

# PEMOGRAMAN DASAR DALAM BIDANG FISIKA DENGAN TURBO PASCAL



PERPUSTAKAAN IKIP PADANG  
Jl. Pemuda No. 10 Padang

PERPUSTAKAAN IKIP PADANG	
DITERIMA TGL. :	15 JUN 1998
SUMBER / HARGA :	H /
Oleh KOLEKSI :	K
Amali Putra Akman	NO. INVENTARIS : 542/k/98-P2 (1)
	KLASIFIKASI : 530.0285 put P 2

**JURUSAN PENDIDIKAN FISIKA  
FAKULTAS PENDIDIKAN MATEMATIKA DAN IPA  
INSTITUT KEGURUAN DAN ILMU PENDIDIKAN PADANG  
1998**

## KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Ilahi, karena atas rahmat dan izin Nya jua lah tulisan ini dapat diselesaikan dengan judul *“Pemograman Dasar Dalam Bidang Fisika dengan Turbo Pascal “*

Tulisan ini dimulai dengan peranan komputer dalam pengajaran fisika, pengenalan sistem komputer, bahasa pemograman Turbo Pascal, dan langkah-langkah membuat membuat grafik sederhana, serta diberikan contoh-contoh program sederhana.

Pada kesempatan ini penulis menyampaikan rasa terima kasih dan penghargaan yang setinggi-tingginya kepada para senior dan teman sejawat yang ikut serta membantu untuk terselesaikannya tulisan ini. semoga segala amal baik beliau diterima Allah S.W.T dengan pahala yang setimpal.

Akhirnya penulis merasakan bahwa tulisan ini masih jauh dari kesempurnaan, oleh sebab itu kritik dan saran yang bersifat membangun sangat penulis harapkan dari pembaca semua, untuk kesempurnaan di masa mendatang.

Padang, Februari 1998  
Penulis,

# DAFTAR ISI

	HAL
KATA PENGANTAR .....	i
DAFTAR ISI .....	ii
BAB I. PENDAHULUAN .....	1
A. Peranan Komputar Dalam Pelajaran Fisika .....	1
B. Mengenal System Komputer .....	4
C. System Operasi (DOS).....	8
D. Bahasa Pemograman .....	14
BAB II. BAHASA PEMOGRAMAN TURBO PASCAL .....	15
A. Elemen-elemen Dasar Turbo Pacal .....	16
B. Persiapan Membuat Program Sederhana .....	17
C. Struktur Program Turbo Pascal .....	22
D. Input dan Output .....	23
E. Pendeklarasian .....	28
F. Struktur Kontrol .....	31
G. Algoritma .....	46
H. Pembuatan Diagram Alir (Flow Chart).....	47
I. Program Pembantu (Sub Program) .....	51
J. Data Terstruktur .....	53
BAB III. GRAFIK .....	56
A. Mode Teks dan Mode Grafik .....	56
B. System Koordinat Layar Monitor .....	57
C. Membuat Visualisasi Sedrerhana .....	59
D. Transformasi Koordinat .....	60
E. Membagi Layar Monitor Menjadi Beberapa Jendela .....	63
F. Sumbu Koordinat .....	65
G. Skala .....	67
DAFTAR PUSTAKA .....	72
LAMPIRAN-LAMPIRAN .....	73

## BAB I. PENDAHULUAN

### A. PERANAN KOMPUTER DALAM PELAJARAN FISIKA

Fisika sebagai bagian dari sains, tidak akan maju dan berkembang jika hanya berdasarkan dogmatis belaka. Dasar dari fisika adalah eksperimen dan percobaan dalam menyelidiki berbagai gejala alam sehingga muncul berbagai konsep, hukum dan teori yang menambah khasanah keilmuan fisika itu sendiri. Selanjutnya konsep dan teori tersebut diaplikasikan dalam berbagai teknologi yang dimanfaatkan untuk keperluan hidup umat manusia.

Konsekwensinya, dalam setiap pelajaran fisika harus dirancang adanya praktikum atau percobaan mulai dari yang sederhana sampai pada penelitian pengembangan fisika tersebut. Ada kalanya praktikum atau percobaan sukar dilakukan karena berbagai kendala, seperti ; kekurangan alat peraga, bahaya yang ditimbulkannya, atau keterbatasan berbagai pengetahuan, maka diciptakanlah berbagai pemodelan, diantaranya pemodelan matematis, simulasi dan animasi dengan menggunakan perangkat lunak komputer, seperti yang dikemukakan Zarlis dan Syahyar (1993 : 2) yang menyatakan bahwa :

" Untuk mengatasi beberapa kendala yang terjadi dalam menjelaskan atau menganalisis gejala fisis yang dikerjakan secara eksperimen dapat digunakan perangkat lunak komputer".

Dengan demikian keberadaan penelaahan fisika secara eksperimen, pengkajian secara teoritis, analisis matematik, dan penyelesaian berbagai persoalan fisis dengan menggunakan perangkat lunak komputer merupakan saling mendukung dalam

pengembangan fisika.

Dalam pengajaran fisika disekolah, pemakaian komputer dapat dianjurkan untuk mengatasi berbagai kendala yang dihadapi melalui eksperimen dalam menjelaskan atau menganalisis gejala fisis seperti :

1. Gejala yang diamati terlalu cepat untuk diukur dan diamati visualisasinya.
2. Ukuran gejala (benda) yang diamati terlalu kecil untuk ukur
2. Gejala yang diteliti cenderung berbahaya
4. Peralatan yang diperlukan untuk analisis gejala sangat mahal dan berbahaya
5. Data hasil eksperimen cukup besar dan sukar untuk dianalisis secara manual
6. dan lain-lain.

Kendala-kendala yang dihadapi tersebut, mengakibatkan karakteristik suatu gejala fisis tidak dapat terungkap secara tuntas sehingga akan mengganggu terhadap pemahaman dan pengembangan fisika itu sendiri.

Dalam mempelajari fisika, pembahasan dapat dilakukan dengan analisis secara makroskopik atau secara mikroskopik. Sehubungan dengan ini D. Halliday dan Resnick (1987:694) mengemukakan bahwa :

"Sifat-sifat makroskopik suatu zat, benda atau sistem adalah sifat-sifat yang langsung dapat diasosiasikan dengan tanggapan indra. dan sifat-sifat mikroskopik tidaklah langsung dapat diasosiasikan dengan tanggapan indra"

sifat-sifat mikroskopik yang masih berada pada taraf abstrak ini dapat dianalisis menjadi bentuk yang nyata dengan simulasi melalui komputer. Penerapan perangkat lunak

komputer dalam fisika dapat terlihat pada pemecahan masalah masalah analitik yang kompleks, numerikal, simulasi, dan animasi berbagai gejala secara interaktif. Oleh karena ini dalam menerapkan perangkat lunak komputer dalam fisika paling tidak merupakan gabungan dari tiga disiplin ilmu, yaitu : ilmu fisika, pemrograman komputer dan analisis numerik.

Secara makro, peranan komputer dalam aplikasi fisika seperti dalam berbagai bidang seperti geofisika, fisika reaktor, fisika semikonduktor dsb. telah mencakup pengembangan program-program aplikasi bidang yang bersangkutan. Selain hal-hal tersebut diatas, dalam bidang pendidikan kehadiran komputer dapat membuat suatu konsep lebih menarik sehingga dapat menambah motivasi untuk mempelajari dan menguasainya.

Komputer sebagai salah satu media interaktif dalam bidang pendidikan, dapat membuat peserta didik bersifat aktif dan reaktif, sehingga memungkinkan siswa selalu berinisiatif dalam kegiatan belajar mengajar. Saat ini sistem komputer yang canggih disertai dengan CDROM atau Video Cassette Recorder yang berisi gambar hidup, media suara dll, sehingga sistem komputer ini semakin ideal untuk dijadikan sebagai multimedia dalam kegiatan belajar mengajar.

Dengan bahasa pemrograman seperti BASIC, FORTRAN, COBOL, PASCAL, dll, memungkinkan kita untuk membuat sendiri program-program khusus untuk pendidikan. Selain penguasaan salah satu bahasa pemrograman tersebut, penguasaan terhadap

berbagai teknik numerik, grafik, dan animasi juga diperlukan. Dengan pemrograman sederhana dapat dibuat berbagai tampilan seperti grafik, simulasi dan animasi yang diharapkan dapat memperjelas berbagai konsep. Untuk menciptakan program pengajaran yang menarik dengan menggunakan media komputer ini diperlukan latihan dalam menggunakan bahasa pemrograman.

## B. MENGENAL SYSTEM KOMPUTER

Komputer dapat diartikan sebagai suatu sistem elektronik yang dapat digunakan untuk memanipulasi data secara cepat dan tepat dan dapat secara otomatis menerima, menyimpan, memproses data input serta menghasilkan data output berdasarkan perintah program yang diterima. Beberapa kemampuan komputer adalah :

- fast reliable and precise computations
- non-analytic solution of equation
- simulations of processes and experiments
- graphics solution of problems
- interactives programs
- direct interfacing with measuring instruments for treatment of data
- assignments and record that are easy to retrieve

Komputer mikro tersusun dari bagian-bagian yang disebut perangkat keras (hardware) dan perangkat lunak (software)

1. Perangkat Keras (hardware) : yaitu bagian sistem komputer yang menjalankan perangkat lunak (software). Perangkat keras ini paling tidak terdiri dari :
  - a. Unit sistem
  - b. Unit masukan
  - c. Unit keluaran

2. Perangkat Lunak (software): yaitu program yang menterjemahkan suatu instruksi yang diberikan oleh si pemrogram dari bahasa pemrograman ke bahasa mesin yang dimengerti oleh komputer. Ada dua bagian sistem operasi yang ditangani oleh perangkat lunak komputer ini, yaitu :

- a. Operasi dasar ; yang dilakukan oleh sistem operasi (operating system) DOS ataupun MS-DOS. Saat ini DOS yang tersedia sudah banyak, yaitu versi 3,00 keatas.
- b. Operasi lanjutan ; yaitu berbagai bahasa pemrograman seperti Lotus, DBase III+, Basic, QuickBasic, Turbo Pascal, Fortran. dll. yang dirancang penggunaannya untuk berbagai tujuan dan keperluan.

Unit system paling tidak berisi bagian-bagian yang disebut

1. CPU (Central Processing Unit); merupakan bagian dari Unit system yang wujudnya berupa IC atau Chip dan berfungsi sebagai bagian yang memproses informasi.
2. Processor; merupakan pusat pemroses yang terdiri dari :
  - Unit penghitung (Arithmetic Unit) ; yaitu bagian yang melakukan proses perhitungan
  - Unit Logika (Logic Unit) ; yaitu bagian yang melakukan proses perbandingan
  - Unit kendali (Controll Unit) ; yaitu bagian yang mengatur aliran pelaksanaan program dalam system komputer.

Untuk komputer mikro, mikroprocessor yang ada tipenya bermacam-macam, diantaranya yang sudah dikenal adalah mikroprocessor tipe ; 8088, 80286, 80386SX, 80386DX, 80486SX, 80486DX66, 80486DX100, pentium dsb.

3. Unit memori (Memory) ; ada 2 macam, yaitu unit memori dalam dan unit memori luar



a. Unit memori dalam terdiri dari :

- ROM (Read Only Memory) yang berisi program atau instruksi yang digunakan untuk mengendalikan pengoperasian komputer dan menafsirkan perintah yang dimasukkan pada keyboard. Informasi dalam ROM bersifat permanen (tidak dapat diubah oleh sipemakai)
- RAM (Random Access Memory) merupakan tempat menyimpan program yang dimuat dalam komputer dan data yang dimasukkan sipemakai. Processor dapat membaca informasi atau perintah dari RAM dan melaksanakannya serta menulis informasi pada RAM. Si pemakai komputer dapat membuat, mengubah dan menghapus informasi didalam RAM. Kapasitas RAM diukur dalam satuan byte. Kapasitas 1 byte adalah kapasitas penyimpanan untuk menyimpan 1 karakter.

1 Kilo byte (KB) = 1024 Byte (B)

1 Mega byte (MB) = 1024 KB

1 Giga byte (GB) = 1024 MB

b. Unit Memori Luar, yaitu media tempat menyimpan data yaitu berupa disket (floppy disk) atau hard disk.

4. Disk Drive (Penggerak disket) ; yaitu bagian dari unit system yang berfungsi untuk menggerakkan disket disaat proses pemindahan isi memori ke disket atau pemindahan isi disket ke memori berlangsung. Umumnya komputer dilengkapi dengan 2 buah disk drive, yaitu bagian atas disebut drive A dan sebelah bawah disebut drive B. Sesuai dengan tipe disket yang ada maka macam-macam ukuran dan kapasitas penggerak disket ini adalah sbb :

No	Kapasitas	Ukuran
1.	360 KB	5 1/4 inc
2.	1,2 MB	5 1/4 inc
3.	720 KB	3 1/2 inc
4.	1,4 MB	3 1/2 inc

Disket (floppy disk) adalah media magnetik berbentuk cakram yang lembut tempat menyimpan secara permanen isi memori komputer. Tipe disket yang ada dipasaran diantaranya adalah :

1. 360 KB (5 1/4" Double Density)
2. 1,2 MB (5 1/4" High Density)
3. 720 KB (3 1/2" Double Density)
4. 1,4 MB (3 1/2" High Density)

Dalam memilih disket yang hendak dipakai supaya disesuaikan dengan ukuran dan kapasitas disk drive yang tersedia pada komputer.

Unit Masukan ; yaitu berupa papan ketik (keyboard), merupakan sarana untuk berkomunikasi dengan komputer. Karena melalui keyboard dapat dimasukkan perintah atau data. Jumlah tombol keyboard bermacam-macam, ada yang berjumlah 83, 84, 101. atau 102. Tombol-tombol keyboard ini secara garis besarnya dibedakan atas :

- a. Tombol fungsi (terletak dibagian kiri atau atas papan ketik dengan tanda F1 s.d F12 ). Fungsi tombol ini tergantung pada program yang sedang dijalankan.
- b. Tombol mesin ketik (terletak dibagian tengah), digunakan untuk menulis huruf abjad , angka, dan beberapa tanda tertentu.
- c. Tombol numerik (numeric keypad), terletak dibagian kanan papan ketik, digunakan jika pemakai sering menuliskan angka-angka. Tombol ini akan berfungsi jika diaktifkan, yaitu jika menghidupkan tombol Num-Lock.
- d. Tombol panah (arrow).

Selain tombol-tombol yang disebutkan diatas, peralatan

masukkan dapat juga berupa ; mouse, joystick, lightpen, touch screen, scanner dsb.

Unit Keluaran ; dapat berupa monitor (layar peraga) dan mikmikrophon untuk tampilan sementara, atau printer untuk menghasilkan tampilan permanen (hasil cetak). monitor yaitu berupa perangkat keras komputer yang mirip TV berfungsi melayani proses interaktif antara manusia dan komputer. Artinya instruksi yang diberikan pada komputer akan diproses oleh mikprocessor dan sekaligus diperagakan pada layar monitor. Jenis-jenis monitor yang telah dikenal diantaranya adalah :

- monochrome
- CGA (Color Graphics Adapter)
- MDA (Monochrome Display Adapter)
- HGA (Hercules Graphic Adapter)
- EGA (Enhanced Graphics Adapter)
- MCGA (Multi Color Graphics Adapter)
- VGA (Video Graphics Array)
- SVGA (Super VGA)

Jenis printer yang digunakan dapat dikelompokkan menjadi :

- Printer daisy wheel
- Printer dot matrix
- Printer inkjet
- Printer termal
- Printer laser

Berdasarkan kemampuan kerjanya komputer dapat dibedakan atas :

- Super komputer
- Mainframe
- Mini Computer
- Micro Computer

### C. SISTEM OPERASI (DOS)

Perangkat keras komputer saja tidak dapat digunakan, baru setelah sistem operasi (DOS) diberikan kepada komputer,

maka komputer baru dapat dipergunakan untuk berbagai tujuan dan keperluan.

Sistem operasi adalah kumpulan program yang berfungsi menyiapkan perangkat keras untuk berbagai keperluan, penghubung antara perangkat keras dan perangkat lunak lainnya serta sebagai penunjang untuk penggunaan perangkat lunak lainnya.

Sistem operasi DOS atau MS-DOS ada beberapa versi mulai dari versi 2 keatas. Pada tulisan ini dipergunakan MS-DOS versi 5.0 yang dipakai untuk komputer berbasis mikroprosesor 80386 keatas. Bagian-bagian dari MS-DOS ini ada 4 bagian penting, yaitu :

1. Boot-Loader ; bertugas memuat file-file IO.SYS, MSDOS.SYS dan COMMAND.COM ke memori
2. IO.SYS ; file yang berisi program yang bertanggung-jawab membentuk sistem komunikasi antar perangkat keras komputer.
3. MSDOS.SYS ; file berisi program yang mengelola file dan memori
4. COMMAND.COM ; File yang bertugas menginterpretasikan dan melaksanakan perintah pemakai komputer terhadap komputer.

#### a. Persiapan Untuk Memakai Komputer

Untuk menggunakan komputer mikro dengan memakai sistem operasi DOS versi 5.0 ini, langkah-langkah yang dilakukan adalah :

1. Cocokkan tegangan jala-jala listrik di ruangan anda bekerja dengan tegangan listrik yang dibutuhkan oleh perangkat komputer yang digunakan, bila tidak cocok pakailah trafo untuk menaikkan atau menurunkan tegangan, dan bila tegangan listrik yang digunakan tidak stabil pakailah stabilizer.
2. Hubungkan kabel keyboard, monitor, printer, ke portnya masing-masing yang terdapat pada bagian belakang unit

sistem, serta kabel printer, monitor, dan unit sistem ke-sumber tegangan yang sudah melalui stabilizer tadi.

3. Anda masukkan disket DOS (MS-DOS) pada disk drive A, kemudian ON-kan saklar stabilizer, monitor, dan unit sistem, sehingga komputer melakukan uji diri sendiri (POST = Power On Self Test), yang maksudnya program yang tersimpan dalam BIOS pada komputer melakukan pengujian terhadap perangkat keras yang digunakan. Tunggu sejenak sampai pada layar monitor muncul tulisan :

```
Current date is Thus 07-28-1994
Enter new date :
```

4. Masukkan data tanggal hari ini seperti format di layar, kemudian tekan <ENTER>, sehingga pada layar muncul tulisan :

```
Curren time is 4:15:15:23
Enter new time :
```

5. Ketikkan data jam saat ini kemudian tekan <ENTER>sampai di layar monitor muncul prompt A>, yang menandakan komputer siap kerja dengan disk drive A yang aktif.

Dengan demikian berarti komputer sudah siap digunakan untuk berbagai maksud dan tujuan.

#### b. Perintah-Perintah Minimum DOS atau MS-DOS

DOS atau MS-DOS menyediakan banyak perintah yang dapat mewujudkan keinginan kita dalam menggunakan komputer untuk berbagai keperluan. Perintah minimum yang harus dikuasai agar kita tidak menemui kesulitan dalam menggunakan komputer dapat dibagi 2 bagian, yaitu : perintah internal minimum dan perintah eksternal minimum.

Perintah internal adalah perintah-perintah yang ada dalam file COMMAND.COM dan akan berada di memori sampai komputer dimatikan. Sedangkan perintah eksternal disimpan DOS dalam disket dengan extension .COM atau .EXE.

Perintah internal minimum tersebut adalah :

1. DIR : perintah untuk menampilkan daftar isi disket pada layar monitor. Perintah ini digunakan dengan syntax

DIR [drive:] [pathname] [nama\_file] [/P] [/W]

Contoh :

- a. DIR menampilkan isi disket pada drive aktif
- b. DIR B: menampilkan isi disket pada drive B
- c. DIR/W menampilkan file-file pada drive aktif secara horizontal
- d. DIR/P menampilkan file-file per halaman
- e. DIR \*.COM menampilkan file-file dengan type COM

2. COPY : perintah untuk menggandakan suatu file pada lokasi lain. Perintah ini digunakan dengan syntax :

Copy [drive:] [nama-file] [drive :] [nama-file]

Contoh :

- a. COPY COMMAND.COM B: membuat duplikat file COMMAND.COM yang ada pada disket drive aktif ke disket pada drive B
- b. COPY A:CONT.PAS B: membuat duplikat CONT.PAS yang ada pada disket drive A ke disket pada drive B
- c. COPY SAY.PAS AI.PAS membuat duplikat file SAY.PAS dengan nama AI.PAS pada disket yang sama.
- d. COPY A: \*.CHI B: membuat duplikat semua file yang mempunyai ekstensi CHI pada disket drive A ke disket pada drive B
- e. COPY A: \*.\* B: membuat duplikat semua file pada disket drive A ke disket drive B

4. DEL : perintah untuk menghapus suatu file. Perintah ini dilakukan dengan syntax :

DEL [drive :] <nama\_file>

Contoh :

- a. DEL REMOT.PAS menghapus file REMOT.PAS pada disket drive yang aktif

- b. DEL A:\*.PAS                    menghapus semua file dengan ekstensi PAS yang ada pada drive A
  - c. DEL B:\*. \*                    menghapus semua file yang ada pada drive B
5. REN    : perintah untuk mengubah nama suatu file. Perintah ini menggunakan syntax :
- REN [drive:] <nama\_file> <nama\_file\_baru>

Contoh :

- a. REN WS.EXE ROBOT.EXE        merubah nama WS.EXE menjadi ROBOT EXE.
- b. REN \*.CHI \*.TEX            mengubah semua file yang mempunyai ekstensi CHI menjadi file dengan ekstensi TEX

Perintah eksternal minimum adalah :

- 1. PRINT    : perintah untuk mencetak isi suatu file teks

Contoh :

- a. A> B: PRINT AQUA.BAT        mencetak file teks AQUA.BAT di A
- b. B> PRINT A: AQUA.BAT        mencetak file teks AQUA.BAT di A
- c. B> PRINT AQUA.BAT            mencetak file teks AQUA.BAT di B

- 2. FORMAT    : perintah untuk memformat disket sebelum dipergunakan.

