

**PERANCANGAN & PEMBUATAN LABORATORIUM ELEKTRONIKA VIRTUAL
UNTUK RANGKAIAN RESISTOR-INDUKTOR-KAPASITOR**



Oleh :

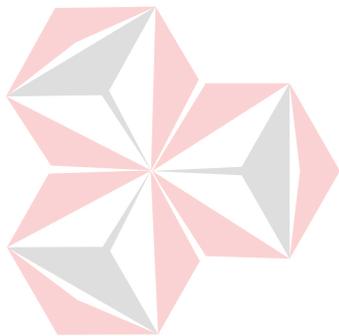
Nama : JANUAR BUDIJANTO
NIM : 91.1002
NIRM : 91.7.085.31132.00512
Program : S1 (Strata Satu)
Jurusan : Manajemen Informatika

**SEKOLAH TINGGI
MANAJEMEN INFORMATIKA & TEKNIK KOMPUTER
SURABAYA
1997**

**PERANCANGAN & PEMBUATAN LABORATORIUM ELEKTRONIKA VIRTUAL
UNTUK RANGKAIAN RESISTOR-INDUKTOR-KAPASITOR**

SKRIPSI

**Diajukan sebagai salah satu syarat untuk menyelesaikan
program Sarjana Komputer**



UNIVERSITAS
Dinamika

Oleh :

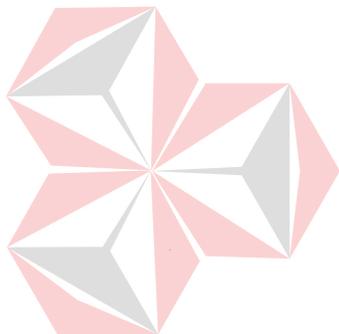
Nama : JANUAR BUDIANTO
NIM : 91.1002
NIRM : 91.7.085.31132.00512
Program : S1 (Strata Satu)
Jurusan : Manajemen Informatika

**SEKOLAH TINGGI
MANAJEMEN INFORMATIKA & TEKNIK KOMPUTER
SURABAYA**

1997

**PERANCANGAN & PEMBUATAN LABORATORIUM ELEKTRONIKA VIRTUAL
UNTUK RANGKAIAN RESISTOR-INDUKTOR-KAPASITOR**

Telah diperiksa, diuji, dan disetujui



UNIVERSITAS
Surabaya, Juni 1997
Dinamika

Mengetahui:

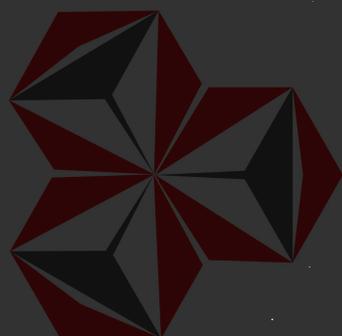
Ir. Ronny S. Susilo, MM
Pembantu Ketua I



Menyetujui:

Ir. Purwadi Agus D., MSc.
Pembimbing

A handwritten signature in black ink, appearing to read 'Purwadi'.



UNIVERSITAS
Dinamika

ABSTRAKSI

Pada era teknologi dewasa ini, komputer bukanlah suatu produk yang hanya dapat digunakan oleh kalangan tertentu saja. Komputer kini telah memasyarakat dan telah dapat dirasakan manfaatnya bagi semua orang. Semua itu didukung oleh perangkat keras dan perangkat lunak yang semakin canggih, semakin mudah digunakan, dengan berbagai fasilitas yang disertakan dalam perangkat-perangkat tersebut.

Salah satu penggunaan komputer yang akhir-akhir ini sering dilakukan adalah untuk mensimulasikan suatu tempat atau situasi tertentu. Misalnya untuk mensimulasikan sebuah terowongan angin untuk mengetahui kondisi aliran pada suatu airfoil. Sementara itu penyelesaian secara eksperimental mampu memberikan jawaban yang cukup memuaskan atas kondisi yang diberikan. Namun cara ini memerlukan peniruan kondisi yang diinginkan dengan baik, dan karena terbatasnya peralatan pengujian, suatu kondisi uji yang diinginkan dapat menjadi terlalu mahal dan memakan banyak waktu untuk dibuat.

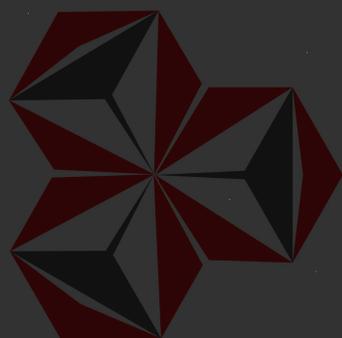
Dalam tugas akhir ini, dirancang dan dibuat sebuah laboratorium elektronika virtual yang berisi sebuah rangkaian objek yang terdiri atas resistor, induktor, dan kapasitor, dan sebuah oscilloscope. Laboratorium elektronika virtual ini dipakai untuk mengetahui

karakteristik dari rangkaian yang telah digambar. Dengan menempatkan kedua *probe* oscilloscope ke lokasi (pada rangkaian) yang diinginkan, bentuk tegangan antara dua titik tersebut dapat dilihat.

Dengan demikian, laboratorium elektronika virtual ini dapat dipakai sebagai sarana belajar atau uji coba tanpa mengharuskan pemakai mempunyai rangkaian dan oscilloscope yang sebenarnya. Hal ini memberikan banyak keuntungan terutama dari segi biaya, lokasi, dan resiko. Pemakai dapat melakukan percobaan berulang-ulang tanpa ada batasan waktu; dan karena disajikan secara interaktif materi yang ada akan lebih mudah dipahami.



UNIVERSITAS
Dinamika



UNIVERSITAS
Dinamika

KATA PENGANTAR

Berkat rahmat Tuhan yang Maha Esa, akhirnya penulis dapat menyelesaikan tugas akhir ini sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer di bidang Manajemen Informatika di Sekolah Tinggi Manajemen Informatika dan Teknik Komputer Surabaya.

Banyak manfaat yang penulis peroleh dari ilmu atau pelajaran yang telah penulis terima waktu kuliah selama ini untuk menyusun tugas akhir ini. Besar harapan penulis semoga tugas akhir ini bermanfaat bagi rekan-rekan mahasiswa lainnya.

Berkat bimbingan, nasehat, dorongan serta bantuan moril maupun materiil yang telah penulis terima dari berbagai pihak, maka penyusunan tugas akhir ini dapat berjalan lancar.

Pada kesempatan ini penulis ucapkan banyak terima kasih kepada:

1. Bapak Ir. Purwadi Agus Darwito, MSc. selaku dosen pembimbing yang telah memberi pengarahan dan saran dalam menyelesaikan tugas akhir ini.
2. Bapak Drs. Antok Supriyanto yang telah memberi ijin penulis untuk mengambil topik tugas akhir ini.
3. Pimpinan dan seluruh staf pengajar STIKOM Surabaya, yang telah memberikan bekal ilmu kepada penulis selama menempuh kuliah.

4. Rekan dan sahabat, terutama angkatan '91, dan semua pihak yang telah banyak memberi saran dan bantuan dalam mengerjakan tugas akhir ini.

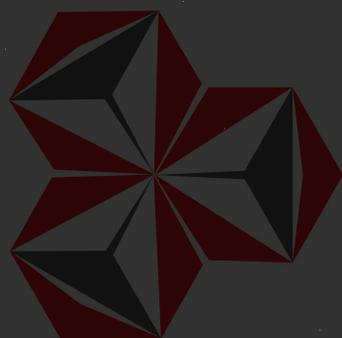
Semoga Tuhan yang Maha Esa memberikan berkat dan rahmat-Nya atas segala kebaikan budi mereka.

Penulis menyadari bahwa dalam tugas akhir ini masih banyak terdapat kekurangan, oleh karena itu saran dan kritik yang membangun sangat penulis harapkan sebagai bahan pertimbangan dan perbaikan di masa yang akan datang.

Akhir kata penulis berharap semoga tugas akhir ini dapat berguna bagi Pembaca semua.



UNIVERSITAS
Dinamika
Surabaya, Juni 1997
Penulis

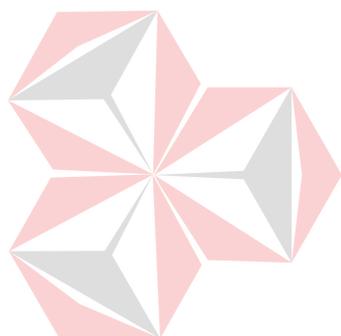


UNIVERSITAS
Dinamika

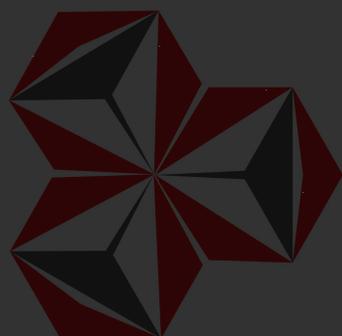
DAFTAR ISI

	Halaman
ABSTRAKSI.....	iii
KATA PENGANTAR.....	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	ix
DAFTAR TABEL.....	xi
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Permasalahan.....	2
1.3. Tujuan.....	3
1.4. Metode Penelitian.....	4
1.5. Sistematika.....	4
BAB II LANDASAN TEORI.....	6
2.1. Laboratorium Virtual.....	6
2.2. Rangkaian Resistor-Induktor-Kapasitor.....	9
2.3. Komputasi Numerik Yang Dipakai.....	22
2.4. Grafik Turbo Pascal.....	24
BAB III PERMASALAHAN.....	32
BAB IV PERANCANGAN DAN IMPLEMENTASI.....	35
4.1. Perangkat Pendukung.....	35
4.2. Rancangan Perangkat Lunak.....	36
4.3. Analisa Hasil.....	53
BAB V KESIMPULAN DAN SARAN.....	56
5.1. Kesimpulan.....	56
5.2. Saran.....	56

DAFTAR PUSTAKA.....	58
LAMPIRAN.....	59



UNIVERSITAS
Dinamika

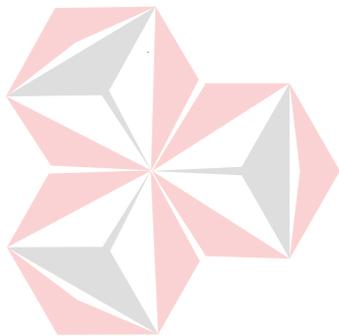


UNIVERSITAS
Dinamika

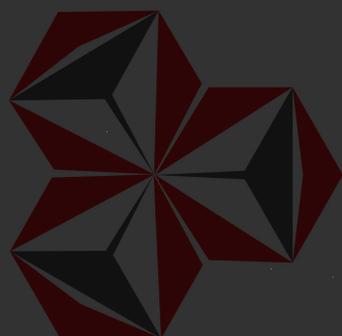
DAFTAR GAMBAR

	Halaman
Gambar 2.1. Simbol sumber tegangan sinusoidal.....	10
Gambar 2.2. Simbol resistor.....	10
Gambar 2.3. Simbol induktor.....	11
Gambar 2.4. Simbol kapasitor.....	11
Gambar 2.5. Formasi paralel dan ekuivalen-nya.....	13
Gambar 2.6. Formasi <i>star</i> dan <i>mesh</i> ekuivalen-nya.....	14
Gambar 2.7. Rangkaian dengan empat simpul.....	15
Gambar 2.8. Rangkaian dengan dua tegangan simpul.....	16
Gambar 2.9. Rangkaian dengan simpul-pembantu.....	19
Gambar 2.10. Rangkaian ter-transformasi.....	20
Gambar 2.11. Rangkaian dengan ground bukan di terminal sumber tegangan.....	21
Gambar 2.12. Sistem koordinat (modus 640x480).....	26
Gambar 4.1. Bentuk visual perangkat lunak.....	36
Gambar 4.2. Rancangan karakter R, L, C, V horisontal..	38
Gambar 4.3. Rancangan karakter kabel.....	38
Gambar 4.4. Pembagian bit kode ASCII karakter.....	38
Gambar 4.5. Struktur file skema.....	40
Gambar 4.6. Struktur list.....	42
Gambar 4.7. Struktur variabel record pada Pascal.....	43
Gambar 4.8. Rangkaian dengan dan tanpa sumber tegangan	45
Gambar 4.9. Struktur matriks tegangan simpul.....	48
Gambar 4.10. Reduksi rangkaian untuk memperoleh nilai admitansi antara dua simpul	49
Gambar 4.11. Contoh <i>probing</i> pada rangkaian.....	51

Gambar 4.12. Dua cara *probing*..... 52
Gambar 4.13. Rangkaian uji coba..... 54



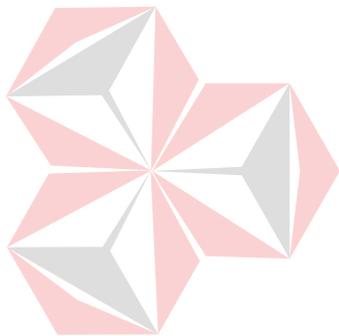
UNIVERSITAS
Dinamika



UNIVERSITAS
Dinamika

DAFTAR TABEL

	Halaman
Tabel 2.1. Driver grafik.....	25
Tabel 2.2. Kode kesalahan hasil fungsi <i>GraphResult</i>	30
Tabel 4.1. Perbandingan perhitungan $ V $	55
Tabel 4.2. Perbandingan perhitungan Z.....	55



UNIVERSITAS
Dinamika

BAB I

PENDAHULUAN

1.1. Latar Belakang

Keberadaan perangkat komputer di rumah atau instansi sudah merupakan hal yang umum. Hal itu beranjak dari kemampuan komputer itu sendiri yang membantu meningkatkan efisiensi waktu, tenaga, dan ruang, baik di bidang perindustrian, pendidikan, dan bidang-bidang lainnya.

