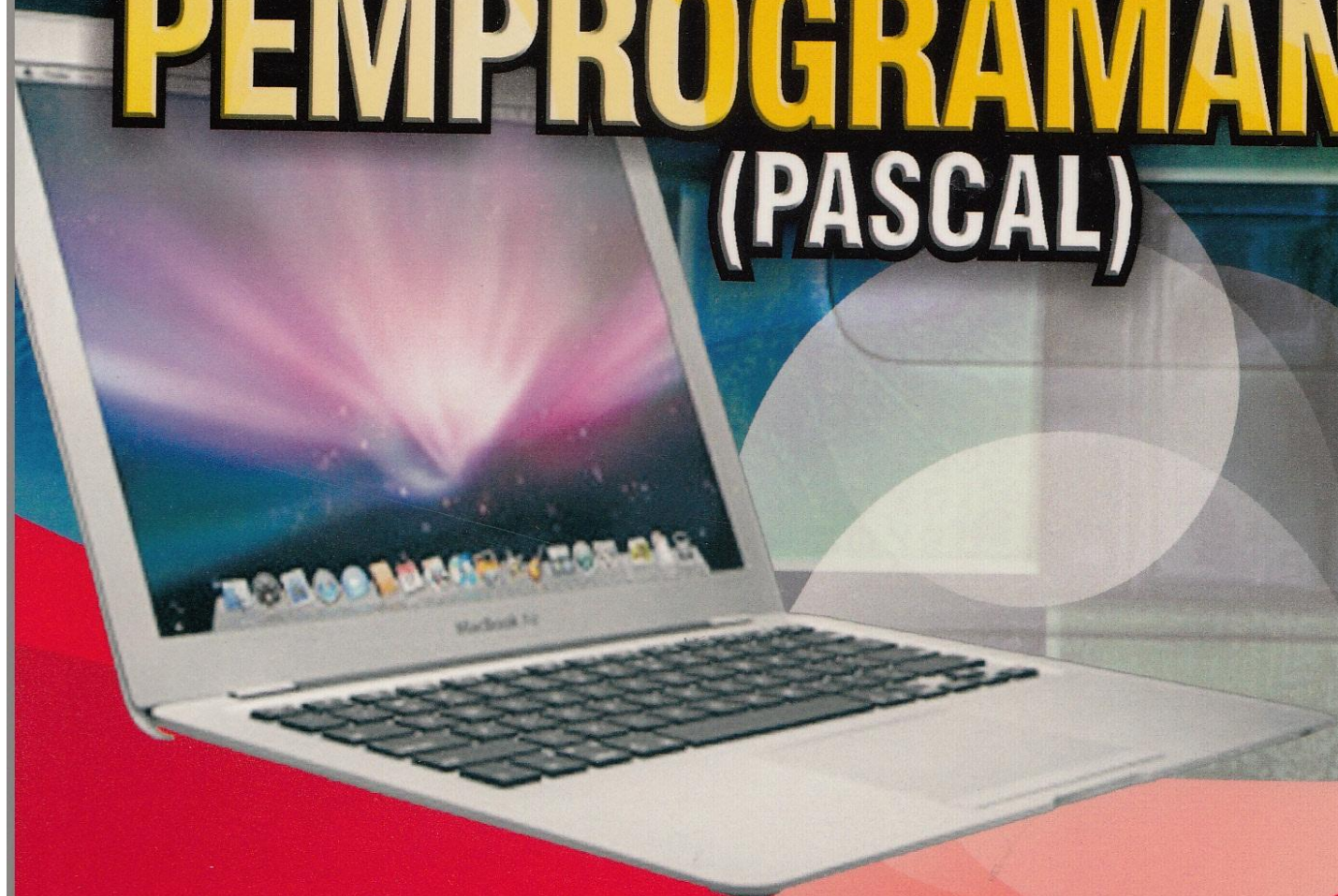


Eko Purwanto

BAHASA PEMPROGRAMAN (PASCAL)



Katalog Dalam Terbitan (KDT)

Bahasa Pemrograman Pascal ; Eko Purwanto

Penulis : Eko Purwanto

Editor : Tim Duta Publishing Indonesia

Tata Letak : Ach. Khotim

Desain Sampul : Tim Desain DPI

Diterbitkan Oleh :



DUTA PUBLISHING INDONESIA
JL. PULANGGENI NO. 8 SOLO Telp. (0271) - 712432

Cetakan I : JUNI 2011

Perpustakaan Nasional RI :

Eko Purwanto, S.Kom

148 hlm., 20, 5 cm

ISBN: 978-602-99181-7-5

KATA PENGANTAR

Puji Syukur ke hadirat Tuhan Yang Maha Esa karena izin dan rahmat-Nya laah buku ini dapat terselesaikan dengan baik.

Pascal merupakan bahasa pemrograman yang terpopuler di lingkungan yang terpopuler di lingkungan akademis dan banyak digunakan di sebagian besar universitas dan institusi pendidikan lainnya untuk menerangkan dasar-dasar logik pemrograman.

Dalam buku ini, pembahasan akan dimulai dari pemahaman konsep dan esensi yang terdapat dalam Bahasa Pascal sebagai bekal dasar untuk melangkah ke tahap selanjutnya. Untuk menguji ketajaman anda dalam menerima setiap materi yang disampaikan, di dalam setiap bab disertakan juga soal-soal yang disuguhkan dalam bentuk langsung ke pembuatan program.

Buku ini dikhususkan untuk mahasiswa STMIK Duta Bangsa Surakarta yang ingin menjadi seorang programmer Pascal, juga bagi anda yang merupakan para pengguna Delphi, buku ini akan memberikan konsep dan dasar yang kuat untuk kemudian dikembangkan dan diimplementasikan ke dalam aplikasi visual.

Tak lupa penulis mengucapkan terima kasih kepada rekan-rekan kerja di STMIK Duta Bangsa serta rekan-rekan pengajar STMIK Duta Bangsa atas bantuan yang telah diberikan. Terakhir, sebagai karya manusia biasa, penulis juga menyadari bahwa dalam buku ini tentu masih terdapat adanya kelemahan. Walaupun demikian, penulis berharap semoga buku ini dapat bermanfaat bagi anda.

Surakarta, Agustus 2011

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
KATA PENGANTAR	ii
DAFTAR ISI	iii
DAFTAR TABEL	v
DAFTAR GAMBAR	vi
BAB I PENDAHULUAN	1
A. Konsep Dasar Program, Pemrograman dan Bahasa Pemrograman	1
B. Sejarah Pascal	5
C. Instalasi Turbo Pascal 7.0	6
D. Latihan Bab I	8
BAB II EDITOR PASCAL	9
A. Menjalankan Editor Turbo Pascal 7.0	9
B. Mengenal Lingkungan Editor Turbo Pascal 7.0	11
C. Mengenal Unit Standar Turbo Pascal 7.0	15
D. Membuat Program Pertama	17
E. Latihan Bab II	24
BAB III DASAR PEMROGRAMAN PASCAL.....	25
A. Susunan Program Pascal	25
B. HotKey	29
C. Aturan Pemrograman Pascal	29
D. Komponen Dasar Program Pascal	30
E. Perintah Dasar Pemrograman Pascal	32
F. Komentar	38
G. Latihan Bab III	39
BAB IV TIPE DATA DAN OPERATOR	40
A. Tipe data yang disediakan oleh PASCAL	40
B. Operator	44

	C. Latihan Bab IV	53
BAB V	LOGIKA PERCABANGAN DAN PENGULANGAN	55
	A. Percabangan	55
	B. Latihan Bab Va	62
	C. Pengulangan	63
	D. Latihan Vb	68
	E. Latihan Vc	71
	F. Peloncatan	72
BAB VI	PROCEDURE DAAN FUNCTION	76
	A. Procedure	76
	B. Function	83
	C. Rekursi	87
	D. Mengenal Kata Kunci Forward	89
	E. Overload Prosedur dan Fungsi	92
BAB VII	ARRAY & RECORD	36
	A. Array	36
	B. Array Dua Dimensi	41
	C. Record	45
	D. Array Berisi record	56
	E. Record Berisi Array	58
	F. Sorting	58
BAB VIII	FILE	118
	A. Text File	118
	B. Typed File	118
	C. Untyped File	118
	D. Contoh Program	119
DAFTAR PUSTAKA		

Kompetesi Dasar :

1. Mahasiswa dapat memahami konsep dasar program, pemrograman dan bahasa pemrograman
2. Mahasiswa dapat mengetahui perkembangan bahasa pascal
3. Mahasiswa dapat mmenginstall program Turbo Pascal 7.0
4. Mahasiswa memahami konsep dasar pemrograman dengan menggunakan Turbo Pascal 7.0

A. Konsep Dasar Program, Pemrograman dan Bahasa Pemrograman

1. Program, Pemrograman dan Bahasa Pemrograman

Istilah program atau program komputer mungkin sudah tidak asing lagi ditelinga kita, begitu juga istilah pemrograman. **Program atau program komputer** adalah instruksi-instruksi yang diberikan kepada komputer untuk melaksanakan tugas tertentu. sedangkan **pemrograman** adalah upaya untuk membuat kode kode yang dapat dikenal oleh komputer.

Seperti halnya manusia yang diciptakan dengan berbagai bahasa, agama, suku dan golongan, komputer juga memiliki bahasa yang dikenal dengan bahasa pemrograman. Perbedaan bahasa manusia dan bahasa komputer terletak pada keanekaragaman bahasa yang dimiliki. Manusia memiliki ribuan bahkan jutaan bahasa yang digunakan untuk berkomunikasi antar manusia sedangkan komputer hanya mengerti bahasa satu bahasa saja yaitu bahasa mesin. Bahasa mesin terdiri dari digit biner yaitu 0 dan 1. Bahasa dengan kode seperti ini terntulah menyulitkan manusia untuk mempelajarinya sehingga diciptakanlah bahasa pemrograman komputer yang lain untuk membantu manusia membuat kode program yang dikenal dengan **bahasa pemrograman**.

Bahasa pemrograman terbagi atas dua tingkatan (walaupun terdapat beberapa buku yang mendefinisikan 3 tingkatan) yaitu : **Bahasa Pemrograman Tingkat Rendah** (*Low Level Language*) dan **Bahasa Pemrograman Tingkat Tinggi** (*High Level Language*).

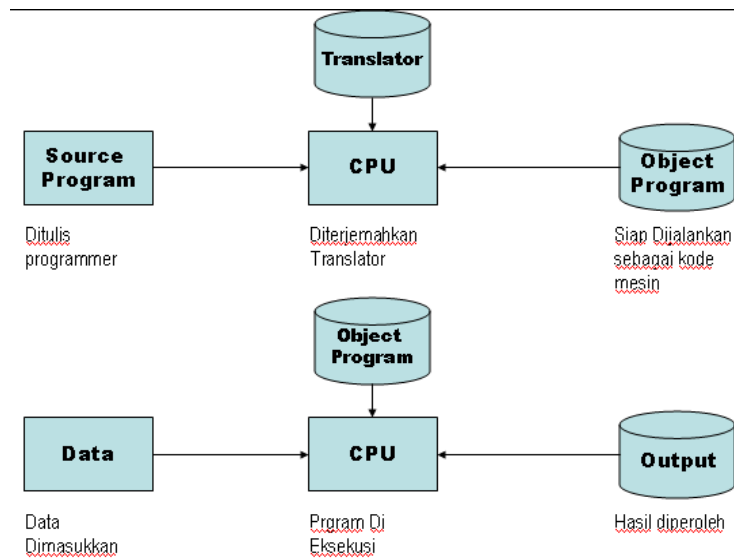
Bahasa pemrograman tingkat rendah adalah bahasa yang tidak bisa dimengerti oleh manusia karena hanya merupakan deret angka 0 dan 1. Contoh : Bahasa Mesin, Bahasa Rakitan.

Sedangkan bahasa pemrograman tingkat tinggi adalah bahasa yang mendekati bahasa manusia sehingga mudah dimengerti oleh manusia. Contoh : FORTRAN, PASCAL, ALGOL, COBOL dll

2. Language Translator

Bahasa tingkat tinggi tidak serta merta dimengerti oleh komputer karena komputer hanya mengerti bahasa mesin. Untuk mengatasi hal tersebut kita dapat menggunakan translator yang dikenal dengan istilah *language translator*. *Language translator* bertugas untuk menterjemahkan bahasa tingkat tinggi kedalam bahasa mesin sehingga dapat dijalankan oleh komputer. Proses menterjemahkan bahasa pemrograman tersebut dikenal dengan istilah Kompilasi Program. Perlu diketahui bahwa language translator yang digunakan haruslah sesuai dengan kode yang ingin diterjemahkan. Jika kita menulis program dengan menggunakan bahasa PASCAL, maka language translator yang kita gunakan juga harus sama yaitu language translator untuk bahasa PASCAL.