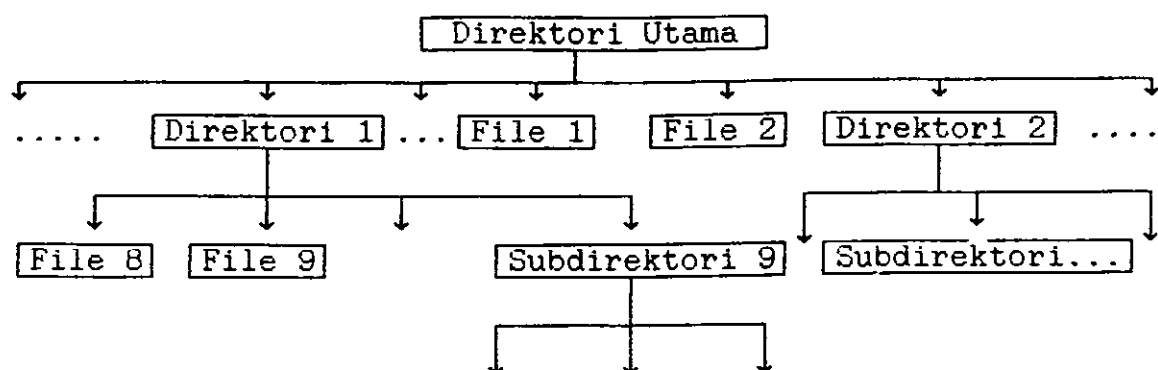
Contoh :

- a. A>FORMAT B:                    memformat disket yang ada di drive B
- A>FORMAT B :/s                    memformat disket yang ada di drive B dengan memasukkan system

### c. Direktori dan Sub Direktori

Satuan terkecil dari daftar isi disket disebut file. Kumpulan file dapat membentuk direktori. Adakalanya isi disket tersebut dibagi atas ; Direktori Utama, Direktori dan

Sub Direktori. Struktur yang demikian disebut Struktur Pohon seperti bagan pada Gambar 1.1 berikut ini :



Gambar 1.1. Struktur Daftar Isi Disket

Direktori utama akan dibuat pada saat disket diformat sedangkan direktori dan subdirektori dibuat setelah disket diformat. Cara pemberian nama direktori dan subdirektori sama dengan cara pemberian nama file, Untuk direktori utama tidak bernama, langsung diberi tanda "backslash" (\).

Misalkan disket di drive B: memiliki daftar isi :

a. Direktori utama berisi :

- File : UTAMA.COM, SISIPAN.EXE  
 -Direktori : IKIP, FPMIPA.TXT, EASTRA.TXT

b. Direktori IKIP berisi :

-File : JURIPA.DAT, JURMAT.DAT  
 -Direktori : SILABUS

c. Direktori SILABUS berisi file : ELKA.CHI, ELDIN.CHI

Untuk mengetahui isi direktori IKIP maka path yang harus diberikan adalah ; \IKIP, dan untuk mendapatkan file



ELKA.CHI maka path-nya ditulis sebagai ; \IKIP\SILABUS\  
ELKA.CHI ;

#### D. BAHASA PEMROGRAMAN

Dalam bidang komputasi, akan terjadi komunikasi antara drang dengan komputer. Orang mempunyai bahasanya sendiri yang tidak dimengerti oleh komputer, dan komputer juga mempunyai bahasanya sendiri pula yaitu bahasa mesin. Oleh sebab itu untuk berkomunikasi dengan komputer orang harus mempelajari bahasa mesin yang cukup rumit sehingga tidak semua orang yang mampu untuk mempelajarinya. Atas dasar ini para ahli berusaha menciptakan suatu bahasa yang mudah dipelajari orang dan dapat dimengerti komputer, yang disebut *bahasa pemrograman*. Dewasa ini telah banyak bahasa pemrograman yang diciptakan, seperti : Basic, Fortran, Prolog, Pascal dll. Masing-masing bahasa pemrograman diatas masih terdiri dari beberapa versi. Banyaknya bahasa pemrograman diciptakan, adalah untuk memenuhi berbagai keperluan dan permasalahan, masing-masing masalah tersebut akan dapat diselesaikan secara efisien dengan menggunakan bahasa pemrograman tertentu.

## BAB II. BAHASA PEMOGRAMAN TURBO PASCAL

Turbo Pascal, merupakan nama bahasa pemrograman yang dewasa ini banyak digunakan untuk keperluan memecahkan persoalan-persoalan fisika, matematika, dan dalam bidang teknik. Nama ini diambil dari nama seorang fisikawan terkenal Prancis yaitu Blaise Pascal (1623-1662). Bahasa pemrograman Turbo Pascal ini dirancang secara sistematis, terstruktur dan menggunakan editor mirip dengan wordstar, yaitu salah satu program pengolah kata yang cukup populer. Karena sifat dan keistimewaan Turbo Pascal yang demikian, maka bahasa pemrograman ini banyak dipakai dalam bidang pendidikan, karena sangat mudah dalam hal membuat suatu program atau saat mencari kesalahan.

- a. sistematis : maksudnya disusun sedemikian rupa langkah demi langkah yang secara keseluruhan merupakan satu sistem yang lengkap yaitu sebuah program.
- b. terstruktur : maksudnya terdiri dari bagian-bagian kecil yang disebut blok program. Tiap blok akan menyelesaikan suatu pekerjaan secara tuntas sehingga mudah menelusurinya seandainya memerlukan perbaikan atau perubahan. Masing-masing blok berdiri sendiri sehingga memudahkan dalam perakitan dan tidak mengganggu blok yang lain.

Dalam tulisan ini Turbo Pascal yang diperkenalkan adalah Turbo Pascal ciptaan Borland International Inc. yang versinya bermacam-macam, yaitu mulai dari versi 1.0 dan saat ini sudah sampai versi 7.0. Yang akan dibahas adalah Turbo Pascal versi 5.5

## A. ELEMEN-ELEMEN DASAR TURBO PASCAL

Turbo Pascal tersusun atas 3 elemen dasar yaitu ; simbol dasar (karakter), kata-kata simpanan (kata perbendaharaan) dan pengenalan (identifikasi)

### 1. Simbol dasar (Karakter) ; terdiri dari :

- a. Alfabet (simbol huruf), terdiri dari :  
huruf A ... Z dan a ... z
- b. Numerik (simbol angka), terdiri dari angka 0 ... 9
- c. Karakter Khusus (simbol tanda), terdiri dari tanda-tanda ; + - \* / := = . : ; ' ^ < <= > >=  
<> [] {} # \$ @ dan spasi

### 2. Kata-kata simpanan (Kata Perbendaharaan).

Yaitu nama variabel yang mempunyai arti tertentu dalam bahasa pemrograman Pascal dan tidak dapat diubah atau didefinisikan kembali. Kata-kata perbendaharaan Turbo Pascal versi 5.5 adalah sbb :

absolute	end	inline	procedure	type
and	external	interface	program	unit
array	file	interrupt	record	until
begin	for	label	repeat	uses
case	forward	mod	set	var
const	function	nil	shl	while
div	goto	not	shr	with
do	if	of	string	xor
downto	implementation	or	then	
else	in	packed	to	

### 3. Identifier (Pengenalan)

Yaitu nama pengenalan dalam bahasa Pascal yang digunakan untuk memberi nama terhadap suatu : tetapan (konstanta), tipe, variabel, prosedur, fungsi, unit dan field dalam record. Sebuah pengenalan haruslah memenuhi syarat :

- a. karakter pertama haruslah huruf atau garis bawah
- b. tidak mengandung karakter khusus
- c. bukan merupakan salah satu kata simpanan
- d. tidak berisi spasi

Sebuah pengenalan dapat terdiri dari sejumlah karakter yang tak terbatas banyaknya, tetapi maksimal hanya 63 karakter pertama yang diakui, artinya karakter ke 64 dan seterusnya akan diabaikan oleh Turbo Pascal.

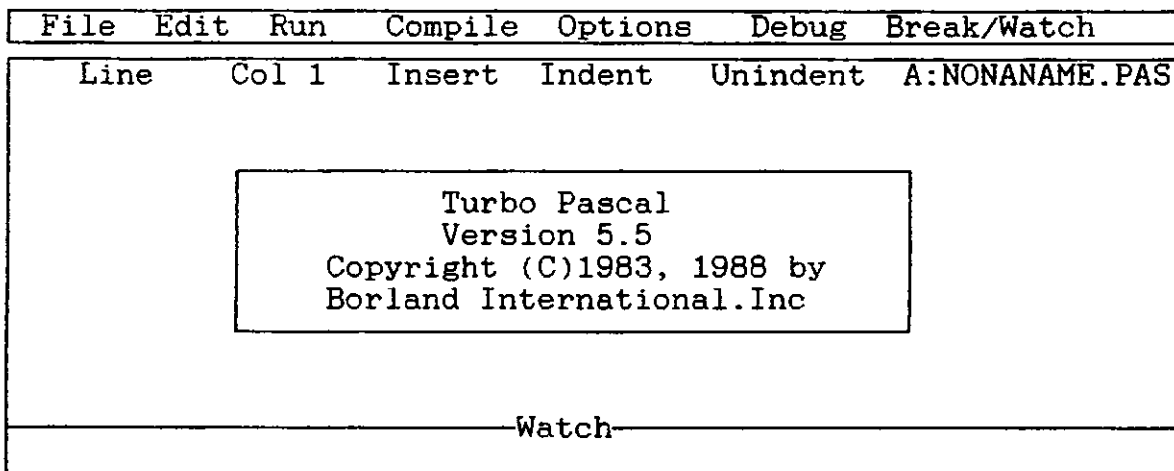
## B. PERSIAPAN MEMBUAT PROGRAM SEDERHANA

### 1. Persiapan Awal Menjalankan Turbo Pascal

- a. Masukkanlah disket DOS pada drive A, kemudian ON kan saklar stabilizer, monitor, dan CPU. Tekan <ENTER> sampai muncul A prompt atau : A>  
Selanjutnya dikeluarkan disket DOS dan diganti dengan disket program TURBO PASCAL Versi 5.00
- b. Ketikkan Turbo dibelakang prompt A tersebut kemudian tekan <ENTER> atau :

A>Turbo <ENTER>

Tunggu beberapa saat sampai pada layar monitor muncul "banner" yang bentuknya seperti gambar 2.1 berikut ini



F1-Help F5-Zoom F6-Switch F7 Trace F8-Step F9-Make F10-Menu

Gambar 2.1 Banner Turbo Pascal Versi 5.5

- c. Tekan <ENTER> maka banner akan menghilang dari layar

542 | k/90 (2)

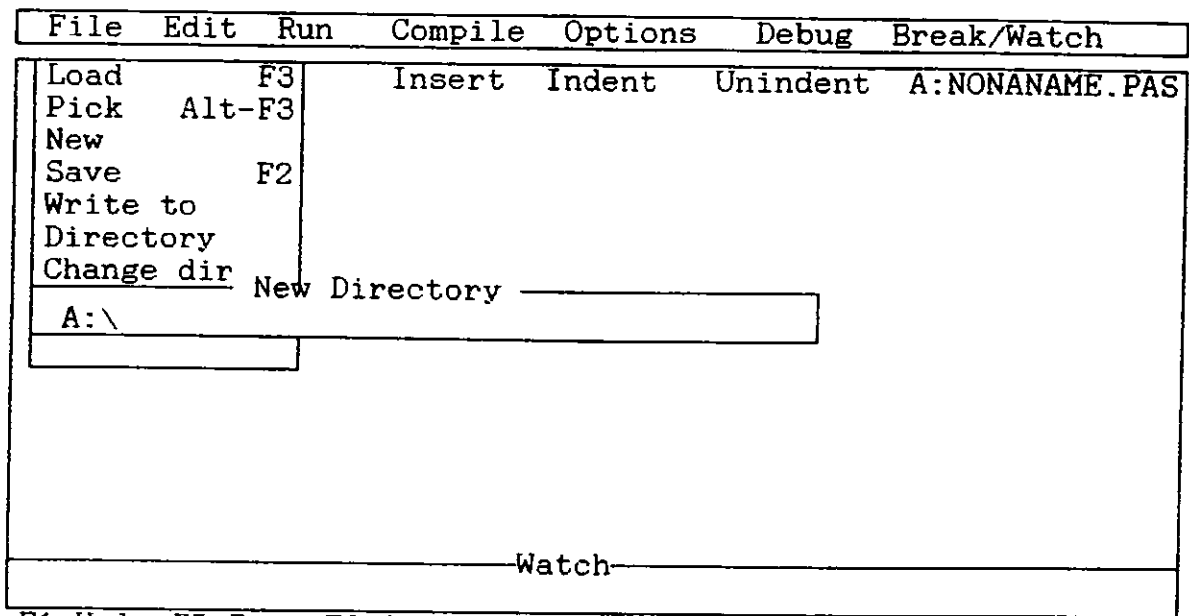
monitor, selanjutnya jika ditekan <ENTER> maka pada layar akan muncul seperti gambar 2.2 :

File	Edit	Run	Compile	Options	Debug	Break/Watch
Load		F3	Insert	Indent	Unindent	A:NONANAME.PAS
Pick	Alt-F3					
New						
Save		F2				
Write to						
Directory						
Change dir						
OS shell						
Quit	Alt-x					
Watch						
F1-Help F5-Zoom F6-Switch F7 Trace F8-Step F9-Make F10-Menu						

Gambar 2.2

Baris paling atas disebut menu utama (berisi 7 pilihan). Kecuali "Edit" maka setiap menu utama tersebut akan menampilkan pilihannya. Cara menggunakannya dapat dilakukan dengan menggunakan tombol <→> untuk menggeser kursor ke tempat menu yang diinginkan. Pada gambar atas, dibawah menu "File" tersedia 9 pilihan lagi, merupakan pilihan yang tersedia dari menu "File" Cara menggunakannya adalah dengan menekan tombol yang sesuai dengan huruf kapital yang ada pada awal kalimat pilihan tersebut, disaat kita berada pada menu "File"

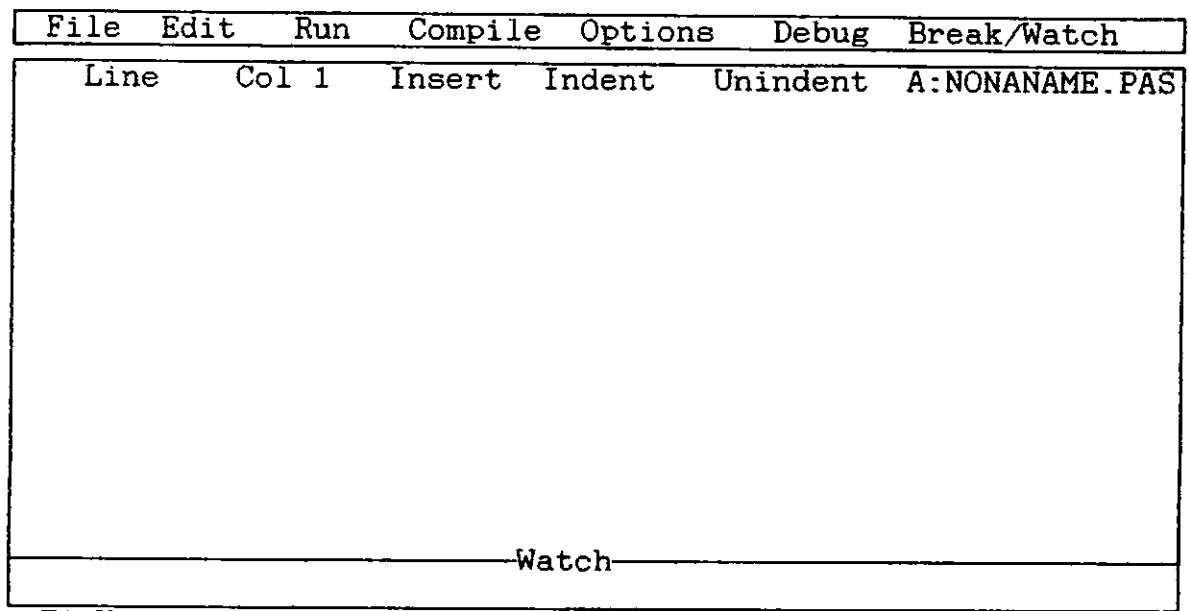
- d. Saat ini disk drive yang aktif adalah drive A. Masukkanlah disket kerja yang telah diformat pada drive B, dan tukarlah pemutar disket yang aktif menjadi drive B dengan cara menekan <C> = Change directory (Change dir), selanjutnya pada layar akan nampak seperti gambar 2.3 berikut ini :



F1-Help F5-Zoom F6-Switch F7 Trace F8-Step F9-Make F10-Menu

Gambar 2.3

Tekanlah <B> dan <:> untuk mengaktifkan disk drive B, selanjutnya tekan <ENTER>, maka pada layar monitor sudah siap untuk ditulisi seperti ditunjukkan pada gambar berikut ini :



F1-Help F5-Zoom F6-Switch F7 Trace F8-Step F9-Make F10-Menu

Gambar 2.4

- e. Kutiplah program berikut ini dengan baik, usahakan jangan membuat kesalahan, buatlah sebagaimana aslinya.

```
Program Volume_Balok ;
Uses Crt;
Var
  Panjang, Lebar, Tebal, Volume :Real;
Begin
  Clrscr;
  Writeln ('Menghitung Volume Balok');
  Writeln;
  Writeln ('Diketahui :(masukkan nilai panjang, lebar & tebal)');
  Writeln;
  Write ('panjang  :');Readln (panjang);
  Write ('lebar    :');Readln (lebar  );
  Write ('tebal    :');Readln (tebal  );
  Volume:= Panjang*Lebar*Tebal;
  Writeln;
  Writeln('Maka setelah proses : Volume=panjang x lebar x tebal');
  Writeln;
  Writeln ('Diperoleh : "Volume" :',Volume:10:2);
  Writeln;
  Write ('jika hendak keluar tekan <ENTER>');
End.
```

Bila menurut anda sudah tidak ada kesalahan dan yang terlewatkan dari program yang anda kutip, berikutnya tekanlah <F10> untuk mengaktifkan menu utama. Selanjutnya gunakan <—> untuk menggerakkan kursor ke menu Compile, atau menekan <C> selanjutnya tekan <ENTER> sehingga menu "Compile" ditampilkan. Tekanlah <C> atau <ENTER> untuk mengaktifkan pilihan "Compile" maka selanjutnya "Turbo Pascal" akan memeriksa program anda. barangkali masih ada kesalahan syntax. Bila ternyata masih ada terdapat kesalahan syntax yang ditemukan Turbo Pascal, maka pesan kesalahan itu akan dimunculkan pada layar monitor. Bila tidak ada kesalahan syntax, maka pada layar monitor akan ditampilkan pesan seperti gambar 2.5 berikut ini

```

File Edit Run Compile Options Debug Break/Watch
Line Col 1 Insert Indent Unindent A:NONANAME.PAS

      Compiling to Memory
Main file :A:\NONANAME.PAS
Compiling :EDITOR + NONANAME.PAS

                Total      File
Line compiled :   22       22

Available memory : 256 K
Success : Press any key

      Watch

F1-Help F5-Zoom F6-Switch F7 Trace F8-Step F9-Make F10-Menu

```

Gambar 2.5 Menandakan Program Terbebas Dari Kesalahan Syintak

Program yang sudah tidak ada terdapat kesalahan syntax ini dapat dijalankan dengan menekan <F10> untuk mengaktifkan menu utama, selanjutnya menekan <R> = Run atau menggunakan tombol <→> untuk pindah ke menu "Run" selanjutnya tekan <ENTER>, Hasilnya setelah ditekan <Run> adalah sbb :

Menghitung Volume Balok

Diketahui :(masukkan nilai panjang, lebar & tebal)

panjang :

Apa yang terjadi pada layar monitor dapat anda lihat. Bila ternyata program tersebut meminta sesuatu (Input) melalui papan ketik (key board) penuhilah, selanjutnya tekan <ENTER>, maka hasil tampilan adalah sbb :

Menghitung Volume Balok

Diketahui :(masukkan nilai panjang, lebar & tebal)

panjang :20  
lebar :30  
tebal :40



Maka setelah proses :  $\text{Volume} = \text{panjang} \times \text{lebar} \times \text{tebal}$

Diperoleh : "Volume" :24000.00

Jika hendak keluar tekan <ENTER>

Simpanlah program yang sudah jadi ini pada disket kerja di drive B, yaitu dengan cara menekan <F10> untuk mengaktifkan menu utama,

menekan <F> menuju ke menu "File" dan menekan <W> untuk mengaktifkan pilihan "Write to" dan tuliskanlah nama file yang akan anda simpan ini, misalnya B:COBA1.PAS, selanjutnya tekan <ENTER>.

Bila anda sudah selesai membuat dan mengedit suatu program, setelah anda berada pada menu "File" tekanlah <Q> = Quit. berarti anda sudah berada di DOS kembali.

### C. STRUKTUR PROGRAM TURBO PASCAL

Struktur program Turbo Pascal terdiri dari 3 blok utama yaitu : blok judul program, blok deklarasi dan blok badan program.

- a. Blok Judul Program ; Blok yang berisi nama program sebagai pengenalan jika program ini akan dipanggil atau digunakan. Blok judul program terdiri dari kata simpan program dan diikuti oleh nama program.
- b. Blok Deklarasi ; yaitu blok yang berisi pendefinisian tipe variabel yang dipakai dalam program bila diperlukan. Untuk program sederhana terdiri dari ; deklarasi variabel deklarasi tipe, deklarasi label, deklarasi konstanta dan deklarasi unit terpakai.
- c. Blok Badan Program ; Yaitu bagian yang berisi perintah-perintah untuk melakukan sesuatu yang diletakkan diantara kata simpanan BEGIN dan END serta diikuti oleh tanda titik (.). Bila perintah-perintah yang dibuat lebih dari satu, maka setiap perintah diakhiri tanda titik koma (;). Tanda titik (.) sesudah kata simpan END menandakan perintah telah selesai.

Berdasarkan struktur program yang telah diutarakan, dapat dibuat bentuk umum suatu program sederhana yaitu :

PROGRAM nama_program ;	BLOK JUDUL PROGRAM
USES nama_unit1, nama_unit2, ..., ...; LABEL nama_label1 ; nama_label2 ;  CONST nama_tetapan1 = nilai_tetapan1 ; nama_tetapan2 = nilai_tetapan2 ;  TYPE nama_tipe = jenis_tipe ; nama_tipe = tipe1, tipe2, ..., ...);  VAR pengenal1                    :tipe_pengenal ; pengenal1, pengenal2, ... :tipe_pengenal ;	BLOK DEKLARASI
BEGIN .... ; .... ; END.	BLOK BADAN PROGRAM

Gambar 2.6

#### D. INPUT DAN OUTPUT

Tugas sebuah program adalah menghasilkan sesuatu yang berguna sebagai jawaban permasalahan yang dihadapi sehubungan dengan program yang dibuat itu. Pada sebuah program, terdapat perintah khusus untuk mengeluarkan hasilnya yang disebut bagian keluaran (output). Adakalanya program tersebut juga membutuhkan data untuk dapat melakukan sesuatu. Bagian program yang bertugas untuk menerima data ini disebut

bagian masukan (input). Dalam bidang komputasi kedua bagian ini disebut I/O ( singkatan Input / Output ). Contoh tempat memasukkan data adalah keyboard, dan keluaran pada layar monitor.