Kenyataan yang ada bahwa perangkat komputer umumnya hanya digunakan sebagai pengolah kata, lembar kerja, atau sebagai pengolah file. Padahal komputer juga dapat digunakan untuk meniru perangkat lain, misalnya oscilloscope (seperti yang telah dilakukan oleh National Instruments Inc. yang berupa perangkat keras dan perangkat lunak driver-nya), yang dipakai untuk melihat bentuk tegangan listrik. Keuntungan yang didapatkan di sini terutama dari segi ekonomis karena komputer dapat berperan sebagai oscilloscope, sedangkan oscilloscope tidak dapat menggantikan kedudukan komputer untuk peran yang lainnya, disamping itu sekarang ini komputer sudah memasyarakat.

Melihat hal di atas, akan lebih baik lagi jika penggunaan komputer lebih dioptimalkan, yaitu dipakai untuk melakukan simulasi. Salah satunya adalah untuk mensimulasikan sebuah laboratorium elektronika yang terdiri dari oscilloscope dan rangkaian percobaannya,

sehingga laboratorium elektronika virtual ini dapat dipakai sebagai sarana belajar atau sarana praktikum tanpa mengharuskan pemakai mempunyai peralatan yang sebenarnya. Hal ini juga akan memberikan banyak keuntungan daripada mengadakan laboratorium yang sesungguhnya, baik dari segi biaya, lokasi, dan keamanan.

1.2. Permasalahan

1.2.1. Identifikasi

Berdasarkan latar belakang yang ada, akan timbul suatu permasalahan tentang bagaimana menciptakan suatu alat bantu belajar dengan menggunakan sebuah komputer, sehingga lebih bermanfaat bagi pemakai dalam memahami suatu objek permasalahan yang dalam hal ini tentang karakteristik rangkaian resistor-induktor-kapasitor.

Perangkat yang disimulasikan meliputi oscilloscope, rangkaian percobaan, dan sumber tegangan. Jadi memungkinkan untuk membuatnya dalam bentuk perangkat lunak saja. Dengan demikian hal ini memberikan banyak keuntungan karena pengadaan sebuah laboratorium disamping membutuhkan biaya yang cukup besar juga memerlukan tempat cukup luas dan mungkin juga membutuhkan biaya tambahan yang disebabkan kerusakan peralatan baik secara alami maupun karena kesalahan pemakai. Dengan menggunakan laboratorium virtual, selain penghematan biaya, dan tempat, percobaan dapat dilakukan sendiri oleh pemakai

kapan saja diinginkan dan di mana saja, juga tanpa resiko yang mungkin membahayakan pemakai karena penggunaan tegangan tinggi.

1.2.2. Ruang lingkup

Ruang lingkup dari tugas akhir ini adalah:

1. Visualisasi yang dibuat berfungsi sebagai sebuah laboratorium untuk bidang elektronika.
2. Visualisasi yang dibuat ditekankan untuk melihat bentuk tegangan listrik pada rangkaian resistor-induktor-kapasitor (RLC) yang dihubungkan dengan sebuah sumber tegangan sinusoidal.
3. Rangkaian RLC yang dibentuk adalah rangkaian sebidang (jika diagram rangkaian dapat digambar pada sebuah permukaan bidang sedemikian rupa sehingga tak ada cabang yang melalui di atas atau dibawah cabang lain, maka rangkaian tersebut dinamai rangkaian sebidang).
4. Perangkat lunak yang dibuat, membutuhkan perangkat keras dengan spesifikasi minimal sebagai berikut:
 - a. IBM PC/AT 386, memori 640 KB.
 - b. VGA card dan monitor VGA.
 - c. Mouse (kompatibel dengan Microsoft-Mouse).

1.3. Tujuan

Tujuan dari tugas akhir ini adalah merancang dan membuat program yang berfungsi sebagai laboratorium

elektronika virtual, yang dapat membantu pemakai dalam mempelajari elektronika khususnya tentang karakteristik rangkaian RLC.

1.4. Metode Penelitian

Dari masalah yang akan dipecahkan maka metode yang akan digunakan untuk menyelesaikan masalah tersebut adalah:

1. Studi literatur pada beberapa buku yang berkaitan dengan masalah ini.
2. Melakukan studi banding terhadap beberapa perangkat lunak yang sudah ada sebagai bahan pertimbangan dengan memperhatikan kelebihan dan kekurangan dalam perangkat lunak tersebut.

1.5. Sistematika

Untuk mempermudah pemahaman laporan tugas akhir ini, penyusunan dibuat dalam bab-bab sebagai berikut:

Bab I, menjelaskan secara ringkas mengenai isi tugas akhir yang meliputi latar belakang dan permasalahan pembuatan laboratorium elektronika virtual, ruang lingkup dari perangkat lunak yang dibuat, tujuan pembuatan perangkat lunak ini, metode penelitian untuk menyelesaikan tugas akhir ini, serta sistematika laporan tugas akhir ini.

Bab II, membahas tentang teori-teori yang dipakai dalam tugas akhir ini, yang meliputi teori mengenai

laboratorium virtual, teori elektronika dan komputasi yang digunakan dalam perangkat lunak ini, serta fasilitas untuk pemrograman grafik yang dimiliki Turbo Pascal.

Bab III, membahas tentang beberapa permasalahan yang perlu diantisipasi dalam tugas akhir ini, meliputi permasalahan visualisasi, kondisi rangkaian yang terbentuk, maupun permasalahan pemrograman.

Bab IV, membahas tentang perangkat yang dibutuhkan (perangkat lunak dan perangkat keras), perancangan visualisasi laboratorium elektronika, perancangan struktur data untuk menggambar rangkaian maupun untuk proses perhitungan, serta analisa perangkat lunak ini.

Bab V, berisi kesimpulan-kesimpulan yang di dapat dengan adanya perangkat lunak ini serta saran-saran yang diharapkan dapat dipakai untuk pengembangan lebih lanjut.

BAB II

LANDASAN TEORI

Laboratorium elektronika ini adalah sebuah perangkat lunak yang berfungsi sebagai sebuah laboratorium untuk bidang elektronika, khususnya tentang karakteristik rangkaian RLC (resistor-induktor-kapasitor). Oleh karena itu, dalam perancangan perangkat lunak ini diperlukan teori-teori yang mendukung, baik mengenai pemrograman yang digunakan maupun teori-teori elektronika khususnya mengenai karakteristik rangkaian RLC.

2.1. Laboratorium Virtual

Dewasa ini pemanfaatan komputer dalam dunia pendidikan sudah semakin luas. Ini menunjukkan bahwa kemajuan teknologi akan banyak berpengaruh pada kemajuan di bidang ilmu pengetahuan. Demikian juga sebaliknya, kemajuan di bidang ilmu pengetahuan juga akan meningkatkan kemajuan dalam dunia teknologi.

Proses belajar-mengajar misalnya, komputer dapat digunakan seperti seorang pakar yang akan memberikan informasi-informasi yang berhubungan dengan ilmu yang dikuasainya. Jadi, fungsi komputer di sini adalah sebagai seorang guru. Disamping itu, dengan adanya alat bantu komputer, akan memberikan dorongan dan rangsangan terhadap pemakai untuk mau belajar. Dengan teknologi

jaringan, kita dapat membuat suatu kelas elektronik di mana antara guru dengan siswa dan antar siswa dapat saling berkomunikasi melalui komputer pada suatu jaringan.

Selain itu, komputer juga dapat digunakan sebagai suatu laboratorium, di mana komputer akan mensimulasikan dan merepresentasikan sebuah laboratorium. Fungsi komputer seperti tersebut di atas disebut laboratorium virtual. Contoh-contoh laboratorium virtual adalah laboratorium virtual kimia, laboratorium virtual fisika, laboratorium virtual alat ukur, dan lain sebagainya.

Dalam suatu perangkat lunak laboratorium virtual, ada beberapa hal yang perlu diperhatikan:

1. *Kemampuan pemakai.* Keberhasilan penyampaian materi sangat tergantung pada tingkat kemampuan pemakai. Pemakai dapat terdiri dari berbagai kalangan sehingga penyampaian materi harus dibuat sederhana dan mudah untuk dimengerti.
 2. *Rentang perhatian pemakai.* Sebagai salah satu perangkat ajar, materi yang disampaikan diupayakan dapat mengarahkan dan mempertahankan perhatian pemakai dengan merancang bentuk tampilan juga memperhatikan jangka waktu belajar yang berpengaruh terhadap konsentrasi pemakai.
 3. *Basis pengetahuan.* Agar materi dalam tugas akhir ini mudah dimengerti dan dipahami oleh pemakai, maka
-

materi yang disampaikan hendaknya merupakan sesuatu yang menjadi dasar dan sudah dimiliki oleh pemakai. Dengan demikian materi tersebut dapat menambah pengetahuan khususnya tentang karakteristik rangkaian resistor-induktor-kapasitor.

4. *Pendekatan dalam pengajaran.* Pendekatan dalam pengajaran tergantung pada tujuan dari belajar tersebut. Dalam tugas akhir ini, tujuan pengajaran adalah pemahaman konsep dalam mempelajari karakteristik rangkaian resistor-induktor-kapasitor. Penggunaan komputer sebagai perangkat ajar, memungkinkan adanya komunikasi dua arah dengan pemakai, meskipun reaksi dari komputer terhadap respon pemakai terlihat kaku dan kurang alamiah.
 5. *Penerapan prinsip belajar.* Pada dasarnya prinsip belajar adalah mencoba dan mengulangi suatu aktifitas, karena dengan sering mencoba dan mengulangi aktifitas tersebut, dan membandingkannya, maka pemakai akan lebih memahami tentang aktifitas yang sedang mereka pelajari. Oleh karena itu dalam tugas akhir ini pemakai diarahkan untuk mencoba dan mengulangi materi yang disajikan. Jadi perlu diupayakan agar perangkat ajar ini tidak membosankan sehingga pemakai akan sering belajar.
 6. *Perancangan visualisasi.* Dalam tugas akhir ini, visualisasi, sebagai salah satu media komunikasi
-

antara komputer dengan pemakai, dibuat semenarik mungkin sehingga dapat mendukung kegiatan belajar yang diinginkan. Visualisasi dirancang sedemikian rupa agar dapat menyampaikan materi khususnya tentang karakteristik rangkaian resistor-induktor-kapasitor, dan membuat gagasan abstrak menjadi konkrit. Layar komputer tidak dapat memuat informasi sekaligus sehingga layar yang terlalu penuh akan sulit dibaca dan dipahami yang dapat mengakibatkan penyampaian informasi tidak efektif. Jadi perlu adanya perancangan visualisasi yang baik.