Cara kerja language translator dapat dilihat pada skema berikut :



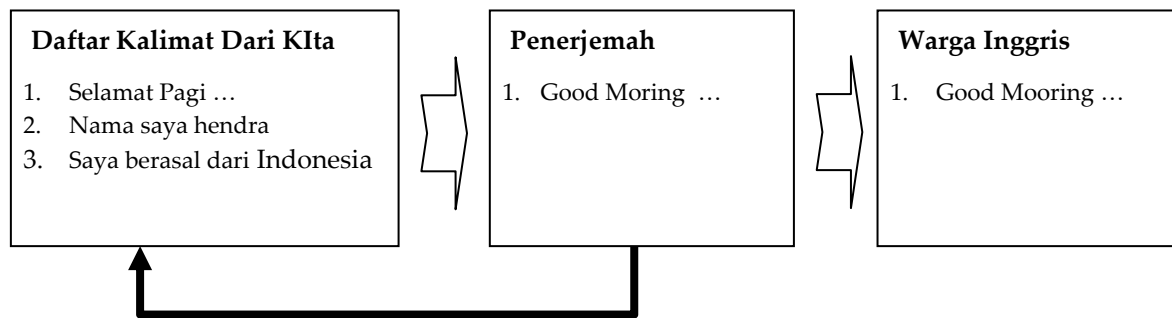
Gambar 1.1 Skema Kerja Language Translator

Language translator terbagi menjadi 2 yaitu : *Interpreter* dan *Compiler*. Walaupun prinsip kerjanya sama, compiler dan interpreter memiliki perbedaan yang cukup mencolok. Pada interpreter, kode akan diterjemahkan perbaris sehingga apabila menemui kesalahan, baris sebelumnya akan tetap dieksekusi. Hal ini sangat berbeda dengan compiler yang menterjemah semua kode program terlebih dahulu sebelum akhirnya menjalankannya, sehingga apabila terdapat kesalahan pada baris tertentu, Compiler tidak akan mengeskusi baris-baris sebelumnya. Untuk lebih jelasnya perhatikan ilustrasi berikut :

Anda tengah berkunjung ke Inggris, sedangkan anda tidak bisa berbahasa Inggris. Anda mengambil inisiatif untuk menyewa seorang penerjemah untuk menyapa salah seorang warga yang anda temui. Daftar kalimatnya adalah :

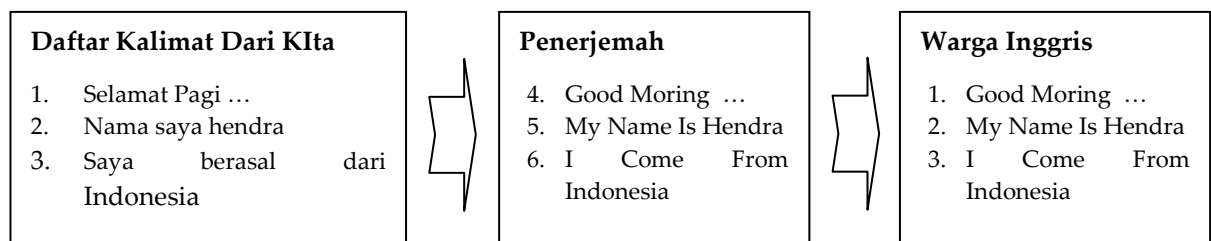
1. Selamat Pagi ...
2. Nama saya hendra
3. Saya berasal dari Indonesia

Konsep Interpreter



Gambar 1.2 Ilustrasi Interpreter

Konsep Compiler



Gambar 1.3 Ilustrasi Compiler

3. Langkah Pemrograman

Tahap-tahap yang harus kita tempuh pada saat akan melakukan pemrograman adalah :

1. Membuat Algoritma dan Flowchart
2. Menuliskan program dengan menggunakan bahasa pemrograman
3. Menguji kebenaran program
4. Perbaiki program jika ada kesalahan
5. Menjalankan program

Khusus pemrograman compiler, sebelum program dapat dijalankan, program dapat dikompilasi dahulu untuk mengetahui baris mana saja yang terdapat kesalahan sintaks (kesalahan dalam penulisan perintah). Selain

kesalahan sintaks terdapat kesalahan lain yang disebut yang kesalahan logika, kesalahan pada tahap ini umumnya menimbulkan *bug* (baca : bag). Sedangkan upaya untuk mencari kesalahan program disebut dengan *debugging*.

B. Sejarah Pascal

Pascal adalah bahasa tingkat tinggi (*high level language*) yang orientasinya pada segala tujuan, dirancang oleh **Profesor Niklaus Wirth** dari *Technical University of Zurich*, Switzerland. Nama Pascal diambil sebagai penghargaan terhadap *Blaise Pascal*, ahli matematik dan filosofer terkenal abad 17 dari Perancis. Profesor Niklaus Wirth memperkenalkan kompiler bahasa Pascal pertama kali untuk komputer *CDC 6000* (Control Data Corporation) yang dipublikasikan pada tahun 1971 dengan tujuan membantu mengajar program komputer secara sistematis, khususnya untuk memperkenalkan pemrograman terstruktur.

Dalam waktu singkat, Pascal telah menjadi bahasa yang populer di kalangan pelajar universitas dan merupakan bahasa yang diajarkan di beberapa perguruan tinggi. Beberapa profesional komputer juga mulai beralih ke bahasa Pascal. Kenyataannya, Pascal merupakan bahasa yang paling cepat populer dibandingkan dengan bahasa-bahasa komputer tingkat tinggi yang lainnya.

Standar Pascal adalah bahasa Pascal yang didefinisikan oleh **K.Jensen** dan **Niklaus Wirth**. Penerapan nyata dari standar Pascal banyak yang berbeda dengan seperti apa yang telah didefinisikan oleh K. Jensen dan Niklaus Wirth. Standar Pascal di Eropa didefinisiakn oelh *ISO* (International Standards Organization) dan di Amerika oleh kerjasama antara *ANSI* (American National Standar Institute) dengan *IEEE* (Intitute of Electrical and Electronic Engineer).

Beberapa versi dari Pascal yang telah beredar di pasaran, di antaranya *UCSD pascal* (University of California at San Diego Pascal), *MS-Pascal*

(Microsoft Pascal), *Apple Pascal*, *Turbo Pascal* dan lain sebagainya. Sampai saat ini untuk komputer-komputer mikro dan personal, Turbo Pascal merupakan versi bahasa Pascal yang paling populer dan banyak digunakan. Kompiler Turbo Pascal banyak digemari, karena terutama bersifat interaktif, seperti interpreter saja layaknya. Selain itu Turbo Pascal mengikuti definisi dari standar Pascal seperti yang didefinisikan oleh K. Jensen dan Niklaus Wirth di Pascal User Manual dan Report. Turbo PASCAL adalah copyright dari Borland Inc. dan dapat digunakan pada sistem operasi PC-DOS, MS-DOS, CPM-86 dan CP/M-80.

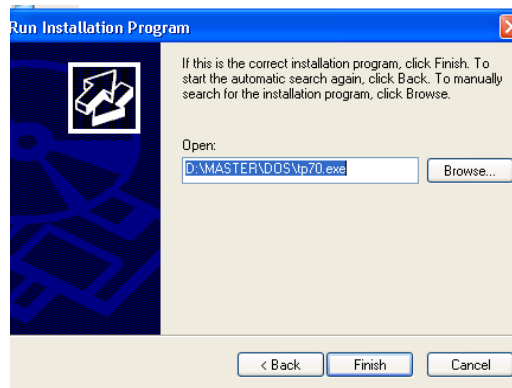
Kepopuleran PASCAL berkembang seiring dengan perkembangan PC. Terdapat beberapa Co mpiler PASCAL, salah satunya adalah TURBO PASCAL yang dikeluarkan oleh Borland International. TURBO PASCAL 7.0 dirilis pada tahun 1992 yang mempunyai :

1. Kompiler (untuk mengkompilasi program menjadi suatu objek yang dipahami oleh komputer)
2. Debugger (untuk membantu mencari kesalahan program)
3. Linker (untuk menggabungkan file objek dan file pustakadan membentuk File Executable)
4. Editor (untuk menulis program sumber/ *source code*)

C. Instalasi Turbo Pascal 7.0

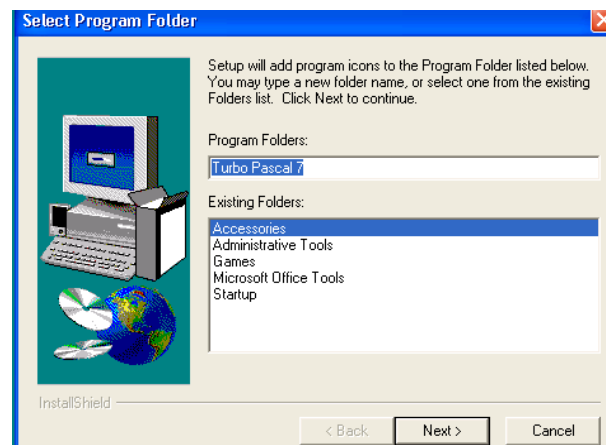
Untuk melakukan instalasi Turbo Pascal 7.0 anda harus mempunyai mastermya dulu. Lakukan langkah instalasi program dengan memilih menu **Control Panel** kemudian menu **Add and Remove Program** kemudian **Add New Program** dan langkah selanjutnya adalah :

1. Pilih CD or Floppy kemudian pilih **Browse** untuk mencari letak master **Turbo Pascal 7.0** yang akan anda install kemudian pilih **Finish**



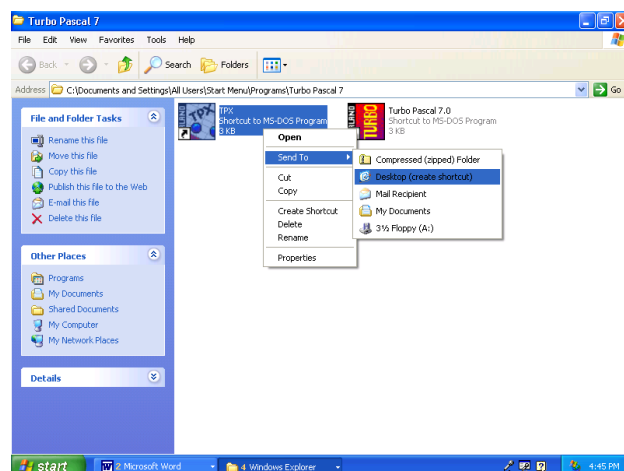
Gambar 1.4 Install Pascal 1

2. Tunggu Proses Instalasi berjalan kemudian Klik **Next** setelah tampil konfirmasi penambahan icon pascal. Setelah itu tunggu proses instalasi akan dilanjutkan sampai selesai.



Gambar 1.5 Install Pascal 2

3. Pindahkan Icon TPX yang terbentuk ke Desktop



Gambar 1.6 Install Pascal 3

Daftar Pustaka

Abdul Kadir, 2004, *Pemrograman Pascal*, Edisi I, Penerbit Andi, Yogyakarta
Aji Supriyanto, 2005, *Pengantar Teknologi Informasi*, , Salemba Infotek, Jakarta
Budi Raharjo, 2005, *Teknik Pemrograman Pascal*, Informatika, Bandung
<http://www.boxpascal.110mb.com/Contoh%20Program%20Pascal.html> (Diakses Tanggal 14 Januari 2008)
<http://www.hu.freepascal.org/faq.html.id> (Diakses Tanggal 14 Januari 2008)

Latihan Bab 1

1. Apa yang ada ketahui tentang program, pemrograman dan bahasa pemrograman ?
2. Sebutkan dan beri contoh bahasa pemrograman !
3. Apa yang dimaksud dengan language translator dan jelaskan cara kerjanya!
4. Apa perbedaan Compiler dan Interpreter ?
5. Apa yang menjadi kelebihan Turbo Pascal 7.0 ?