Bagian masukan mempunyai 2 macam perintah, yaitu Read, dan Readln. Bentuk umumnya adalah :

```
READ (variabel1, variabel2, ... );  
READLN (variabel1, variabel2, ...);
```

1. Perintah READ, bila dipakai maka setelah membaca data, kursor akan tetap berada disamping data tersebut.
2. Perintah READLN, bila dipakai, maka setelah membaca data, kursor akan pindah ke kolom baris berikutnya.

Bagian keluaran mempunyai 2 macam perintah, yaitu Write, dan Writeln Bentuk umumnya adalah :

```
WRITE (variabel1, variabel2, ... );  
WRITELN (variabel1, variabel2, ...);
```

1. Perintah WRITE, bila dipakai maka setelah menulis data, kursor akan tetap berada disamping data tersebut.
2. Perintah WRITELN, bila dipakai, maka setelah menulis data kursor akan pindah ke kolom baris berikutnya.

Sebelum mulai merakit suatu program seserhana, sebaiknya membuat rancangan yang sebaik mungkin agar pemakaian komputer menjadi efisien.

Contoh 1 : Buatlah suatu program menentukan besarnya volume balok jika diketahui: Panjang = 5  
Lebar = 3  
Tebal = 2

Program tersebut dapat dibuat sebagai berikut :

```
Program Volume_Balok_01{Perintah Write(ln)};
Var
  Panjang, Lebar, Tebal, Volume :Real;
Begin
  panjang :=5;
  lebar   :=3;
  tebal   :=2;
  Volume := Panjang*Lebar*Tebal;
  Writeln (Volume);
End.
```

Jika program di eksekusi <RUN>, hasilnya adalah :

3.0000000000E+01

Program tersebut hanya dapat menghitung isi (volume) satu jenis balok saja, yaitu balok yang panjang, lebar dan tebalnya masing masing 5, 3, dan 2. Program sederhana yang dapat menghitung isi dari berbagai jenis balok dapat dilihat contoh berikut ini

Contoh 2 :

```
Program Volume_Balok_02{Read(ln) dan Write(ln)};
Var
  Panjang, Lebar, Tebal, Volume :Real;
Begin
  Read (panjang);
  Read (lebar  );
  Read (tebal  );
  Volume := Panjang*Lebar*Tebal;
  Writeln (Volume);
End.
```

Program contoh 2 tersebut, jika dieksekusi dan memasukkan nilai panjang = 2, lebar = 3 dan tebal = 4, diperoleh hasilnya sebagai berikut :

2  
3  
4  
2.4000000000E+01

Hasil eksekusi contoh 1 dan 2 kurang informatif, karena angka-angka hasil eksekusi tidak mencantumkan besaran apa yang ditampilkan itu. Agar hasil eksekusi lebih informatif, dapat dikutip program berikut ini :

Contoh 3 :

```
Program Volume_Balok_03{Write(ln) dan Read(ln)};
Var
  Panjang, Lebar, Tebal, Volume :Real;
Begin
  Write ('Panjang : ');Readln (panjang);
  Write ('lebar   : ');Readln (lebar  );
  Write ('tebal   : ');Readln (tebal  );
  Volume := Panjang*Lebar*Tebal;
  Writeln ('Volume : ',volume);
End.
```

Program contoh 3 tersebut, jika dieksekusi dan memasukkan nilai panjang = 2, lebar = 3 dan tebal = 4, diperoleh hasilnya sebagai berikut :

```
panjang :2
lebar   :3
tebal   :4
Volume  : 2.4000000000E+01
```

Lengkapilah contoh 3 tersebut dengan sarana format, sehingga perogramnya menjadi sbb:

Contoh 4 :

```
Program Volume_Balok_04{Sarana Format};
Var
  Panjang, Lebar, Tebal, Volume :Real;
Begin
  Write ('Panjang : ');Readln (panjang);
  Write ('lebar   : ');Readln (lebar  );
  Write ('tebal   : ');Readln (tebal  );
  Volume := Panjang*Lebar*Tebal;
  Writeln ('Volume : ',volume:10:2);
End.
```

Program contoh 4 tersebut, jika dieksekusi dan memasukkan nilai panjang = 2, lebar = 3 dan tebal = 4, diperoleh hasilnya

sebagai berikut :

```
panjang :2
lebar   :3
tebal   :4
Volume  :      24.00
```

Bandingkan hasil tampilan contoh 3 dengan hasil tampilan contoh 4 ternyata hasil tampilan contoh 4 lebih mudah dibaca dibandingkan hasil tampilan contoh 3.

Setelah mengutip beberapa contoh, nampaknya hasil eksekusi pada layar masih menampilkan hasil eksekusi contoh-contoh sebelumnya. Supaya layar hanya berisi dengan hasil eksekusi program yang sedang di jalankan, maka digunakan perintah Uses CRT dan Clrscr

Contoh 5 :

```
Program Volume_Balok_05{Uses Crt dan Clrscr};
Uses Crt;
Var
  Panjang, Lebar, Tebal, Volume :Real;
Begin
  Clrscr;
  Write ('Panjang : ');Readln (panjang);
  Write ('lebar   : ');Readln (lebar  );
  Write ('tebal   : ');Readln (tebal  );
  Volume := Panjang*Lebar*Tebal;
  Writeln ('Volume : ',volume:10:2);
End.
```

Jika program contoh 5 dieksekusi dan dimasukkan nilai panjang, lebar dan tebal, maka pada layar tidak akan ditampilkan hasil eksekusi sebelumnya.

Latihan : Rancanglah suatu program sederhana dengan menggunakan perintah read(ln), Write(ln), sarana format, Uses Crt dan Clrscr yang dapat menghitung massa jenis berbagai silinder materi jika : massa, jari-jari, dan panjang cilinder diketahui.

## E. PENDEKLARASIAN

### 1. Deklarasi Variabel (VAR)

Yaitu bagian yang memberi tahu program tentang tipe-tipe variabel yang dipakai dalam program yang bersangkutan. Turbo Pascal mengenal dua macam variabel yaitu ; variabel standar dan variabel tidak standar.

Variabel standard adalah variabel yang langsung dikenal oleh Turbo Pascal sehingga pemakaian variabel ini tidak perlu pendeklarasian terlebih dahulu.

Yang termasuk kedalam variabel standard adalah :

- Bilangan Bulat, yang dibagi menjadi 5 kelompok yaitu

No.	Type Data dan Jangkauan Numerik	Memori Dibutuhkan
1	Byte 0 sampai 255	1 byte
2	Shortint -128 sampai 127	1 byte
3	Integer -32768 sampai 32767	1 Byte
4	Word 0 sampai 65535	2 Byte
5	Longint -2147483648 s.d 214748647	4 Byte

Tabel 1. Bilangan Bulat

- Bilangan Nyata. dibagi menjadi 5 sub kelompok yaitu

No.	Type Data dan Jangkauan Numerik	Memori Dibutuhkan
1	Real s.d $1,7 \cdot 10^{+38}$ $2,9 \cdot 10^{-39}$	6 byte
2	Single s.d $3,4 \cdot 10^{+38}$ $1,5 \cdot 10^{-45}$	4 byte
3	Double s.d $1,7 \cdot 10^{308}$ $5,0 \cdot 10^{-324}$	8 Byte
4	Extended s.d $1,1 \cdot 10^{4932}$ $3,4 \cdot 10^{-4932}$	10 Byte
5	Comp s.d $2^{+63} - 1$ $-2^{+63} + 1$	8 Byte

Tabel 2. Bilangan Nyata

- Karakter ; membutuhkan memori 1 byte
- Boolean, yaitu tipe data yang terdiri dari kondisi True dan False . Boolean butuh memori 1 byte

Variabel tidak standard adalah variabel yang tidak dikenal Turbo Pascal, sehingga memerlukan pendeklarasian terlebih dahulu yang dilakukan oleh deklarasi TYPE.

## 2. Deklarasi Tipe (TYPE)

"TYPE" digunakan untuk pendeklarasian variabel yang tidak masuk variabel standard.

Turbo Pascal menyediakan tipe-tipe dasar terstruktur untuk dapat dipakai dalam pendeklarasian TYPE yaitu :

- \* String : kumpulan karakter-karakter
- \* Array : kumpulan variabel-variabel yang tipenya sama
- \* Record : kumpulan variabel-variabel yang tipenya berbeda
- \* Set : kumpulan karakter atau bilangan bulat dengan urutan tertentu
- \* file : tempat menyimpan data



### 3. Deklarasi Label (LABEL)

"LABEL" merupakan tanda pengenal berbentuk angka yang digunakan untuk tujuan eksekusi program meloncat dari suatu perintah ke perintah lain yang letaknya mungkin berjauhan. Deklarasi label biasanya berpasangan dengan perintah GOTO.

Karena loncatan tersebut, eksekusi dapat terjadi tidak dalam satu blok, maka akan sukar dalam penelusuran pencarian kesalahan. Dianjurkan untuk pemrogram pemula tidak menggunakan deklarasi label.

### 4. Deklarasi Konstanta (CONST)

"CONST" merupakan pendeklarasian sebuah tetapan dengan suatu nilai tertentu dan tetap. Penggunaan konstanta ini sangat memudahkan dalam memodifikasi program.

### 5. Deklarasi Unit Terpakai (USES)

"USES" dipakai untuk pemakaian unit yang berisi kumpulan prosedur dan fungsi pelaksana yang telah disiapkan dan dapat langsung dipakai dalam program. Unit bisa dibuat sendiri oleh pemrogram untuk suatu keperluan yang sering digunakan.

Turbo Pascal versi 5.5 juga menyediakan unit yang siap pakai seperti ; CRT, DOS, PRINT, GRAPH, TURBO dsb.

## F. STRUKTUR KONTROL

Kemampuan Turbo Pascal untuk melakukan pengulangan (iterasi) serta memilih suatu tindakan pencabangan (seleksi) dalam suatu tindakan disebut struktur kontrol.

### 1. Proses Pengulangan (Iterasi)

Dalam bahasa Pascal dikenal 4 macam pernyataan untuk melakukan proses pengulangan, yaitu pernyataan :

```
* FOR ....TO....DO          * FOR ....DOWNTO ....DO
* REPEAT ....UNTIL ....     * WHILE .....DO.....
```

#### a. Pernyataan FOR ...TO ...DO dan FOR ... DOWNTO ....DO

Digunakan untuk melakukan proses pengulangan bila kondisi pengulangan sudah diketahui sebelumnya.

Bentuk Umum :

1. FOR pengenal := nilai awal TO nilai akhir DO

Syarat :

1. nilai awal < nilai akhir
2. pengenal harus bertipe integer
3. penambahan otomatis dengan nilai 1

2. FOR pengenal := nilai awal DOWNTO nilai akhir DO

Syarat :

1. nilai awal > nilai akhir
2. pengenal harus bertipe integer
3. pengurangan otomatis dengan nilai 1

#### b. Pernyataan REPEAT .....UNTIL dan WHILE .....DO

Digunakan untuk melakukan pengulangan bila kondisi pengulangan (jumlah proses pengulangan) tidak diketahui dengan pasti. Parameter iterasi dapat berupa bilangan nyata atau kalimat.

Perbedaan pemakaian pernyataan REPEAT .....UNTIL dan WHILE ... DO adalah sebagai berikut :

<u>REPEAT .....UNTIL ...</u>	<u>WHILE .....DO</u>
1. Proses pengulangan setelah pernyataan REPEAT berakhir pada pernyataan UNTIL	1. Proses pengulangan setelah pernyataan DO selama kondisi pada pernyataan WHILE terpenuhi.
2. Tidak memerlukan pasangan pernyataan BEGIN ... END; untuk kumpulan pernyataan proses pengulangan.	2. Memerlukan pasangan pernyataan BEGIN ... END; bila pengulangan dilakukan lebih dari 1 kali.
3. Pemeriksaan kondisi pengulangan pada akhir blok ( sesudah pernyataan UNTIL )	3. Pemeriksaan kondisi pengulangan pada awal blok ( sesudah pernyataan WHILE ).

Bentuk Umumnya :

- |   |   |
|---|---|
| <p>1. REPEAT<br/>             &lt;pernyataan 1&gt;;<br/>             &lt;pernyataan 2&gt;;<br/>             .....<br/>             &lt;pernyataan n&gt;<br/>             UNTIL &lt;ungkapan nalar&gt;</p> | <p>2. WHILE &lt;ungkapan nalar&gt; DO<br/>             &lt;pernyataan&gt;</p> |
|---|---|

Berikut ini diberikan 4 buah contoh program sederhana penggunaan fasilitas iterasi diatas ( contoh 6, 7, 8, dan 9 ) dengan tujuan yang sama, yaitu menghitung Rapat Massa Silinder Materi yang jumlah silinder materinya diketahui. Rapat massa tersebut dihitung untuk masing-masing silinder materi sampai semuanya dihitung.

-----  
Contoh 6 : Pemakaian Iterasi : FOR ...TO...DO  
-----

```
Program Silinder_Materi_06{Iterasi :FOR ...TO..DO};
Uses Crt;
Var
  Radius,Panjang,Massa,Luas,Volume,Kerapatan:Real;
  Silinder_Ke,n :Integer;
Begin
  Clrscr;
  Writeln('Menghitung Rapat Massa Silinder Materi');
  Writeln;
  FOR Silinder_Ke := 1 TO n DO
  Begin
    Write('Diketahui : ');
    Writeln;
    Write('Jari-jari Alas : ');Readln (Radius);
    Write('Panjang Silinder : ');Readln (Panjang);
    Write('Massa Silinder : ');Readln (Massa );
    Writeln;
    Luas      := 3.14*Radius*Radius;
    Volume    := Luas*Panjang
    Kerapatan := Massa/Volume;
    Writeln('Rapat Massa Silinder Materi : ',Kerapatan:4:2);
    Writeln;
  End;
End.
```

Tampilan Pada Layar (Contoh 6) :

Menghitung Rapat Massa Silinder Materi

Banyak Silinder Materi :2

Diketahui :  
Jari-jari Alas :2  
Panjang Silinder :3  
Massa Silinder :4

Rapat Massa Silinder Materi : 0.11

Diketahui :  
Jari-jari Alas :2  
Panjang Silinder :3  
Massa Silinder :2

Rapat Massa Silinder Materi : 0.05

-----  
Contoh 7 : Pemakaian Iterasi : FOR ..DOWNTO...DO  
-----

```
Program Silinder_Materi_07{Iterasi :FOR ...DOWNTO...DO};
Uses Crt;
Var
  Radius,Panjang,Massa,Luas,Volume,Kerapatan:Real;
  Silinder_Ke,n :Integer;
Begin
  Clrscr;
  Writeln('Menghitung Rapat Massa Silinder Materi');
  Writeln;
  FOR Silinder_Ke := 1 DOWNTO n DO
  Begin
    Write('Diketahui : ');
    Writeln;
    Write('Jari-jari Alas   : ');Readln (Radius);
    Write('Panjang Silinder : ');Readln (Panjang);
    Write('Massa Silinder   : ');Readln (Massa );
    Writeln;
    Luas      := 3.14*Radius*Radius;
    Volume    := Luas*Panjang
    Kerapatan := Massa/Volume;
    Writeln('Rapat Massa Silinder Materi : ',Kerapatan:4:2);
    Writeln;
  End;
End.
```

Tampilan Pada Layar (Contoh 7) :

Menghitung Rapat Massa Silinder Materi

Banyak Silinder Materi :2

Diketahui :  
Jari-jari Alas :2  
Panjang Silinder :3  
Massa Silinder :4

Rapat Massa Silinder Materi : 0.11

Diketahui :  
Jari-jari Alas :2  
Panjang Silinder :3  
Massa Silinder :2

Rapat Massa Silinder Materi : 0.05

Catatan : Hasil tampilan contoh 6 = hasil tampilan contoh 7

UNIVERSITAS  
10

-----  
Contoh 8 : Pemakaian Iterasi : REPEAT....UNTIL  
-----

```
Program Silinder_Materi_08{Iterasi :REPEAT....UNTIL};
Uses Crt;
Var
  Radius,Panjang,Massa,Luas,Volume,Kerapatan:Real;
  Silinder_Ke,n :Integer;
Begin
  Clrscr;
  Writeln('Menghitung Rapat Massa Silinder Materi');
  Writeln;
  Write ('Banyak Silinder Materi : ');Readln(n)
  Writeln;
  Silinder_Ke := 0;
  REPEAT
    Begin
      Write('Diketahui : ');
      Writeln;
      Silinder_Ke := Silinder_Ke + 1;
      Write('Jari-jari Alas : ');Readln (Radius);
      Write('Panjang Silinder : ');Readln (Panjang);
      Write('Massa Silinder : ');Readln (Massa );
      Writeln;
      Luas      := 3.14*Radius*Radius;
      Volume    := Luas*Panjang
      Kerapatan := Massa/Volume;
      Writeln('Rapat Massa Silinder Materi : ',Kerapatan:4:2);
      Writeln;
    End;
  UNTIL Silinder_Ke = n;
End.
```

Tampilan Pada Layar (Contoh 8) :

Menghitung Rapat Massa Silinder Materi

Banyak Silinder Materi :2

Diketahui :

Jari-jari Alas :2

Panjang Silinder :3

Massa Silinder :4

Rapat Massa Silinder Materi : 0.11

Diketahui :

Jari-jari Alas :2

Panjang Silinder :3

Massa Silinder :2

Rapat Massa Silinder Materi : 0.05

-----  
Contoh 9 : Pemakaian Iterasi : WHILE....DO  
-----

```
Program Silinder_Materi_09{Iterasi :WHILE ....DO};
Uses Crt;
Var
  Radius,Panjang,Massa,Luas,Volume,Kerapatan:Real;
  Silinder_Ke,n :Integer;
Begin
  Clrscr;
  Writeln('Menghitung Rapat Massa Silinder Materi');
  Writeln;
  Write (' Banyak Silinder Materi : ');Readln(n)
  Writeln;
  Silinder_Ke := 0;
  WHILE Silinder_Ke < n DO
    Begin
      Write('Diketahui : ');
      Writeln;
      Silinder_Ke := Silinder_Ke + 1;
      Write('Jari-jari Alas : ');Readln (Radius);
      Write('Panjang Silinder : ');Readln (Panjang);
      Write('Massa Silinder : ');Readln (Massa );
      Writeln;
      Luas      := 3.14*Radius*Radius;
      Volume    := Luas*Panjang
      Kerapatan := Massa/Volume;
      Writeln('Rapat Massa Silinder Materi : ',Kerapatan:4:2);
      Writeln;
    End;
  UNTIL Silinder_ke = n;
End.
```

Tampilan Pada Layar (Contoh 9) :

Menghitung Rapat Massa Silinder Materi

Banyak Silinder Materi :2

Diketahui :  
Jari-jari Alas :2  
Panjang Silinder :3  
Massa Silinder :4

Rapat Massa Silinder Materi : 0.11

Diketahui :  
Jari-jari Alas :2  
Panjang Silinder :3  
Massa Silinder :2

Rapat Massa Silinder Materi : 0.05

Catatan : Hasil tampilan contoh 8 = hasil tampilan contoh 9

## 2. Proses Pencabangan (Seleksi)

Adakalanya komputer harus mengambil suatu keputusan, dimana proses pengambilan keputusan itu dilakukan bila suatu kondisi tertentu terpenuhi. Dalam bahasa Pascal dikenal beberapa pernyataan untuk proses pengambilan keputusan tersebut (seleksi) yaitu pernyataan :

- a. IF...THEN ... dan IF ....THEN ....ELSE
- b. CASE .....OF dan CASE .....OF .....ELSE

- a. Pernyataan IF ...THEN ..... dan IF ...THEN .....ELSE  
Pernyataan IF ...THEN ...digunakan untuk menentukan pilihan dengan 2 kemungkinan. Ekpressi yang diuji haruslah ekpressi boolean yang mempunyai 2 kemungkinan pilihan True dan False, untuk persyaratan terpenuhi atau tidak terpenuhi. Bila pilihan lebih dari 2 kemungkinan, dipakai pernyataan IF ...THEN ....ELSE secara bertingkat.

Bentuk Umumnya :

1. IF ekpressi boolean THEN blok pernyataan ;
2. IF ekpressi boolean THEN blok pernyataan 1  
ELSE blok pernyataan 2;

Ekpressi boolean merupakan ungkapan nalar yang dipakai sebagai kondisi pengatur eksekusi terhadap pernyataan sesudah IF. Pernyataan sesudah THEN dieksekusi bila ekpressi boolean bernilai True (benar) dan pernyataan sesudah ELSE dieksekusi jika ekpressi boolean bernilai False (salah). Syarat pemakaian, pernyataan sebelum



ELSE tidak boleh diakhiri dengan tanda titik koma (;).  
Berikut ini diberikan contoh 10 dan 11, yaitu penca-  
bangan (seleksi) dengan memakai pernyataan :

IF ....THEN dan IF THEN ELSE

-----  
Contoh 10 : Pencabangan : IF .....THEN  
-----

```
Program Luas_Permukaan_Silinder;
Uses
  Crt;
Var
  Radius, Panjang, Luas : Real;
  Silinder_ke, n : Integer;
Begin
  Clrscr;
  Writeln ('Menghitung Luas Permukaan Silinder');
  Writeln;
  Write ('Banyak Silinder : '); Readln (n);
  Writeln;
  For Silinder_ke := 1 To n Do
  Begin
    Writeln ('Bila diketahui : ');
    Writeln;
    Write ('Jari-jari :? '); Readln (Radius );
    Write ('Panjang :? '); Readln (Panjang);
    Writeln;
    If Silinder_ke mod 2 = 0 Then
    Begin
      Luas := 3.14*Radius*Radius+(2*Panjang);
      Writeln ('Luas PermukaanSilinder :', Luas:10:2);
      Writeln;
    End;
  End;
End.
```

Hasil Eksekusi (tampilan pada layar ):

Menghitung Luas Permukaan Silinder

Banyak Silinder : 2

Bila diketahui :

Jari-jari : ? 3

Panjang : ? 4

Bila diketahui :

Jari-jari : ? 4

Panjang : ? 3

Luas Permukaan Silinder : 56.24

-----  
Contoh 11 : Pencabangan : IF .....THEN.....ELSE  
-----

```
Program Luas_Permukaan_Silinder;
Uses
  Crt;
Var
  Radius, Panjang, Luas : Real;
  Silinder_ke, n : Integer;
Begin
  Clrscr;
  Writeln ('Menghitung Luas Permukaan Silinder');
  Writeln ('Dengan nomor urut genab');
  Writeln;
  Write ('Banyak Silinder : '); Readln (n);
  Writeln;
  For Silinder_ke := 1 To n Do
  Begin
    Writeln ('Bila diketahui : ');
    Writeln;
    Write ('Jari-jari : '); Readln (Radius);
    Write ('Panjang : '); Readln (Panjang);
    Writeln;
    If Silinder_ke mod 2 = 0 Then
    begin
      Luas := 3.14*Radius*Radius+(2*Panjang);
      Writeln ('Luas PermukaanSilinder : ', Luas:10:2);
    End;
    Else Writeln ('Nomor urut silinder ganjil');
    Writeln;
  End;
End.
```

Hasil Eksekusi (tampilan pada layar) :

Menghitung Luas Permukaan Silinder  
Dengan nomor urut genab

Banyak Silinder : 3

Bila diketahui :

Jari-jari : ? 2

Panjang : ? 3

Nomor urut silinder ganjil

Bila diketahui :

Jari-jari : ? 2

Panjang : ? 5

Luas Permukaan Silinder : 22.56

Bila diketahui :

Jari-jari : ?

b. Pernyataan CASE ... OF ... dan CASE ...OF ...ELSE.