2.2. Rangkaian Resistor-Induktor-Kapasitor

2.2.1. Elemen-elemen dasar elektronika

Elemen rangkaian dapat dibedakan sebagai elemen rangkaian umum dan elemen rangkaian sederhana. Sebuah elemen rangkaian umum terdiri lebih dari satu elemen rangkaian sederhana, dan sebuah elemen rangkaian sederhana tidak dapat dibagi lagi menjadi elemen-elemen rangkaian lain.

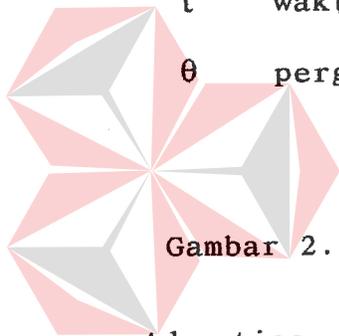
Sambungan antara dua atau lebih elemen rangkaian sederhana dinamai jaringan (*network*). Jika jaringan mengandung sedikitnya satu jalur tertutup, jaringan itu disebut rangkaian (*circuit*) listrik.

Sebuah sumber tegangan bebas adalah sumber yang ditandai dengan sebuah tegangan terminal yang sama sekali tidak bergantung pada arus yang melaluinya.

Untuk sumber tegangan sinusoidal (tegangan bolak-balik) dinyatakan dengan:

$$v = V_m \sin(\omega t + \theta) \quad (2.1)$$

- v tegangan saat t (Volt)
 V_m tegangan maksimum (Volt)
 ω ($= 2\pi f$) frekuensi angular (radian/detik)
 t waktu (detik)
 θ pergeseran fase (radian)



UNIVERSITAS



Gambar 2.1. Simbol sumber tegangan sinusoidal

Ada tiga jenis elemen elektrik dasar dari suatu rangkaian listrik.

1. *Resistor*. Resistor adalah elemen yang menghamburkan energi sebagai panas atau yang lain. Hubungan antara tegangan, arus, dan resistansi:

$$v = R i \quad (2.2)$$

- v tegangan pada ujung-ujung elemen (Volt)
 i arus yang melalui elemen (Ampere)
 R resistansi (Ohm)



Gambar 2.2. Simbol resistor

2. *Induktor*. Induktor adalah elemen yang menyimpan energi yang bergantung pada aliran arus. Hubungan antara tegangan, perubahan arus, dan induktansi:

$$v = L \frac{di}{dt} \quad (2.3)$$

v tegangan pada ujung-ujung elemen (Volt)

di/dt rate perubahan arus (Ampere/detik)

L induktansi (Henry)



Gambar 2.3. Simbol induktor

3. *Kapasitor*. Kapasitor adalah elemen yang menyimpan energi yang bergantung pada tegangan. Hubungan antara perubahan tegangan, arus, dan kapasitansi:

$$i = C \frac{dv}{dt} \quad (2.4)$$

dv/dt rate perubahan tegangan (Volt/detik)

i arus yang melalui elemen (Ampere)

C kapasitansi (Farad)



Gambar 2.4. Simbol kapasitor

Impedansi (Z) adalah perbandingan tegangan kepada arus yang berkuantitas kompleks yang berdimensi ohm. Untuk ketiga elemen dasar resistor, induktor, dan kapasitor:

$$\begin{aligned} V &= Z I \\ Z_R &= R \\ Z_L &= j \omega L \\ Z_C &= -\frac{j}{\omega C} \end{aligned} \quad (2.5)$$

dan karena Z bekerja dibidang kompleks, maka V perlu ditransformasikan ke bidang kompleks juga.

$$V = V_m \cos(\omega t + \theta) + j V_m \sin(\omega t + \theta) \quad (2.6)$$

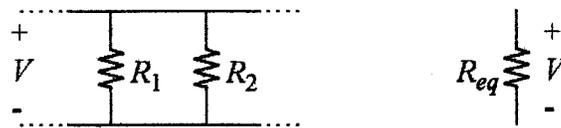
Kebalikan dari resistansi adalah *konduktansi* yang mempunyai simbol g , dan kebalikan dari impedansi adalah *admitansi* yang mempunyai simbol Y . Satuan dari konduktansi dan admitansi adalah *mho*.

2.2.2. Resistansi ekuivalen

Resistansi ekuivalen dapat didefinisikan sebagai suatu resistansi baru yang memakai arus yang sama seperti pada rangkaian yang asli jika pada dua titik terminal dari resistansi baru diberi tegangan yang sama.

Berdasarkan formasi susunan elemen-elemen, dapat dibedakan menjadi dua:

1. *Formasi paralel.* Dua buah elemen atau lebih dikatakan mempunyai formasi paralel jika dua terminal dari masing-masing elemen berhubungan langsung antara elemen yang satu dengan elemen yang lain. Resistansi ekuivalen dari formasi ini berupa sebuah elemen saja. Secara umum, diperlihatkan seperti pada Gambar berikut:



Gambar 2.5. Formasi paralel dan ekuivalen-nya

Persamaan untuk mencari nilai R ekuivalen:

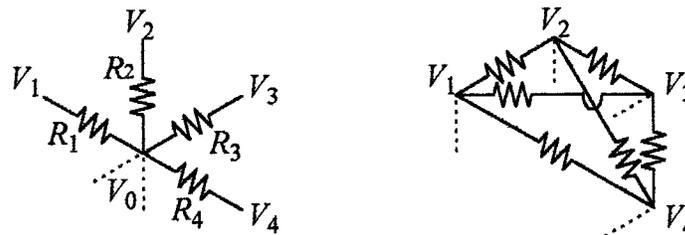
$$\frac{1}{R_{eq}} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} + \dots \quad (2.7)$$

atau jika menggunakan konduktansi:

$$G_{eq} = G_1 + G_2 + G_3 + \dots \quad (2.8)$$

2. *Formasi star.* Dua buah elemen atau lebih dikatakan mempunyai formasi *star* jika salah satu terminal dari masing-masing elemen berhubungan langsung antara elemen yang satu dengan elemen yang lain, dan terminal yang satu lagi tidak berhubungan langsung antara elemen yang satu dengan elemen yang lain. Jadi, formasi seri dan Y sudah tercakup. Resistansi

ekuivalen dari formasi star ini menghasilkan $n(n-1)/2$ elemen dengan formasi *mesh*. Secara umum, formasi *star* diperlihatkan seperti pada rangkaian kiri dari Gambar berikut, sedangkan rangkaian kanan adalah rangkaian *mesh*, yaitu rangkaian hasil transformasi rangkaian *star*.



Gambar 2.6. Formasi *star* dan *mesh* ekuivalen-nya

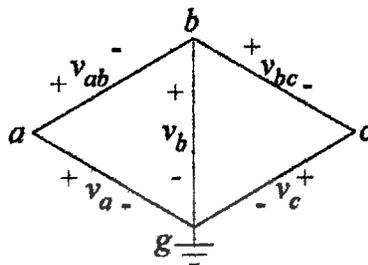
Persamaan untuk mencari R_{mesh} ekuivalen:

$$G_{\text{mesh}} = \frac{\text{Perkalian } G_{\text{star}} \text{ yang berdampingan}}{\text{Jumlah semua } G_{\text{star}}} \quad (2.9)$$

2.2.3. Metode tegangan-simpul

Tegangan di sebuah simpul (titik cabang) dapat diumpamakan seperti menyatakan elevasi atau ketinggian suatu tempat, yang biasanya diukur relatif terhadap permukaan laut yang berperan sebagai ketinggian acuan. Hal serupa juga ada dalam jaringan elektrik; level tegangan di suatu simpul atau titik dinyatakan berdasarkan titik acuan yang lebih umum disebut "ground".

Jika sebuah rangkaian mempunyai empat simpul, maka ada tiga tegangan yang tak diketahui. Dari Gambar 2.7, tiga simpul yang dimaksud adalah simpul a, b, dan c. Sebuah rangkaian yang mempunyai sepuluh simpul, maka ada sembilan tegangan yang tak diketahui; dan sebuah rangkaian yang mempunyai n simpul, ada (n-1) tegangan yang tak diketahui.

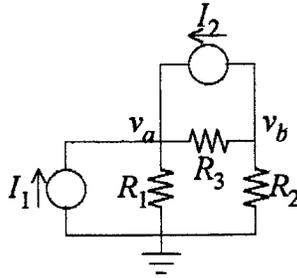


Gambar 2.7. Rangkaian dengan empat simpul

Dari rangkaian di atas, tegangan di simpul a, b, dan c yang berhubungan langsung dengan simpul acuan g adalah v_a , v_b , dan v_c . Tegangan antara a dan b, berbeda dengan tegangan di a atau tegangan di b; begitu juga tegangan antara b dan c, berbeda dengan tegangan di b atau tegangan di c. Berdasarkan hukum tegangan Kirchoff:

$$v_{ab} = v_a - v_b \quad v_{bc} = v_b - v_c \quad (2.10)$$

Sebagai contoh, kita tinjau rangkaian pada Gambar 2.8, yang mempunyai dua tegangan simpul yang tak diketahui.



Gambar 2.8. Rangkaian dengan dua tegangan simpul

Persamaan hukum-arus untuk rangkaian ini adalah:

$$v_a \frac{1}{R_1} + (v_a - v_b) \frac{1}{R_2} - I_1 - I_2 = 0 \quad (2.11)$$

dan

$$(v_b - v_a) \frac{1}{R_2} + v_b \frac{1}{R_3} + I_2 = 0 \quad (2.12)$$

Persamaan (2.11) menyatakan jumlah arus yang keluar dari simpul v_a . Suku pertama menyatakan arus yang melalui R_1 , dan suku kedua menyatakan arus yang melalui R_2 . Suku-suku sisanya menyatakan sumber arus. Persamaan (2.12) menyatakan jumlah arus yang keluar dari simpul v_b . Suku pertama menyatakan arus yang melalui R_2 , suku kedua menyatakan arus yang melalui R_3 , dan suku ketiga menyatakan sumber arus (sebagai catatan, tegangan di R_2 dapat berupa $v_a - v_b$ atau $v_b - v_a$, tergantung polaritas yang diasumsikan). Biasanya persamaan (2.11) dan (2.12) ditulis dengan bentuk berikut:

$$v_a \left(\frac{1}{R_1} + \frac{1}{R_2} \right) - v_b \frac{1}{R_2} = I_1 + I_2 \quad (2.13)$$

dan

$$-v_a \frac{1}{R_1} + v_b \left(\frac{1}{R_2} + \frac{1}{R_3} \right) = -I_2 \quad (2.14)$$

Suku pertama dalam persamaan (2.13) menyatakan jumlah arus yang keluar dari simpul v_a jika simpul-simpul lainnya dianggap nol atau dihubungkan dengan ground. Koefisien yang dikalikan dengan v_a adalah jumlah konduktansi yang menyebar dari simpul. Dengan kata lain, semua hambatan yang berhubungan langsung dengan simpul ini, diikutkan ke persamaan sebagai koefisien. Suku kedua adalah arus yang keluar dari simpul v_a jika v_b dianggap positif dan v_a di-ground-kan. Yang terjadi, suku ini bernilai negatif karena arus mengalir masuk ke simpul v_b , bukan keluar. Suku-suku di sisi kanan menyatakan sumber arus yang masuk ke simpul v_a .

Ada hal khusus yang berkenaan dengan persamaan-persamaan di atas, yaitu berbentuk simetris dan elemen-elemen di diagonal utama (dari kiri atas sampai kanan bawah) bernilai positif. Elemen-elemen inilah yang akan memberikan nilai tegangan pada simpul yang bersangkutan. Elemen-elemen lain di luar diagonal utama, bernilai negatif. Adanya hal ini, sangat membantu dalam

mengurangi kesalahan-kesalahan kecil selama mengerjakan perhitungan. Dan dari kedua persamaan di atas, matriks konduktansinya:

$$\begin{bmatrix} \frac{1}{R_1} + \frac{1}{R_2} & -\frac{1}{R_2} \\ -\frac{1}{R_1} & \frac{1}{R_2} + \frac{1}{R_3} \end{bmatrix} \quad (2.15)$$

Jika terdapat k simpul yang tak diketahui maka persamaan yang didapat:

$$\begin{aligned} v_1(g_{11}) - v_2(g_{12}) - \dots - v_k(g_{1k}) &= I_1 \\ -v_1(g_{12}) + v_2(g_{22}) - v_k(g_{2k}) &= I_2 \\ &\vdots \\ -v_1(g_{1k}) - v_2(g_{2k}) + v_k(g_{kk}) &= I_k \end{aligned} \quad (2.16)$$

di mana

v_1 tegangan simpul di simpul 1,

I_1 sumber arus yang masuk ke simpul 1,

g_{11} total konduktansi yang menyebar dari simpul 1,

g_{12} total konduktansi antara simpul 1 dan 2,

dan seterusnya.