Kompetesi Dasar :

1. Mahasiswa mengenal editor PASCAL untuk membuat kode program
2. Mahasiswa dapat menggunakan editor PASCAL untuk membuat kode program

A. Menjalankan Editor Turbo Pascal 7.0

Setelah program Turbo Pascal 7.0 kita install di komputer kita, langkah selanjutnya adalah membuka editor PASCAL untuk dapat menuliskan program. Langkah-langkah yang dapat kita lakukan untuk menjalankan Editor Turbo Pascal 7.0 adalah sebagai berikut :

1. Sebelum anda bekerja sebaiknya anda membuat satu folder khusus untuk menyimpan latihan kita misal folder bernama : LATIH (misal di direktori D:\ atau yang lain).
2. Jalankan program Turbo Pascal 7.0 dari desktop anda atau **Start --**

Program – Turbo Pascal 7 – TPX



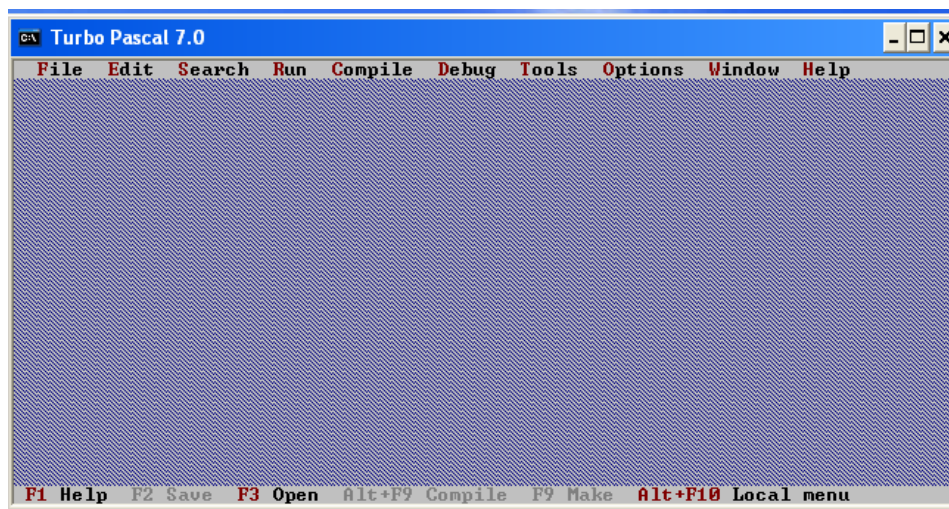
Gambar 2.1 Menu Editor TURBO PASCAL 7.0 Desktop

Atau



Gambar 2.2 Menu Editor TURBO PASCAL 7.0 Start

3. Akan muncul tampilan awal Editor Turbo Pascal 7.0 sebagai berikut :



Gambar 2.3 Tampilan Awal Editor Turbo Pascal

4. Langkah selanjutnya adalah mengarahkan ke direktori kerja yang sudah kita buat.

B. Mengetahui Lingkungan Editor Turbo Pascal 7.0

Turbo Pascal mempunyai sebaris menu yang berada dibagian paling atas yang terdiri dari :

1. File

- a. New, membuat program baru
- b. Open, Memanggil program yang disimpan pada disk
- c. Save, Menyimpan program dengan nama yang sama
- d. Save As, Menyimpan program dengan nama yang lain
- e. Save All, menyimpan semua program yang ada dilayar
- f. Change Dir, mengubah drive aktif untuk penyimpanan dan pemanggilan file.
- g. Print, untuk mencetak
- h. Printer Setup, pengaturan printer
- i. DOS Shell, Masuk ke DOS untuk sementara, untuk kembali lagi ke Turbo Pascal Ketik EXIT kemudian Enter
- j. Exit, Keluar dari Turbo Pascal

2. Edit

- a. Undo, mengembalikan yang terhapus terakhir
- b. Redo, menghapus kembali yang di Undo
- c. Cut, menghapus blok untuk dipindah ke tempat lain
- d. Copy, mengcopy blok
- e. Paste, menampilkan isi dari cut atau copy blog
- f. Show Clipboard, menampilkan isi clipboard

3. Search

- a. Find, untuk mencari atau perintah yang tertulis pada program yang sedang dikerjakan
- b. Replace, untuk mengganti kata atau perintah yang tertulis pada program yang sedang dikerjakan

- c. Search Again, mencari lagi kata atau perintah yang tertulis pada program yang sedang dikerjakan
- d. Go to line number, menuju ke baris yang dimaksud
- e. Show last compiler error, menunjukkan kesalahan kompilasi yang terakhir
- f. Find error, mencari kesalahan
- g. Find procedure, mencari procedure yang dimaksud
- h. Previous Browser dan Globals, menunjukkan berbagai fungsi dan prosedur yang ada di pascal
- i. Objects dan Unit, untuk mencari nama objek dan unit
- j. Symbol, mencari simbol

4. **Run**

- a. Run, untuk menjalankan atau mengeksekusi program
- b. Step Over, Step over ini memiliki cara kerja yang hampir sama dengan Trace Into, hanya saja jika statement adalah sebuah prosedur atau fungsi, maka keseluruhan subrutin akan dikerjakan sekaligus dan debugger akan berhenti pada statement sesudah pemanggilan ke subrutin
- c. Trace into, Perintah ini digunakan untuk mengeksekusi baris berikutnya, jika ada pemanggilan ke suatu subrutin, maka pelacakan akan dimulai dari statement pertama dari subrutin tersebut.
- d. Go to cursor, Pilihan ini digunakan untuk memulai / melanjutkan eksekusi program dimulai dari posisi saat eksekusi dimulai sampai tempat kursor berada.
- e. Program reset, Pilihan ini digunakan untuk memberitahukan Turbo Pascal bahwa anda selesai dengan pembetulan-pembetulan dan menginisialisasi debugger untuk operasi yang lain.
- f. Parameter, menjalankan program parameter

5. Compile

- a. Compile, Perintah ini digunakan untuk mengkompile program (mencari kesalahan) atau bisa juga kita bilang kalau program yang telah kita buat tadi dengan bahasa pascal di terjemahkan kedalam bahasa komputer agar si komputer ini mengerti apa-apa saja perintah kita
- b. Make, Pilihan ini digunakan untuk mengaktifkan perintah make, jika file yang primer telah diberi nama, maka file ini akan dikompilasi, jika tidak maka file yang terakhir yang sedang dimuat di editor akan dikompilasi.. Pilihan ini sangat berguna jika terdapat beberapa program yang secara fisik terpisah tetapi secara logikanya merupakan suatu kesatuan.
- c. Build, Perintah ini mirip dengan perintah make, hanya saja perintah build ini sifatnya bebas, sedangkan perintah make hanya akan mengkompilasi file-file selain file yang digunakan.
- d. Primary File, Perintah ini digunakan untuk menyatakan bahwa file .PAS akan dikompilasi pada saat make [F9] atau build [Alt + C + B] diaktifkan.
- e. Clear primary file, menghapus file primer
- f. Information, perintah ini digunakan untuk menampilkan jendela informasi dari program .PAS yang sedang anda gunakan, termasuk ukuran kode sumber [dalam sumber dan baris], ukuran file .EXE atau .TPU dalam byte dan data.

6. Debug

- a. Breakpoints, mengatur suatu baris sebagai breakpoint. Dalam editor breakpoint ditunjukkan dengan highlight text.
- b. Call stack, pada saat anda melakukan debugging, suatu jendela yang menunjukan daftar pemanggil prosedur dan fungsi akan terlihat.

- c. Register, menampilkan Informasi CPU.
- d. Watch, menampilkan jendela watch.
- e. Output, menampilkan output program.
- f. User Screen, Pada umumnya perintah ini digunakan untuk melihat hasil dari program yang telah compile dan kita run.
- g. Evaluate/ modify, Dengan pilihan ini anda akan dibawa ke suatu jendela dengan 3 kotak yang memungkinkan anda untuk menuliskan sembarang nama perubah / ungkapan, menunjukan nilai perubah / ungkapan saat itu an memungkinkan anda untuk memberikan nilai yang baru untuk sembarang perubah
- h. Add watch, digunakan untuk menambahkan rinci data, perubah atau ungkapan kedalam jendela watch.
- i. Add breakpoint, menambah breakpoint.

7. **Tools**

- a. Messages, menampilkan jendela pesan kesalahan
 - b. Go to Next, ke baris selanjutnya
 - c. Go to previous, ke bari sebelumnya
 - d. Grep, digunakan untuk menyaring masukan atau menampilkan baris-baris yang mengandung kriteria tertentu
8. **Option**, untuk mengatur berbagai pilihan compiler, memori size, linker, debugger dan lain-lain
9. **Window**, untuk mengatur tampilan windows
10. **Help**, berisi berbagai bantuan yang disediakan Turbo Pascal

C. **Mengenal Unit Standar Turbo Pascal 7.0**

Unit adalah kumpulan konstanta, tipe data, variabel, *procedur* dan *function*. Semua perintah yang terdapat dalam turbo pascal disimpan dalam

bentuk unit-unit. Turbo pascal telah menyediakan beberapa unit-unit standar antara lain :

1. System

Unit sistem adalah perintah-perintah utama dalam turbo pascal sehingga kita tidak perlu menggunakan perintah Uses System pada awal program. Berikut ini contoh perintah yang masuk dalam unit system.

Tabel 2.1 Contoh Perintah Unit System

Perintah	Fungsi
INT	Mengabaikan angka dibelakang koma atau membulatkan angka.
Frac	Mengambil nilai desimal suatu angka
Exp	Mencari ekponensial dari suatu bilangan
Ln	Mencari logaritma Natural dari suatu bilangan
Pi	Memasukkan bilangan 3.14
Sin	Mencari SINUS dari suatu bilangan dalam satuan radian
Cos	Mencari COSINUS dari suatu bilangan dalam satuan radian
Abs	Mencari absolute (menghilangkan tanda negatif) dari suatu bilangan
Inc	Menambahkan angka dengan angka 1 atau angka lain sesuai dengan perintah
Dec	Mengurangi dengan angka 1 atau angka lain sesuai dengan perintah
Odd	Melakukan pelacakan apakah suatu bilangan ganjil atau tidak
Succ	Untuk mencari pengikut (successor atau sesudahnya) dari suatu angka atau argumen
Pred	Untuk mencari pendahulu (predecessor atau argumen sebelumnya) dari suatu angka atau argumen
Chr	Perintah untuk mencetak karakter tambahan atau alkim disebut Kode ASCII
Ord	Untuk mencari letak suatu argumen atau mencari nomor ASCII dari Suatu Argumen
Round	Membulatkan angka real menjadi integer
Trunc	Menghilangkan desimal dari suatu bilangan
Upcase	Mengubah karakter huruf kecil menjadi huruf besar
Lo	Untuk mencari byte terendah dari suatu argumen
Hi	Untuk mencari byte tertinggi dari suatu argumen
Swap	Untuk membalikkan byte tertinggi dan terendah dari suatu argumen
Low	Mengambil nilai terendah dari range suatu argumen
High	Mengambil nilai tertinggi dari range suatu argumen

2. Crt

Unit Crt digunakan untuk memanipulasi layer teks (windowing, peletakkan cursor dilayar, color untuk teks, kode extended keyboard dan lainnya). Untuk menggunakan perintah yang terdapat dalam unit Crt harus menggunakan Uses Crt pada awal program. Berikut ini diberikan contoh perintah yang tergabung dalam unit Crt.

Tabel 2.2 Contoh Perintah Unit Crt

Perintah	Fungsi
ClrScr	Perintah untuk membersihkan layar
GoToXY	Untuk meletakkan pada posisi tertentu di layar
TextColor	Untuk membuat warna pada teks
TextBackGround	Memberi warna latar belakang pada teks
TextAttr	Gabungan dari TextColor dan TextBacground
Sound	Untuk membuat suara
No Sound	Untuk menghentikan suara
Delay	Untuk membarikan waktu tunda
KeyPressed	Untuk mengetahui sembarang tombol sudah ditekan
Readkey	Untuk membaca tombol yang ditekan oleh pemakai
Window	Untuk membuat jendela teks pada layar
ClrEol	Untuk menghapus posisi kursor hingga akhir baris

3. DOS

Digunakan bila akan menggunakan prosedur-prosedur dan fungsi-fungsi standar yang berhubungan dengan DOS call, semacam DetTime, SetTime, DiskSize, DiskFree dan lainnya.

4. WinDOS

Unit yang berkaitan implementasi sistem operasi dan dan menangani file-file rutin

5. Graph

menyediakan suatu kumpulan rutin grafik yang canggih, sehingga dapat memanfaatkannya untuk keperluan-keperluan pembuatan grafik.

6. Graph3

Unit yang dirancang untuk mendukung aplikasi grafis pada turbo pascal 3.0

7. Overlay

Unit yang berisi *procedure*, *function* dan *symbol* untuk yang digunakan dengan *overlay manager Borland Pascal*

8. Printer

Merupakan unit yang sangat kecil dirancang untuk penggunaan printer didalam program

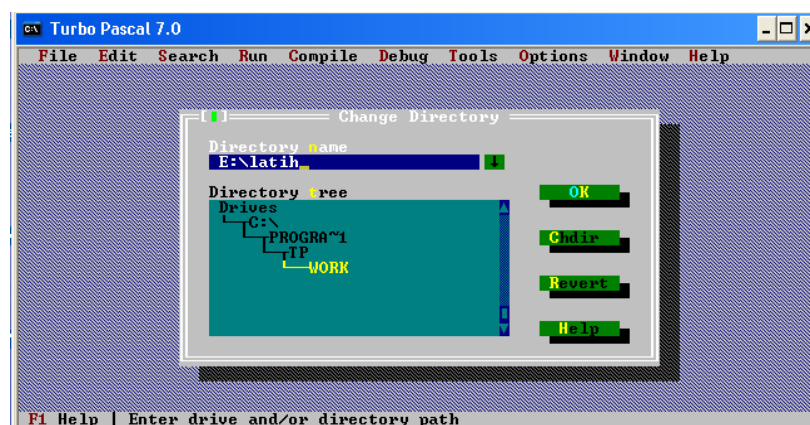
9. Turbo3

Unit yang dikembangkan hanya untuk mendukung Turbo Pascal 3.0

D. Membuat Program Pertama

1. Mengubah Direktori Kerja

Sebelum kita membuat program, aturlah terlebih dahulu direktori kerja kita. Setelah tampilan editor muncul, pilih menu **File – Change directory** kemudian ketikkan Direktori kerja anda dan tekan **Enter** atau klik **OK**.