Pernyataan CASE digunakan dalam mengambil keputusan untuk memilih salah satu menu (pilihan) dari sejumlah menu yang disediakan.

Bentuk Umumnya

1. CASE pengenal OF

```
label1 :pernyataan1;  
label2 :pernyataan2;  
dst.
```

2. CASE pengenal OF

```
label1 :pernyataan1;  
label2 :pernyataan2;  
dst.  
ELSE  
pernyataan_a;  
pernyataan_b;  
dst.
```

Keterangan :

1. pengenal dapat sebarang ungkapan seperti Char, Byte, boolean
2. label 1.2. dst. boleh berupa satu atau beberapa nilai bertipe ordinal
3. pernyataan 1.2. dst. akan dieksekusi jika nilai dari pengenal tercantum dalam daftar label (1,2 dan seterusnya). Apabila salah satu pernyataan tersebut sudah dijalankan, eksekusi selanjutnya dilakukan terhadap perintah sesudah kata simpan END. yang merupakan penutup pernyataan CASE.
4. ELSE dalam pernyataan CASE hanya bersifat opsional

dan pernyataan ELSE ini akan dieksekusi hanya bila nilai pengenal tidak terdapat dalam daftar label yang disediakan.

Berikut ini diberikan contoh program yang memakai pernyataan CASE ...OF dan CASE ... OF ...ELSE

-----  
Contoh 12 : Pemakaian Pernyataan : CASE ... OF  
-----

```
Program Hitung_Silinder {Pernyataan CASE...OF};
Uses
  Crt;
Const
  Pilihan = 'Pilih salah satu';
Var
  Radius,Panjang,Massa,La,Ls,Lp,Vs,Rm:Real;
  pilihlah : Char;
Begin
  Clrscr;
  Writeln ('Menghitung Data Silinder');
  Writeln;
  Write ('Jari-jari   : ? ');Readln (Radius );
  Write ('Panjang     : ? ');Readln (Panjang);
  Write ('Massa        : ? ');Readln (Massa );
  Writeln;
  Writeln ('Petunjuk :','Pilihlah A B C D atau E');
  Writeln ('A = luas alas silinder');
  Writeln ('B = luas selimut silinder');
  Writeln ('C = luas permukaan silinder');
  Writeln ('D = volume silinder');
  Writeln ('E = Rapat massa silinder');
  Writeln;
  Write ('pilihan : ');Readln (pilihlah);
  Writeln;
  Case pilihlah Of
    'A' :Begin
      La := 3.14*Radius*Radius;
      Writeln ('Luas Alas Silinder  :',La:10:2);
    End;
    'B' :Begin
      Ls := 2*3.14*Radius*Panjang;
      Writeln ('Luas Selimut Silinder  :',Ls:10:2);
    End;
    'C' :Begin
      Lp := 3.14*Radius*(Radius+(2*Panjang));
      Writeln ('Luas Permukaan Silinder  :',Lp:10:2);
    End;
```

```

'D' :Begin
    Vs := 3.14*Radius*Radius*Panjang;
    WriteLn ('Volume Silinder :',Vs:10:2);
End;
'E' :Begin
    Rm := Massa/Vs;
    WriteLn ('Rapat Massa Silinder :',Rm:10:2);
End;
End;
ReadLn:
End.

```

Hasil Eksekusi Contoh di atas adalah sbb :

Menghitung Data Silinder

```

Jari-jari : ? 3
Panjang   : ? 2
Massa     : ? 4

```

```

Petunjuk :Pilihlah A B C D atau E
A = luas alas silinder
B = luas selimut silinder
C = luas permukaan silinder
D = volume silinder
E = Rapat massa silinder

```

pilihan : A

Luas Alas Silinder : 28.26

Menghitung Data Silinder

```

Jari-jari : ? 3
Panjang   : ? 2
Massa     : ? 4

```

```

Petunjuk :Pilihlah A B C D atau E
A = luas alas silinder
B = luas selimut silinder
C = luas permukaan silinder
D = volume silinder
E = Rapat massa silinder

```

pilihan : B

Luas Selimut Silinder : 37.68

Menghitung Data Silinder

Jari-jari : ? 3  
Panjang : ? 2  
Massa : ? 4

Petunjuk :Pilihlah A B C D atau E  
A = luas alas silinder  
B = luas selimut silinder  
C = luas permukaan silinder  
D = volume silinder  
E = Rapat massa silinder

pilihan : C

Luas Permukaan Silinder : 32.26

Menghitung Data Silinder

Jari-jari : ? 3  
Panjang : ? 2  
Massa : ? 4

Petunjuk :Pilihlah A B C D atau E  
A = luas alas silinder  
B = luas selimut silinder  
C = luas permukaan silinder  
D = volume silinder  
E = Rapat massa silinder

pilihan : D

Volume Silinder : 56.52

Menghitung Data Silinder

Jari-jari : ? 3  
Panjang : ? 2  
Massa : ? 4

Petunjuk :Pilihlah A B C D atau E  
A = luas alas silinder  
B = luas selimut silinder  
C = luas permukaan silinder  
D = volume silinder  
E = Rapat massa silinder

pilihan : E

Rapat Masa Silinder : 0.07

## Menghitung Data Silinder

Jari-jari : ? 3  
Panjang : ? 2  
Massa : ? 4

Petunjuk :Pilihlah A B C D atau E  
A = luas alas silinder  
B = luas selimut silinder  
C = luas permukaan silinder  
D = volume silinder  
E = Rapat massa silinder

pilihan : E

Ternyata jika pilihan tidak sesuai dengan menu yang disediakan, program tidak menampilkan hasil eksekusi

-----  
Contoh 13 : Pemakaian Pernyataan : CASE ... OF ....ELSE  
-----

```
Program Hitung_Silinder {Pernyataan CASE...OF...ELSE};
Uses
  Crt;
Var
  pilihan = 'Pilih salah satu';
  R, P, M, La, Ls, Lp, Vs, Rm: Real;
  pilihlah : Char;
Begin
  Clrscr;
  Writeln ('Menghitung Data Silinder');
  Writeln;
  Write ('Jari-jari : ? '); Readln (Radius );
  Write ('Panjang : ? '); Readln (Panjang);
  Write ('Massa : ? '); Readln (Massa );
  Writeln;
  Writeln ('Petunjuk :'. 'Pilihlah A B C D atau E');
  Writeln ('A = luas alas silinder');
  Writeln ('B = luas selimut silinder');
  Writeln ('C = luas permukaan silinder');
  Writeln ('D = volume silinder');
  Writeln ('E = Rapat massa silinder');
  Writeln;
  Write ('pilihan : '); Readln (pilihlah);
  Writeln;
  Case pilihlah Of
    'A' :Begin
      La := 3.14*Radius*Radius;
```

```

        Writeln ('Luas Alas Silinder  :',La:10:2);
    End;
'B' :Begin
    Ls := 2*3.14*Radius*Panjang;
    Writeln ('Luas Selimut Silinder  :',Ls:10:2);
    End;
'C' :Begin
    Lp := 3.14*Radius*(Radius+(2*Panjang));
    Writeln ('Luas Permukaan Silinder  :',Lp:10:2);
    End;
'D' :Begin
    Vs := 3.14*Radius*Radius*Panjang;
    Writeln ('Volume Silinder  :',Vs:10:2);
    End;
'E' :Begin
    Rm := Massa/Vs;
    Writeln ('Rapat Massa Silinder  :',Rm:10:2);
    End;
Else
    Begin
        Writeln ('Pilihan Anda Salah !  Ulangi lagi');
    End;
End;
End.

```

Hasil Eksekusi program contoh diatas, jika pilihan yang dimasukkan melalui papan ketik (keyboard) tidak sesuai dengan menu yang tersedia pada pernyataan CASE.....OF, yaitu pilihan (A, B, C, D, atau E) maka program perintah sesudah pernyataan ELSE, yaitu : "Pilihan Anda Salah ! Ulangi lagi". Perhatikanlah hasil eksekusi berikut ini dari program contoh diatas.

#### Menghitung Data Silinder

```

Jari-jari : ? 3
Panjang   : ? 2
Massa     : ? 4

```

```

Petunjuk :Pilihlah A B C D atau E
A = luas alas silinder
B = luas selimut silinder
C = luas permukaan silinder
D = volume silinder
E = Rapat massa silinder

```



pilihan : k

Pilihan Anda Salah ! Ulangi lagi

Menghitung Data Silinder

Jari-jari : ? 5

Panjang : ? 6

Massa : ? 3

Petunjuk :Pilihlah A B C D atau E

A = luas alas silinder

B = luas selimut silinder

C = luas permukaan silinder

D = volume silinder

E = Rapat massa silinder

pilihan : 10

Pilihan Anda Salah ! Ulangi Lagi

## G. ALGORITMA

Dalam menyelesaikan suatu problem, sering kali kita perlu merencanakan langkah demi langkah secara sistematis agar setiap urutan kegiatan yang dilakukan selalu mempunyai arti penting dalam menyelesaikan problem tersebut. Dalam kegiatan pemrograman dengan komputer dikenal istilah algoritma. Mengutip pendapat Sumartono (1989 :44), yang dimaksud dengan algoritma adalah : " suatu daftar urutan instruksi untuk menyelesaikan suatu problem khusus, atau suatu proses selangkah demi selangkah (step by step)".

Umumnya sebuah algoritma untuk suatu proses dapat disusun dengan berbagai cara, pada dasarnya algoritma ini merupakan perencanaan yang matang agar dalam menyelesaikan suatu problem dapat dilakukan langkah-langkah yang efektif dan efisien. Untuk suatu proses berskala besar algoritmanya disusun dengan langkah-langkah yang banyak dan sangat rumit,

contohnya algoritma yang dilaksanakan sistem komputer terdiri dari berjuta-juta langkah keunsuran (elementer).

Berikut ini diberikan sebuah algoritma untuk menentukan nilai rata-rata kelas mata pelajaran fisika di kelas II SMU :

1. Tuliskan nama seluruh siswa secara berurutan dalam suatu daftar nilai sehingga diketahui jumlah siswa seluruhnya
2. Tuliskan nilai tiap siswa pada baris yang sesuai dengan nama siswa
3. Jumlahkan nilai seluruh siswa
4. Bagi jumlah seluruh nilai siswa dengan jumlah siswa sehingga diperoleh nilai rata-rata kelas siswa.

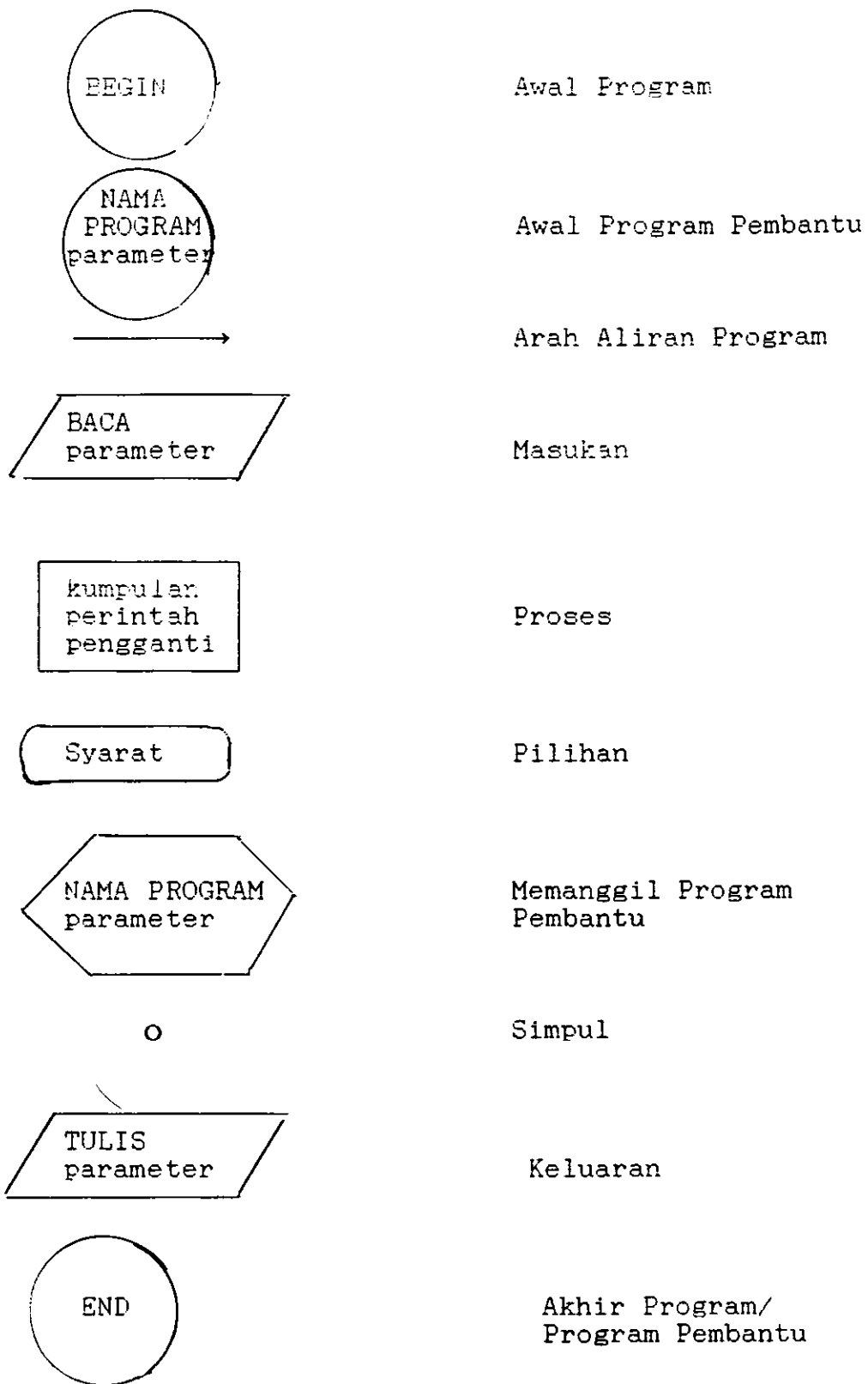
Untuk suatu proses berskala besar, pembuatan algoritma semacam contoh diatas sangat membosankan dan sangat rumit. Untuk menuangkan ide suatu algoritma secara lebih cepat dan tepat, dapat dituangkan dalam sebuah diagram alir (flow chart).

#### H. PEMBUATAN DIAGRAM ALIR (FLOW CHART)

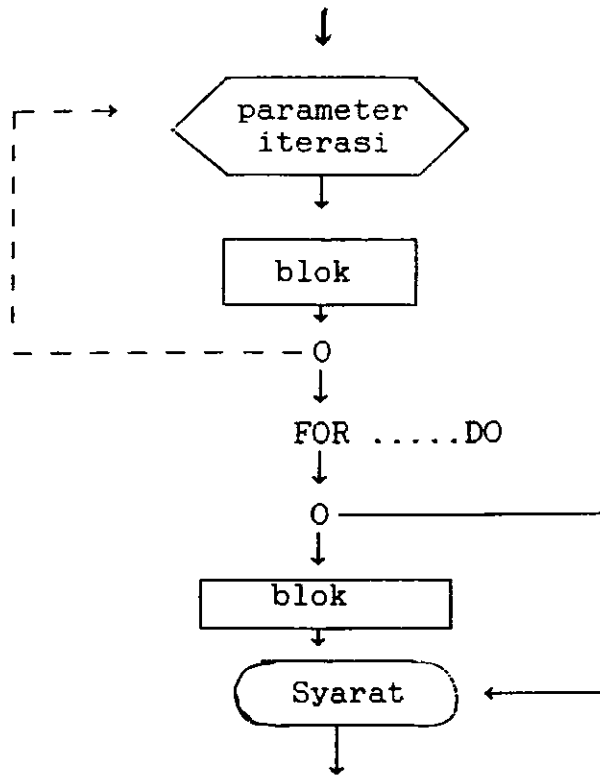
"Suatu diagram alir adalah suatu diagram yang mempergunakan peta simbol-simbol dan menunjukkan sebuah algoritma untuk sebuah proses" (Sumartono, 1989 : 44). Diagram alir ini dimaksudkan untuk mengatasi algoritma yang sangat rumit sehingga, apabila pembuatan program dimulai dengan membuat diagram alir ini akan memberikan kemudahan bagi sipemrogram dalam menyelesaikan programnya, juga kemudahan dalam menelusuri dan memperbaiki program. dibandingkan pembuatan program secara langsung.

Suatu diagram alir dapat dipakai untuk menterjemahkan suatu algoritma kedalam setiap bahasa pemrograman, karena simbol-simbol yang digunakan dalam suatu diagram alir memberikan pandangan menyeluruh dari sebuah prosedur dan berlaku umum untuk semua bahasa pemrograman. Simbol-simbol dalam suatu diagram alir mempunyai arti tertentu seperti halnya dengan rambu-rambu lalu lintas di jalan. Para ahli komputer menyepakati bahwa paling tidak ada 7 (tujuh) simbol pokok yang dipakai dalam suatu diagram, yaitu :

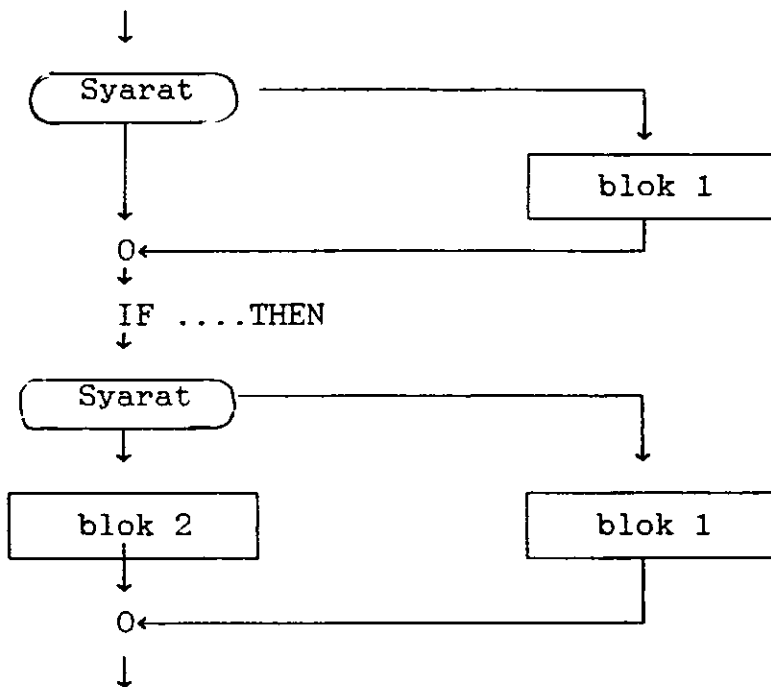
1. Lingkaran : simbol menunjukkan awal/akhir suatu program
2. Kotak : simbol menyatakan proses/ instruksi yang memerlukan suatu daftar perintah untuk mengimplemen-  
tasikannya.
3. Jajaran genjang : simbol menyatakan masukan atau keluaran
4. Lingkaran Oval/ellips : simbol me-  
nyatakan persyaratan/pilihan
5. Garis pakai panah/arrow : simbol menyatakan arah aliran program
6. Lingkaran kecil : simbol simpul/  
penghubung.
7. Segi enam : simbol untuk memanggil program pembantu.



Gambar 2.7. Simbol Diagram Alir  
(Soengeng, 1992 : 35)



Gambar 2.8. Repeat ...until atau While ....do  
(Soegeng, 1992 : 36)



Gambar 2.9. Repeat ...until atau While ....do  
(Soegeng, 1992 : 36)

Untuk penyelesaian promblem-problem kecil pembuatan diagram alir bersifat opstional (boleh diabaikan) namun untuk pemula dan melatih sistematika kerja yang baik dianjurkan menyisihkan sedikit waktu untuk membuat diagram alir ini Hal ini dimaksudkan agar si pemrogram dapat terbantu dalam memperbaiki program-program yang rumit di masa mendatang.

Untuk mendisain sebuah diagram alir yang baik ada beberapa prinsip dan teknik yang dapat diikuti :

1. Hindari pemakaian istilah yang hanya berlaku untuk suatu jenis komputer dan tidak berlaku umum untuk berbagai jenis komputer.
2. Dalam menyiapkan diagram alir, alur utama penalaran (logika) yaitu penalaran normal harus dilengkapi terlebih hulu.
3. Setiap simbol/blok harus mewakili suatu program lengkap dan mempunyai arti bagi pemrogram. Pecahlah langkah-langkah rumit (komplek) menjadi langkah-langkah yang lebih kecil/sederhana.
4. Disain dasar suatu diagram alir biasanya mengandung 3 (tiga) struktur kontrol yaitu :
  - a. struktur urutan (sequens): aliran biasanya dari atas kebawah dan dari kiri ke kanan.
  - b. struktur persyaratan (conditional) : pencabangan dapat muncul dan tergantung pada beberapa persyaratan.
  - c. struktur iterasi (loop) : proses-proses tertentu dalam loop diulang sampai suatu persyaratan terpenuhi.