2.2.4. Tegangan simpul-pembantu

Persamaan tegangan simpul ditulis untuk jumlah arus yang meninggalkan simpul. Sumber-sumber arus yang ada dapat dimasukkan langsung ke persamaan. Lain halnya

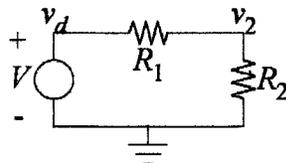
dengan sumber tegangan. Sumber tegangan membutuhkan langkah-langkah tambahan agar dapat dimasukkan ke persamaan. Ada dua metode:

1. Transformasi sumber, yaitu mengkonversi sumber-sumber tegangan menjadi sumber-sumber arus.
2. Menggunakan tegangan simpul-pembantu.

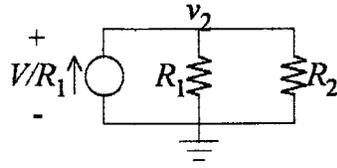
Transformasi sumber, selalu dapat diaplikasikan, tetapi cara ini mengubah bentuk jaringan atau rangkaian. Sedangkan metode tegangan simpul-pembantu tidak mengubah bentuk jaringan atau rangkaian.

Setiap sumber tegangan, dapat dipertimbangkan untuk dipakai untuk memunculkan sebuah simpul pembantu. Setiap simpul pembantu dapat dimasukkan ke dalam persamaan dengan berperan mirip sebuah simpul tegangan biasa, bedanya simpul tegangan ini sudah diketahui nilainya.

Sebagai contoh, rangkaian yang sebenarnya adalah seperti pada Gambar 2.9 dan rangkaian yang tertransformasi ada pada Gambar 2.10.



Gambar 2.9. Rangkaian dengan simpul-pembantu



Gambar 2.10. Rangkaian ter-transformasi

Untuk rangkaian yang kedua, persamaan yang diperoleh:

$$v_2 \left(\frac{1}{R_1} + \frac{1}{R_2} \right) = \frac{V}{R_1} \quad (2.17)$$

atau

$$v_2 = V \frac{R_2}{R_1 + R_2} \quad (2.18)$$

Tegangan v_2 yang dihasilkan dari persamaan (2.18) adalah benar untuk rangkaian pertama dan kedua, tetapi tegangan pada resistor R_1 untuk kedua rangkaian, tidak sama.

Rangkaian pada Gambar 2.9, metode tegangan simpul-pembantu diaplikasikan secara langsung. Sebuah tegangan simpul-pembantu diasumsikan terletak di terminal sumber tegangan. Persamaan kendalanya sederhana:

$$v_d = V \quad (2.19)$$

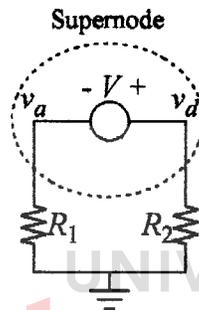
dan persamaan hukum arusnya:

$$v_2 \left(\frac{1}{R_1} + \frac{1}{R_2} \right) - v_d \frac{1}{R_1} = 0 \quad (2.20)$$

dari dua persamaan di atas diperoleh:

$$v_2 = V \frac{R_2}{R_1 + R_2} \quad (2.21)$$

Masalah yang lain adalah jika ground tidak diletakkan di terminal sumber tegangan. Metode ini masih dapat diaplikasikan, tetapi persamaan kendalanya menjadi lebih kompleks. Contohnya seperti rangkaian berikut:



Gambar 2.11. Rangkaian dengan ground bukan di terminal sumber tegangan

Rangkaian ini hanya mempunyai sebuah simpul tegangan yang tak diketahui dan sebuah simpul ground, sebuah simpul lagi adalah simpul-pembantu. Tidak menjadi masalah, simpul mana yang merupakan simpul-pembantu dan simpul mana yang merupakan simpul tegangan yang tak diketahui. Persamaan kendala untuk rangkaian ini:

$$v_d = v_a + V \quad (2.22)$$

dan persamaan hukum arusnya:

$$\frac{v_a}{R_1} + \frac{v_d}{R_2} = 0 \quad (2.23)$$

Persamaan di atas diperoleh dengan menganggap sumber tegangan sebagai sebuah simpul saja, yaitu sebuah simpul-super (*supernode*), sehingga jumlah arus yang keluar dari simpul-super ini sama dengan nol. Dan, dari kedua persamaan di atas diperoleh:

$$v_a = -\frac{R_1}{R_1 + R_2}V \quad v_d = \frac{R_2}{R_1 + R_2}V \quad (2.24)$$

2.3. Komputasi Numerik Yang Dipakai

2.3.1. Operasi perhitungan dasar bilangan kompleks

Jika diketahui dua bilangan kompleks:

$$A = a + jb \quad B = c + jd$$

maka, penjumlahan/pengurangan kedua bilangan ini adalah:

$$(a + jb) \pm (c + jd) = (a \pm c) + j(b \pm d) \quad (2.25)$$

perkalian:

$$(a + jb)(c + jd) = (ac - bd) + j(bc + ad) \quad (2.26)$$

pembagian:

$$\frac{a + jb}{c + jd} = \frac{(ac + bd) + j(bc - ad)}{c^2 + d^2} \quad (2.27)$$

2.3.2. Penyelesaian persamaan linier simultan

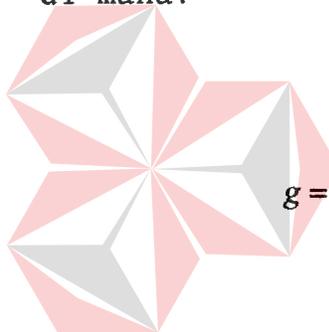
Diketahui ada k persamaan linier seperti berikut:

$$\begin{aligned} g_{11}v_1 \quad g_{12}v_2 \quad \dots \quad g_{1k}v_k &= I_1 \\ g_{21}v_1 \quad g_{22}v_2 \quad \dots \quad g_{2k}v_k &= I_2 \\ \vdots & \\ g_{k1}v_1 \quad g_{k2}v_2 \quad \dots \quad g_{kk}v_k &= I_k \end{aligned} \quad (2.28)$$

Persamaan-persamaan di atas dapat dinotasikan sebagai:

$$gv = I \quad (2.29)$$

di mana:



$$g = \begin{bmatrix} g_{11} & g_{12} & \dots & g_{1k} \\ g_{21} & g_{22} & & g_{2k} \\ \vdots & & & \\ g_{k1} & g_{k2} & & g_{kk} \end{bmatrix} \quad v = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_k \end{bmatrix} \quad I = \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_k \end{bmatrix} \quad (2.30)$$

dan v adalah variabel yang dicari dengan:

$$v = g^{-1} I \quad (2.31)$$

Salah satu metode yang sering di pakai untuk menyelesaikan persamaan linier simultan adalah eliminasi Gauss, yang mempunyai algoritma seperti berikut:

1. [Eliminasi maju]

Ulangi sampai langkah (3) untuk $i := 1, 2, \dots, n$

2. [Membagi setiap elemen dalam baris dengan pivot]

Ulangi untuk $j := i, i+1, \dots, m$

$g(i,j) := g(i,j) / g(i,i)$

3. [Me-nol-kan elemen di bawah diagonal primer]

Ulangi untuk $k := i+1, i+2, \dots, n$

Ulangi untuk $j := i, i+1, \dots, m$

$g\{k,j\} := g(k,j) - g(k,j) * g(i,j)$

4. [Substitusi kembali]

Ulangi untuk $i := n, n-1, \dots, 2$

Ulangi untuk $k := 1, 2, \dots, i-1$

$g(k,m) := g(k,m) - g(k,i) * g(i,m)$

$g(k,i) := 0$

5. [Selesai]

Keluar

2.4. Grafik Turbo Pascal

2.4.1. Unit Graph

Unit Graph berisi lebih dari 50 rutin grafik dari *high-level*, seperti *SetViewPort*, *Circle*, *Bar3D*, *DrawPoly*, sampai rutin-rutin yang berorientasi bit, seperti *GetImage*, dan *PutImage*. Berbagai-macam pengisian (*fill*) dan bentuk garis (*line style*) juga didukung, dan ada cukup font yang dapat diperbesar, diatur perataannya, dan orientasi horisontal atau vertikal.

Untuk menjalankan program yang menggunakan unit *Graph*, dibutuhkan sebuah atau beberapa file driver grafik



(.BGI). Jika program menggunakan font vektor, dibutuhkan juga sebuah atau beberapa file font (.CHR).

2.4.2. Pengendali (Driver)

File-file driver grafik yang ada ditujukan untuk adapter grafik: CGA, MCGA, EGA, VGA, Hercules, AT&T 400 baris, 3270 PC, IBM-8514, dan VESA 16 warna.

Setiap driver berisi kode dan data yang disimpan dalam file yang terpisah (Tabel 2.1).

Tabel 2.1. Driver grafik

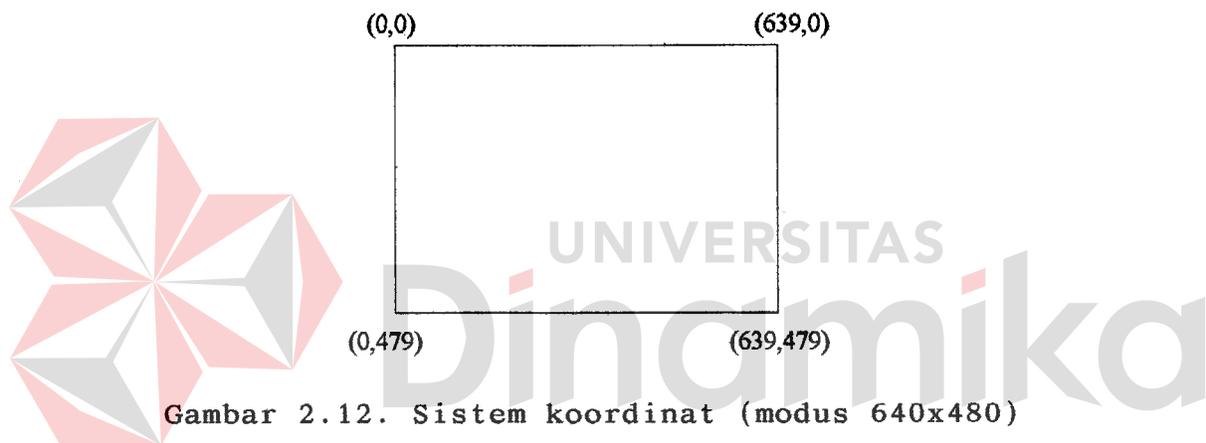
File driver	Adapter yang didukung
CGA.BGI	IBM CGA, MCGA
EGAVGA.BGI	IBM EGA, VGA
HERC.BGI	Monokrom Hercules
ATT.BGI	AT&T 6300 (400 baris)
PC3270.BGI	IBM 3270 PC
IBM8514.BGI	IBM 8514
VESA16.BGI	IBM VESA 16 warna

Pada saat dijalankan, prosedur *InitGraph* mengidentifikasi adapter grafik, mengambil dan menginisialisasi dengan driver grafik yang sesuai, dan mengeset video ke modus grafik. Prosedur *CloseGraph* bertindak berlawanan, yaitu melepas driver dari memori dan mengembalikan video ke modus sebelumnya. Untuk

berpindah-pindah antara modus teks dan grafik dapat digunakan rutin *RestoreCrtMode* dan *SetGraphMode*.