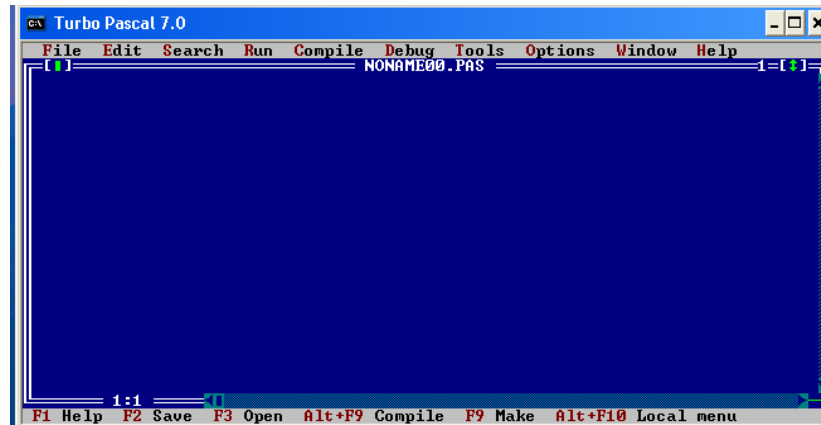
Gambar 2.4 Mengubah Direktori Kerja

2. Menulis Program Baru

Lembar editor baru dapat dibuat dengan langkah sebagai berikut:

(a) Pilih menu File (Alt + F)

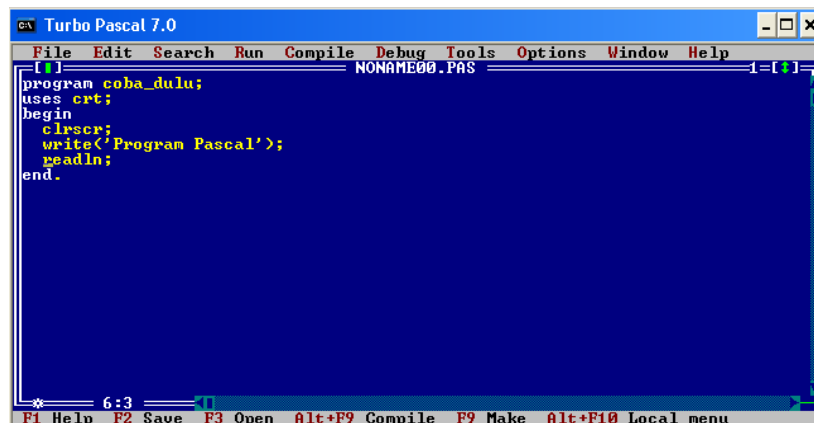
(b) Pilih sub menu New sehingga tampil layar sebagai berikut :



Gambar 2.5 Lembar Editor Baru

(c) Ketik program berikut :

```
program coba_dulu;  
uses crt;  
begin  
  clrscr;  
  write ('Program Pascalku');  
  readln;  
end.
```

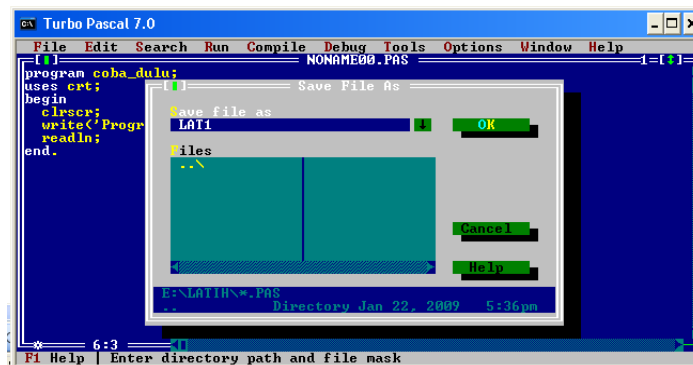


Gambar 2.6 Kode Program yang Diketik pada Lembar Editor

3. Menyimpan Program

Kode program yang sudah kita ketik atau dimodifikasi akan lebih baik jika langsung disimpan, hal ini untuk berjaga-jaga jika listrik tiba-tiba mati atau karena sebab yang lain. Cara menyimpan program adalah sebagai berikut :

- (1) Pilih Menu File
- (2) Pilih Save kemudian beri nama LAT1 dan tekan Enter

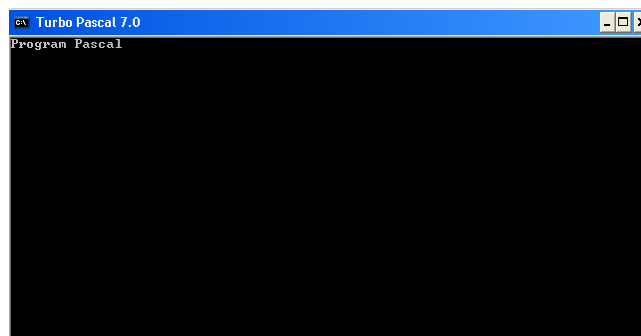


Gambar 2.7 Jendela untuk Menyimpan File Program

4. Menjalankan Program

Program yang telah ditulis pada editor pascal dapat dijalankan atau dieksekusi dengan cara sebagai berikut :

- (1) Pilih Menu Run (Alt + R)
- (2) Pilih RUN atau perintah tersebut dapat diganti dengan menekan Ctrl+F9 secara bersamaan dan tekan Enter untuk kembali ke Editor Pascal

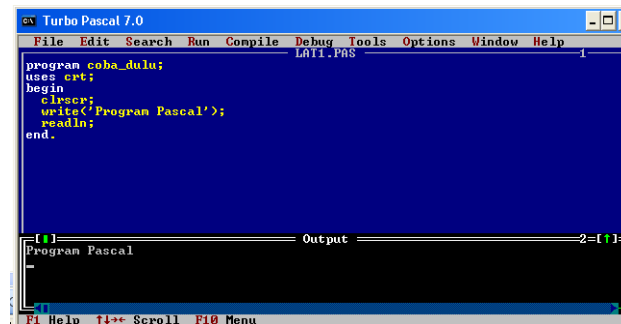


Gambar 2.8 Program yang telah dieksekusi

5. Melihat Hasil Program

Untuk melihat hasil program dan kode program secara bersamaan dapat dilakukan dengan langkah berikut :

- (1) Pilih Menu Debug
- (2) Pilih Output kemudian tampil output program dan klik di area editor pascal untuk menghilangkan output program



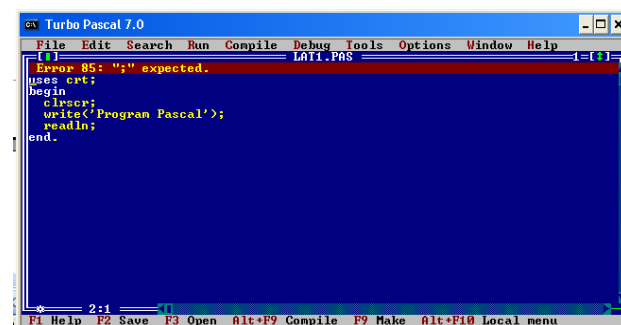
Gambar 2.9 Melihat Ouput Program

6. Mengkompilasi Program

Sebelum dijalankan, program dapat dikompilasi dulu untuk mengetahui kesalahan yang mungkin terjadi. Cobalah menghilangkan tanda (;) pada akhir baris pertama program yang telah dibuat, kemudian lakukan langkah berikut :

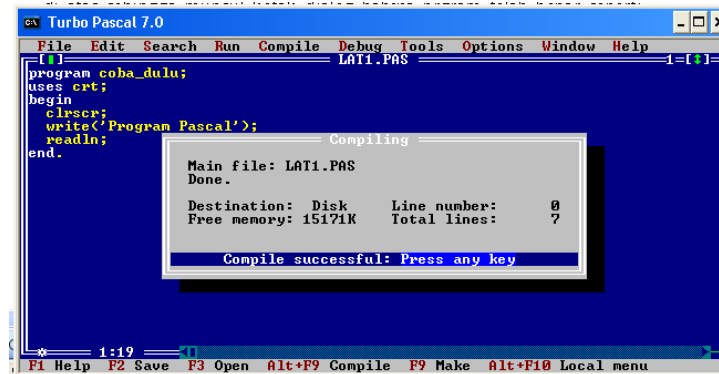
- (1) Pilih Menu Compile (Alt + C)
- (2) Pilih Compile

Akan tampil pesan kesalahan seperti berikut ini :



Gambar 2.10 Kesalahan Sintaks Program

Kembalikan kode program seperti semua dan lakukan lagi langkah-langkah di atas sehingga muncul kotak dialog bahwa program telah benar seperti berikut ini :



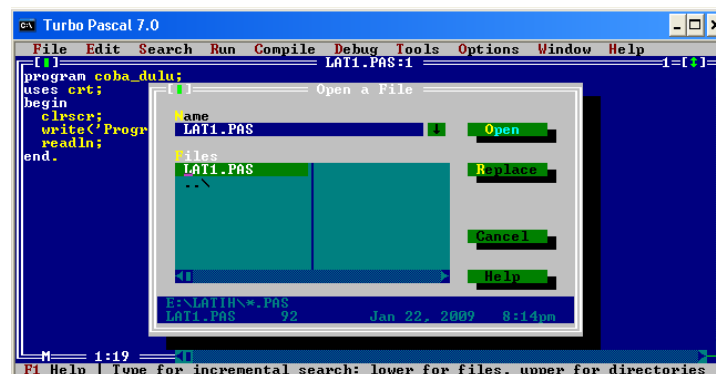
Gambar 2.10 Hasil Kompilasi Program

Proses kompilasi program juga dapat dilakukan dengan menekan tombol Alt + F9 secara bersamaan.

7. Membuka File Program

File yang telah tersimpan di harddisk atau media penyimpanan lain dapat kita buka kembali dengan :

- (1) Pilih menu File (Alt+F)
- (2) Pilih Open (F3) kemudian cari dimana lokasi penyimpanan file dan juga nama filenya
- (3) Tekan Enter atau klik Open

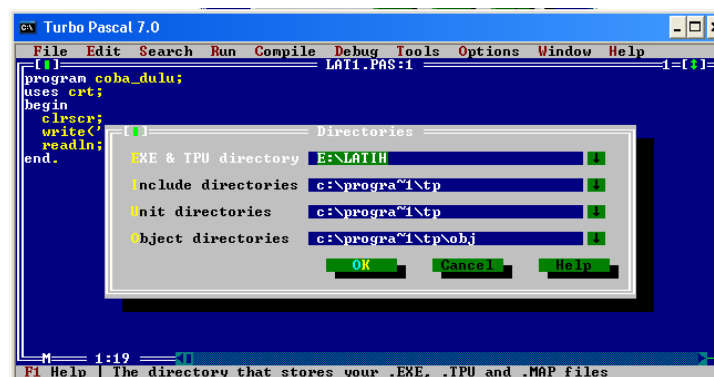


Gambar 2.11 Membuka File

8. Membuat File .EXE

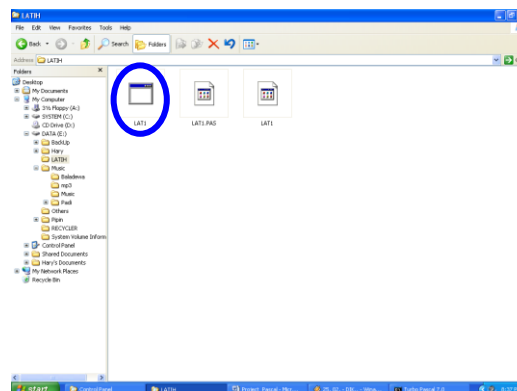
File .EXE atau executable file adalah program yang dapat dijalankan secara langsung melalui prompt DOS tanpa harus membuka program Turbo Pascal terlebih dahulu. Untuk membuat file .EXE dapat dilakukan dengan cara :

- (1) Pilih Menu Option
- (2) Pilih Directories, kemudian ubah direktori EXE dan TPU menjadi Direktori kerja kita lalu klik OK



Gambar 2.11 Mengatur Sub Menu Directries pada Menu Option

- (3) Pilih Menu Option lagi dan Pilih Sub Menu Save
- (4) Pilih Menu Compile kemudian sub menu Compile atau tekan Alt + F9 bersamaan
- (5) Buka direktori kerja kita lewat Windows Explorer, akan tampak satu file berekstensi .EXE yang baru saja kita buat



Gambar 2.12 File .Exe

9. Keluar dari Turbo Pascal

Untuk keluar dari editor pascal dapat dilakukan dengan cara berikut :

- (1) Pilih Menu File (Alt+F) kemudian pilih sub menu Exit
- (2) atau tekan Alt+X secara bersamaan