#### I PROGRAM PEMBANTU (SUB PROGRAM)

Dalam bahasa pemrograman Turbo Pascal, menurut jumlah keluarannya dikenal 2 (dua) macam program pembantu (sub program) yaitu :

- a. Program Pembantu Prosedur, bila banyak keluarannya lebih dari satu,
- b. Program Pembantu Fungsi, bila keluarannya hanya satu (Soegeng, 1992 :57)

Kedua sub program tersebut tidak dapat berdiri sendiri, artinya hanya dapat dijalankan dengan adanya program utama. Tugas sub program adalah melakukan sebahagian tugas program utama. Soegeng (1992 :57) membagi sub program ini menjadi 3 (tiga) kelompok, yaitu :

1. Yang bergantung pada program utama. Sub program ini mendapat masukan data dari program utama dan kemudian mengeluarkannya pula ke program utama. Masukan disebut parameter nilai dan keluaran disebut parameter variabel.
2. Yang setengah bebas. Sub program ini mendapat masukan dari program utama, tapi hasilnya tidak dikembalikan ke program utama, tetapi ke alat keluaran seperti monitor, atau mesin cetak. Masukannya disebut parameter nilai.
3. Yang bebas. Sub program ini tidak mendapat masukan dari program utama dan tidak mengeluarkan hasilnya ke program utama. (sub program tanpa parameter).

Pada dasarnya bagian-bagian dari suatu sub program hampir sama dengan bagian-bagian suatu program sederhana, yaitu terdiri dari bagian *judul*, *deklarasi*, dan *bagian badan* sub program.

Bagian judul, terdiri dari kata simpan *procedure* untuk program pembantu prosedur, dan kata simpan *function*, untuk program pembantu fungsi. Untuk program pembantu setengah bebas, dan yang bergantung pada program utama, setelah kata simpan diikuti dengan nama parameter yang diletakkan diantara kurung biasa.

Bagian deklarasi, sama dengan bagian deklarasi pada program

seederhana dan

Bagian badan program pembantu, juga sama dengan pada program sederhana, cuma setelah diakhiri dengan kata simpan END diikuti dengan tanda titik koma (;)

Program pembantu (sub program) diletakkan pada bagian deklarasi program utama, dan dipanggil dengan menyebutkan nama sub program tersebut dan diikuti dengan nama parameternya jika ada. Secara sederhana, letak sub program dalam program utama, seperti urutan berikut :

```
PROGRAM Judul Program Utama ;  
    PROCEDURE Judul prosedur ;  
    BEGIN  
        Badan Sub Program  
    END;  
BEGIN {Program Utama }  
    Badan Program Utama  
END.{Akhir Preogram Utama}
```

Dalam satu program utama dapat diisi dengan lebih dari satu program pembantu (sub program) baik untuk program pembantu prosedur atau program pembantu fungsi, dan dalam program pembantu juga dapat diisi dengan program pembantu yang lain.

## J. DATA TERSTRUKTUR

Kita telah mengenal berbagai tipe data sederhana, se-



perti integer, real, string dan boolean. Disamping itu dalam bahasa pemrograman Turbo Pascal juga dikenal data terstruktur yang terdiri dari *Array*, *Set*, *Record*, dan *File*.

1. Array (Larik); pengenalan untuk menampung sekumpulan data yang tipenya sama. Setiap elemen data dibuat dengan nama pengenalan yang sama, dengan indeks berbeda, seperti ;  $X_1, X_2, X_3, \dots$ . Contoh, Vektor dan matriks adalah array 1 dimensi, dan determinan dapat berupa array 2 dimensi atau lebih.

Ada 2 cara mendeklarasikan array, yaitu :

a. TYPE  
    tipe\_pengenal = ARRAY [batasan indeks ]OF tipe data;  
VAR  
    nama\_variabel : tipe\_pengenal ;

b. VAR  
    nama\_variabel = ARRAY [batasan indeks ]OF tipe data;

2. Set (Himpunan) ; diartikan sebagai kumpulan objek atau elemen yang bertipe sama dan merupakan data sederhana (tidak data terstruktur), contohnya ; himpunan nama hari nama bulan, alamat, bilangan bulat, dsb. Dalam mendeklarasikan suatu set, diletakkan diantara tanda kurung siku [ ` dan ` ], dan urutan anggota set tidak terlalu penting, boleh tidak teratur. Bentuk deklarasi himpunan/set adalah sbb ;

a. TYPE   pengenal\_1 = tipe\_data;  
          pengenal\_2 = SET OF pengenal\_1;  
VAR    pengenal\_3 : pengenal\_2;

b. TYPE   pengenal\_1 = tipe\_data;  
VAR    pengenal\_2 : SET OF pengenal\_1;

c. VAR   pengenal : SET OF tipe\_data;

3. Record ; merupakan data terstruktur yang terdiri atas sekumpulan variabel data terstruktur, yang disebut 'field' yang tipenya boleh sama atau berlainan. Data boleh bertipe integer, real, character, byte atau boolean. Misalnya record mengenai data nama, alamat, pekerjaan dsb. Bentuk pendeklarasian record, adalah sbb :

```
TYPE
  nama_record = RECORD
    field_1 : tipe_data_1;
    field_2 : tipe_data_2;
    .....
    .....
    field_n : tipe_data_n;
END;
VAR pengenal : nama_record;
```

4. File ; diartikan sebagai kumpulan data yang bertipe sama dalam jumlah tertentu. Tipe file dapat berbentuk teks, integer, real, array ataupun record. Dikenal 2 macam file, yaitu *file program* yang bertugas menyimpan program dan *file data* dan disebut file saja. Tugas suatu file adalah menyimpan data (informasi). File perlu dideklarasikan terlebih dahulu dengan menggunakan kata simpan '*file*' dan diikuti oleh tipe data yang disimpan didalamnya. Contoh :

```
finteg = FILE OF INTEGER ;
```

artinya filinteg adalah file untuk menyimpan bilangan bulat (integer).

## BAB III. GRAFIK

### A. MODE TEKS DAN MODE GRAFIK

Layar monitor dapat ditulisi dengan kata-kata atau untuk melakukan perhitungan-perhitungan numerik, penggunaan yang demikian disebut "mode teks". Disamping itu layar monitor dapat pula digunakan untuk menggambar atau membuat grafik, pada keadaan ini dikatakan layar monitor berada dalam "mode grafik".

Dalam keadaan biasa layar monitor siap untuk mode teks, dan bila hendak digunakan sebagai tempat menggambar atau membuat grafik, terlebih dahulu harus dirubah dari mode teks ke mode grafik dengan menggunakan "unit graph" atau file Graph.TPU dengan beberapa perintah, yaitu :

1. perintah untuk mengenali jenis monitor yang dipakai
2. perintah untuk memilih mode grafik yang dikehendaki
3. perintah untuk menemukan letak dari "interface" grafik

Untuk kembali dari mode grafik ke mode teks hanya ada satu perintah saja, misalnya dengan menekan tombol <esc>.

Kumpulan perintah-perintah ini selalu dipakai bila hendak menggambar grafik, maka dianjurkan dibuat dalam bentuk unit, seperti unit grafiki berikut ini :

```
Unit Grafiki;  
Interface  
  Uses  
    Crt, Graph;  
  Procedure Membuka_grafik;  
  Procedure Menutup_grafik;  
Implementation  
  Procedure Membuka_grafik;  
  Var  
    gd, gm, Errorcode : Integer;  
  Begin  
    gd := Detect;
```

```

    Initgraph (gd, gm, "");
    Errorcode := Graphresult;
    If Errorcode <> Grok Then Halt;
End;

Procedure Menutup_grafik;
Begin
    Repeat until Readkey = #27;
    Closegraph;
End;
End.

```

Perintah "detect" pada unit grafik tersebut dimaksudkan agar program yang menggunakan unit ini akan berlaku untuk segala jenis monitor.

Unit Grafik ini berisi 2 program pembantu, yaitu :  
 procedure membuka grafik dan procedure menutup grafik serta perintah repeat until readkey = #27; yang maksudnya supaya menekan <esc> untuk kembali ke mode teks.

Sebelum bekerja dengan mode grafik, haruslah disalin terlebih dahulu file "Graph.TPU" serta file-file :

- CGA.BGI ; bila menggunakan monitor CGA
- EGAVGA.BGI ; bila menggunakan monitor EGA atau VGA
- dengan tambahan .CHR ; yang berisi font.

## B. SISTEM KOORDINAT LAYAR MONITOR

Monitor jenis apapun dalam keadaan normal dapat memuat 80 buah huruf setiap barisnya, dan jumlah barisnya ada 25 buah ( 80 kolom dan 25 baris).

Layar monitor terdiri dari pixel-pixel (picture element), yaitu kotak-kotak kecil yang dibentuk oleh kolom dan baris. Jumlah kolom dan baris pada layar monitor tidak tetap, tergantung pada jenis monitornya. Macam-macam jumlah kolom dan baris untuk berbagai jenis monitor adalah sebagai berikut :

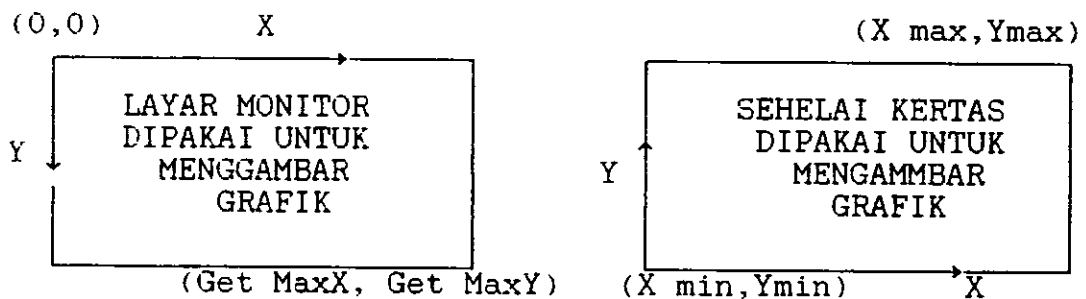
No.	Jenis Monitor	Jml.Kolom	Jml.Baris	Jumlah Pixel
1	CGA	320	200	320 X 200
2	EGA	640	480	640 X 480
3	VGA	640	480	640 X 480
		800	600	800 X 600
		1024	768	1024 X 768

Tabel 3. Jenis Monitor dan Jumlah Pixel

Catatan : Jumlah pixel = hasil kali jumlah kolom dan jumlah baris

Beragamnya jumlah pixel untuk tiap jenis monitor menimbulkan kesukaran bila layar monitor tersebut hendak digunakan untuk menggambar grafik. Misalnya posisi pixel di koordinat (319,199) untuk monitor CGA berbeda dengan posisi koordinat tersebut untuk monitor VGA, sistem koordinat yang demikian disebut koordinat absolut. Dianjurkan untuk menggambar grafik pada layar monitor gunakanlah koordinat relatif.

Untuk menggambar pada sehelai kertas, dan menggambar pada layar monitor terdapat perbedaan sistem koordinatnya seperti ditunjukkan pada gambar berikut ini :



Gambar 3.1 Perbedaan Sistem Koordinat Layar Monitor Dengan Koordinat Sehelai Kertas Jika Digunakan Untuk Menggambar Grafik

Keterangan : Dalam Turbo Pascal, Koordinat X maximum dan Y maximum (titik sudut kanan bawah layar monitor) ditulis (Get maxX, Get maxY)

### C. MEMBUAT VISUALISASI SEDERHANA

Didalam file Graph.TPU banyak terdapat program pembantu (sub program), diantaranya adalah :

#### 1. Program Pembantu Melukis Segi Empat (Rectangle)

Untuk melukis segi empat diperlukan 2 koordinat titik sudut yang berhadapan

Contoh : Rectangle (X1,Y1 , X2,Y2) ; artinya menggambar segi empat dengan titik sudut kiri atas (X1,Y1) dan titik sudut kanan bawah (X2,Y2)

#### 2. Program Pembantu Melukis Garis Lurus (Line)

Untuk melukis garis lurus, diperlukan koordinat titik kedua ujung garis tersebut.

Contoh : Line (X1,Y1 , X2,Y2) ; artinya menggambar sebuah garis lurus dari koordinat (X1,Y1) ke koordinat (X2,Y2)

#### 3. Program Pembantu Melukis Lingkaran (Circle)

Untuk melukis sebuah lingkaran, diperlukan informasi tentang koordinat titik pusat dan jari-jarinya.

Contoh : Circle (GetmaxX div 4, 3 \* GetmaxY div 4, GetmaxX div 6) ; artinya menggambar lingkaran dengan pusat dikoordinat (1/4 Xmax, 3/4 Ymax) dan jari-jarinya (1/6 Xmax)

#### 4. Program Pembantu Melukis Sebuah Titik (Putpixel)

Untuk melukis sebuah titik diperlukan informasi tentang koordinat titik tersebut dan warnanya.

Contoh : Putpixel (3 \* GetmaxX div 4, GetmaxY div 4, White) ; artinya menggambar sebuah titik yang berada di koordinat (3/4 Xmax, 1/4 Ymax) dan warnanya putih.

Berikut ini diberikan contoh membuat program sederhana dengan memakai Unit Grafiki dan Unit Graph.

```
Program Menggambar_Grafik1;
Uses
  Crt, Graph, Grafiki;
Begin
  Membuka_grafik ;

  Rectangle (0,0 , GetmaxX,GetmaxY);
  Line (0,0 , GetmaxX,GetmaxY);
  Circle (GetmaxX div 4,3 * GetmaxY div 4 , GetmaxX div 6);
  Putpixel (3 * GetmaxX div 4, GetmaxY div 4, White);

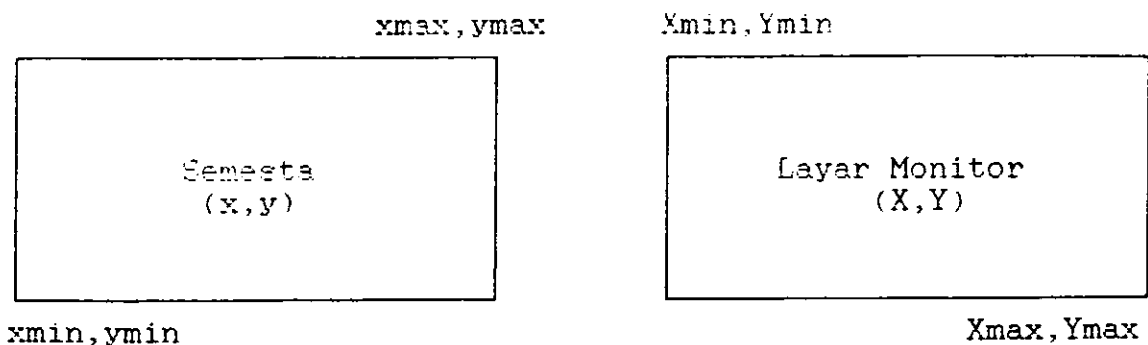
  Menutup_grafik
End.
```

#### D. TRANSFORMASI KOORDINAT

Adanya perbedaan sistem koordinat antara bidang gambar pada layar monitor dan bidang gambar pada sehelai kertas saat layar monitor berada pada "mode grafik", mengharuskan kita melakukan transformasi koordinat dari koordinat semesta ke koordinat layar monitor, jika kita hendak menggunakan layar monitor untuk menggambar grafik, sehingga kita tidak mengalami kesukaran dalam hal penskalaan. Transformasi yang mudah dilakukan adalah transformasi linier.

Misalkan koordinat suatu titik pada semesta adalah (x,y) dan pada layar monitor (X,Y), maka menurut transformasi linier dapat ditulis :

$$X = A + Bx \quad \text{dan} \quad Y = C + Dy$$



Gambar 3.2. Transformasi Koordinat Semesta ke Koordinat Layar Monitor

1. Untuk  $X = 0$  maka  $x = x_{min}$  dan  $X = GetmaxX$  maka  $x = x_{max}$  sehingga memenuhi persamaan-persamaan :

a.  $0 = A + Bx_{min}$  atau  $B = -A/(x_{min}) \dots\dots\dots(1)$

b.  $GetmaxX = A + Bx_{max}$  atau  $B = (GetmaxX-A)/x_{max} \dots(2)$

Persamaan (1) = Persamaan (2)

maka

$A = - (GetmaxX)(x_{min})/(x_{max} - x_{min}) \dots\dots\dots(3)$

$B = (GetmaxX)/(x_{max}-x_{min}) \dots\dots\dots(4)$

sehingga :

$X = (GetmaxX)(-x_{min} + x)/(x_{max}-x_{min}) \dots\dots\dots(5)$

2. Untuk  $Y = 0$  maka  $y = y_{max}$  dan  $Y = GetmaxY$  maka  $y = y_{min}$  sehingga memenuhi persamaan-persamaan :

a.  $0 = C + Dy_{max}$  atau  $D = -C/(y_{max}) \dots\dots\dots(6)$

b.  $GetmaxY = C + Dy_{min}$  atau  $D = (GetmaxY-C)/y_{min} \dots(7)$

Persamaan (6) = Persamaan (7)

maka

$C = (GetmaxY)(y_{max})/(y_{max} - y_{min}) \dots\dots\dots(8)$

$D = -(GetmaxY)/(y_{max}-y_{min}) \dots\dots\dots(9)$



sehingga :

$$X = (\text{GetmaxY})(y_{\text{max}} - y)/(y_{\text{max}} - y_{\text{min}}) \dots\dots(10)$$

Transformasi ini akan selalu dipakai jika daerah semesta hendak dipindahkan ke layar monitor, maka sebaiknya prosedur transformasi ini dimasukkan kedalam unit grafik1 yang sudah dibuat tadi, sehingga unit grafik1 menjadi sebagai berikut

Unit Grafik2;

Interface

Uses

Crt, Graph;

Procedure Membuka\_grafik;

Procedure Bidang\_gambar (Val x\_min,y\_min,x\_max,y\_max:Real);

Procedure Transformasi (x\_min,y\_min,x\_max,y\_max,  
x\_gambar,y\_gambar : Real;  
Var x\_layar,y\_layar :Real);

Procedure Menutup\_grafik;

Implementation

Procedure Membuka\_grafik;

Var

gd, gm, Errorcode : Integer;

Begin

gd := Detect;

Initgraph (gd,gm,"");

Errorcode := Graphresult;

If Errorcode <> Grok Then Halt;

End;

Procedure Bidang\_gambar (Val x\_min,y\_min,x\_max,y\_max:Real);

Begin

Writeln;Writeln;

Writeln ('Bidang gambar :');

Write ('Xmin :');Readln (x\_min);

Write ('Ymin :');Readln (y\_min);

Write ('Xmax :');Readln (x\_max);

Write ('Ymax :');Readln (x\_max);

End;

Procedure Transformasi (x\_min,y\_min,x\_max,y\_max,

x\_gambar,y\_gambar : Real;

Var x\_layar,y\_layar :Real);

Begin

x\_layar := Getmaxx/(x\_max - x\_min) \* (-x\_min + x\_gambar);

y\_layar := Getmaxy/(y\_max - y\_min) \* ( y\_max - y\_gambar);

End;

```

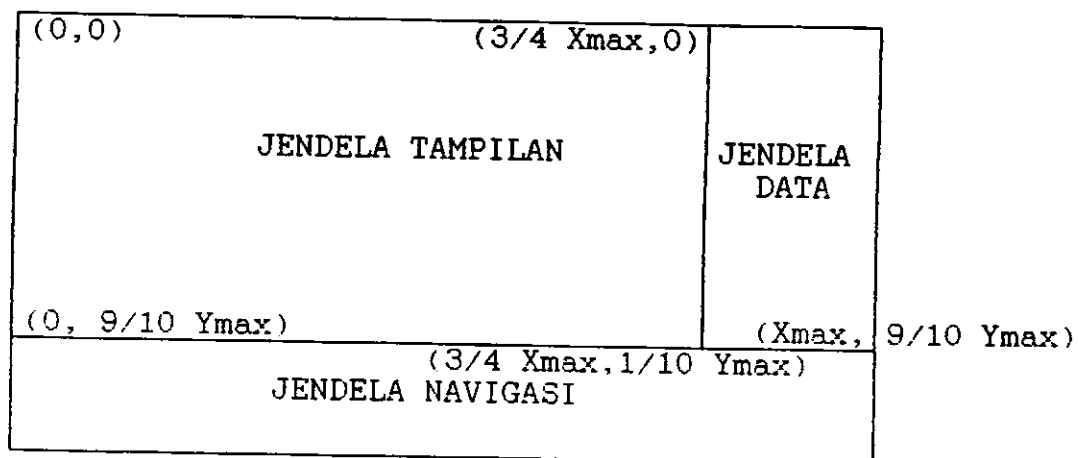
Procedure Menutup_grafik;
Begin
  Repeat until Readkey = #27;
  Closegraph;
End;
End.

```

### E. MEMBAGI LAYAR MONITOR MENJADI BEBERAPA JENDELA

Unit grafik2 yang ditampilkan pada bagian transformasi koordinat diatas memanfaatkan seluruh layar monitor sebagai jendela, sehingga layar tidak dapat lagi memuat informasi tentang data masukan dan langkah kerja yang akan dilakukan selanjutnya. Agar layar dapat difungsikan secara informatif, maka perlu dibagi atas beberapa ruang (jendela), misalnya layar dibagi menjadi 3 jendela, yaitu ; jendela gambar (tampilan, jendela data, dan jendela navigasi, yang besarnya masing-masing sesuai dengan selera dan kebutuhan.