2.4.3. Sistem koordinat

Berdasarkan ketentuan yang ada, pojok kiri atas dari layar grafik adalah (0, 0). Nilai x (kolom), bertambah jika ke kanan. Nilai y (baris), bertambah jika ke bawah. Sebagai contoh, di dalam modus 640x480, koordinat layar (x, y) adalah seperti gambar berikut:



Gambar 2.12. Sistem koordinat (modus 640x480)

2.4.4. Lokasi aktif (current pointer)

Current Pointer (CP) mirip dengan konsep cursor di modus teks, hanya saja CP tidak tampak.

```
Write('ABC');
```

Di dalam modus teks, pernyataan *Write* di atas akan menempatkan cursor di kolom tepat setelah huruf C. Jika C di kolom 80, maka cursor akan berada di kolom 1 di baris berikutnya. Jika C di kolom 80 baris 25, seluruh

isi layar akan naik ke atas satu baris, dan cursor berada di kolom 1 baris 25.

```
MoveTo(0,0);  
LineTo(20,20);
```

Di dalam modus grafik, pernyataan *LineTo* di atas akan menempatkan CP di posisi (20,20).

Prosedur *MoveTo* ekivalen dengan *GotoXY*. Hanya saja digunakan untuk memindahkan CP.

Prosedur-prosedur ini, selain menggunakan CP, juga memindahkannya: *MoveRel*, *LineTo*, *LineRel*, *OutText*, *SetGraphMode**, *GraphDefaults**, *ClearDevice**, *SetViewPort**, dan *ClearViewPort** (prosedur-prosedur yang diikuti tanda asterisk (*) memindahkan CP ke (0,0)).

2.4.5. Teks

Sebuah font bitmap 8x8 dan beberapa font vektor, tersedia untuk penulisan selama di modus grafik. Huruf bitmap didesain dengan matriks pixel 8x8. Font vektor didesain dengan sederet vektor yang akan memberitahukan sistem bagaimana suatu huruf digambar.

Kelebihan font vektor adalah saat menggambar suatu huruf dengan ukuran yang diperbesar. Karena font vektor didesain dengan vektor-vektor, maka gambar huruf yang dihasilkan masih mempunyai resolusi dan kualitas yang cukup baik saat huruf itu dicetak dengan ukuran yang diperbesar.

Jika font bitmap diperbesar, matriks font akan dikalikan dengan skala yang diinginkan, akibatnya resolusi huruf yang dihasilkan terlihat kasar.

Perataan teks grafik diatur dengan prosedur *SetTextJustify*. Penentuan skala dan jenis font dilakukan dengan prosedur *SetTextStyle*. Untuk menampilkan teks grafik itu sendiri dapat menggunakan prosedur *OutText* atau *OutTextXY*. Informasi tentang setting teks saat itu digunakan prosedur *GetTextSettings*. Ukuran dari font vektor dapat diatur lebih lanjut dengan prosedur *SetUserCharSize*.

Sebuah program bantu, BINOBJ.EXE, dapat digunakan untuk mengkonversikan sebuah file font (atau file data biner) ke sebuah file .OBJ yang dapat di-link ke sebuah unit atau program menggunakan *compiler-directive* {\$L}. Hal ini akan memungkinkan sebuah program mempunyai semua file font di dalam file .EXE-nya.

2.4.6. Gambar dan corak (style)

Hampir semua rutin-rutin yang ada dibuat untuk menggambar dan mengisi (warna) suatu gambar. *SetLineStyle* digunakan untuk mengatur apakah bentuk garis diinginkan tebal atau tipis, atau apakah ingin digambar solid, berupa titik-titik putus, atau menggunakan pola-pola yang dapat didefinisikan sendiri.

SetFillStyle, *SetFillPattern*, *FillPoly*, dan *FloodFill* digunakan untuk keperluan mengisi sebuah bidang atau sebuah poligon dengan garis-garis silang atau dengan pola-pola lainnya.

2.4.7. Bidang gambar (viewport) dan citra bit (bit image)

Prosedur *ViewPort* membuat semua perintah output beroperasi pada sebuah bidang segi empat di layar. Titik, garis, gambar -- semua output grafik -- adalah relatif terhadap viewport sampai viewport tersebut diubah. Jika pemotongan diaktifkan, semua output grafik akan terpotong jika keluar dari viewport yang aktif saat itu.

GetPixel dan *PutPixel* digunakan untuk membaca dan menggambar titik (pixel). *GetImage* dan *PutImage* digunakan untuk menyimpan dan menggambar image dalam suatu bidang segi empat di layar. Prosedur dan fungsi di atas mendukung penuh operasi-operasi *BitBlt* (*copy*, *xor*, *or*, *and*, dan *not*).

2.4.8. Penanganan kesalahan

Kesalahan internal dalam unit Graph diperoleh dengan fungsi *GraphResult*. *GraphResult* menghasilkan kode kesalahan yang menunjukkan status operasi grafik terakhir. Kode-kode kesalahan tersebut didefinisikan seperti berikut:

Tabel 2.2. Kode kesalahan hasil fungsi *GraphResult*

Konstanta	Nilai	Keterangan/arti
<i>grOk</i>	0	Tidak ada kesalahan
<i>grNoInitGraph</i>	-1	Grafik belum diinstall
<i>grNotDetected</i>	-2	Adapter tidak terdeteksi
<i>grFileNotFound</i>	-3	File driver tidak ditemukan
<i>grInvalidDriver</i>	-4	File driver tidak sesuai
<i>grNoLoadMem</i>	-5	Memori tidak cukup untuk me-load driver
<i>grNoScanMem</i>	-6	Memori tidak cukup untuk scan fill
<i>grNoFloodMem</i>	-7	Memori tidak cukup untuk flood fill
<i>grFontNotFound</i>	-8	File font tidak ditemukan
<i>grNoFontMem</i>	-9	Memori tidak cukup untuk me-load font
<i>grInvalidMode</i>	-10	Modus grafik tidak sesuai untuk driver yang sedang aktif
<i>grError</i>	-11	Kesalahan grafik lainnya
<i>grIOError</i>	-12	Kesalahan I/O grafik
<i>grInvalidFont</i>	-13	File font tidak benar
<i>grInvalidFontNum</i>	-14	Nomor font tidak benar

Rutin-rutin berikut ini mempengaruhi hasil *GraphResult*:

<i>Bar</i>	<i>Bar3D</i>	<i>ClearViewport</i>
<i>CloseGraph</i>	<i>DetectGraph</i>	<i>DrawPoly</i>
<i>FillPoly</i>	<i>FloodFill</i>	<i>GetGraphMode</i>
<i>ImageSize</i>	<i>InitGraph</i>	<i>InstallUserDriver</i>
<i>InstalluserFont</i>	<i>PieSlice</i>	<i>RegisterBGIdriver</i>
<i>RegisterBGIfont</i>	<i>SetAllPalette</i>	<i>SetFillPattern</i>

<i>SetFillStyle</i>	<i>SetGraphBufSize</i>	<i>SetGraphMode</i>
<i>SetLineStyle</i>	<i>SetPalette</i>	<i>SetTextJustify</i>
<i>SetTextStyle</i>		

Perlu diketahui *GraphResult* akan menghasilkan nol setelah fungsi ini dipanggil. Jadi perlu variabel untuk menampung hasil *GraphResult* jika ingin memakai nilai ini lebih dari sekali pada suatu saat.

2.4.9. Prosedur dan fungsi lainnya

<i>CloseGraph</i>	Mengakhiri sistem grafik.
<i>GetImage</i>	Menyimpan sebuah image-bit pada suatu daerah ke sebuah buffer.
<i>InitGraph</i>	Menginisialisasi sistem grafik dan masuk ke modus grafik.
<i>PutImage</i>	Menggambar image-bit ke layar.
<i>SetRGBPallette</i>	Mengubah warna palet untuk IBM-8514 dan VGA dengan format RGB.
<i>SetWriteMode</i>	Mengeset modus penggambaran garis yang digambar dengan <i>DrawPoly</i> , <i>Line</i> , <i>LineRel</i> , <i>LineTo</i> , dan <i>Rectangle</i> .
<i>ImageSize</i>	Menghasilkan angka banyaknya byte untuk menyimpan image pada sebuah daerah kotak di layar.
<i>RegisterBGIDriver</i>	Meregister sebuah driver BGI yang cocok dengan sistem grafik.

BAB III

PERMASALAHAN

Sebagai suatu perangkat ajar, bentuk visual dan kemudahan pemakaian perangkat merupakan salah satu hal yang cukup penting, karena dalam proses belajar ada aspek-aspek yang perlu diperhatikan:

1. Tingkat kecerdasan dan minat ikut menentukan keberhasilan penyampaian materi. Perangkat ajar ini diharapkan dapat menimbulkan minat belajar.
2. Lingkungan yang menyenangkan, akan membantu pemakai dalam memahami suatu materi. Jadi, perangkat ajar ini diharapkan dapat digunakan di mana saja.
3. Dengan perangkat ajar ini, diharapkan dapat tercipta metode belajar yang lebih efektif dan efisien, karena dapat diulang, lebih aman, dan dapat digunakan kapan saja.

Dalam pembuatan perangkat lunak ini, ada beberapa hal yang perlu diperhatikan sehubungan dengan visualisasi yang akan dibuat, antara lain:

1. Penguasaan materi yang akan disajikan dalam pembuatan perangkat lunak ini sangat diperlukan untuk membentuk perangkat lunak yang cukup berkualitas, karena visualisasi yang akan dibuat sangat bergantung kepada materi yang akan disajikan.

2. Fungsi dan manfaat dari perangkat lunak yang akan dibuat juga perlu dipertimbangkan, apakah akan lebih menambah pemahaman bagi pemakai atau tidak. Hal ini karena sudah banyak alat bantu proses belajar-mengajar yang sering digunakan di lembaga-lembaga pendidikan formal maupun informal, misalnya laboratorium elektronika, dan lainnya. Dan sudah pasti bahwa semua perangkat tersebut sangatlah terbatas baik segi jumlah pemilik maupun waktu pemakaiannya.
3. Pengadaan perangkat keras dan perangkat lunak perlu mendapat perhatian karena merupakan tulang punggung dalam pembuatan alat bantu ini. Maksud dari perangkat lunak di sini adalah bahasa pemrograman yang sesuai dengan kebutuhan aplikasi yang akan dibuat. Sedangkan perangkat keras adalah perangkat keras minimal yang dibutuhkan untuk menunjang pembuatan visual serta untuk mengoperasikannya.
4. Pembuatan visual ini perlu mempertimbangkan ilmu pengetahuan dan teknologi yang dikuasai untuk membuat visual ini. Ilmu pengetahuan yang dimaksud adalah ilmu pengetahuan untuk mengoperasikan perangkat keras yang akan digunakan serta kemampuan pemrograman dalam modus grafik.

Visualisasi suatu laboratorium virtual perlu diusahakan sedemikian rupa agar menyerupai perangkat

aslinya sehingga dalam mempraktekkannya tidak terlalu jauh berbeda. Dalam hal ini, pendekatan yang paling baik adalah menggunakan modus grafik, terutama visualisasi skema dan grafik tegangan.

Gambar rangkaian dapat dibentuk dengan beberapa cara, antara lain dengan metode vektor, bitmap, atau kombinasi keduanya. Dalam memilih metode penggambaran skema ini perlu diperhatikan memori yang digunakan karena untuk perhitungan-perhitungan yang ada membutuhkan memori yang cukup besar.

Struktur data, baik list elemen maupun untuk proses perhitungan, perlu dirancang dengan baik agar dapat diimplementasikan mengingat keterbatasan kapasitas memori; terutama struktur data untuk matriks yang dipakai dalam perhitungan tegangan simpul.

Permasalahan yang lain adalah mengenai kondisi rangkaian yang digambar. Perlu adanya kemampuan untuk memeriksa rangkaian yang telah digambar, untuk memperingatkan pemakai jika pemakai membuat rangkaian yang membahayakan kalau dipraktekkan, yaitu jika pemakai menghubungkan-singkatkan sumber tegangan.

Saat melakukan penyidikan (*probing*) perlu adanya suatu cara agar komputer tahu elemen-elemen mana saja yang berhubungan dengan suatu probe.

