/2;
       }
     }
```

Untuk saluran tunggal nilai dl diberikan satu, sedang jika terdapat saluran ganda nilai dl menjadi dua dan impedansi saturan yang teridentifikasi ganda akan dibagi dua. Dan dalam program ditambahkan kode program berikut untuk memastikan bahwa saluran ganda tidak perlu diproses dua kali:

```
for (l=1;l<=nl;l++){
   i = sl[l];
   j = el[l];
   if (dl[l]==1) //Support double line
     {
       Program perhitungan ....
       }
}
```

Sehingga tampilan hasil program menjadi:

```
Line Flow Solution using Fast Decoupled Method
  ------------------------------------------------------------------
   From    To         Pij        Pji        Qij          Qji
   Bus.i   Bus.j     (MW)       (MW)       (MVar)       (MVar)
  ------------------------------------------------------------------
    1       2        21.8202   -21.3640     3.1310    -6.7197
    1       4        29.2881   -28.7347    16.8381   -19.1028
    1       5        26.7695   -26.0636    12.6187   -16.5420
    2       3         4.4694    -4.4594    -3.9717    -2.2209
    2       4        16.6309   -16.4372    10.2083   -11.9455
    2       5        12.3640   -12.1616     5.4699    -8.9878
    2       6        17.9044   -17.6430     5.6065   -10.0503
```