Berikut ini diberikan contoh pembagian layar monitor menjadi beberapa jendela :



Gambar 3.3 Contoh Pembagian Layar Monitor Menjadi 3 Jendela

Karena jendela tampilan tidak sebesar layar monitor, maka fungsi transformasi ikut berubah. Untuk kondisi seperti di-

tunjukkan pada Gambar 6.3, fungsi transformasinya menjadi :

$$X = (3 * \text{GetmaxX} \text{ div } 4) (-x_{\min} + x) / (x_{\max} - x_{\min})$$

$$Y = (9 * \text{GetmaxY} \text{ div } 10) (y_{\max} - y) / (y_{\max} - y_{\min})$$

Jika model pembagian layar monitor seperti ini yang hendak dipakai pada program membuat grafik, maka haruslah dimasukkan kedalam unit grafik2 sehingga menjadi Unit Grafik3 sebagai berikut :

```
Unit Grafik3;
Interface
  Uses
    Crt, Graph;
  Procedure Membuka_grafik;
  Procedure Bidang_gambar (Val x_min,y_min,x_max,y_max:Real);
  Procedure Kerangka;
  Procedure Transformasi (x_min,y_min,x_max,y_max,
    x_gambar,y_gambar : Real;
    Var x_layar,y_layar :Real);
  Procedure Menutup_grafik;

Implementation
  Procedure Membuka_grafik;
  Var
    gd, gm, Errorcode : Integer;
  Begin
    gd := Detect;
    Initgraph (gd, gm, "");
    Errorcode := Graphresult;
    If Errorcode <> Grok Then Halt;
  End;

  Procedure Bidang_gambar (Val x_min,y_min,x_max,y_max:Real);
  Begin
    Writeln;Writeln;
    Writeln ('Bidang gambar :');
    Write ('Xmin  :');Readln (x_min);
    Write ('Ymin  :');Readln (y_min);
    Write ('Xmax  :');Readln (x_max);
    Write ('Ymax  :');Readln (y_max);
  End;

  Procedure Kerangka;
  Begin
    Rectangle (0,0,3*Getmaxx div 4,9*Getmaxy div 10);
    Rectangle (3*Getmaxx div 4,0,Getmaxx,9*Getmaxy div 10);
    Rectangle (0,9*Getmaxy div 10, Getmaxx, Getmaxy);

    Settextstyle (Smalfont, Horizdir,5);
    Settextjustify (Centertext, Centertext);
```

```

    Outtextxy (7*Getmaxx div 8, Getmaxy div 10, "DATA :");
End;

Procedure Transformasi (x_min,y_min,x_max,y_max,
    x_gambar,y_gambar : Real;
    Var x_layar,y_layar :Real);

Begin
    x_layar :=(3*Getmaxx div 4)/(x_max-x_min)*(-x_min+x_gambar);
    y_layar :=(9*Getmaxy div 10)/(y_max-y_min)*(y_max-y_gambar);
End;

Procedure Menutup_grafik;
Begin
    Setcolor (White);
    Settextstyle (Smallfont, Horizdir,5);
    Settextjustify (Centertext, Centertext);
    Outtextxy (Getmaxx div 2,19* Getmaxy div 20, 'Tekan<Esc>');
    Repeat until Readkey = #27;
    Closegraph;
End;
End.

```

Keterangan :

1. Setcolor (White); perintah menyiapkan warna putih untuk pesan yang muncul di jendela navigasi
2. Settextstyle (Smallfont, Horizdir, 5); perintah untuk gunakan huruf jenis small arahnya mendatar dan besarnya 5
3. Settextjustify (Centertext, Centertext); perintah untuk menuliskan pesan, yang diletakkan ditengah-tengah koordinat yang ditentukan.
4. Outtextxy (Getmaxx div 2, 19\*Getmaxy div 20, 'Tekan <esc>'); perintah untuk menuliskan pesan "Tekan <esc>" pada jarak 1/2 panjang layar dari sisi kiri dan 19/20 lebar layar dari sisi atas.

## F. SUMBU KOORDINAT

Sumbu koordinat banyak dipakai dalam visualisasi, sehingga perintah-perintah untuk melukis sumbu koordinat dapat dihimpun dalam suatu program pembantu pada unit grafik

Sumbu koordinat dibuat pada layar monitor harus sesuai dengan keadaan semesta, oleh sebab itu sebelum sumbu koordinat dibuat, koordinat batas-batas semesta harus diubah

kedalam koordinat layar monitor terlebih dahulu melalui program pembantu transformasi.

Program pembantu sumbu koordinat mempunyai 2 perintah yaitu :

1. perintah memanggil program pembantu transformasi yang ada dalam unit grafik
2. perintah untuk melukis garis lurus.

Unit grafik4 berikut ini sudah ditambah dengan program pembantu sumbu koordinat.

```
Unit Grafik4;
Interface
  Uses
    Crt, Graph;
  Procedure Membuka_grafik;
  Procedure Bidang_gambar (Var x_min,y_min,x_max,y_max:Real);
  Procedure Kerangka;
  Procedure Transformasi (x_min,y_min,x_max,y_max,
    x_gambar,y_gambar : Real;
    Var x_layar,y_layar :Real);
  Procedure Sumbu_koordinat (x_min,y_min,x_max,y_max:Real);
  Procedure Menutup_grafik;

Implementation
  Procedure Membuka_grafik;
  Var
    gd, gm, Errorcode : Integer;
  Begin
    gd := Detect;
    Initgraph (gd,gm,"");
    Errorcode := Graphresult;
    If Errorcode <> Grok Then Halt;
  End;

  Procedure Bidang_gambar (Var x_min,y_min,x_max,y_max:Real);
  Begin
    Writeln;Writeln;
    Writeln ('Bidang gambar :');
    Write ('Xmin :');Readln (x_min);
    Write ('Ymin :');Readln (y_min);
    Write ('Xmax :');Readln (x_max);
    Write ('Ymax :');Readln (x_max);
  End;

  Procedure Kerangka;
  Begin
```

```

Rectangle (0,0,3*Getmaxx div 4,9*Getmaxy div 10);
Rectangle (3*Getmaxx div 4,0,Getmaxx,9*Getmaxy div 10);
Rectangle (0,9*Getmaxy div 10, Getmaxx, Getmaxy);

Settextstyle (Smalfont, Horizdir,5);
Settextjustify (Centertext, Centertext);
Outtextxy (7*Getmaxx div 8, Getmaxy div 10, "DATA :");
End;

Procedure Transformasi (x_min,y_min,x_max,y_max,
                       x_gambar,y_gambar : Real;
                       Var x_layar,y_layar :Real);

Begin
  x_layar :=(3*Getmaxx div 4)/(x_max-x_min)*(-x_min+x_gambar);
  y_layar :=(9*Getmaxy div 10)/(y_max-y_min)*(y_max-y_gambar);
End;

Procedure Sumbu_koordinat (x_min,y_min,x_max,y_max:Real);
Var
  x1,y1,x2,y2:Real;
Begin
  Transformasi (x_min,y_min,x_max,y_max,x_min,0,x1,y1);
  Transformasi (x_min,y_min,x_max,y_max,x_max,0,x2,y2);
  Line (Trunc (x1),Trunc (y1), Trunc (x2), Trunc (y2));

  Transformasi (x_min,y_min,x_max,y_max,0,y_min,x1,y1);
  Transformasi (x_min,y_min,x_max,y_max,0,y_max,x2,y2);
  Line (Trunc (x1),Trunc (y1), Trunc (x2), Trunc (y2));
End;

Procedure Menutup_grafik;
Begin
  Setcolor (White);
  Settextstyle (Smalfont, Horizdir,5);
  Settextjustify (Centertext, Centertext);
  Outtextxy (Getmaxx div 2,19* Getmaxy div 20, 'Tekan<Esc>');
  Repeat until Readkey = #27;
  Closegraph;
End;
End.

```

## G. S K A L A

Untuk lebih sempurnanya visualisasi yang dibuat, perlu dilengkapi dengan "skala". Perintah pemberian skala ini sebaiknya ditempatkan pada program pembantu tersendiri, dan bila tidak diperlukan tidak usah dipanggil.

Sebaiknya skala ditampilkan sepanjang tepi kiri dan

tepi bawah dari jendela tampilan (gambar) sehingga jendela gambar jendela gambar menjadi agak sempit (mengecil) dari semula.

Misalnya koordinat-koordinat jendela menjadi sebagai berikut :

- a.  $X_{min} = \text{GetmaxY} \text{ div } 10$
- b.  $X_{max} = 3 * \text{GetmaxX} \text{ div } 4$
- c.  $Y_{min} = 0$
- d.  $Y_{max} = 8 * \text{GetmaxY} \text{ div } 10$

sehingga fungsi transformasinya menjadi :

$$X = [(X_{min} * x_{max} - X_{max} * x_{min}) + (X_{max} - X_{min})x] / (x_{max} - x_{min})$$

$$Y = [(Y_{max} * y_{min} - Y_{min} * y_{min}) - (Y_{max} - Y_{min})y] / (y_{max} - y_{min})$$

Perintah untuk pembuatan skala ini adalah dalam hal :

- a. melukis garis lurus sepanjang tepi kiri dan tepi bawah gambar
- b. melukis angka-angka skala sepanjang tepi kiri dan tepi bawah gambar.

Sebagai masukan adalah batas-batas semesta. Agar tidak merepotkan, maka antara  $x_{min}$  dan  $x_{max}$ ,  $y_{min}$  dan  $y_{max}$  dibagi menjadi 5 bagian. Program pembantu ini disimpan dalam unit grafik.

Berikut ini Unit Grafik4 yang telah mengalami perubahan yaitu pada program pembantu transformasinya ditambah dengan program pembantu skala, dan dinamakan Unit Grafik5.

```
Unit Grafik5;
Interface
  Uses
    Crt, Graph;
  Procedure Membuka_grafik;
  Procedure Bidang_gambar (Var x_min,y_min,x_max,y_max:Real);
  Procedure Kerangka;
  Procedure Transformasi (x_min,y_min,x_max,y_max,
    x_gambar,y_gambar : Real;
    Var x_layar,y_layar :Real);
```

```

Procedure Sumbu_koordinat (x_min,y_min,x_max,y_max:Real);
Procedure Skala (x_min,y_min,x_max,y_max:Real);
Procedure Menutup_grafik;

```

#### Implementation

```

Procedure Membuka_grafik;

```

```

Var
  gd, gm, Errorcode : Integer;
Begin
  gd := Detect;
  Initgraph (gd,gm,"");
  Errorcode := Graphresult;
  If Errorcode <> Grok Then Halt;
End;

```

```

Procedure Bidang_gambar (Var x_min,y_min,x_max,y_max:Real);

```

```

Begin
  Writeln;Writeln;
  Writeln ('Bidang gambar :');
  Write ('Xmin :');Readln (x_min);
  Write ('Ymin :');Readln (y_min);
  Write ('Xmax :');Readln (x_max);
  Write ('Ymax :');Readln (x_max);
End;

```

```

Procedure Kerangka;

```

```

Begin
  Rectangle (0,0,3*Getmaxx div 4,9*Getmaxy div 10);
  Rectangle (3*Getmaxx div 4,0,Getmaxx,9*Getmaxy div 10);
  Rectangle (0,9*Getmaxy div 10, Getmaxx, Getmaxy);

  Settextstyle (Smalfont, Horizdir,5);
  Settextjustify (Centertext, Centertext);
  Outtextxy (7*Getmaxx div 8, Getmaxy div 10, "DATA :");
End;

```

```

Procedure Transformasi (x_min,y_min,x_max,y_max,
  x_gambar,y_gambar : Real;
  Var x_layar,y_layar :Real);

```

```

Var

```

```

  xmin,ymin,xmax,ymax :Integer;

```

```

Begin

```

```

  xmin := Getmaxx div 10;
  xmax := 3*Getmaxx div 4;
  ymin := 0
  ymax := 8*Getmaxy div 10;

```

```

  x_layar := (xmin * x_max - xmax * x_min)/(x_max - xmin)
            + x_gambar * (xmax - xmin)/(x_max - x_min);
  y_layar := (ymax * y_max - ymin * y_min)/(y_max - ymin)
            - y_gambar * (ymax - ymin)/(y_max - y_min);

```

```

End;

```



```

Procedure Sumbu_koordinat (x_min,y_min,x_max,y_max:Real);
Var
  x1,y1,x2,y2:Real;
Begin
  Transformasi (x_min,y_min,x_max,y_max,x_min,0,x1,y1);
  Transformasi (x_min,y_min,x_max,y_max,x_max,0,x2,y2);
  Line (Trunc (x1),Trunc (y1), Trunc (x2), Trunc (y2));

  Transformasi (x_min,y_min,x_max,y_max,0,y_min,x1,y1);
  Transformasi (x_min,y_min,x_max,y_max,0,y_max,x2,y2);
  Line (Trunc (x1),Trunc (y1), Trunc (x2), Trunc (y2));
End;

```

```

Procedure Skala (x_min, y_min, x_max, y_max :Real);

```

```

Var
  del,del_1,x1,y1,x2,y2:Real;
  n : Integer;
  dtx : String[5];
Begin
  Transformasi (x_min,y_min,x_max,y_max,x_min,y_min,x1,y1);
  Transformasi (x_min,y_min,x_max,y_max,x_min,y_max,x2,y2);
  Line (Trunc (x1),Trunc (y1), Trunc (x2), Trunc (y2));

  Settextstyle (Smalfont, Horizdir,5);
  Settextjustify (Centertext, Centertext);

  del := (y_max - y_min)/5;
  del_1 := (y2 - y1)/5;
  For n := 1 To 6 Do
  Begin
    If n = 6 Then
      Outtextxy (Getmaxx div 20,Trunc (y1+del_1*4.75),'Y');
    Else
      Begin
        Str (y_min + del* (n - 1):5:2,dtx);
        Outtextxy (Getmaxx div 20,Trunc(y1+del_1*(n-1)),dtx);
        If n <> 1 Then
          Outtextxy (Getmaxx div 10,Trunc(y1+del_1*(n-1)),'-');
      End;
  End;

  Transformasi (x_min,y_min,x_max,y_max,x_max,y_min,x2,y2);
  Line (Trunc (x1),Trunc (y1), Trunc (x2), Trunc (y2));
  del := (x_max - x_min)/5;
  del_1 := (x2 - x1)/5;
  For n := 1 To 6 Do
  Begin
    If n = 6 Then
      Outtextxy (Trunc(x1 + del_1*4.75),17*Getmaxy div 20,'X');
    Else
      Begin
        Str (x_min + del* (n - 1):5:2,dtx);
        Outtextxy (Trunc(x1+del_1*(n-1)),17*Getmaxy div 20,dtx);
        If n <> 1 Then
          Outtextxy (Trunc(x1+del_1*(n-1)),8*Getmaxy div 10,'|');
      End;
  End;

```

```
End:
End:

Procedure Menutup_grafik:
Begin
  Setcolor (White);
  Settextstyle (Smalfont, Horizdir,5);
  Settextjustify (Centertext, Righttext);
  Outtextxy (Getmaxx div 2,19* Getmaxy div 20, 'Tekan<Esc>');
  Repeat until Readkey = #27;
  Closegraph;
End;
End.
```

## DAFTAR PUSTAKA

- Halliday D & Resnick (1987). Fisika Jilid I, (Terjemahan P. Silaban & Erwin Sucipto Edisi ke 3) Penerbit : Erlangga Jakarta.
- Soegeng (1992). Diktat Kuliah Fisika Komputasi, Bandung : ITE
- Soegeng (1993). Visualisasi Dalam Fisika ( Materi Kursus Singkat Fisika Komputasi di Univ. Bengkulu).
- Sumartono Prawirosusanto dkk (1989), Apresiasi Komputer, Penerbit : UPT Komputer Univ. Gajah Mada Yogyakarta
- Sutiono Gunadi FX, dkk (1990), Belajar Sendiri Turbo Pascal 5.5, Penerbit : Gramedia Jakarta.
- Zarlis & Syahyar (1993), Pemakaian Perangkat Lunak Komputer Dalam Fisika (Materi Kursus Singkat Fisika Komputasi di Univ. Bengkulu),

LAMPIRAN-LAMPIRAN  
CONTOH-CONTOH PROGRAM SEDERHANA

```

Program parabola:
Uses
  Crt;
Var
  t,X,Y,Vx,Vy,Vox,Voy,V:Real;
  Waktu_ke,n:integer;
  Pilihlah : char;
const
  Pilihan = 'pilih salah satu';
  Vo      = 100 ;
  g       = 10;
  alpha   = 45/57.3;
Begin
  Clrscr;
  Writeln (' Menentukan Posisi atau Kecepatan Partikel ');
  Writeln ('      Setiap Saat Dalam Lintasan Parabola      ');
  Writeln (' ===== ');
  Writeln;
  Writeln ('                2 ');
  Writeln ('Percep.Grafitasi = 10 m/s ');
  Writeln ('Kec. Awal       = 100 m/s ');
  Writeln ('Sudut elevasi   = 45 derajat');
  Writeln;
  Writeln ('Pilihlah A atau B');
  Writeln;
  Writeln ('A = Posisi Partikel (X dan Y) ');
  Writeln ('B = Kecepatan Partikel      ');
  Writeln;
  Write('pilihan : ');Readln (pilihlah);
  Writeln;
  Case pilihlah of
    'A':Begin
      Write('lama bergerak (dlm detik) : ');Readln(t);
      Vox := Vo*cos(alpha);
      Voy := Vo*sin(alpha);
      Vx  := Vox;
      Vy  := (Voy*t)-(g*t);
      X   := Vox*t;
      Y   := (Voy*t)-(0.5*g*sqr(t));
      V   := sqrt(sqr(Vx) +sqr(Vy));
      Writeln('diperoleh koordinat "X" (dlm meter) : ', X:10:2);
      Writeln('diperoleh koordinat "Y" (dlm meter) : ', Y:10:2);
    end;
    'B':Begin
      Write('lama bergerak (dlm detik) : ');Readln(t);
      Vox := Vo*cos(alpha);
      Voy := Vo*sin(alpha);
      Vx  := Vox;
      Vy  := (Voy*t)-(g*t);
      V   := sqrt(sqr(Vx) +sqr(Vy));
      Write ('kecepatan pada posisi "X,Y" (dlm m/s) : ', V:10:2);
    end;
  Else
    Begin
      Writeln ('Pilihan anda salah ! Ulangi lagi');
    end;
  End;
  Readln;
End.

```

Hasil eksekusi "Program Parabola" adalah sbk :

Menentukan Posisi atau Kecepatan Partikel  
Setiap Saat Dalam Lintasan Parabola

Percep. Gravitasi = 10 m/s<sup>2</sup>  
Keo. Awal = 100 m/s  
Sudut elevasi = 45 derajat

Pilihlah A atau B

A = Posisi Partikel (X dan Y)  
B = Kecepatan Partikel

pilihan : B

lama bergerak (dlm detik) : 5  
kecepatan pada posisi "X,Y" (dlm m/s) : 311.66

Program Konversi\_Suhu:

Uses

Crt:

Var

K,C,F,R,Suhu:real;

Jumlah,n:integer;

Begin

Cursor;

Begin

Writeln(' PROGRAM KONVERSI SUHU ');

Writeln:

Write('Jumlah Yang Dicari :'):Readln(n);

Writeln:

Writeln('Masukkan Nilai yang Anda Inginkan');

Writeln;

Writeln:

For jumlah :=1 to n do

Begin

Write ('Suhu Dalam Derajat Kelvin = ');Readln(K);

Writeln;

C:= (K-273);

Writeln('Suhu Dalam derajat Celcius adalah:'.C:6:2);

R:= 4/5 \*(K-273);

Writeln('Suhu Dalam Derajat Reamur adalah:'.R:6:2);

F:= (9/5\*(K-273))+32;

Writeln('Suhu Dalam Derajat Fahrenheit adalah:'.F:6:2);

Writeln;

End;

Readln;

End;

End.

Hasil eksekusi "Program Konversi Suhu" adalah sbb :

### PROGRAM KONVERSI SUHU

Jumlah Yang Dicari :2

Masukkan Nilai yang Anda Inginkan

Suhu Dalam Derajat Kelvin = 373

Suhu Dalam derajat Celcius adalah:100.00

Suhu Dalam Derajat Reamur adalah: 80.00

Suhu Dalam Derajat Fahrenheit adalah:212.00

Suhu Dalam Derajat Kelvin = 233

Suhu Dalam derajat Celcius adalah:-40.00

Suhu Dalam Derajat Reamur adalah:-32.00

Suhu Dalam Derajat Fahrenheit adalah:-40.00

Program Akar\_PK:

Uses Crt;

Var

A,B,C,D, X1,X2,im1,im2 : Real;

Begin

Clrscr;

Read (A,B,C);

D := SQR(B)-4\*A\*C;

If D=0 then

Begin

X1 := -B/(2\*A);

X2 := X1;

Writeln('X1 = X2 = ',X2:4:2);

Writeln('Merupakan akar-akar kembar');

end;

If D>0 then

Begin

X1 := (-B+SQRT(D))/(2\*A);

X2 := (-B-SQRT(D))/(2\*A);

Writeln ('X1 = ',X1:4:2);

Writeln ('X2 = ',X2:4:2);

Writeln ('merupakan akar-akar berlainan');

End;

If D<0 then

Begin

Im1 := -B/(2\*A);

Im2 := SQRT(ABS(D))/(2\*A);

Writeln ('X1 = ',Im1:4:2,'+',Im2:4:2,'i');

Writeln ('X2 = ',Im1:4:2,'-',Im2:4:2,'i');

Writeln ('merupakan akar akar kompleks');

End;

Readln;

End.