Daftar Pustaka

Abdul Kadir, 2004, *Pemrograman Pascal*, Edisi I, Penerbit Andi, Yogyakarta
Budi Raharjo, 2005, *Teknik Pemrograman Pascal*, Informatika, Bandung
Mico Pardosi, 1999, *Bahasa Pemrograman Turbo Pascal 7.0*, Penerbit Indah, Surabaya
<http://www.a-rang.info> (diakses tanggal 22 Januari 2009)
<http://materi-praktek.blogspot.com/2007/03/bab-1-mengenal-tampilan-turbo-pascal.html>
(diakses tanggal 22 Januari 2009)

Latihan Bab 2

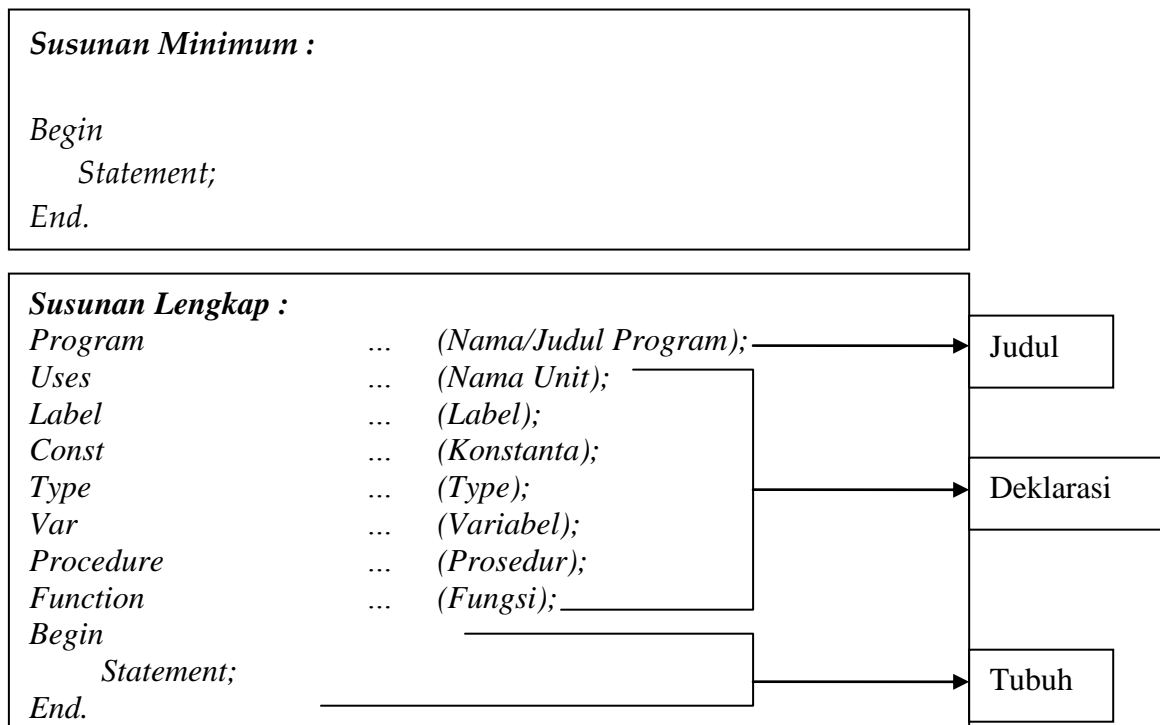
1. Sebutkan menu yang terdapat dalam editor Turbo Pascal 7.0
2. Apa yang dimaksud dengan Unit dan apa fungsi unit Crt?
3. Apa yang kau ketahui tentang Unit System dan mengapa untuk menggunakannya tidak memerlukan perintah Uses (Uses System)?
4. Apa perbedaan Run dan Compile?
5. Apa yang dimaksud dengan File .EXE dan bagaimana cara membuatnya ?

Kompetesi Dasar :

3. Mahasiswa dapat mengetahui dan memahami susunan program pascal
4. Mahasiswa dapat membuat program dengan menggunakan bahasa pascal

A. Susunan Program Pascal

Program bahasa pascal memiliki susunan program sebagai berikut :



Judul Program

Judul program digunakan sebagai nama atau penanda unik terhadap suatu program.

Bentuk Umum : **PROGRAM Nama_Program;**

Contoh : **PROGRAM Contoh1;**

Deklarasi

Tempat untuk mendefinisikan atau mendeklarasikan sesuatu obyek yang akan kita gunakan dalam membuat program pascal. Adapun jenis deklarasi adalah sebagai berikut :

1. Uses

Perintah uses digunakan untuk mendefinisikan unit yang akan kita gunakan dalam membuat program. Unit adalah kumpulan konstanta, tipe data, variable, procedure dan function. Dalam pascal ada beberapa unit yang telah disediakan antara lain : System, Crt, Dos, WinDos, Printer, Graph, Graph3, Turbo3, Overlay, String, WinCrt.

Contoh : **Uses Crt;**

2. Label

Digunakan apabila program menggunakan statement goto untuk melompat ke statement tertentu.

Contoh:

```
Program Contoh_Label;  
Uses Crt;  
Label 10, 20, Keluar;  
Begin  
  clrscr;  
  writeln('STMIK Duta Bangsa');  
  writeln('Surakarta');  
  goto 20;  
10:  
  writeln('Program Pascal');  
  goto Keluar;  
20:  
  Writeln('Ayo Belajar Bersama');  
  goto 10;  
keluar:  
  readln;  
end.
```

3. Const

Adalah suatu besaran yang nilainya tetap atau tidak berubah-ubah.


```

Program Contoh_Label;
Uses Crt;
Const
NIM ='070101002';
Nama='Aditya';
Begin
clrscr;
  writeln('Nomor Induk Mahasiswa :', NIM);
  writeln('Nama Mahasiswa      :', Nama);
  readln;
end.

```

4. Deklarasi Variabel

Adalah suatu besaran yang nilainya dapat berubah-ubah. Variabel yang akan digunakan harus didefinisikan terlebih dahulu.

```

Program Contoh_Variabel;
uses crt;
Var p,l,kel,luas:integer;
begin
  Write('Panjang Persegi Panjang      :'); readln(p);
  writeln;
  write('Lebar Persegi Panjang        :'); readln(l);
  writeln;
  luas:=p*l;
  kel:=2*(p+l);
  Writeln('Luas Persegi Panjang      :',luas:2);
  writeln('Keliling Persegi Panjang :',kel:2);
  readln;
end.

```

5. Type

Adalah definisi nilai dengan membatasi nilai awal maupun nilai akhir dari suatu data.

```

Program Contoh_Type;
uses crt;
Type
pasaran=(Pon,Wage,Kliwon,Legi,Paing);
Var
nama_psr:pasaran;
n:byte;
ket:string[25];
begin
  clrscr;
  Write('Kode Pasaran      :');
  readln(n);

```

```

writeln;
nama_psr:=pasaran(n-1);
case nama_psr of
  Pon,Wage,Kliwon,Legi:ket:='Pasar Ramai Pengunjung'
else
  ket:='Pasar Sepi Pengunjung';
end;
write ('Hari itu  :',ket);
readln;
end.

```

6. Procedure

Adalah suatu program terpisah dalam blok sendiri yang dapat diaktifkan dimanapun dalam program utama.

Contoh Procedure :

```

Program Contoh_Procedure;
Uses Crt;
Procedure Tambah (A,B:byte; Var C:integer);
begin
  C:=A+B;
end;
{***** Program Utama*****}
Var hasil:integer;
begin
  Tambah(4,5,hasil);
  write('4',' ','+', '5',' ','=',' ',hasil);
  readln;
end.

```

7. Function

Seperti juga procedure, function adalah suatu program terpisah dalam blok sendiri yang dapat diaktifkan dimanapun dalam program utama.

Contoh Function :

```

Program Contoh_Function;
Uses Crt;
Function Tambah(A,B:byte):integer;
begin
  Tambah:=A+B;
end;
{***** Program Utama*****}
Var hasil:integer;
    bil1,bil2:byte;
begin
  Clrscr;

```

```

write('Bilangan 1:');readln(bil1);
write('Bilangan 2:');readln(bil2);
writeln;
Hasil:=Tambah(bil1,bil2);
write(bil1,' ','+',bil2,' ','=',' ',hasil);
readln;
end.

```

B. HotKey

Gunakan tombol hotkey dibawah ini untuk mempermudah anda mengoperasikan program Pascal.

Tabel 3.1 Contoh Tombol HotKey di Turbo Pascal

Tombol	Fungsi
Ctrl + F9	Compile dan Run
Alt + F9	Compile
F2	Save File
F3	Open File
Alt + F3	Close Active Window
Alt + F5	OutPut
F6	Toggle Active Window
Shift + panah	Blok
Ctrl + Ins	Copy Blok
Shift+ Ins	Paste Blok
Shift+ Del	Cut Blok
Alt + Backspace	Undo
Alt + X	Keluar dari editor

C. Aturan Pemrograman Pascal

1. Akhir sebuah program Pascal ditandai dengan tanda baca titik (.) setelah END yang paling akhir.
2. Tanda titik koma (;) merupakan pemisah antar instruksi satu dengan lainnya.
3. Beberapa statement boleh ditulis menjadi satu baris dipisahkan dengan tanda baca titik koma (;)

4. Bersifat *non case sensitive* atau tidak membedakan huruf besar dan huruf kecil dalam hal penulisan perintah.
5. Contoh : `simpan := akhir; akhir := simpan + awal;`
6. Baris komentar diletakkan diantara tanda (* dan *) atau diantara tanda { dan }

Contoh :

```
Var    rerata : real;      (*nilai rata-rata*)
Nil1      : real;      {nilai ujian}
```

E. Komponen Dasar Program Pascal

Pola susun bahasa Pascal dibentuk dengan menggunakan komponen bahasa pemrograman yang umum, yaitu :

1. Simbol Dasar

Simbol dasar terdiri atas :

- a. Simbol huruf, yaitu huruf A sampai dengan Z atau a sampai dengan z.
- b. (huruf besar dan kecil).
- c. Simbol angka atau digit yaitu : 0,1,2,3,4,5,6,7,8,9.
- d. Simbol khusus, yaitu : + - * / ; := , ' = < > <= >= <> : { }
() []

2. Reserved Word (kata pasti)

Reserved Word adalah suatu kata yang secara mutlak tidak boleh diartikan lain dan harus digunakan sebagaimana yang telah didefinisikan atau ditentukan kegunaanya oleh bahasa Pascal. Reserved word ini tidak boleh didefinisikan ulang oleh pemakai, sehingga tidak dapat dipergunakan sebagai pengenalan (identifikasi). Reserved Word ini jumlahnya berbeda untuk masing-masing bahasa Pascal.

Contoh beberapa reserved word yang telah didefinisikan oleh bahasa pascal antara lain :

AND	ELSE	LABEL	SET
ARRAY	END	OF	TYPE
BEGIN	FUNCTION	OR	UNTIL
CASE	FOR	PROCEDURE	VAR
CONST	GOTO	PROGRAM	WHILE
DO	IF	RECORD	WITH
DOWNTON	IN	REPEAT	dsb

3. Identifier (penyebut)

Identifier merupakan sebuah kata yang digunakan sebagai nama atau sebutan terhadap sesuatu didalam program. Pemakai dapat mendefinisikan sendiri suatu nama sebagai identifier. Identifier ini terdiri atas :

- a. Identifier Standar, yaitu identifier yang telah didefinisikan oleh bahasa pascal. Contoh dari Identifier standar ini antara lain:

ABS	LN
ARCTAN	ODB
BOOLEAN	PRED
CHAR	ROUND
CHR	READ
COS	READLN
EOF	SQR
EOLN	SQRT
EXP	SUCC

Dan masih banyak lagi.

- b. Identifier Non Standar; yaitu identifier yang didefinisikan oleh pemakai bahasa pascal; misalnya;
 - 1) nama suatu program
 - 2) nama suatu konstanta
 - 3) nama suatu variabel
 - 4) nama suatu procedure

Identifier ini bebas, tetapi dengan ketentuan-ketentuan sebagai berikut :

- a. terdiri dari gabungan huruf dan angka dengan karakter pertama harus berupa huruf. Huruf besar dan huruf kecil dianggap sama.
- b. Tidak boleh mengandung blank.
- c. Tidak boleh mengandung simbol-simbol khusus, kecuali garis bawah.
- d. Panjangnya bebas, tetapi hanya 63 karakter pertama yang dianggap signifikan.

Contoh :

Tabel 3.2 Contoh Pemberian Nama Identifier

Identifier	Keterangan
GajiKaryawan	Benar
No_Mhs	Benar
P3K	Benar
1X	Salah, karakter pertama harus huruf
A&B	Salah, tidak boleh mengandung simbol khusus
A B	Salah, tidak boleh mengandung blank

F. Perintah Dasar Pemrograman Pascal

1. Writeln

Writeln digunakan untuk menampilkan informasi ke layar. Writeln juga dapat menerima argumen bertipe data dasar (real, integer) dan bahkan string. Contoh :

```
Writeln(5.5);           {tipe data real}
Writeln(67);            {tipe data integer}
Writeln(false);         {tipe data boolean}
Writeln('K');           {tipe data Char}
Writeln('STMIK Duta Bangsa') {tipe data string}
```

2. Write

Fungsi dari perintah write sama dengan perintah writeln, perbedaan perintah write dan writeln adalah letak kursor setelah text ditampilkan

ke layar atau penambahan LF (*linefeed*) dan CR (*Carriage Return*). Jika menggunakan perintah `write`, setelah text tampil dilayar maka kursor akan terletak dikanan text. Tetapi, perintah `writeln` akan meletakkan kursor dibawah text yang ditampilkan.