```
   3       5        10.9004   -10.6823     5.7436   -10.3929
   3       6        23.5595   -23.3417    22.9701   -24.0054
   4       5         5.1709    -5.1147    -2.8453    -4.9656
   5       6        -0.9777     0.9849    -6.5424     0.6960
   5       6        -0.9777     0.9849    -6.5424     0.6960
   5       7        15.0001   -15.0001    10.0034   -10.0000
-----------------------------------------------------------------
```

5.3 Kapasitor Bank

Pemasangan kapasitor bank sering dilakukan dalam rangka perbaikan profil tegangan sistem tenaga, mengurangi rugi-rugi saluran dan drop tegangan. Kapasitor bank dalam program aliran daya dapat dimodelkan sebagi admitansi shunt yang mempengaruhi nilai admitansi bus berikut:

$$b_{ii} = \sum_{\substack{j=1 \\ j \neq i}}^{N} \left(\frac{-x_{ij}}{r_{ij}^2 + x_{ij}^2} + \frac{y_{ij}^{Shunt}}{2} + y_i^{Cap} \right) \qquad (5.1)$$

Setelah program menghitung admitansi matrik Y = G+jB, tambahkan perintah berikut untuk menelusuri bus yang memiliki koneksi kapasitor dan tambahkan nilai kapasitor tersebut kedalam admitansi bus.

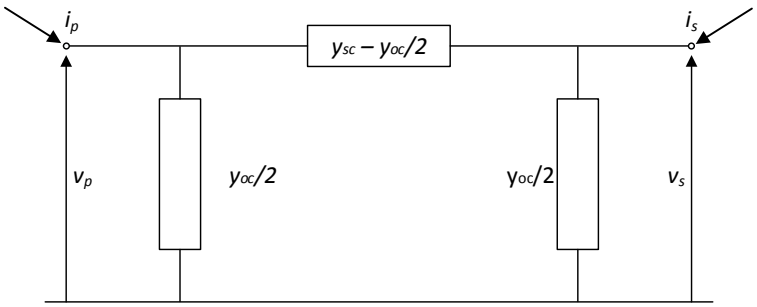
```
for (i=1;i<=nb;++i)
  if (Xcap[i]!=0)
         b[i][i]+=-1/Xcap[i];
```

Atau model kompensasi injeksi daya reaktif pada bus yang berkaitan dengan lokasi pemasangan kapasitor sebagai berikut:

$$Q = Q_i - Q_{cap} \qquad (5.2)$$

5.4 Tap Transformator

Selain dengan kapasitor bank, pengaturan tap transformator sering dilakukan dalam rangka perbaikan profil tegangan sistem tenaga, mengurangi rugi-rugi saluran dan drop tegangan. Rangkaian ekivalen trasformator diperlihatkan pada gambar 5.3 berikut:



Gambar 5.3 Rangkaian ekivalen transformator

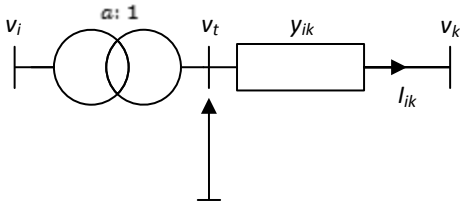
dimana:

y_{oc} adalah kebalikan dari z_{oc} yaitu impedansi magnetisasi

y_{sc} adalah kebalikan dari z_{sc} yaitu impedansi bocor

Nilai z_{oc} dan z_{sc} diperoleh dari pengujian rangkaian terbuka dan hubung singkat standar.

Suatu transformator dengan rasio belitan a yang menghubungkansimpul i dan k dapat digambarkan dengan model trafo ideal seri dengan admitansi bocor trafo nominal seperti gambar 5.4 (a).



Gambar 5.4 Rangkaian ekivalen transformator dengan pengaturan tap

Jika trafo dalam kondisi tap nominal (a=1) persamaan yang berlaku adalah:

$$I_{ik} = y_{ik} V_i - y_{ik} V_k \quad (5.3)$$

$$I_{ki} = y_{ik} V_k - y_{ik} V_i \quad (5.4)$$

dimana: $I_{ik} = -I_{ki}$

tetapi dalam kondisi tap off nominal berlaku:

$$V_t = \frac{V_i}{a} \quad (5.5)$$

$$I_{ki} = y_{ik} (V_k - V_i) \quad (5.6)$$

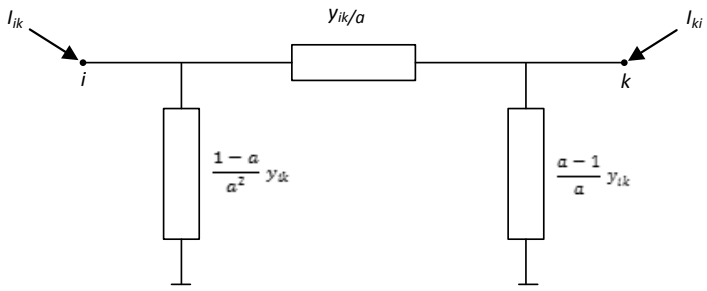
$$I_{ik} = -\frac{I_{ki}}{a} \quad (5.7)$$

Dengan mengelimnasi V_t diperoleh:

$$I_{ik} = \frac{y_{ik}}{a^2} V_i - \frac{y_{ik}}{a} V_k \quad (5.8)$$

$$I_{ki} = -\frac{y_{ik}}{a} V_i + y_{ik} V_k \quad (5.9)$$

Sehingga model transfomator dengan pengaturan tap dapat digambarkan sebagai berikut:



Gambar 5.5 Rangkaian ekivalen transformator dengan tap off nominal

Persamaan tersebut dapat ditulis dalam bentuk matrik berikut:

$$\begin{bmatrix} I_{ik} \\ I_{ki} \end{bmatrix} = \begin{bmatrix} \frac{y_{ik}}{a^2} & -\frac{y_{ik}}{a} \\ -\frac{y_{ik}}{a} & y_{ik} \end{bmatrix} \begin{bmatrix} V_i \\ V_k \end{bmatrix} \quad (5.10)$$

Berdasarkan persamaan (5.10) diatas maka untuk pengaturan tap trafo perlu ditambahkan variabel tap dalam kode program perhitungan admitansi bus sebagai berikut:

```
for (l=1;l<=nl;l++)
   {
   i = sl[l];
   j = el[l];
   if (dl[l]==1) //Support double line
     {
     aa = rl[l]*rl[l] + xl[l]*xl[l];
     if (tipebranch[l]==0) //jika saluran
       {
       g[i][j] = -rl[l]/aa;
       g[j][i] = g[i][j];
       b[i][j] = xl[l]/aa;
       b[j][i] = b[i][j];
       g[i][i] += rl[l]/aa;
       g[j][j] += rl[l]/aa;
       b[i][i] += -xl[l]/aa + bl[l]/2;
       b[j][j] += -xl[l]/aa + bl[l]/2;
       }
       else if (tipebranch [l]==1) //jika trafo
       {
       g[i][j] = (-rl[l]/aa)/tap[l];
       g[j][i] = g[i][j];
       b[i][j] = (xl[l]/aa)/tap[l];
       b[j][i] = b[i][j];
```