BUK UPT PERPUSTAKAAN  
1991/1992

Hasil eksekusi "Program Akar\_PK" adalah :

```
1
5
6
X1 = -2.00
X2 = -3.00
merupakan akar-akar berlainan
```

```
Program Suhu_dan_Olah_Raga{Pernyataan : IF.....THEN};
Uses
  Crt;
Var
  Suhu:Real;
  Jumlah,n : Integer;
Begin
  Clrscr;
  Writeln('PROGRAM MENENTUKAN MAKANAN YANG COCOK ');
  Writeln('      MENURUT SUHU LINGKUNGAN ');
  Writeln('      Oleh : ANONIM ');
  Writeln('=====');
  Writeln:
  Writeln('BAKSO           : Jika suhu > 15 C ');
  Writeln('NASI SOTO        : Jika suhu > 16 dan <= 20 C');
  Writeln('BAKMI            : Jika suhu > 21 dan <= 30 C');
  Writeln('ES POKAT         : Jika suhu > 31 dan <= 35 C');
  Writeln('ES TELLER        : Jika suhu <= 35 C ');
  Writeln;
  Write ('Jumlah Yang Dicari : ');Readln(n);
  Writeln;
  For Jumlah := 1 to n Do
  Begin
    Write('Jika ');
    Write ('Suhu Lingkungan (Celcius) : ');Readln (Suhu);
    Write ('Makanan yang baik adalah : ');
    IF Suhu > 15 Then
      Writeln ('bakso ');
    Else
      IF (suhu >16) and (suhu <= 20) THEN
        Writeln ('nasi soto ');
      Else
        IF (suhu >21) and (suhu <= 30) THEN
          Writeln ('bakmi ');
        Else
          IF (suhu >31) and (suhu <= 35) THEN
            Writeln ('es pokat ');
          Else
            IF suhu <= 35 THEN
              Writeln ('es teller');
            Writeln:
  End;
  Readln:
End
```



Hasil eksekusi "Program Suhu dan Olah Raga" adalah :

PROGRAM MENENTUKAN MAKANAN YANG COCOK  
MENURUT SUHU LINGKUNGAN  
Oleh : ANONIM

=====

BAKSO : Jika suhu > 15 C  
NASI SOTO : Jika suhu > 16 dan <= 20 C  
BAKMI : Jika suhu > 21 dan <= 30 C  
ES POKAT : Jika suhu > 31 dan <= 35 C  
ES TELLER : Jika suhu <= 35 C

Jumlah Yang Dicari : 5

Jika Suhu Lingkungan (Celcius) : 1  
Makanan yang baik adalah : es teller

Jika Suhu Lingkungan (Celcius) : 15  
Makanan yang baik adalah : es teller

Jika Suhu Lingkungan (Celcius) : 34  
Makanan yang baik adalah : bakso

Jika Suhu Lingkungan (Celcius) :

Program Hitung\_Nilai:

Uses

Ort:

Const

Jml\_Test = 2;

Max\_Mhs = 40;

Type

Data\_Mhs = Record

No : Integer;

Nama : String[20];

Nilai : Array[1..Jml\_Test] OF Real;

Rerata : Real;

End;

Array\_Mhs = Array [1..Max\_Mhs] OF Data\_Mhs;

Var

Jml : 1..Max\_Mhs;

i, j : Integer;

Total\_Nilai : Real;

Mhs : Array\_Mhs;

Temporer : Data\_Mhs;

```

Begin
  Clrscr;
  Write('Jumlah Data : ');Readln(Jml);Writeln;
  FOR i := 1 TO Jml DO
  Begin
    Write('No      : ');Readln(Mhs[i].No);
    Write('Nama    : ');Readln(Mhs[i].Nama);
    Writeln;
    Writeln('Nilai tes yang ke : ');
    Writeln;
    For j := 1 TO Jml_Test DO
    Begin
      Write(j:1, ' ');Readln(Mhs[i].Nilai[j]);
    End;
  Writeln;
  End;
  For i := 1 TO Max_Mhs Do
  Begin
    Total_Nilai := 0;
    For j := 1 To Jml_Test Do
      Total_Nilai := Total_Nilai + Mhs[i].Nilai[j];
      Mhs[i].Rerata := Total_Nilai / Jml_Test;
  End;
  For i := 1 To Jml - 1 Do
    For j := i + 1 To Jml Do
      [if Mhs[i].Rerata < Mhs[j].Rerata Then
      Begin
        Temporer := Mhs[i];
        Mhs[i] := Mhs[j];
        Mhs[j] := Temporer;
      End;
  Writeln;
  Writeln('=====');
  Writeln(' NO.URUT      N A M A          NILAI RATA-RATA ');
  Writeln('=====');
  For i := 1 To Jml Do
    Writeln(Mhs[i].No:6.Mhs[i].Nama:20.Mhs[i].Rerata:10:1);
  Writeln('=====');
  Readln;
End.

```

Hasil eksekusi "Program Hitung Nilai" adalah sbb :

No : 1  
Name : Badu

Nilai tes yang ke :

1. 65
2. 78

No : 2  
Nama : Amir

Nilai tes yang ke :

1. 56
2. 76

NO.URUT	N A M A	NILAI RATA-RATA
1	Badu	71.5
2	Amir	66.0

Program parabola:

Uses

Crt:

const

Vo = 100 :

g = 10;

alpha = 45/57.3:

Var

t,X,Y,V:Real;

Waktu\_ke,n:integer;

Function Posisi\_x(t:real):real;

Begin

X := Vo\*cos(alpha)\*t;

end;

Function Posisi\_y(t:real):Real;

Begin

Y := Vo\*sin(alpha)\*t-(0.5\*g\*sqr(t));

end;

Function kec\_xy(t:real):Real;

Begin

V := sqrt(sqr(Vo) +sqr(Vo\*sin(alpha)\*t-g\*t));

End;

```

Begin
  Clrscr;
  Writeln (' Menentukan Posisi dan Kecepatan Partikel ');
  Writeln ('      Setiap Saat Dalam Lintasan Parabola      ');
  Writeln ('=====');
  Writeln:
  Writeln ('                2 ');
  Writeln ('Percep.Grafitasi = 10 m/s ');
  Writeln ('Kec. Awal       = 100 m/s ');
  Writeln ('Sudut elevasi   = 45 derjat');
  Writeln:
  Write ('Jumlah Data : ');Readln(n);
  Waktu_ke := 0;
  Repeat
  Begin
    Writeln:
    Waktu_ke := waktu_ke+1;
    Write('lama bergerak (dlm detik) : ');Readln(t);
    X:=posisi_x(t);
    Y:=posisi_y(t);
    V:=kec_xy(t);
    Writeln('diperoleh koordinat "X" (dlm meter) : ', X:10:2);
    Writeln('diperoleh koordinat "Y" (dlm meter) : ', Y:10:2);
    Write ('kecepatan pada posisi "X.Y" (dlm m/s) : ', V:10:2);
    Writeln:
  End;
  Until Waktu_ke = n;
  Readln;
End.

```

Hasil eksekusi contoh Prog. Pembantu "Fungsi" diatas adalah sbb :

```

Menentukan Posisi dan Kecepatan Partikel
Setiap Saat Dalam Lintasan Parabola
=====

                2

Percep.Grafitasi = 10 m/s
Kec. Awal       = 100 m/s
Sudut elevasi   = 45 derjat

Jumlah Data : 2

lama bergerak (dlm detik) : 5
diperoleh koordinat "X" (dlm meter) : 1897894514300000.00
diperoleh koordinat "Y" (dlm meter) : 1897894556800000.00
kecepatan pada posisi "X.Y" (dlm m/s) : 1897894619200000.00

lama bergerak (dlm detik) : 12
diperoleh koordinat "X" (dlm meter) : 1897894514300000.00
diperoleh koordinat "Y" (dlm meter) : 1897894556800000.00
kecepatan pada posisi "X.Y" (dlm m/s) : 1897894619200000.00

```

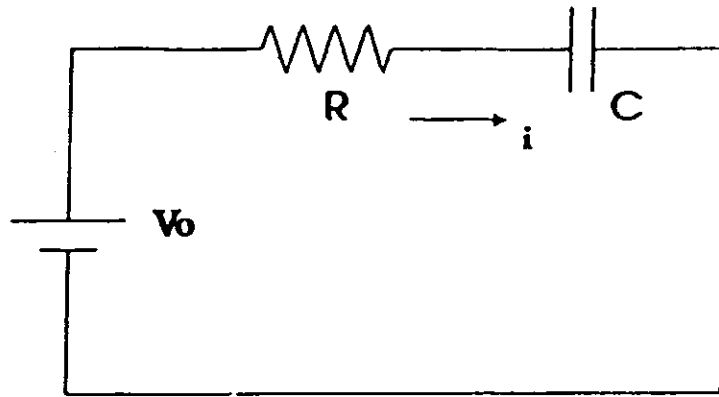
```

Program parabola:
Uses
  Crt;
const
  Vo      = 100 ;
  g       = 10;
  alpha   = 45/57.3:
Var
  t,X,Y,Vx,Vy,Vox,Voy,V:Real;
  Waktu_ke,n:integer;
Procedure Posis_kec(t:real;var x,y,vx,vy,vox,voy,v:real);
Begin
  Vox     := Vo*cos(alpha);
  Voy     := Vo*sin(alpha);
  Vx      := Vox;
  Vy      := (Voy*t)-(g*t);
  X       := Vox*t;
  Y       := (Voy*t)-(0.5*g*sqr(t));
  V       := sqrt(sqr(Vx) +sqr(Vy));
End;

Begin
  Clrscr;
  Writeln (' Menentukan Posisi dan Kecepatan Partikel ');
  Writeln ('   Setiap Saat Dalam Lintasan Parabola   ');
  Writeln ('=====');
  Writeln;
  Writeln ('                2 ');
  Writeln ('Percep.Grafitasi = 10 m/s ');
  Writeln ('Kec. Awal       = 100 m/s ');
  Writeln ('Sudut elevasi   = 45 derjat');
  Writeln;
  Write ('Jumlah Data : ');Readln(n);
  Waktu_ke := 0;
  Repeat
  Begin
    Writeln;
    Waktu_ke := waktu_ke+1;
    Write('lama bergerak (dlm detik) : ');Readln(t);
    Posis_kec(t,x,y,vx,vy,vox,voy,v);
    Writeln('diperoleh koordinat "X" (dlm meter) : ', X:10:2);
    Writeln('diperoleh koordinat "Y" (dlm meter) : ', Y:10:2);
    Write ('kecepatan pada posisi "X,Y" (dlm m/s) : ', V:10:2);
    Writeln;
  End;
  Until Waktu_ke = n;
  Readln;
End.

```

Hasil eksekusi contoh program pembantu "Prosedur" ini sama dengan hasil eksekusi contoh program pembantu "Fungsi"



## RANGKAIAN R-C SERI

Persamaan rangkaian

$$iR + \frac{q}{C} = V(t)$$

$$R \frac{dq}{dt} + \frac{q}{C} = V(t)$$

$$\frac{dq}{dt} = \frac{1}{R} \left[ V(t) - \frac{q}{C} \right]$$

Runge Kutta order 4

$$k_1 = \frac{\Delta t}{R} \left[ V(t) - \frac{q}{C} \right]$$

$$k_2 = \frac{\Delta t}{R} \left[ V(t + \frac{1}{2}\Delta t) - \frac{q + \frac{1}{2}k_1}{C} \right]$$

$$k_3 = \frac{\Delta t}{R} \left[ V(t + \frac{1}{2}\Delta t) - \frac{q + \frac{1}{2}k_2}{C} \right]$$

$$k_4 = \frac{\Delta t}{R} \left[ V(t + \Delta t) - \frac{q + k_3}{C} \right]$$

$$q_{n+1} = q_n + \frac{1}{6} [k_1 + 2k_2 + 2k_3 + k_4]$$

$$V_c(t) = \frac{q(t)}{C} \quad V_r(t) = V(t) - V_c(t)$$

## RANGKAIAN R-C

Program Rangkaian\_RC\_arus\_bolak\_balik\_sinusoida;

Uses

Crt, Graph, Grafik;

Const

sbx = 't (ms)';

sby = 'V (mV)';

x\_min = 't min';

y\_min = 'V min';

x\_max = 't max';

y\_max = 'V max';

Var

xmin, ymin, xmax, ymax,

xxmin, yymin, xxmax, yymax,

k1, k2, k3, k4,

x1, y1, t, i, v, vo, frek, farad, ohm, delt : Real;

dtx : String [10];

judul : String [70];

Procedure Garis (warna : Word; keterangan : String);

Begin

SetColor (Getmaxcolor - warna);

Line (3 \* Getmaxx div 4 + Getmaxx div 30,

Getmaxy div 10 + (9 + warna) \* Getmaxy div 30,

3 \* Getmaxx div 4 + 5 \* Getmaxx div 30,

Getmaxy div 10 + (9 + warna) \* Getmaxy div 30);

Outtextxy (3 \* Getmaxx div 4 + 6 \* Getmaxx div 30,

Getmaxy div 10 + (9 + warna) \* Getmaxy div 30, keterangan);

End;

Begin

Clrscr;

judul := 'RANGKAIAN RC SERI ARUS BOLAK-BALIK';

Gotoxy ((79 - Length (judul)) div 2, Wherey); Writeln (judul);

judul := 'Kurva potensial fungsi waktu untuk fungsi sinusoida';

Gotoxy ((79 - Length (judul)) div 2, Wherey); Writeln (judul);

Writeln;

Writeln ('Dalam SI.');

Write ('Kapasitansi kapasitor : '); Readln (farad);

Write ('Hambatan : '); Readln (ohm);

Write ('Potensial maksimum : '); Readln (vo);

Write ('Frekuensi : '); Readln (frek);

Bidang\_gambar (x\_min, y\_min, x\_max, y\_max,

xmin, ymin, xmax, ymax);

delt := (xmax - xmin) / 5000;

Membuka\_grafik;

Kerangka;

Sismbu\_koordinat (xmin, ymin, xmax, ymax);

xxmin := 1000 \* xmin;

xxmax := 1000 \* xmax;

yymin := 1000 \* ymin;

```
yymax := 1000 * ymax;
Skala (xmin, ymin, xmax, ymax, sbx, sby);
```

```
Settextstyle (Smallfont, Horizdir, 5);
Settextjustify (Lefttext, Centertext);
```

```
Str (farad:8, dtx);
Outtextxy (3 * Getmaxx div 4, Getmaxy div 10 + 2 * Getmaxy div 30,
  'C : ' + dtx);
```

```
Str (ohm:8, dtx);
Outtextxy (3 * Getmaxx div 4, Getmaxy div 10 + 3 * Getmaxy div 30,
  'R : ' + dtx);
```

```
Str (vo:8:2, dtx);
Outtextxy (3 * Getmaxx div 4, Getmaxy div 10 + 4 * Getmaxy div 30,
  'Vp : ' + dtx);
```

```
Str (frek:8:2, dtx);
Outtextxy (3 * Getmaxx div 4, Getmaxy div 10 + 5 * Getmaxy div 30,
  'f : ' + dtx);
```

```
Str (farad * ohm:8, dtx);
Outtextxy (3 * Getmaxx div 4, Getmaxy div 10 + 6 * Getmaxy div 30,
  'RC : ' + dtx);
```

```
Garis (1, 'Vr');
Garis (2, 'Vs');
Garis (3, 'Vc');
```

```
t := 0;
l := 0;
frek := 2 * pi * frek;
```

```
Tulis_pesan;
Hapus_pesan;
Tulis_tunggu;
Repeat
  If t >= xmin Then
  Begin
    Hapus_tunggu;
    v := vo * sin (frek * t) - l * ohm;
    If (v >= ymin) And (v <= ymax) Then
    Begin
      Transformasi (xmin, ymin, xmax, ymax, t, v, x1, y1);
      Putpixel (Trunc (x1), Trunc (y1), Getmaxcolor - 3);
    End;

    v := vo * sin (frek * t);
    If (v >= ymin) And (v <= ymax) Then
    Begin
      Transformasi (xmin, ymin, xmax, ymax, t, v, x1, y1);
      Putpixel (Trunc (x1), Trunc (y1), Getmaxcolor - 2);
    End;
```



```
v := i * ohm;
If (v >= ymin) And (v <= ymax) Then
Begin
  Transformasi (xmin, ymin, xmax, ymax, t, v, x1, y1);
  Putpixel (Trunc (x1), Trunc (y1), Getmaxcolor - 1);
End;
End;

k1 := delt * (frek * vo * cos (frek * t) - i / farad) / ohm;
k2 := delt * (frek * vo * cos (frek * (t + 0.5 * delt)) - (i + 0.5 * k1) / farad) / ohm;
k3 := delt * (frek * vo * cos (frek * (t + 0.5 * delt)) - (i + 0.5 * k2) / farad) / ohm;
k4 := delt * (frek * vo * cos (frek * (t + delt)) - (i + k3) / farad) / ohm;

i := i + (k1 + 2 * k2 + 2 * k3 + k4) / 6;

t := t + delt;
Until Keypressed Or (t > xmax);
Hapus_tunggu;

Menutup_grafik;
End.
```

```
Program Rangkaian_RC_urus_bolak_balik_kotak;
Uses
```

```
  Crt, Graph, Grafik;
```

```
Const
```

```
  sbx = 't (ms)';
```

```
  sby = 'V (mV)';
```

```
  x_min = 't min';
```

```
  y_min = 'V min';
```

```
  x_max = 't max';
```

```
  y_max = 'V max';
```

```
Var
```

```
  xmin, ymin, xmax, ymax,
```

```
  xxmin, yymin, xxmax, yymax,
```

```
  k1, k2, k3, k4,
```

```
  x1, y1, t, q, v, vr, vo, vc, frek,
```

```
  farad, ohm, delt : Real;
```

```
  j : Integer;
```

```
  dtx : String [10];
```

```
  judul : String [70];
```

```
Procedure Garis (warna : Word; keterangan : String);
```

```
Begin
```

```
  Setcolor (Getmaxcolor - warna);
```

```
  Line (3 * Getmaxx div 4 + Getmaxx div 30,
```

```
    Getmaxy div 10 + (9 + warna) * Getmaxy div 30,
```

```
    3 * Getmaxx div 4 + 5 * Getmaxx div 30,
```

```
    Getmaxy div 10 + (9 + warna) * Getmaxy div 30);
```

```
  Outtextxy (3 * Getmaxx div 4 + 6 * Getmaxx div 30,
```

```
    Getmaxy div 10 + (9 + warna) * Getmaxy div 30, keterangan);
```

```
End;
```

```
Begin
```

```
  Clrscr;
```

```
  judul := 'RANGKAIAN RC SERI ARUS BOLAK-BALIK';
```

```
  Gotoxy ((79 - Length (judul)) div 2, Wherey); Writeln (judul);
```

```
  judul := 'Kurva potensial fungsi waktu untuk fungsi kotak';
```

```
  Gotoxy ((79 - Length (judul)) div 2, Wherey); Writeln (judul);
```

```
  Writeln;
```

```
  Writeln ('Dalam SI.');
```

```
  Write ('Kapasitansi kapasitor : '); Readln (farad);
```

```
  Write ('Hambatan : '); Readln (ohm);
```

```
  Write ('Potensial maksimum : '); Readln (vo);
```

```
  Write ('Frekuensi : '); Readln (frek);
```

```
  Bidang_gambar (x_min, y_min, x_max, y_max,
```

```
    xmin, ymin, xmax, ymax);
```

```
  delt := (xmax - xmin) / 5000;
```

```
  Membuka_grafik;
```

```
  Kerangka;
```

```
  Sumbu_koordinat (xmin, ymin, xmax, ymax);
```

```
  xxmin := 1000 * xmin;
```

```

xxmax := 1000 * xmax;
yymin := 1000 * ymin;
ymax := 1000 * ymax;
Skala (xxmin, yymin, xxmax, ymax, sbx, sby);

Settextstyle (Smallfont, Horizdir, 5);
Settextjustify (Lefttext, Centertext);

Str (farad:10, dtx);
Outtextxy (3 * Getmaxx div 4, Getmaxy div 10 + 2 * Getmaxy div 30,
  'C : ' + dtx);

Str (ohm:10, dtx);
Outtextxy (3 * Getmaxx div 4, Getmaxy div 10 + 3 * Getmaxy div 30,
  'R : ' + dtx);

Str (vo:8:2, dtx);
Outtextxy (3 * Getmaxx div 4, Getmaxy div 10 + 4 * Getmaxy div 30,
  'Vp : ' + dtx);

Str (frek.o.2, dtx);
Outtextxy (3 * Getmaxx div 4, Getmaxy div 10 + 5 * Getmaxy div 30,
  'f : ' + dtx);

Str (farad * ohm:8, dtx);
Outtextxy (3 * Getmaxx div 4, Getmaxy div 10 + 6 * Getmaxy div 30,
  'RC : ' + dtx);

Garis (1, 'Vr');
Garis (2, 'Vs');
Garis (3, 'Vc');

t := 0;
q := 0;
vc := 0;
frek := 2 * pi * frek;

Tulis_pesan;
Hapus_pesan;
Tulis_tunggu;
Repeat
  If sin (frek * t) >= 0 Then v := vo
  Else v := -vo;

  If t >= xmin Then
  Begin
    Hapus_tunggu;

    If (vc >= ymin) And (vc <= ymax) Then
    Begin
      Transformasi (xmin, ymin, xmax, ymax, t, vc, x1, y1);
      Putpixel (Trunc (x1), Trunc (y1), Getmaxcolor - 3);
    End;

    Transformasi (xmin, ymin, xmax, ymax, t, v, x1, y1);

```

```
Putpixel (Trunc (x1), Trunc (y1), Getmaxcolor - 2);

If (vr >= ymin) And (vr <= ymax) Then
Begin
  Transformasi (xmin, ymin, xmax, ymax, t, vr, x1, y1);
  Putpixel (Trunc (x1), Trunc (y1), Getmaxcolor - 1);
End;
End;

k1 := delt * (v - q / farad) / ohm;
k2 := delt * (v - (q + 0.5 * k1) / farad) / ohm;
k3 := delt * (v - (q + 0.5 * k2) / farad) / ohm;
k4 := delt * (v - (q + k3) / farad) / ohm;

q := q + (k1 + 2 * k2 + 2 * k3 + k4) / 6;
vc := q / farad;
vr := v - vc;

t := t + delt;
Until Keypressed Or (t > xmax);
Hapus_tunggu;

Menutup_grafik;
End.
```

Program Rangkaian\_RC\_ arus\_searah;

Uses

Crt, Graph, Grafik;

Const

sbx = 't (ms)';

sby = 'V (mV)';

x\_min = 't min';

y\_min = 'V min';

x\_max = 't max';

y\_max = 'V max';

Var

xmin, ymin, xmax, ymax,

xxmin, yymin, xxmax, yymax,

k1, k2, k3, k4,

x1, y1, t, q, v, vr, vo, vc,

farad, ohm, delt : Real;

] : Integer;

dtx : String [10];

judul : String [70];

Procedure Garis (warna : Word; keterangan : String);

Begin

SetColor (Getmaxcolor - warna);

Line (3 \* Getmaxx div 4 + Getmaxx div 30,

Getmaxy div 10 + (9 + warna) \* Getmaxy div 30,

3 \* Getmaxx div 4 + 5 \* Getmaxx div 30,

Getmaxy div 10 + (9 + warna) \* Getmaxy div 30);

Outtextxy (3 \* Getmaxx div 4 + 6 \* Getmaxx div 30,

Getmaxy div 10 + (9 + warna) \* Getmaxy div 30, keterangan);

End;

Begin

Clrscr;

judul := 'RANGKAIAN RC SERI ARUS SEARAH';

Gotoxy ((79 - Length (judul)) div 2, Wherey); Writeln (judul);

judul := 'Kurva potensial fungsi waktu';

Gotoxy ((79 - Length (judul)) div 2, Wherey); Writeln (judul);

Writeln;

Writeln ('Dalam SI.');

Write ('Kapasitansi kapasitor : '); Readln (farad);

Write ('Hambatan : '); Readln (ohm);

Write ('Potensial maksimum : '); Readln (vo);

Bidang\_gambar (x\_min, y\_min, x\_max, y\_max,

xmin, ymin, xmax, ymax);

delt := (xmax - xmin) / 5000;