Contoh :

Program Contoh_Write;

Uses Crt;

Begin

Clrscr;

Write ('STMIK DUTA BANGSA SURAKARTA');

Readln;

End.

Jalankan dan apa yang terjadi?

Program Contoh_Writeln;

Uses Crt;

Begin

Clrscr;

Writeln ('STMIK DUTA BANGSA SURAKARTA');

Readln;

End.

Jalankan dan apa yang terjadi?

Kedua program diatas menghasilkan dua tampilan yang berbeda yaitu :

STMIK DUTA BANGSA SURAKARTA _

Dan

STMIK DUTA BANGSA SURAKARTA

—

Kemudian buat program dibawah ini dan apa hasilnya : ???

Program Contoh_1;

Uses Crt;

Begin

Clrscr;

Write ('Pergi ke Makasar Naik Kapal');

Writeln ('Hati-hati jangan terjepit');

Write ('Ayo Belajar Bahasa Pascal');

Writeln ('Kata Upik Tidaklah Sulit');

Readln;

End.

3. Format write atau writeln

Turbo pascal menyediakan cara yang mudah untuk menformat tampilan bilangan pada layar sesuai dengan notasi yang kita pakai sehari-hari. Hal ini sangat penting untuk memperjelas hasil kelaran dari suatu progra,m, apalagi jika bilang bertipe data real. Untuk bilang tipe data integer dapat kita tulis dengan format :

Writeln(data:lebar_data);

Dengan menggunakan format ini maka barisan data yang ingin kita tampilkan akan ditampilkan dengan rata kanan. Contoh :

```
Writeln (12356:8);
```

```
Writeln (12:8);
```

```
Writeln (1235:8);
```

Hasilnya :

```
12356
```

```
12
```

```
1235
```

Pengaturan yang sama juga dapat kita lakukan untuk tipe data real.

Untuk tipe data real dapat kita format dengan perintah berikut :

Writeln(data:lebar_data:lebar_desimal);

Contoh :

```
Writeln(1253.53:5:2);
```

Contoh :

```
1253.53
```

4. GotoXY

Fungsi perintah ini adalah untuk menempatkan kursor pada koordinat tertentu. Sintaks :GotoXY(kolom,baris)

Contoh :

```
Program Contoh_2;  
Uses Crt;  
begin  
  Clrscr;  
  gotoXY(10,10);  
  writeln('STMIK Duta Bangsa');  
end.
```

5. Readln

Fungsi ini berarti meminta input nilai dari keyboard dan berfungsi untuk menghentikan proses eksekusi program selama belum ada penekanan tombol ENTER.

Contoh :

```
Program Contoh_3;  
Uses Crt;  
Var nama:string[25];  
begin  
  Clrscr;  
  gotoXY(25,10);  
  Write('Masukkan Nama Anda :');  
  readln(nama); {menerima input dari user}  
  gotoXY(15,15);  
  Write('Saudara',' ',nama,' ','Anda Memasuki Dunia Tanpa Batas');  
  GotoXY(15,17);  
  Write('Anda Sudah Siap ???');  
  readln; {berhenti sebentar sampai tekan ENTER}  
end.
```

6. TextColor

Fungsi perintah ini adalah untuk memberi warna pada text.

Sintak: TextColor(Red); atau TextColor(4);

```
Program Contoh_4;  
Uses Crt;  
Var nama:string[25];  
begin  
  Clrscr;  
  gotoXY(25,10);  
  textcolor(3);  
  Write('Masukkan Nama Anda :');readln(nama);  
  gotoXY(15,15);  
  textcolor(12);  
  Write('Saudara',' ',nama,' ','Anda Memasuki Dunia Tanpa Batas');
```

```
GotoXY(15,17);
textcolor(13);
Write('Anda Sudah Siap ???');
readln;
end.
```

Tabel 3.3 Kode Warna TextColor

Kode	Warna	Kode	Warna
0	Black	8	Darkgray
1	Blue	9	LightBlue
2	Green	10	LightGreen
3	Cyan	11	LightCyan
4	Red	12	LightRed
5	Magenta	13	LightMagenta
6	Brown	14	Yellow
7	Lightgray	15	White

7. TextBackground

Fungsi perintah ini adalah untuk memberi warna latar belakang dari suatu tulisan.

Contoh :

Program Contoh_5;

```
Uses Crt;
Var nama:string[25];
begin
  Clrscr;
  gotoXY(25,10);
  textcolor(13);textbackground(1);
  Write('Masukkan Nama Anda :');readln(nama);
  gotoXY(15,15);
  textcolor(12);textbackground(3);
  Write('Saudara',' ',nama,' ','Anda Memasuki Dunia Tanpa Batas');
  GotoXY(15,17);
  textcolor(14); textbackground(2);
  Write('Anda Sudah Siap ???');
  readln;
end.
```

Tabel 3.4 Kode Warna TextBackGround

Kode	Warna
0	Black
1	Blue
2	Green
3	Cyan
4	Red
5	Magenta
6	Brown
7	lightgray

8. DelLine

Digunakan untuk menghapus satu baris pada layar pada posisi kursor.

9. InsLine

Digunakan untuk menyisipkan satu baris kosong pada posisi kursor.

10. Readkey

Digunakan untuk membaca sebuah karakter yang diketikkan, bisa juga menghentikan sementara sampai tekan sembarang karakter.

11. KeyPressed

Bertujuan untuk memeriksa apakah telah dilakukan penekanan terhadap tombol tertentu.

12. Sound

Bertujuan untuk membuat Suara

Contoh :

Program Contoh_Suara;

```
Uses Crt;
Var j :integer;
begin
  Clrscr;
  gotoXY(25,10);
  textcolor(14);
  Write('Anda Sudah Siap ???');
  j:=10;
  textAttr:=7;
  textcolor(14);gotoxy(15,20);
  Write('Tekan Sembarang Tombol untuk keluar....');
  while not keypressed do
  begin
    sound(j*10);sound(j+10*10);
```

```

    sound(j+20*10);delay(30);
    j:=j+10;
    if j>=300 then j:=10;
    end;
    NoSound;
    readln;
end.

```

G. Komentar

Program yang kita buat, bukan hanya dibaca oleh komputer melainkan bisa juga dibaca oleh orang lain. Karena itu, kejelasan program menjadi satu hal yang penting agar program dapat dimengerti oleh orang lain atau pemrogram itu sendiri. Kejelasan program akan menolong anda jika suatu saat anda lupa apa maksud program yang telah anda buat. Untuk memberikan kejelasan program yang anda buat, anda dapat menambahkan komentar atau keterangan.

Komentar adalah suatu tulisan pada program yang tidak berarti apa-apa pada komputer tetapi sangat berguna bagi orang karena berfungsi sebagai dokumentasi. Komentar dalam pascal biasa diletakan diantara kurung kurawal {komentar} atau dengan tanda (*Komentar*).

Contoh :

```

{----- Menulis Program Pascal -----}
{Program ini adalah contoh program pascal sederhana}
{Oleh : }
{NIM      : 080101001}
{Nama     : Abram}

```

Daftar Pustaka

Abdul Kadir, 2004, *Pemrograman Pascal*, Edisi I, Penerbit Andi, Yogyakarta
Budi Raharjo, 2005, *Teknik Pemrograman Pascal*, Informatika, Bandung
Mico Pardosi, 1999, *Bahasa Pemrograman Turbo Pascal 7.0*, Penerbit Indah, Surabaya
<http://www.a-rang.info> (diakses tanggal 22 Januari 2009)
<http://materi-praktek.blogspot.com/2007/03/bab-1-mengenal-tampilan-turbo-pascal.html>
(diakses tanggal 22 Januari 2009)

Latihan Bab 3

1. Bagaimana susunan bahasa pascal?
2. Apa perbedaan variabel dan konstanta ?
3. Perhatikan Program dan apa hasilnya?

```
Program Coba Dulu;  
Begin  
    Write('Halo');  
    Readln;  
End.
```

4. Bagaimana aturan dalam pemrograman pascal?
5. Tunjukkan letak kesalahan program berikut dan betulkanlah !

Program 1

```
Program Coba Dulu;  
Begin  
    Write('Halo');  
    Readln;  
End;
```

Program 4

```
{Program Pertama};  
Begin;  
    Write('Halo');  
    Readln;  
End.
```

Program 2

```
Program Coba2;  
Begin  
    Clrscr;  
    Write('Halo');  
    Readln;  
End;
```

Program 3

```
Program Coba3;  
Var mod:integer;  
Begin  
    Clrscr;  
    Write('angka :',mod);  
    Readln;  
End;
```

Kompetesi Dasar :

1. Mahasiswa dapat mengetahui dan menggunakan tipe data yang ada dalam bahasa pemrograman pascal
2. Mahasiswa dapat mengetahui dan menggunakan jenis operator yang ada dalam bahasa pemrograman pascal

A. Tipe data yang disediakan oleh PASCAL meliputi:

1. Tipe Data Sederhana

merupakan tipe data dasar yang sering dipakai oleh program, meliputi: integer (bilangan bulat), real (bilangan pecahan), char (alphanumeric dan tanda baca), dan boolean (logika). Untuk data integer dan real masing-masing terbagi menjadi beberapa kategori

a. Bilangan Integer

merupakan tipe data berupa bilangan bulat, terbagi atas beberapa kategori seperti terlihat dalam tabel 1. tabel 1 menunjukkan jenis data, ukuran dalam memori dan rentang nilainya. Berikut ini adalah rentang nilai tipe data integer :

Tabel 4.1 Tipe Data Integer

Tipe Data	Ukuran Tempat	Rentang Nilai
Byte	1 byte	0 s/d +255
Shortint	1 byte	-28 s/d +127
integer	2 bytes	-32768 s/d 32767
Word	2 bytes	0 s/d 65535
Longint	4 bytes	2147483648 s/d 2147483647

b. Bilangan Real

Bilangan real atau nyata merupakan jenis bilangan pecahan, dapat dituliskan secara biasa atau model scientific . Contoh bilangan real: 34.265 -

3.55 0.0 35.997E+11, dimana E merupakan simbol perpangkatan 10. Jadi 452.13 mempunyai nilai sama dengan 4.5213e2. Penggolongan tipe data bilangan real adalah sebagai berikut :

Tabel 4.2 Tipe Data Real

Tipe Data	Ukuran Tempat	Rentang Nilai
real	6 bytes	2.9×10^{-39} s/d 1.7×10^{38}
single	4 bytes	1.5×10^{45} s/d 3.4×10^{38}
double	8 bytes	5.0×10^{-324} s/d 1.7×10^{308}
extended	10 bytes	3.4×10^{-4932} s/d 1.1×10^{4932}
comp	8 bytes	-9.2×10^{18} s/d 9.2×10^{18}

c. Char

Tipe data ini menyimpan karakter yang diketikkan dari keyboard, memiliki 266 macam yang terdapat dalam tabel ASCII (American Standard Code for Information Interchange). Contoh: 'a' 'B' '+', dsb. Yang perlu diingat bahwa dalam menuliskannya harus dengan memakai tanda kutip tunggal. Jenis data ini memerlukan alokasi memori sebesar 1(satu) byte untuk masing-masing data.

d. Tipe Data Boolean

Merupakan tipe data logika, yang berisi dua kemungkinan nilai: TRUE (benar) atau FALSE (salah). Turbo Pascal for Windows memiliki tiga macam jenis ini yaitu: Boolean, WordBool, dan LongBool. Tipe boolean memakai memori paling kecil, sedangkan WordBool dan LongBool dipakai untuk menulis program yang sesuai dengan lingkungan Windows.

2. Tipe Data Terstruktur

Tipe ini terdiri atas : array, record, set, dan file. String adalah tipe data jenis array, tetapi karena string memiliki kekhasan tersendiri sebagai array dari karakter maka penulis perlu memberikan penjelasan tersendiri.

a. String

Merupakan suatu data yang menyimpan array (larik), sebagai contoh 'ABCDEF' merupakan sebuah konstanta string yang berisikan 6 byte karakter. Ukuran Tempat untuk tipe data ini adalah 2 s/d 256 byte, dengan jumlah elemen 1 s/d 255. String dideklarasikan dengan string [konstanta] atau string. Bila ukuran string tidak didefinisikan maka akan banyak memakan ruang, karena ukuran string menyesuaikan dengan defaultnya. Misalkan

var kata: string [20]; atau var kata: string; karena string merupakan array dari karakter. Maka kata[1] merupakan karakter pertama dari string, kemudian kata[2], merupakan elemen kedua, dst.

Routines pada Pascal untuk manipulasi string :

1) Length : menghasilkan panjang string.

Syntax : length(s)

Contoh : n:=length(s);

Misal s:='pemrograman'; n akan sama dengan 11.