```
      g[i][i] += (rl[l]/aa)/(tap[l]*tap[l]);
      g[j][j] += rl[l]/aa;
      b[i][i] += (-xl[l]/aa)/(tap[l]*tap[l]) + bl[l]/2;
      b[j][j] += (-xl[l]/aa) + bl[l]/2;
      }
    }
  }
```

5.5 Alokasi Memori Dinamis

Data bertipe array harus dideklarasikan terlebih dahulu dengan menyebutkan ukurannya sebelum digunakan. Ukuran variabel tersebut tidak dapat kita ubah selama program dijalankan. Penggunaan array dengan ukuran tetap untuk sistem skala besar akan membutuhkan alokasi memori yang cukup besar dan memperlabat proses perhitungan.

Jika menggunakan pointer array maka dapat dialokasikan memori sesuai ukuran yang diinginkan atau disebut dengan alokasi memori dinamis. Dengan alokasi memori dinamis, ukuran sesuai dengan yang dibutuhkan dan akan mempercepat proses perhitungan.

Sebagai contoh perhatikan deklarasi array satu dimensi berikut:

Alokasi memori tetap sebesar 100:

```
int nl=15; // nl adalah jumlah line
double rl[100], xl[100],bl[100];
```

Jika menggunakan alokasi memori dinamis ditulis menjadi:

```
int nl=15; // nl adalah jumlah line
double *rl,*xl,*bl;
rl=(double*)calloc(nl+1,sizeof(double));
xl=(double*)calloc(nl+1,sizeof(double));
bl=(double*)calloc(nl+1,sizeof(double));
```

Dengan alokasi memori dinamis, array rl, xl, dan bl masing masing dialokasikan memori sebesar 15 sesuai jumlah line.

Untuk array dua dimensi dengan memori fix seperti:

```
int nb=10; // nb adalah jumlah bus
double g[100][100],b[100][100];
```

karena dimensi matrik g dan b adalah 100 kali 100, maka sistem akan mengalokasikan sebanyak 100 kali 100 memori untuk menampung variabel g atau b tersebut.

Sedangkan jika menggunakan deklrasi matrik g dan b menggunakan alokasi memori dinamis seperti berikut:

```
int nb=10; // nb adalah jumlah bus
double **g,**b;
try
 {
   g = new  double*[nb];         // SET UP THE ROWS.
   for (int i = 1; i < (nb+1); i++)
   g[i] = new double[nb];        // SET UP THE COLUMNS
 }
 catch (std::bad_alloc)        // ENTER THIS BLOCK ONLY IF bad_alloc IS
THROWN.
 {
      cout << "Could not allocate";
      exit(-1);
 }
 try
 {
   b = new  double*[nb];         // SET UP THE ROWS.
   for (int i = 1; i < (nb+1); i++)
   b[i] = new double[nb];        // SET UP THE COLUMNS
 }
 catch (std::bad_alloc)        // ENTER THIS BLOCK ONLY IF bad_alloc IS
THROWN.
 {
      cout << "Could not allocate";
      exit(-1);
 }
```

Ukuran matrik b dan g akan sebesar perkalian jumlah bus yaitu 10 kali 10 untuk menampung elemen matrik g atau b.

Untuk mebebaskan memori yang telah dialokasikan, pada akhir program tambahkan perintah berikut:

```
free(rl);
free(xl);
free(bl);
free(g);
free(b);
```

Soal Latihan:

1. Bandingkan kecepatan komputasi aliran daya fast decaupled alokasi memori statis dengan alokasi memori dinamis? Berikan contoh perhitungan komputer untuk kasus sistem tenaga yang besar?

2. Evaluasi sistem tenaga dari data standar IEEE (dapat didownload melalui website berikut: https://www.ee.washington.edu/research/pstca/) berikut:

a. IEEE 14 bus

b. IEEE 30 bus

c. IEEE 57 bus

d IEEE 118 bus

e IEEE 300 bus