BAB IV

PERANCANGAN DAN IMPLEMENTASI

Perangkat lunak yang dibuat dalam tugas akhir ini terdiri atas dua bagian utama, yaitu bagian untuk menggambar rangkaian (skema), dan bagian untuk melakukan pengukuran (perhitungan). Kedua bagian tersebut akan dijelaskan dalam sub bab tersendiri.

4.1. Perangkat Pendukung

Perangkat lunak yang baik adalah perangkat lunak yang sanggup memenuhi kebutuhan pemakai secara optimal. Agar dapat memenuhi kebutuhan tersebut, maka perangkat lunak harus didukung oleh perangkat pendukung yang baik, antara lain bahasa pemrograman dan perangkat keras.

Untuk membuat perangkat lunak ini, digunakan bahasa pemrograman Pascal dengan *compiler* Turbo Pascal 7.0 dengan pertimbangan strukturnya sistematis, mendukung pemrograman grafik, dan bahasa inilah yang paling dikuasai.

Untuk dapat menjalankan perangkat lunak ini, dibutuhkan perangkat keras dengan spesifikasi minimal:

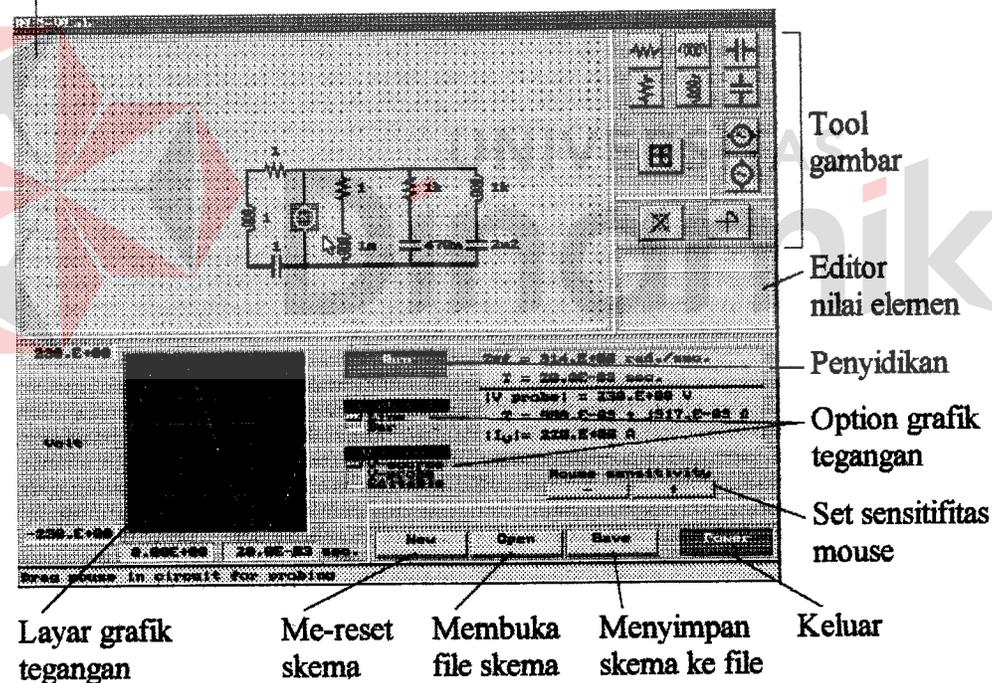
- a. IBM PC/AT 386, memori 640KB.
- b. VGA card dan monitor VGA.
- c. Mouse (kompatibel dengan Microsoft-Mouse).

4.2. Rancangan Perangkat Lunak

Visualisasi yang dibuat merupakan visualisasi laboratorium elektronika khususnya untuk memahami karakteristik rangkaian resistor-induktor-kapasitor. Rancangan dibuat dalam modus grafik dengan harapan dapat menimbulkan minat untuk memakai perangkat ini sehingga pemakai lebih memahami hal-hal yang dipelajarinya.

Garis besar sistem perangkat lunak ini adalah sebagai berikut:

Editor skema



Gambar 4.1. Bentuk visual perangkat lunak

4.2.1. Skema

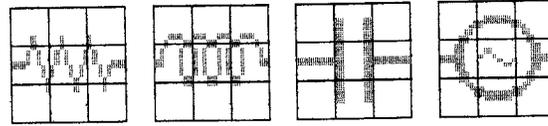
Bagian skema ini adalah bagian yang berkaitan langsung dengan proses pembentukan gambar skema.

Bagian ini mempunyai beberapa fungsi, yaitu:

1. Me-reset skema, yaitu: membersihkan editor skema, dan menghapus list elemen yang lama.
2. Membuka sebuah file skema, yaitu: me-reset skema, dan membaca file. Jika pembacaan file sukses, skema ditampilkan di editor; jika terjadi kesalahan pembacaan file, maka akan diberitahukan ke pemakai.
3. Menyimpan skema ke sebuah file. Jika penulisan file ke media terjadi kesalahan, maka akan diberitahukan ke pemakai.
4. Menggambar skema di editor menggunakan gambar-gambar elemen yang sudah ada.

A. Gambar elemen

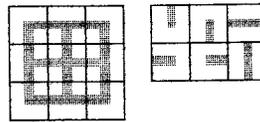
Seperti cara card grafik menampilkan tulisan ke monitor, gambar elemen R, L, C, V, dan gambar kabel, memakai cara yang serupa. Gambar elemen dibentuk dari 3x3 sel, di mana satu sel adalah berupa bitmap 8x8 pixel sehingga satu sel ini seperti sebuah karakter grafik dengan format font 8x8. Lebih jelasnya, rancangan dibuat seperti berikut:



Gambar 4.2. Rancangan karakter R, L, C, V horisontal

Tentu saja selain rancangan di atas masih ada rancangan untuk R, L, C, dan V yang vertikal.

Rancangan untuk kabel adalah seperti berikut:



Gambar 4.3. Rancangan karakter kabel

Setiap sel dari rancangan-rancangan di atas akan diwakili sebuah karakter dengan kode ASCII yang mempunyai struktur sebagai berikut:

7 6 5 4 3 2 1 0

Id elm.	V	U	D	L	R
---------	---	---	---	---	---

Gambar 4.4. Pembagian bit kode ASCII karakter

- Bit 7-5, dipakai untuk identitas jenis elemen, dengan aturan:
 - 000: resistor, dan mungkin juga spasi,
 - 001: induktor,
 - 010: kapasitor,
 - 011: sumber tegangan, dan
 - 100: kabel.
- Bit 4, dipakai untuk flag vertikal (1=vertikal).

- Bit 3-0, dipakai untuk flag arah sambungan kabel; dipakai juga untuk menentukan urutan posisi elemen R, L, C, dan V. Aturan yang dipakai:

1???: ada sambungan kabel ke atas,
?1??: ada sambungan kabel ke bawah,
??1?: ada sambungan kabel ke kiri,
???1: ada sambungan kabel ke kanan.

Misalnya untuk sel kiri-atas dari rancangan kabel 3x3 sel, mempunyai bit 3-0 kode ASCII 0101, dan kode ini juga dipakai untuk sel kiri-atas dari rancangan elemen elektronik baik vertikal maupun horisontal.

Semua rancangan di atas disimpan di sebuah file yang akan diperlakukan seperti data 8x8 font ROM yang ada di card grafik. Jadi editor skema tidak menyimpan data pixel layar, tetapi cukup data karakter yang mewakilinya, dengan demikian hanya dibutuhkan variabel array 2 dimensi untuk menyimpan karakter-karakter tersebut.

Cara yang dipakai ini menguntungkan dalam hal pemakaian memori dibanding jika harus menggambar setiap ruas garis yang ada dengan fasilitas grafik Turbo Pascal.

Dalam perangkat lunak ini, prosedur untuk menampilkan gambar-gambar elemen, adalah prosedur yang sama dengan prosedur untuk menulis teks. Untuk menggambar elemen, alamat font dialihkan ke data karakter elemen; sedangkan untuk menuliskan teks, alamat font dialihkan kembali ke ROM-font.

Rancangan yang dipilih adalah 3x3 sel karena gambar yang dihasilkan dirasa cukup baik. Rancangan 4x4 sel memang memperlihatkan gambar yang lebih jelas tetapi menyulitkan karena tidak ada baris atau kolom sel yang di tengah, yaitu baris atau kolom yang akan dihubungkan dengan kabel agar gambar tampak simetri. Sedangkan rancangan 5x5 sel, membuat layar skema menjadi cepat penuh.

Luas area skema dipilih 30x61 karakter karena jika terlalu luas, area untuk layar oscilloscope (grafik tegangan) terlalu kecil akibatnya grafik sinus yang tergambar menjadi kurang jelas.

B. File skema

Selama menggunakan perangkat lunak ini, sering diinginkan bahwa skema yang ada dapat disimpan ke sebuah file agar dapat digunakan lagi. Untuk keperluan ini struktur file skema dirancang seperti berikut:

Id file	4 byte
Gambar skema	1830 byte
Koord. V-sumber	2 byte
Amp. V-sumber	6 byte
Frek. V-sumber	6 byte
Fase V-sumber	6 byte
Koord. & nilai elm.-1	8 byte
Koord. & nilai elm.-n	8 byte

Gambar 4.5. Struktur file skema

1. 4 byte: identitas file, dalam tugas akhir ini adalah "RLC1". Jika saat "Open", 4 byte pertama tidak sama dengan "RLC1" maka file tersebut dianggap bukan file skema.
2. 1830 byte: data gambar skema.
3. 2 byte: koordinat sumber tegangan.
4. 6 byte: nilai amplitudo tegangan sumber.
5. 6 byte: nilai frekuensi sumber tegangan.
6. 6 byte: nilai pergeseran fase sumber tegangan.
7. 8 byte yang terdiri dari:
 - 2 byte: koordinat elemen (R, L, atau C).
 - 6 byte: nilai elemen (Ohm, Henry, atau Farad).

Butir ke-7 (8 byte) terdapat beberapa kali sesuai banyaknya elemen R, L, dan C dalam skema.

C. Struktur data

Nilai-nilai dari elemen (R, L, C) ditampung dalam heap dengan metode stack dengan struktur `_t_r_lst_rlc` berikut:

```

_t_r_yx = record
  case integer of
    0 : (x, y : byte);
    1 : (yx : word)
  end;

_t_r_val_rlc = record
  coord : _t_r_yx;
  value : real;
end;

_t_r_val_y = record
  pin_no : _t_r_pin_no;
  value : _t_r_cmplx;
end;

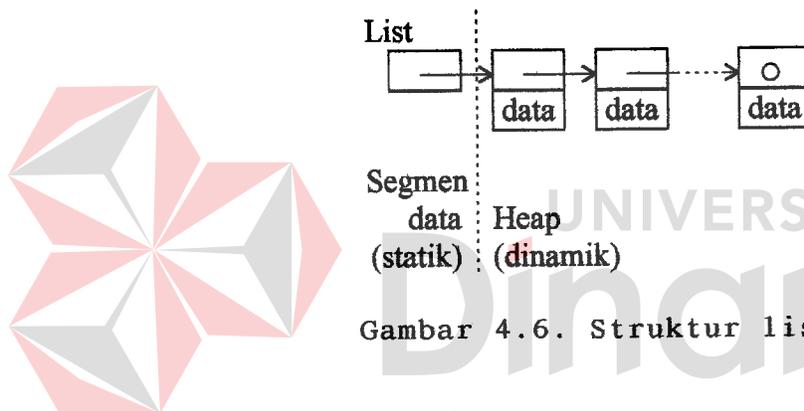
```

```

_t_p_lst_rlc = ^_t_r_lst_rlc;
_t_r_lst_rlc = record
  p_last_node : _t_p_lst_rlc;
  val : _t_r_val_rlc; { file }
  val_y : _t_r_val_y; { exclude file }
end;

```

Metode stack dipilih karena algoritma penambahan simpul data untuk stack (sisip di depan) lebih sederhana daripada algoritma penambahan simpul data dengan metode antrian; lagi pula tak banyak manfaatnya jika list dibentuk sebagai antrian.



Gambar 4.6. Struktur list

Untuk 'memegang' list digunakan variabel pointer, sebut saja `p_lst_rlc`.