Membuka\_grafik;

Kerangka;

Sumbu\_koordinat (xmin, ymin, xmax, ymax);

xxmin := 1000 \* xmin;

xxmax := 1000 \* xmax;

```

yymin := 1000 * ymin;
yymax := 1000 * ymax;
Skala (xmin, yymin, xmax, yymax, sbx, sby);

```

```

Settextstyle (Smallfont, Horizdir, 5);
Settextjustify (Lefttext, Centertext);

```

```

Str (farad:10, dtx);
Outtextxy (3 * Getmaxx div 4, Getmaxy div 10 + 2 * Getmaxy div 30,
  ' C : ' + dtx);

```

```

Str (ohm:10, dtx);
Outtextxy (3 * Getmaxx div 4, Getmaxy div 10 + 3 * Getmaxy div 30,
  ' R : ' + dtx);

```

```

Str (vo:8:2, dtx);
Outtextxy (3 * Getmaxx div 4, Getmaxy div 10 + 4 * Getmaxy div 30,
  ' Vp : ' + dtx);

```

```

Garis (1, 'Vr');
Garis (2, 'Vs');
Garis (3, 'Vc');

```

```

t := 0;
q := 0;
vc := 0;

```

```

Tulis_pesan;
Hapus_pesan;
Tulis_tunggu;
Repeat
  v := vo;

```

```

If t >= xmin Then
  Begin
    Hapus_tunggu;

```

```

  If (vc >= ymin) And (vc <= ymax) Then
    Begin
      Transformasi (xmin, ymin, xmax, ymax, t, vc, x1, y1);
      Putpixel (Trunc (x1), Trunc (y1), Getmaxcolor - 3);
    End;

```

```

  Transformasi (xmin, ymin, xmax, ymax, t, v, x1, y1);
  Putpixel (Trunc (x1), Trunc (y1), Getmaxcolor - 2);

```

```

  If (vr >= ymin) And (vr <= ymax) Then
    Begin
      Transformasi (xmin, ymin, xmax, ymax, t, vr, x1, y1);
      Putpixel (Trunc (x1), Trunc (y1), Getmaxcolor - 1);
    End;
  End;

```

```

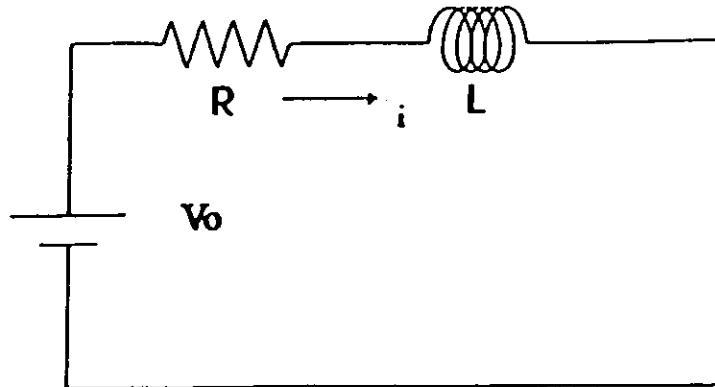
k1 := delt * (v - q / farad) / ohm;
k2 := delt * (v - (q + 0.5 * k1) / farad) / ohm;

```

## RANGKAIAN R-C

---

```
k3 := delt * (v - (q + 0.5 * k2) / farad) / ohm;  
k4 := delt * (v - (q + k3) / farad) / ohm;  
  
q := q + (k1 + 2 * k2 + 2 * k3 + k4) / 6;  
vc := q / farad;  
vr := v - vc;  
  
t := t + delt;  
Until Keypressed Or (t > xmax);  
Hapus_tunggu;  
  
Menutup_grafik;  
End.
```



## RANGKAIAN R-L SERI

Persamaan rangkaian

$$iR + L \frac{di}{dt} = V(t)$$

$$\frac{di}{dt} = \frac{1}{L} [V(t) - Ri]$$

Runge Kutta order 4

$$k_1 = \frac{\Delta t}{L} [V(t) - Ri]$$

$$k_2 = \frac{\Delta t}{L} [V(t + \frac{1}{2}\Delta t) - R(i + \frac{1}{2}k_1)]$$

$$k_3 = \frac{\Delta t}{L} [V(t + \frac{1}{2}\Delta t) - R(i + \frac{1}{2}k_2)]$$

$$k_4 = \frac{\Delta t}{L} [V(t + \Delta t) - R(i + k_3)]$$

$$i_{n+1} = i_n + \frac{1}{6} [k_1 + 2k_2 + 2k_3 + k_4]$$

$$V_r(t) = i(t)R$$

$$V_l(t) = V(t) - V_r(t)$$



```

{SN + }
Program Rangkaian_RL_ arus_bolak_balik_sinusoida;
Uses
  Crt, Graph, Grafik;
Const
  sbx = 't (ms)';
  sby = 'V (mV)';
  x_min = 't min';
  y_min = 'V min';
  x_max = 't max';
  y_max = 'V max';
Var
  xmin, ymin, xmax, ymax,
  xxmin, yymin, xxmax, yymin,
  k1, k2, k3, k4,
  l, v, vo, vr, vl, x1, y1, t, frek,
  henry, ohm, delt : Real;
  dtx : String [10];
  judul : String [70];

Procedure Garis (warna : Word; keterangan : String);
Begin
  Setcolor (Getmaxcolor - warna);
  Line (3 * Getmaxx div 4 + Getmaxx div 30,
    Getmaxy div 10 + (9 + warna) * Getmaxy div 30,
    3 * Getmaxx div 4 + 5 * Getmaxx div 30,
    Getmaxy div 10 + (9 + warna) * Getmaxy div 30);
  Outtextxy (3 * Getmaxx div 4 + 6 * Getmaxx div 30,
    Getmaxy div 10 + (9 + warna) * Getmaxy div 30, keterangan);
End;

Begin
  Clrscr;
  judul := 'RANGKAIAN RL SERI ARUS BOLAK-BALIK';
  Gotoxy ((79 - Length (judul)) div 2, Wherey); WriteLn (judul);
  judul := 'Kurva potensial fungsi waktu untuk fungsi sinusoida';
  Gotoxy ((79 - Length (judul)) div 2, Wherey); WriteLn (judul);

  WriteLn;
  WriteLn ('Dalam SI. ');
  Write ('Induktansi induktor : '); ReadLn (henry);
  Write ('Hambatan : '); ReadLn (ohm);
  Write ('Potensial maksimum : '); ReadLn (vo);
  Write ('Frekuensi : '); ReadLn (frek);

  Bidang_gambar (x_min, y_min, x_max, y_max,
    xmin, ymin, xmax, ymax);
  delt := (xmax - xmin) / 5000;

  Membuka_grafik;
  Kerangka;
  Sumbu_koordinat (xmin, ymin, xmax, ymax);

  xxmin := 1000 * xmin;
  xxmax := 1000 * xmax;

```

```
yymin := 1000 * ymin;
yymax := 1000 * ymax;
Skala (xxmin, yymin, xmax, ymax, sbx, sby);
```

```
Settextstyle (Smallfont, Horizdir, 5);
Settextjustify (Lefttext, Centertext);
```

```
Str (henry:8, dtx);
Outtextxy (3 * Getmaxx div 4, Getmaxy div 10 + 2 * Getmaxy div 30,
  'L : ' + dtx);
```

```
Str (ohm:8, dtx);
Outtextxy (3 * Getmaxx div 4, Getmaxy div 10 + 3 * Getmaxy div 30,
  'R : ' + dtx);
```

```
Str (vo:8:2, dtx);
Outtextxy (3 * Getmaxx div 4, Getmaxy div 10 + 4 * Getmaxy div 30,
  'Vp : ' + dtx);
```

```
Str (frek:8:2, dtx);
Outtextxy (3 * Getmaxx div 4, Getmaxy div 10 + 5 * Getmaxy div 30,
  'f : ' + dtx);
```

```
Garis (1, 'Vr');
Garis (2, 'Vs');
Garis (3, 'Vl');
```

```
t := 0;
l := 0;
frek := 2 * pi * frek;
```

```
Tulis_pesan;
Hapus_pesan;
Tulis_tunggu;
Repeat
  v := vo * sin (frek * t);
```

```
If t >= xmin Then
  Begin
    Hapus_tunggu;
```

```
If (vr >= ymin) And (vr <= ymax) Then
  Begin
    Transformasi (xmin, ymin, xmax, ymax, t, vr, x1, y1);
    Putpixel (Trunc (x1), Trunc (y1), Getmaxcolor - 1);
  End;
```

```
If (v >= ymin) And (v <= ymax) Then
  Begin
    Transformasi (xmin, ymin, xmax, ymax, t, v, x1, y1);
    Putpixel (Trunc (x1), Trunc (y1), Getmaxcolor - 2);
  End;
```

```
If (vl >= ymin) And (vl <= ymax) Then
  Begin
```

---

```

Transformasi (xmin, ymin, xmax, ymax, t, vl, x1, y1);
Putpixel (Trunc (x1), Trunc (y1), Getmaxcolor - 3);
End;
End;

k1 := delt * (vo * sin (frek * t) - i * ohm) / henry;
k2 := delt * (vo * sin (frek * (t + 0.5 * delt)) - (i + 0.5 * k1) * ohm) / henry;
k3 := delt * (vo * sin (frek * (t + 0.5 * delt)) - (i + 0.5 * k2) * ohm) / henry;
k4 := delt * (vo * sin (frek * (t + delt)) - (i + k3) * ohm) / henry;

i := i + (k1 + 2 * k2 + 2 * k3 + k4) / 6;

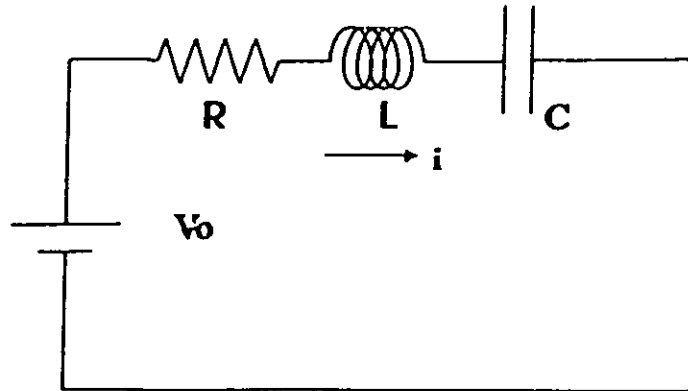
vr := i * ohm;
vl := v - vr;

t := t + delt;
Until Keypressed Or (t > xmax);
Hapus_tunggu;

Menutup_grafik;
End.

```

## RANGKAIAN R-L-C



## RANGKAIAN R-L-C SERI

Persamaan rangkaian

$$iR + L \frac{di}{dt} + \frac{q}{C} = V(t)$$

$$L \frac{d^2q}{dt^2} + R \frac{dq}{dt} + \frac{q}{C} = V(t) \quad \rightarrow \quad \frac{dq}{dt} = i$$

$$\frac{di}{dt} = \frac{1}{L} \left[ V(t) - Ri - \frac{q}{C} \right]$$

Runge Kutta order 4

$$k_{1i} = \frac{\Delta t}{L} \left[ V(t) - Ri - \frac{q}{C} \right]$$

$$k_{2i} = \frac{\Delta t}{L} \left[ V(t + \frac{1}{2}\Delta t) - R(i + \frac{1}{2}k_{1i}) - \frac{q + \frac{1}{2}k_{1i}q}{C} \right]$$

$$k_{3i} = \frac{\Delta t}{L} \left[ V(t + \frac{1}{2}\Delta t) - R(i + \frac{1}{2}k_{2i}) - \frac{q + \frac{1}{2}k_{2i}q}{C} \right]$$

$$k_{4i} = \frac{\Delta t}{L} \left[ V(t + \Delta t) - R(i + k_{3i}) - \frac{q + k_{3i}q}{C} \right]$$

$$i_{n+1} = i_n + \frac{1}{6} \left[ k_{1i} + 2k_{2i} + 2k_{3i} + k_{4i} \right]$$

$$k_{1q} = \Delta t i$$

$$k_{2q} = \Delta t (i + \frac{1}{2} k_{1q})$$

$$k_{3q} = \Delta t (i + \frac{1}{2} k_{2q})$$

$$k_{4q} = \Delta t (i + k_{3q})$$

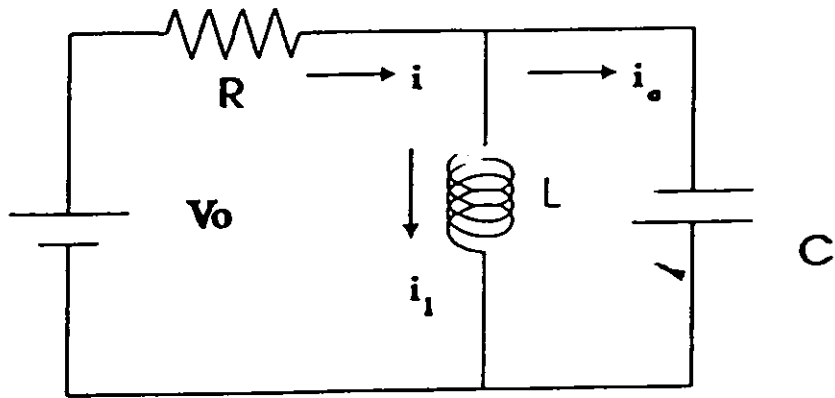
$$q_{n+1} = q_n + \frac{1}{6} [k_{1q} + 2k_{2q} + 2k_{3q} + k_{4q}]$$

$$V_r(t) = i(t)R$$

$$V_c(t) = \frac{q(t)}{C}$$

$$V_l(t) = V(t) - V_r(t) - V_c(t)$$

RANGKAIAN R-L-C



RANGKAIAN R-L-C PARALEL

Persamaan rangkaian

$$i_r R + L \frac{di_l}{dt} = V(t)$$

$$i_r R + \frac{q}{C} = V(t) \quad \leftarrow \quad i_r = i_l + i_c \quad \leftarrow \quad i_c = \frac{dq}{dt}$$

$$i_l R + R \frac{dq}{dt} + \frac{q}{C} = V(t) \quad \rightarrow \quad \frac{dq}{dt} = \frac{1}{R} \left[ V(t) - R i_l - \frac{q}{C} \right]$$

Runge Kutta order 4

$$k_{1q} = \frac{\Delta t}{R} \left[ V(t) - R i_l - \frac{q}{C} \right]$$

$$k_{2q} = \frac{\Delta t}{R} \left[ V(t + \frac{1}{2}\Delta t) - R(i_l + \frac{1}{2}k_{1i}) - \frac{q + \frac{1}{2}k_{1q}}{C} \right]$$

$$k_{3q} = \frac{\Delta t}{R} \left[ V(t + \frac{1}{2}\Delta t) - R(i_l + \frac{1}{2}k_{2i}) - \frac{q + \frac{1}{2}k_{2q}}{C} \right]$$

$$k_{4q} = \frac{\Delta t}{R} \left[ V(t + \Delta t) - R(i_l + k_{3i}) - \frac{q + k_{3q}}{C} \right]$$

$$q_{n+1} = q_n + \frac{1}{6} \left[ k_{1q} + 2k_{2q} + 2k_{3q} + k_{4q} \right]$$

$$i_r R + L \frac{di_l}{dt} = V(t)$$

$$(i_l + i_c)R + L \frac{di_l}{dt} = V(t)$$

$$\left(i_l + \frac{dq}{dt}\right)R + L \frac{di_l}{dt} = V(t)$$

$$\begin{aligned} \frac{di_l}{dt} &= \frac{1}{L} \left[ V(t) - Ri_l - R \frac{dq}{dt} \right] = \frac{1}{L} \left[ V(t) - Ri_l - \left( V(t) - Ri_l - \frac{q}{C} \right) \right] \\ &= \frac{q}{LC} \end{aligned}$$

Runge Kutta order 4

$$k_{1i} = \frac{\Delta t}{LC} q$$

$$k_{2i} = \frac{\Delta t}{LC} (q + \frac{1}{2} k_{1q})$$

$$k_{3i} = \frac{\Delta t}{LC} (q + \frac{1}{2} k_{2q})$$

$$k_{4i} = \frac{\Delta t}{LC} (q + k_{3q})$$

$$i_{n+1} = i_n + \frac{1}{6} [k_{1i} + 2k_{2i} + 2k_{3i} + k_{4i}]$$

Program Rangkaian\_RLC\_arus\_bolak\_balik\_sinusoida:

Uses

Crt, Graph, Grafik;

Const

sbx = 't (ms)';  
sby = 'V (mV)';  
x\_min = 't min';  
y\_min = 'V min';  
x\_max = 't max';  
y\_max = 'V max';

Var

xmin, ymin, xmax, ymax,  
xxmin, yymin, xxmax, yymax,  
k1l, k2l, k3l, k4l, k1q, k2q, k3q, k4q,  
q, l, v, vo, vr, vc, vl, x1, y1, t, frek,  
henry, farad, ohm, d3lt : Real;  
dtx : String [10];  
judul : String [70];

Procedure Garis (warna : Word; keterangan : String):

Begin

SetColor (Getmaxcolor - warna);

Line (3 \* Getmaxx div 4 + Getmaxx div 30,  
Getmaxy div 10 + (9 + warna) \* Getmaxy div 30,  
3 \* Getmaxx div 4 + 5 \* Getmaxx div 30,  
Getmaxy div 10 + (9 + warna) \* Getmaxy div 30);

Outtextxy (3 \* Getmaxx div 4 + 6 \* Getmaxx div 30,  
Getmaxy div 10 + (9 + warna) \* Getmaxy div 30, keterangan);

End;

Begin

Clsrscr;

judul := 'RANGKAIAN RLC SERI ARUS BOLAK-BALIK';

Gotoxy ((79 - Length (judul)) div 2, Wherey); Writeln (judul);

judul := 'Kurva potensial fungsi waktu untuk fungsi sinusoida';

Gotoxy ((79 - Length (judul)) div 2, Wherey); Writeln (judul);

Writeln;

Writeln ('Dalam SI.');

Write ('Kapasitansi kapasitor : '); Readln (farad);

Write ('Induktansi induktor : '); Readln (henry);

Write ('Hambatan : '); Readln (ohm);

Write ('Potensial maksimum : '); Readln (vo);

Write ('Frekuensi : '); Readln (frek);

Bidang\_gambar (x\_min, y\_min, x\_max, y\_max,  
xmin, ymin, xmax, ymax);

delt := (xmax - xmin) / 5000;

Membuka\_grafik;

Kerangka;

Sumbu\_koordinat (xmin, ymin, xmax, ymax);

xxmin := 1000 \* xmin;



```
xxmax := 1000 * xmax;
yymin := 1000 * ymin;
yymax := 1000 * ymax;
Skala (xxmin, yymin, xxmax, ymax, sbx, sby);
```

```
Settextstyle (Smallfont, Horizdir, 5);
Settextjustify (Lefttext, Centertext);
```

```
Str (farad:8, dtx);
Outtextxy (3 * Getmaxx div 4, Getmaxy div 10 + 2 * Getmaxy div 30,
  ' C : ' + dtx);
```

```
Str (henry:8, dtx);
Outtextxy (3 * Getmaxx div 4, Getmaxy div 10 + 3 * Getmaxy div 30,
  ' L : ' + dtx);
```

```
Str (ohm:8, dtx);
Outtextxy (3 * Getmaxx div 4, Getmaxy div 10 + 4 * Getmaxy div 30,
  ' R : ' + dtx);
```

```
Str (vo:8:2, dtx);
Outtextxy (3 * Getmaxx div 4, Getmaxy div 10 + 5 * Getmaxy div 30,
  ' Vp : ' + dtx);
```

```
Str (frek:8:2, dtx);
Outtextxy (3 * Getmaxx div 4, Getmaxy div 10 + 6 * Getmaxy div 30,
  ' f : ' + dtx);
```

```
Garis (1, 'Vr');
Garis (2, 'Vs');
Garis (3, 'Vc');
Garis (4, 'Vl');
```

```
t := 0;
i := 0;
q := 0;
frek := 2 * pi * frek;
```

```
Tulis_pesan;
Hapus_pesan;
Tulis_tunggu;
Repeat
  v := vo * sin (frek * t);
```

```
If t >= xmin Then
  Begin
    Hapus_tunggu;
```

```
If (vr >= ymin) And (vr <= ymax) Then
  Begin
    Transformasi (xmin, ymin, xmax, ymax, t, vr, x1, y1);
    Putpixel (Trunc (x1), Trunc (y1), Getmaxcolor - 1);
  End;
```

```
If (v >= ymin) And (v <= ymax) Then
```

```

Begin
  Transformasi (xmin, ymin, xmax, ymax, t, v, x1, y1);
  Putpixel (Trunc (x1), Trunc (y1), Getmaxcolor - 2);
End;

If (vc >= ymin) And (vc <= ymax) Then
Begin
  Transformasi (xmin, ymin, xmax, ymax, t, vc, x1, y1);
  Putpixel (Trunc (x1), Trunc (y1), Getmaxcolor - 3);
End;

If (vl >= ymin) And (vl <= ymax) Then
Begin
  Transformasi (xmin, ymin, xmax, ymax, t, vl, x1, y1);
  Putpixel (Trunc (x1), Trunc (y1), Getmaxcolor - 4);
End;
End;

k1q := delt * i;
k1l := delt * (vo * sin (fрек * t) - ohm * i - q / farad) / henry;

k2q := delt * (i + 0.5 * k1l);
k2l := delt * (vo * sin (fрек * (t + 0.5 * delt)) - ohm * (i + 0.5 * k1l)
  - (q + 0.5 * k1q) / farad) / henry;

k3q := delt * (i + 0.5 * k2l);
k3l := delt * (vo * sin (fрек * (t + 0.5 * delt)) - ohm * (i + 0.5 * k2l)
  - (q + 0.5 * k2q) / farad) / henry;

k4q := delt * (i + k3l);
k4l := delt * (vo * sin (fрек * (t + delt)) - ohm * (i + k3l) - (q + k3q) / farad) / henry;

i := i + (k1l + 2 * k2l + 2 * k3l + k4l) / 6;
q := q + (k1q + 2 * k2q + 2 * k3q + k4q) / 6;

vr := i * ohm;
vc := q / farad;
vl := henry * (k1l + 2 * k2l + 2 * k3l + k4l) / (6 * delt);

t := t + delt;
Until Keypressed Or (t > xmax);
Hapus_tunggu;

Menutup_grafik;
End.

```