2) Copy : mengkopi suatu ekstrak karakter dari sebuah string.

Syntax : copy(s,dari,banyaknya)

Contoh : st:=copy(s,5,3); mengekstrak 3 karakter dari s, dimulai dari karakter kelima.

Misal s:='Who are you ?'; st adalah 'are'.

catatan: bila indek lebih besar dari panjang string maka hasilnya akan kosong:

Contoh : st:=copy(s,15,4); { kosong} jika banyaknya string yang akan diekstrak lebih besar dari s, maka hasilnya adalah bagian yang tersisa dari string.

Contoh : st:=copy(s,9,10); st hasilnya adalah 'you ?'

- 3) **Pos** : menghasilkan posisi dari suatu substring dari suatu string.
 Syntax : Pos(substr,s)
 Contoh : n:=pos('are','Who are you ?'); { n:=5; }
 Jika substring tidak ditemukan, maka hasilnya 0.
- 4) **Val** : mengkonversi string ke numerik.
 Syntax : val(strvar,numvar,errorcode)
- 5) **strvar** : adalah variabel string yang akan dikonversi, numvar variabel numerik (bisa integer atau real, dan errorcode adalah variabel integer yang menyimpan error code (kode kesalahan). Jika errorcode adalah 0, konversi berhasil. Sebaliknya, hal ini akan menunjukkan posisi dimana pada strvar yang menyebabkan kegagalan.
- 6) **Str** : mengkonversi numerik ke string.
 Syntax : str(numvar,strvar)
- 7) **Concat** : Menggabungkan dua atau lebih string.
 Syntax : concat(s1,s2,...,sn)
- 8) **Insert** : Menyisipkan suatu string ke dalam string yang lain.
 Syntax : insert(asal,target,indek)
- 9) **Delete** : menghapus n karakter dari string s berawal dari indek i.
 Syntax : delete(s,i,n);
- 10) **Fillchar** : mengisi string s dengan karakter c sampai s sama dengan n-1 panjang karakternya.
 Syntax : fillchar(s,n,c); Hati-hati : s[0] akan tertumpuki, jangan lupa menambahkan s[0]:=chr(n-1); untuk menormalkannya.

b. Subrange

Adalah suatu range yang menunjukkan nilai terkecil dan nilai terbesar yang akan digunakan. Deklarasi subranngge berbetuk *Type*

c. Enumerated

Atau tipe data skalar menunjukkan kumpulan dari nilai yang urutannya sudah pasti. Nilai dari tipe yang dideklarasikan ini akan diwakili dengan pengenalan-pengenalan yang akan menjadi suatu nilai konstanta.

d. Tipe File

File atau berkas adalah kumpulan sejumlah komponen yang bertipe data sama, jumlahnya tidak tertentu dan biasanya tersimpan dalam penyimpanan luar.

e. Tipe Pointer

Pointer merupakan variabel khusus yang berisi suatu address (alamat) di lokasi lain didalam memory. Suatu variabel yang points(menunjuk) ke sesuatu sehingga disebut pointer.

B. Operator

Operator dalam pascal dikelompokkan menjadi :

1. Aritmatika

Tabel 4.3 Operator Aritmatika

Operator	Operasi	Tipe Operand	Tipe Hasil Operasi
+	Penjumlahan	Integer, real	Integer, real
-	Pengurangan	Integer, real	Integer, real
*	Perkalian	Integer, real	Integer, real
/	Pembagian	Integer, real	Integer, real
div	Pembagian	integer, integer	integer
mod	Sisa pembagian	integer, integer	integer

2. Boolean

Tabel 4.4 Operator Boolean

Operator	Operasi	Tipe Operand	Tipe Hasil Operasi
not	negasi	boolean	boolean
and	logika 'and'	boolean	boolean
or	logika 'or'	boolean	boolean
xor	logika 'xor'	boolean	boolean

3. Relasional

Tabel 4.5 Operator Relasional

Operator	Operasi	Tipe Operand	Tipe Hasil Operasi
=	Sama dengan	tipe sederhana, string, pointer dan set	boolean
<>	Tidak sama dengan	tipe sederhana, string, pointer dan set	boolean
<	Lebih kecil dari	tipe sederhana, string	boolean
>	Lebih besar dari	tipe sederhana, string	boolean
<=	Lebih kecil atau =	tipe sederhana, string	boolean
>=	Lebih besar atau =	tipe sederhana, string	boolean

4. Set

Tabel 4.6 Operator Set

Operator	Operasi	Tipe Operand	Tipe Hasil Operasi
=	Sama dengan	Set, set	boolean
<>	Tidak sama dengan	Set, set	boolean
<=	Lebih kecil atau =	Set, set	boolean
>=	Lebih besar atau =	Set, set	boolean

5. Fungsi Matematika Standar dalam Pascal

Tabel 4.7 Fungsi Matematika Standar

Nama Fungsi	Deskripsi	Tipe Argumen	Tipe Hasil Operasi
abs	absolute value	real/integer	real/integer
arctan	arctan (radian)	real/integer	real
cos	cosine (radian)	real/integer	real
sin	sin (radian)	real/integer	real
exp	fungsi Perpangkatan e	real/integer	real
ln	ln	real/integer	real

round	Pembulatan terdekat	real	integer
sqr	kuadrat	real/integer	real/integer
sqrt	Akar kuadrat	real/integer	real
trunc	Pembulatan ke bawah	real/integer	integer

6. Simbol yang Memiliki Arti Khusus

Tabel 4.8 Simbol yang memiliki arti khusus

Simbol	Keterangan dan Contoh
:=	Assign Value (Diisi dengan) Contoh : Gol:='A'; gaji_pokok:=2500000;
=	Equal to (sama dengan) Contoh : if gol = 'A' then gaji_pokok:=2500000;
:	Mendefinisikan sebuah nama Contoh : Var A: char;
;	Akhir sebuah kalimat (statement) Contoh : Writeln('STMIK Duta Bangsa Surakarta);
.	Akhir dari sebuah program Contoh : End.

Daftar Pustaka

Abdul Kadir, 2004, *Pemrograman Pascal*, Edisi I, Penerbit Andi, Yogyakarta
Budi Raharjo, 2005, *Teknik Pemrograman Pascal*, Informatika, Bandung
Mico Pardosi, 1999, *Bahasa Pemrograman Turbo Pascal 7.0*, Penerbit Indah, Surabaya
<http://www.boxpascal.110mb.com/Contoh%20Program%20Pascal.html> (Diakses Tanggal 14 Januari 2009)
<http://www.a-rang.info> (diakses tanggal 22 Januari 2009)

Latihan Bab 4

1. Perhatikan program berikut dan apakah hasilnya? kenapa? :

```
Program luas_segitiga;  
Uses crt;  
Var alas,tinggi,luas : integer;  
Begin  
  Clrscr;  
  alas:=5;  
  tinggi:=7;  
  luas:=0.5*alas*tinggi;  
  write('Luas  :',luas);  
  readln;  
end.
```

2. Tuliskan ekspresi berikut dengan menggunakan bahasa pascal :

a. $y = mx + c$

b. $y = x^2 + 3x + 3$

c. $y = mc^2$

d. $c = \sqrt{a^2 + b^2}$

3. Buatlah Program untuk menghitung Luas dan Keliling Lingkaran
4. Buatlah program untuk menampilkan hasil operasi aritmatika dari dua buah bilangan positif. Berikut ini diberikan contoh yang diinginkan bila dimasukkan dua buah bilangan yaitu 15 dan 4.

Masukkan Bilangan 1 : 15

Masukkan Bilangan 2 : 4

Hasil Operasi

15 + 4 = 19

15 - 4 = 11

15 * 4 = 60

$15 \text{ Div } 4 = 3$

$15 \text{ Mod } 4 = 3$

5. Buatlah program untuk menghitung konversi suhu dari **Celcius ke Reamur, Fahrenheit dan Kelvin.**

Celcius : 100 (input)

Reamur : 80

Fahrenheit : 212

Kelvin : 373

6. Buatlah program untuk mengkonversi nilai suatu angka kedalam Kilogram (kg), Ons dan gram. Misal kita masukkan angka 1550 maka bilangan tersebut akan dikonversi menjadi 1 kg 5 Ons 50 gram

Kompetesi Dasar :

3. Mahasiswa dapat memahami dan menggunakan logika percabangan dengan menggunakan cara IF dalam berbagai variasi
4. Mahasiswa dapat memahami dan menggunakan logika percabangan dengan menggunakan cara CASE
5. Mahasiswa dapat memahami dan menggunakan logika AND dan OR

A. Percabangan

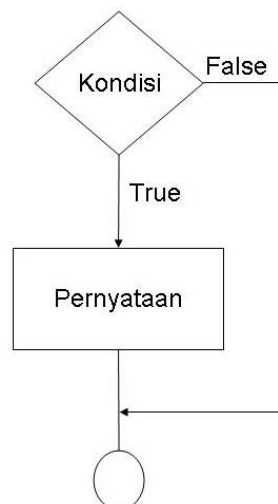
Percabangan adalah suatu pemilihan alur program yang didasarkan atas kondisi tertentu. Percabangan digunakan bila suatu saat kita harus memilih pada suatu kondisi tertentu. Percabangan dalam Turbo Pascal ada dua bentuk :

1. Bentuk IF

Bentuk percabangan IF dibagi dalam beberapa bentuk antara lain :

a. IF (Kondisi) Then (Pernyataan benar);

Merupakan suatu bentuk percabangan yang paling sederhana. Struktur percabangan if memiliki satu buah kondisi yang akan diuji kebenarannya. Jika hasil pengujian bernilai benar, maka program akan melakukan pernyataan yang diberikan.



Gambar 5.1 Flowchart IF dengan satu pernyataan

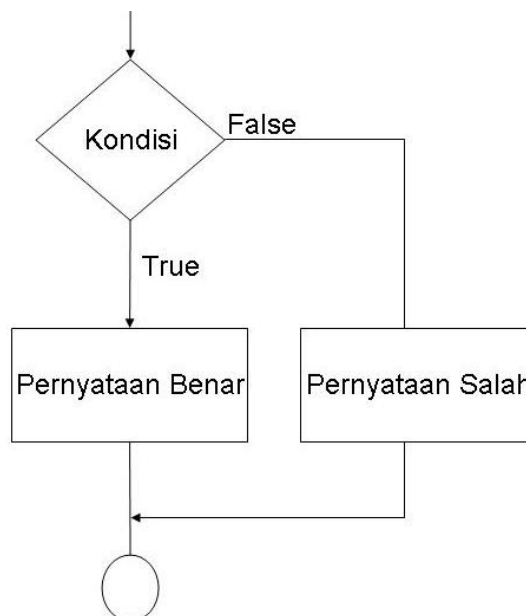
Contoh :

Program Contoh_IF1;

```
uses crt;  
Var nilai:byte;  
begin  
  clrscr;  
  textcolor(13);  
  write('Masukkan Nilai Anda :');  
  readln(nilai);  
  if nilai >=45 then write('LULUS');  
  readln;  
end.
```

b. IF (Kondisi) Then (Pernyataan Benar) else (Pernyataan Salah) ;

Struktur percabangan if..else memiliki satu buah kondisi yang akan diuji kebenarannya dan dua buah pernyataan. Jika hasil pengujian bernilai benar, maka program akan melakukan pernyataan untuk yang benar, dan jika bernilai salah akan melakukan pernyataan untuk yang salah.



Gambar 5.2 Flowchart IF dengan dua pernyataan

Contoh :

Program Contoh_IF2;

```
uses crt;  
Var nilai:byte;  
begin  
  clrscr;  
  textcolor(13);
```

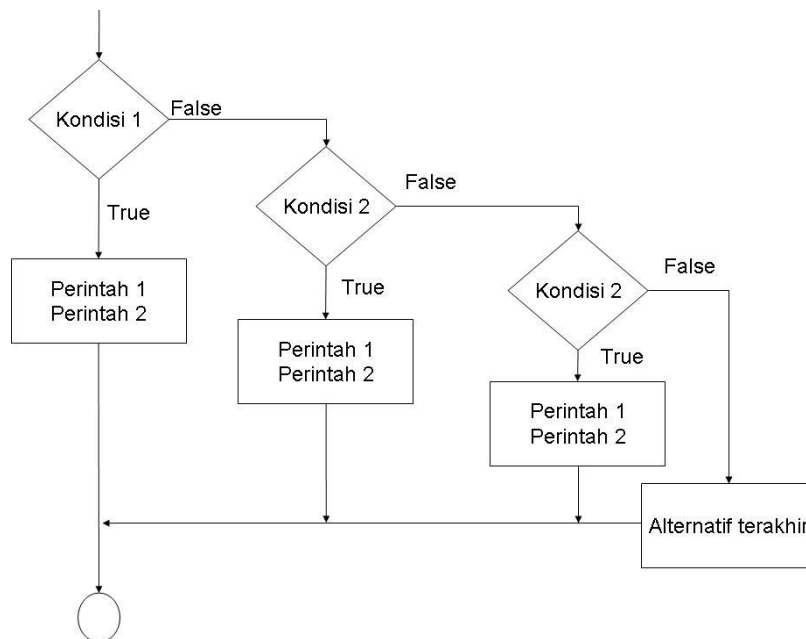
```

write('Masukkan Nilai Anda :');
readln(nilai);
if nilai >=45 then write('Anda Berhasil')
  else write(' Anda Gagal');
readln;
end.