Pointer untuk menunjuk ke simpul yang lain (field `p_last_node`) diletakkan di awal untuk menyederhanakan algoritma untuk menghapus simpul. Inisialisasi proses penghapus simpul:

```

p_node_prev := @p_lst_rlc;
p_node_seek := p_lst_rlc;

```

`p_node_prev` berfungsi sebagai pointer simpul sebelum simpul yang akan dihapus, `p_node_seek` berfungsi sebagai

pointer pencari simpul yang akan dihapus. Jika simpul yang dimaksud sudah ditemukan, hapus.

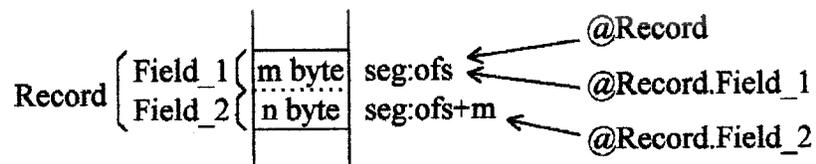
```
if p_node_seek = p_node_want_to_del then begin
  p_node_prev^.p_last_node := p_node_seek^.p_last_node;
  dispose(p_node_seek);
end;
```

Anggap saja saat ini simpul yang akan dihapus sudah ditemukan, dan simpul yang akan dihapus adalah simpul yang pertama. Program di atas mungkin terlihat melakukan kesalahan karena saat ini `p_node_prev = @p_lst_rlc`. Tetapi perlu diingat, Pascal mengakses field pertama tanpa penambahan offset (`offset+0`) seperti halnya variabel-variabel biasa (bukan record). Jadi khusus untuk saat ini, pernyataan:

```
p_node_prev^.p_last_node := ...
```

sama dengan:

```
p_lst_rlc := ...
```



Gambar 4.7. Struktur variabel record pada Pascal

Jadi dengan meletakkan `p_last_node` sebagai field yang pertama, tidak perlu ada pemeriksaan apakah simpul yang akan dihapus adalah simpul yang ditunjuk langsung oleh `p_lst_rlc` atau bukan. Program yang terbentuk memang menjadi lebih kecil; tetapi bagi yang tidak mengetahui

hal ini akan berbahaya jika field ini dipindah, sebab diperkuliahan biasanya field semacam `p_last_node` ini diletakkan di urutan terakhir (field terakhir).

4.2.2. Pengukuran

Bagian pengukuran, secara berurut, mempunyai tugas: mengidentifikasi kabel, memeriksa kondisi rangkaian, melakukan perhitungan, dan menggambar grafik tegangan.

A. Identifikasi kabel

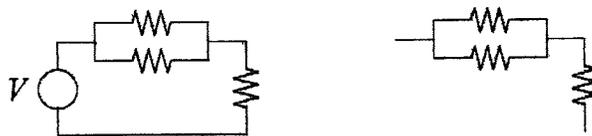
Dalam melakukan pengukuran, ujung *probe* ditempelkan pada bagian logam dari kaki elemen atau kabel yang berhubungan langsung dengan kaki elemen tersebut. Untuk keperluan ini, kabel-kabel diidentifikasi dengan memberi nomor urut, sehingga program dapat mengetahui kabel mana yang berhubungan dengan *probe*.

Untuk memudahkan, penomoran dilakukan terhadap setiap sel kabel. Untuk ini diperlukan variabel array dengan dimensi yang sama dengan dimensi layar skema yang berfungsi menyimpan nomor kabel pada posisi kabel yang bersangkutan; atau menggunakan pointer untuk menyimpan data nomor kabel dan koordinat-koordinat kabel yang berhubungan. Pada beberapa kasus cara pointer ini memang membutuhkan memori yang lebih sedikit, tetapi akan terjadi sebaliknya jika dalam rangkaian terdapat banyak kabel, apalagi jika ada pemakai 'iseng' yang membuat kabel yang berputar-putar padahal ada rute yang lebih

dekat. Jadi dalam tugas akhir ini dipakai variabel array, sebut saja variabel `cell_id`.

Proses penomoran dimulai dari sumber tegangan jika ada sumber tegangan, atau dari simpul elemen yang pertama jika ada elemen dalam rangkaian dan tidak ada sumber tegangan. Sebelum proses `cell_id` harus sudah di-reset (diberi nomor 0). Jika terdapat sumber tegangan maka proses penomoran adalah seperti berikut:

1. Nomori kabel-kabel yang berhubungan dengan sumber tegangan.
2. Cari kabel yang berhubungan dengan elemen yang hanya mempunyai sebuah kaki yang tidak bernomor; jika ada: beri nomor kabel kaki yang belum bernomor, ulangi langkah (2).
3. Cari kabel yang berhubungan dengan elemen yang tidak mempunyai nomor kaki sama sekali; jika ada: beri nomor kabel salah satu kaki elemen, lalu kembali ke langkah (2).



Gambar 4.8. Rangkaian dengan dan tanpa sumber tegangan

Jika tanpa sumber tegangan, prosesnya seperti berikut:

1. Cari kabel yang berhubungan dengan elemen yang tidak mempunyai nomor kaki sama sekali; jika ada: beri

nomor kabel salah satu kaki elemen; jika tidak ada: maka proses ini sudah selesai.

2. Cari kabel yang berhubungan dengan elemen yang hanya mempunyai sebuah kaki yang belum diberi nomor; jika ada: beri nomor kabel kaki yang belum bernomor, ulangi langkah (2); jika tidak ada: ke langkah (1).

Untuk penomoran kabel, perlu diperhatikan jika pelacakan sampai di pertigaan atau di perempatan, urutan prioritas yang diberikan adalah: ke atas, ke kiri, ke kanan, dan ke bawah. Proses lengkapnya adalah sebagai berikut (sel kabel saat ini belum bernomor, dan nomor saat ini adalah nomor yang baru):

1. Beri nomor sel kabel pada posisi saat ini.
2. Hitung sisa jalur kabel dari sel saat ini yang belum bernomor.
3. Jika ada sisa jalur: simpan posisi sel yang baru (berdasarkan prioritas), lanjutkan ke langkah (5); jika tidak ada sisa jalur: *pop* dari stack posisi sel kabel.
4. Jika stack dapat di *pop*: dengan posisi sel hasil *pop* stack, kembali ke langkah (2); jika tidak dapat: maka proses ini sudah selesai (untuk ruas-ruas kabel yang saling berhubungan).
5. Jika sisa jalur lebih dari satu: *push* posisi sel kabel ke stack.

6. Set posisi sel ke posisi sel baru, kembali ke (1).

B. Persiapan nilai-nilai untuk perhitungan

Pemeriksaan sumber tegangan dilakukan setelah penomoran kabel, yaitu pemeriksaan apakah sumber tegangan dihubung singkatkan. Hal ini dapat diketahui dari nomor kabel kaki sumber tegangan, jika mempunyai nomor yang sama berarti sumber tegangan dalam keadaan hubungan singkat.

Untuk mempermudah proses perhitungan selanjutnya, dilakukan proses konversi dari nilai elemen ke nilai admitansi-nya untuk keperluan perhitungan tegangan simpul, juga penomoran kaki elemen-elemen untuk keperluan perhitungan admitansi antara dua simpul.

C. Perhitungan tegangan-simpul

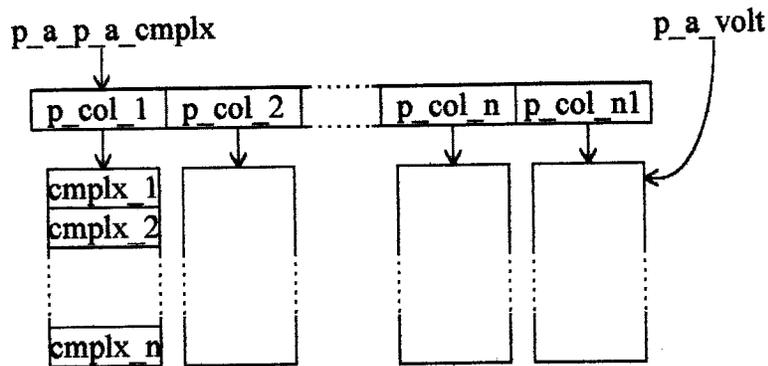
Jika rangkaian tidak mempunyai sumber tegangan atau sumber tegangan yang ada dihubung singkatkan, maka tidak perlu dilakukan perhitungan ini.

Perhitungan tegangan simpul, menggunakan matriks admitansi yang mempunyai struktur seperti yang diperlihatkan pada Gambar 4.9.

Setelah perhitungan selesai, maka array yang ditunjuk oleh p_a_volt , berisi nilai tegangan pada simpul 1 sampai dengan simpul n .

Untuk dapat melanjutkan ke proses perhitungan admitansi, matriks harus dibuang (kecuali kolom n_1 yang

akan digunakan untuk menggambar grafik tegangan), sebab perhitungan admitansi memerlukan memori juga untuk menampung list elemen.



Gambar 4.9. Struktur matriks tegangan simpul

Matriks ini akan diisi dengan nilai admitansi seperti yang dijelaskan dalam teori.

Dalam matriks, nilai sumber tegangan untuk kaki yang mengeluarkan arus dianggap $1+j0$. Hal ini dilakukan untuk mempermudah saat penggambaran grafik tegangan.

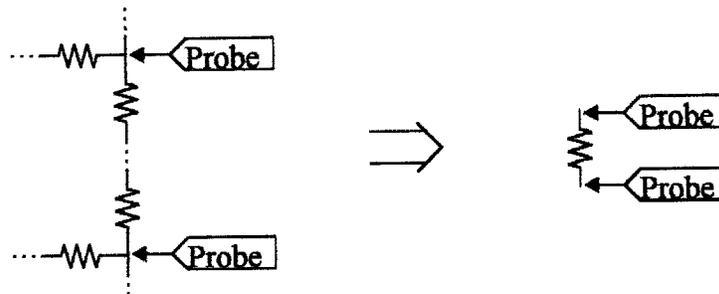
D. Perhitungan admitansi antara dua simpul

Proses perhitungan admitansi ini dilakukan dengan mereduksi rangkaian sampai tidak dapat dilakukan lagi.

Proses diawali dengan memeriksa apakah nomor simpul saat *probing* valid atau tidak. Jika kedua *probe* ada di kabel yang sama maka tidak perlu dihitung; nilai admitansinya adalah tak terhingga (impedansi nol).

Proses selanjutnya adalah meng-*copy* list elemen-elemen yang tidak dihubung-singkatkan; tetapi tidak semua

field, yang diperlukan hanyalah informasi mengenai nomor kaki elemen, dan nilai admitansi elemen, dan tentu saja field pointer untuk membentuk list.



Gambar 4.10. Reduksi rangkaian untuk memperoleh nilai admitansi antara dua simpul

Proses paralel, langsung menggunakan data dari list elemen, yaitu mengambil simpul-simpul yang mempunyai nomor kaki yang sama, kemudian dilakukan penjumlahan. Hasil dari proses paralel adalah sebuah simpul saja; jadi simpul-simpul yang tak terpakai dibuang. Proses paralel ini diulang-ulang sampai tak ada elemen yang mempunyai formasi paralel.

Menilik teori yang digunakan, untuk mempermudah proses *star* ke *mesh*, dibuat sebuah array dengan indeks nomor kaki elemen. Array ini berisi nilai banyaknya elemen yang kakinya mempunyai nomor yang bersangkutan, sebut saja variabel array ini `a_pin_id_cnt`. Variabel `a_pin_id_cnt` digunakan untuk memilih nomor kabel yang paling sedikit elemennya tetapi lebih dari satu. Dari informasi ini diambil simpul-simpul yang mempunyai nomor

kaki yang dimaksud, kemudian dilakukan perhitungan. Proses *star* ke *mesh* ini dilakukan sekali saja, lalu kembali ke proses paralel.

Selama proses reduksi rangkaian, adakalanya ditemukan dalam `a_pin_id_cnt` yang bernilai 1; ini berarti ada elemen dengan kaki yang terbuka (tidak berhubungan dengan elemen manapun). Jika hal ini terjadi, elemen tersebut dibuang dari list jika tidak berpengaruh terhadap nilai akhir nanti.

Jika kedua proses di atas (paralel dan *star* ke *mesh*) sudah tidak dapat dilakukan lagi, maka berarti proses reduksi rangkaian sudah maksimal.

Setelah proses reduksi selesai, nilai admitansi antara dua simpul hanya mempunyai dua kemungkinan:

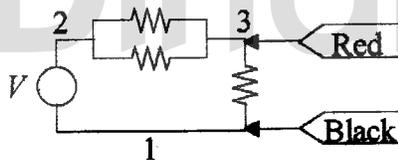
1. Jika elemen dengan nomor kaki yang sama dengan nomor simpul yang dimaksud ada dalam list, maka admitansinya adalah nilai elemen ini.
 2. Jika elemen dengan nomor kaki yang sama dengan nomor simpul yang dimaksud tidak ada dalam list, maka admitansinya adalah nol (impedansi tak terhingga), artinya bahwa antara simpul yang dimaksud, tidak ada hubungan sama sekali.
-

E. Penggambaran grafik tegangan

Jika perhitungan tegangan simpul dilakukan, maka akan digambar grafik tegangannya; jika tidak, maka langsung digambar sebuah garis lurus.

Grafik tegangan mempunyai bentuk sinusiodal. Grafik ini diperoleh dari proyeksi garis bilangan kompleks terhadap sumbu Y atau sumbu imajiner.

Fungsi grafik diperoleh dari selisih tegangan antara dua simpul dari matriks, karena hasil perhitungan yang dilakukan adalah tegangan suatu simpul terhadap sebuah titik acuan. Misalnya *probe-merah* ada di simpul-3 dan *probe-hitam* ada di simpul-1, maka tegangan pada ujung-ujung *probe* adalah tegangan di simpul-3 dikurangi tegangan di simpul-1 ($V_{\text{probe}} = V_3 - V_1$).



Gambar 4.11. Contoh *probing* pada rangkaian

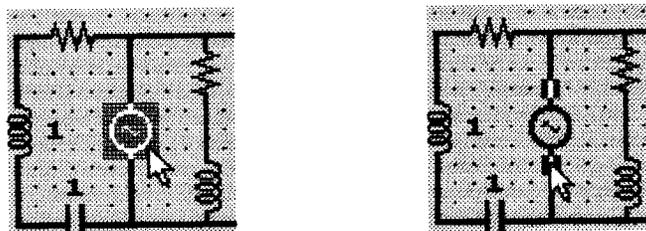
Hasil yang diperoleh belumlah merupakan hasil yang siap pakai, karena hasil ini adalah tegangan-relatif saat $t = 0$ detik. Hasil ini harus dikalikan dengan sumber tegangan asli dengan t yang berjalan, karena yang digambar adalah tegangan dalam suatu interval waktu.

Nilai-nilai pada sumber tegangan asli adalah nilai-nilai dalam sistem koordinat polar. Nilai-nilai ini harus dikonversikan dahulu ke sistem Cartesien sehingga dapat dikalikan dengan tegangan-relatif yang berupa bilangan kompleks. Dari hasil akhir ini, diambil nilai imajinerinya yang merupakan nilai tegangan pada saat t yang dimaksud.

Jika saat perhitungan matriks, dipakai nilai tegangan sumber yang sebenarnya, nilai selisih tegangan yang terjadi memang merupakan nilai yang sebenarnya; tetapi nilai dalam bentuk bilangan kompleks ini lebih panjang prosesnya untuk menjadi nilai akhir yang siap untuk digambar, karena lebih mudah memutar garis dalam sistem polar daripada garis dalam sistem Cartesien.

F. Penyidikan (*probing*) pada rangkaian

Untuk mempermudah pemakaian, *probing* pada perangkat lunak ini mempunyai dua cara seperti yang diperlihatkan dalam gambar berikut:



Gambar 4.12. Dua cara *probing*

Cara pertama, dengan meng-*click* elemen yang ada. Pada cara yang pertama ini, diasumsikan *probe*-merah ada di kaki atas atau kaki kanan dari elemen, dan *probe*-hitam ada di kaki bawah atau kaki kiri dari elemen.

Cara kedua, dengan men-*drag* mouse pada kaki elemen atau kabel yang ada dalam rangkaian. *Probe*-merah ada di posisi saat tombol mouse ditekan (ditandai dengan warna merah pada *background* kabel), dan *probe*-hitam ada di posisi saat tombol mouse dilepas (ditandai dengan warna biru pada *background* kabel).

4.3. Analisa Hasil

Secara keseluruhan, hasil penggambaran skema dan hasil perhitungan yang dilakukan, sesuai dengan yang diharapkan.

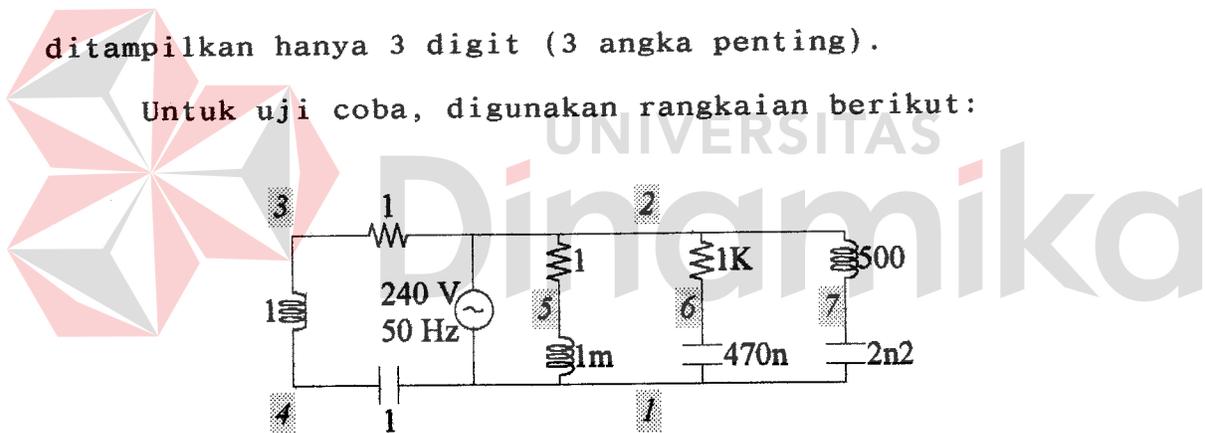
Gambar skema yang dibuat tidak terlalu banyak bisa diharapkan hasil yang sempurna.

Beberapa pemakai mengalami sedikit kebingungan dengan perilaku antarmuka yang berbeda dengan Ms Windows, sehingga untuk dapat mengoperasikan dengan baik perlu latihan untuk memahami perilaku mouse dan tombol-tekan yang ada, karena respon dari tombol-tekan dilaksanakan saat tombol tersebut ditekan, bukan setelah tombol dilepas (seperti yang diterapkan dalam Ms Windows). Hal ini berdasarkan kenyataan yang ada bahwa hampir semua

perangkat elektronik memberikan respon saat suatu tombol ditekan, bukan setelah tombol tersebut dilepas.

Hasil perhitungan tegangan simpul (perhitungan matriks) dan perhitungan untuk mencari admitansi, dibandingkan dengan menggunakan MathCad versi 2.5. Pada beberapa perhitungan memang terjadi sedikit perbedaan yaitu rata-rata mulai pada digit ke-10. Meskipun demikian, hasil yang diperoleh dapat dikatakan cukup memadai, mengingat perangkat lunak ini hanya menggunakan *fast floating-point* dari Turbo Pascal yang mempunyai kapasitas significant 11 sampai 12 digit; lagipula yang ditampilkan hanya 3 digit (3 angka penting).

Untuk uji coba, digunakan rangkaian berikut:



Gambar 4.13. Rangkaian uji coba

Perhitungan yang dilakukan adalah perhitungan $|V|$ yang diukur terhadap *ground* (simpul 1).

Hasil perhitungan $|V|$ dari perangkat lunak ini (nilai internal, bukan nilai yang ditampilkan) dan hasil perhitungan $|V|$ dari MathCad diperlihatkan pada tabel di halaman berikut.

Tabel 4.1. Perbandingan perhitungan $|V|$

Simpul	Lab. (Internal)	MathCad
2	240.0	240
3	239.99878413	239.9987841303972
4	0.0024317207266	0.002431720726564
5	71.932032861	71.93203286140701
6	237.42578186	237.4257818566374
7	269.22902351	269.2290235085284

Berikut ini adalah hasil perhitungan Z (diukur terhadap simpul 1) dari perangkat lunak dan dari MathCad:

Tabel 4.2. Perbandingan perhitungan Z

Simpul	Lab. (Internal)	MathCad
2	0.99805662993	0.998056629928785
	0.31686893557	0.316868935571399
3	1.9959989305	1.995998930464673
	0.32639616421	0.32639616420623
4	2.049130484E-10	2.049130483906788E-10
	-0.0031831310802	-0.003183131080191
5	4.0088194378E-6	4.008819437756714E-6
	0.31385975203	0.313859752032443
6	979.68793064	979.687930634022
	-144.48979464	-144.4897946435043
7	1.2559618191	1.255961863604137
	176210.38258	176210.3825809569

(Baris pertama adalah bagian riil,
baris kedua adalah bagian imajiner)

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

1. Laboratorium elektronika virtual ini dapat membantu para pemakai dalam mendapatkan sebuah laboratorium elektronika untuk melakukan suatu percobaan, karena dalam laboratorium virtual ini sudah terdapat fasilitas untuk membuat rangkaian-elektrik virtual (dengan elemen-elemen: sumber tegangan sinusoidal, resistor, induktor, dan kapasitor), dan oscilloscope virtual; selain itu, biaya untuk membangun sebuah laboratorium elektronika tidaklah sedikit.
2. Dengan adanya laboratorium elektronika virtual ini, pemakai dapat melakukan percobaan berulang-ulang, tanpa ada batasan waktu. Selain itu, juga lebih murah, dan lebih aman.
3. Dengan adanya laboratorium elektronika virtual ini, pemakai akan lebih mudah dalam memahami materi yang dipelajarinya karena disajikan secara interaktif.

5.2. Saran

Adapun saran-saran yang sekiranya berguna untuk pengembangan perangkat lunak ini adalah:

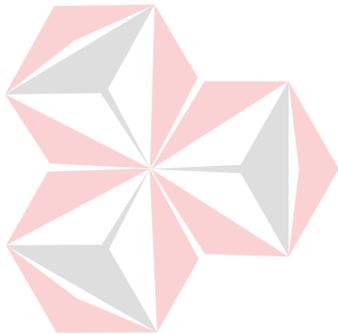
1. Perlu adanya fasilitas *undo* atau sejenisnya dalam editor penggambar rangkaian; juga fasilitas *online-*

help untuk menambah kejelasan pemakaian perangkat lunak ini.

2. Perangkat lunak ini mungkin dapat dikembangkan agar mencakup juga rangkaian-rangkaian *non-planar*, dan penambahan elemen-elemen elektronika yang lain sehingga laboratorium ini menjadi lebih lengkap.
3. Karena kapasitas memori, akan lebih baik jika perangkat ini dikembangkan dalam sistem yang mempunyai manajemen virtual memori sehingga mengurangi beban yang berkenaan dengan kapasitas memori, dan ini bisa digunakan untuk menambah area editor skema. Disamping itu dengan adanya kebebasan memori, akurasi perhitungan dapat ditingkatkan dengan memakai tipe bilangan real yang lebih tinggi misalnya tipe *double-precision* atau *extended-precision*.

DAFTAR PUSTAKA

- Anonim, 1988, Turbo Pascal - Reference Guide, Borland International, Inc., U.S.A.
- Edminister, J.A., 1972, Electric Circuits, McGraw-Hill, New York.
- Hyat dan Kemerly, 1988, Rangkaian Listrik jilid I, edisi IV, Erlangga, Jakarta.
- Mallet, M., 1991, Graphics Programming with Microsoft C 6, M&T Publishing, Inc., California.
- Scout, R.E., 1960, Linear Circuits, Addison-Wesley Publishing, Massachusetts.
- Tischer, M., 1992, PC Intern, Abacus, Grand Rapids.



UNIVERSITAS
Dinamika