```

c. IF (Kondisi) Then Begin End else Begin End;

Digunakan bila pernyataan benar atau pernyataan salah yang akan dikerjakan lebih dari satu perintah.



Gambar 5.3 Flowchart IF dengan lebih dari dua kondisi dan lebih dari satu perintah

Contoh:

Program Contoh_IF3;

```

uses crt;
Var nama:string[25];
    gol :char;
    gaji,tjabat,Tot :real;
begin
  clrscr;
  textcolor(14);
  write('Nama Pegawai :');readln(nama);
  write('Golongan      :');readln(gol);
  if (gol='A') or (gol='a') then
  begin
    gaji:=1000000;
    tjabat:=75000; } Menghitung gaji dan tunjangan jabatan

```

```

end
else
begin
    gaji:=1500000;
    tjabat:=100000;
end;
tot:=gaji+tjabat;
clrscr;
writeln('Nama Pegawai      :',nama);
writeln('Golongan          :',gol);
writeln('Gaji Pokok           :',gaji:7:2);
writeln('Tunjangan Jabatan: ',tjabat:7:2);
readln;
end.

```

d. IF (Kondisi) Then Begin End

Else IF (Kondisi) Then Begin End

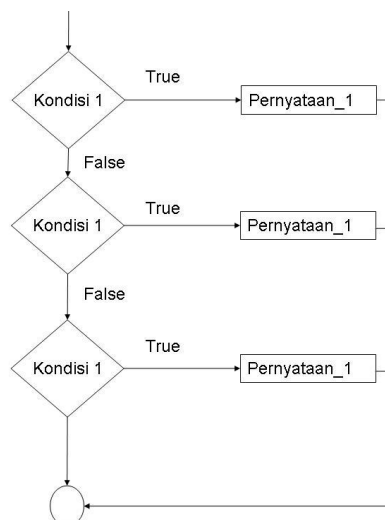
Else IF (Kondisi) Then Begin End

Else BeginEnd;

Digunakan untuk percabangan dengan banyak kondisi dan banyak perintah untuk setiap kondisinya.

2. Bentuk Case

Struktur case adalah generalisasi struktur if, kalau pada struktur if hanya menyediakan dua pilihan yaitu benar dan salah namun dalam struktur case dimungkinkan untuk memilih salah satu diantara banyak pilihan.



Gambar 5.4 Flowchart CASE

Bentuk umum:

Case kondisi of

Kasus 1 : perintah 1;

Kasus 2 : perintah 2;

.....

.....

Kasus n : perintah n;

End;

Atau :

Case kondisi of

Kasus 1 : perintah 1;

Kasus 2 : perintah 2;

.....

.....

Kasus n : perintah n;

Else

Perintah_lainnya;

End;

Agar lebih jelas pemahamannya perhatikan contoh berikut :

Program Contoh_Case;

```
Uses Crt;
Var angka:byte;
begin
  clrscr;
  write('Masukkan angka:');
  readln(angka);
  writeln;
  case angka of
    1:write('Bulan dengan angka :',' ',angka,' ','Januari');
    2:write('Bulan dengan angka :',' ',angka,' ','februari');
    3:write('Bulan dengan angka :',' ',angka,' ','Maret');
    4:write('Bulan dengan angka :',' ',angka,' ','April');
    5:write('Bulan dengan angka :',' ',angka,' ','Mei');
    6:write('Bulan dengan angka :',' ',angka,' ','Juni');
    7:write('Bulan dengan angka :',' ',angka,' ','Juli');
    8:write('Bulan dengan angka :',' ',angka,' ','Agustus');
    9:write('Bulan dengan angka :',' ',angka,' ','September');
```

```

10:write('Bulan dengan angka :',' ',angka,' ','Oktober');
11:write('Bulan dengan angka :',' ',angka,' ','November');
12:write('Bulan dengan angka :',' ',angka,' ','Desember');
else
  write('Tidak ada bulan dengan lambang angka :',' ',angka);
end;
readln;
end.

```

Bentuk Case juga dapat digunakan untuk kondisi yang berupa range yang mempunyai nilai awal dan nilai akhir dengan perintah yang lebih ringkas.

Contoh berikut akan mempertajam pemahaman anda :

Program Contoh_Case_Range;

```

Uses Crt;
Var nilai:byte;
begin
  clrscr;
  write('Masukkan Nilai:');
  readln(nilai);
  writeln;
  case nilai of
    0..40 : write ('Niali Anda E');
    41..60 : write ('Nilai Anda D');
    61..75 : write ('Nilai Anda C');
    76..85 : write ('Nilai Anda B');
    86..100: write ('Nilai Anda A');
  else
    write('Nilai yang anda masukkan menyimpang dari aturan');
  end;
  readln;
end.

```

Catatan : jika perintah yang dikerjakan dalam case lebih dari satu maka harus diawal

Begin dan diakhiri dengan End

3. Pemakaian AND dan OR

Kadang-kadang untuk memutuskan sesuatu, kita tidak cukup satu syarat tetapi dibutuhkan dua syarat atau lebih. Untuk dapat menggunakan syarat tersebut kita bisa menggunakan AND atau OR. Pernyataan AND digunakan jika syarat harus terpenuhi semuanya dan pernyataan OR digunakan bila salah satu syarat sudah terpenuhi.

contoh :

Program Contoh_AND;

```
Uses Crt;  
Var a:integer;  
Begin  
  Clrscr;  
  writeln('Masukkan Sebuah Bilangan :');readln(a);  
  if (a>=1) and (a<=4) then write('Benar')  
    else write('Salah');  
  readln;  
end.
```

Program diatas akan menghasilkan tulisan “Benar” jika angka yang dimasukkan antara 1 sampai dengan 4 dan akan menghasilkan tulisan “Salah” jika angka yang dimasukkan selain 1,2,3, dan 4.

Contoh Program dengan menggunakan pernyataan OR dapat dilihat pada program berikut :

Program Contoh_OR;

```
Uses Crt;  
Var jawab:char;  
Begin  
  Clrscr;  
  writeln('Keluar dari Program [Y/T]:');  
  readln(jawab);  
  if (jawab='T') or (Jawab='t') then write('Jawaban',jawab)  
    else exit;  
  readln;  
end.
```

Daftar Pustaka :

Abdul Kadir, 2004, *Pemrograman Pascal*, Edisi I, Penerbit Andi, Yogyakarta
Budi Raharjo, 2005, *Teknik Pemrograman Pascal*, Informatika, Bandung
Mico Pardosi, 1999, *Bahasa Pemrograman Turbo Pascal 7.0*, Penerbit Indah, Surabaya
<http://www.boxpascal.110mb.com/Contoh%20Program%20Pascal.html> (Diakses tanggal 14 Januari 2009)
<http://rosihanari.net/programming-pascal.php> (Diakses tanggal 22 Januari 2009)

Latihan Bab 5 :

1. Ubahlah sistem penilaian Model **Case** (Studi Kasus Nilai hal. 54) diatas dengan Model **IF**
2. Buatlah program untuk mendeteksi bilangan yang dimasukkan adalah bilangan ganjil, bilangan genap ! (gunakan IF)
3. Buatlah program yang dapat mendeteksi angka yang anda masukkan
 - a. Negatif jika angka kurang dari 0
 - b. Positif jika angka lebih dari 0
 - c. Nol bila angka sama dengan 0 (gunakan IF)
4. Apakah program nomor 2 dan 3 dapat diganti dengan menggunakan CASE? Buatlah programnya !

B. Pengulangan

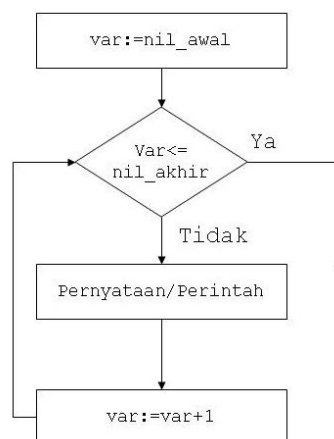
Kompetesi Dasar :

1. Mahasiswa dapat memahami konsep perulangan dengan menggunakan FOR, WHILE dan REPEAT
2. Mahasiswa dapat menyelesaikan soal dengan menggunakan metode pengulangan FOR, WHILE dan REPEAT

Perulangan (*iterasi*) adalah proses yang berulang. Iterasi selalu ada dalam bahasa pemrograman apapun, karena disinilah letak kelebihan komputer dibanding manusia, yaitu mampu melakukan hal yang sama berulang kali tanpa kesalahan akibat bosan atau lelah. Dengan perulangan, program menjadi lebih pendek dan sederhana. Dalam Pascal dikenal tiga macam perintah (*statement*) perulangan, yaitu *statement* for...do, repeat...until dan while...do. Perulangan for...do adalah perulangan dengan penghitung (*counter*), perulangan repeat...until adalah perulangan dengan syarat akhir sedang perulangan while...do adalah perulangan dengan syarat awal.

1. Struktur For .. Do

a) Perulangan Positif



Gambar 5.5 Flowchart Perulangan FOR

Perulangan dilakukan dengan menaikkan counter tiap kali perulangannya. Bentuk umumnya :

```
For Counter:= nilai_awal To nilai_akhir Do
```

Bentuk diatas yang diulang-ulang hanya satu baris perintah setelah Do, jika perintah yang diulang lebih dari satu perintah maka bentuk umumnya menjadi :

```
For Counter:= nilai_awal To nilai_akhir Do  
Begin  
    Perintah1;  
    Perintah2;  
    Perintah3;  
End;
```

Beriku ini adalah contoh-contoh program dengan menggunakan For (perulangan positif) :

```
Program ContohFor1;  
Uses Crt;  
var i:integer;  
Begin  
    clrscr;  
    for i:= 1 to 10 do  
        writeln('STMIK DUTA BANGSA SURAKARTA');  
        readln;  
    end;
```

Contoh program di atas akan menghasilkan tulisan STMIK DUTA BANGSA SURAKARTA sebanyak sepuluh baris. Berikut ini disajikan contoh lain penggunaan for yang lain :

Contoh :

```
Program For_2;  
Uses crt;  
var i,j:integer;  
begin  
    j:=0;  
    clrscr;  
    for i:= 1 to 10 do  
        j:=j+2;  
        write(j, ' ');  
    readln;  
end.
```

```
Program For_3;  
Uses crt;  
var i,j:integer;  
begin  
    j:=0;  
    clrscr;  
    for i:= 1 to 10 do  
        begin  
            j:=j+2;  
            write(j, ' ');  
        end;  
    readln;  
end.
```

Bandingkan hasil akhir dari kedua program diatas dan mengapa bisa demikian ????

b) Perulangan Negatif

Perulangan negatif dilakukan dengan menurunkan counter tiap kali perulangannya. Bentuk umum :

For Counter:= nilai_awal **DownTo** nilai_akhir **Do**

atau

For Counter:= nilai_awal **DownTo** nilai_akhir **Do**

Begin

 Perintah1;

 Perintah2;

 Perintah3;

End;

Perhatikan Contoh berikut :

Program For_1;

Uses crt;

var i,j:integer;

begin

 j:=0;

 clrscr;

 for i:= 7 Downto 2 do

 j:=j+2;

 write(j, ' ');

 readln;

end.

Program For_2;

Uses crt;

var i,j:integer;

begin

 j:=0;

 clrscr;

 for i:= 7 Downto 2

do

 begin

 j:=j+2;

 write(j, ' ');

 end;

 readln;

end.

Bandingkan hasil akhir dari kedua program diatas dan mengapa bisa demikian ????

Contoh program berikut ini akan menampilkan tulisan dengan berbagai warna yang disediakan pascal.

Program TampilWarna;

Uses Crt;

Var i : integer;

```

Begin
    Clrscr;
For i:= 1 to 10 do
    Begin
        Textcolor(i);
        Writeln('Ini warna dengan kode = ',i);
    End;
Readln;
End.

```

Catatan : *variabel counter, nilai awal dan nilai akhir haruslah tipe bilangan bulat jadi tipe data real tidak bisa digunakan untuk mendefinisikan ketiga variabel tersebut.*

c) Perulangan Bersarang

Adalah perulangan yang berada dalam perulangan yang lain. Pada keadaan ini, perulangan yang berada paling dalam akan diselesaikan dulu baru kemudian perulangan yang diluarnya.

Bentuk umum :

```

For Counter:= nilai_awal To nilai_akhir Do
    For Counter:= nilai_awal To nilai_akhir Do
        Perintah;
End;

```

Atau

```

For Counter:= nilai_awal To nilai_akhir Do Begin
    For Counter:= nilai_awal To nilai_akhir Do Begin
        Perintah 1;
        Perintah 2;
        Perintah n;
    End;
End;

```

Contoh : Perulangan Bersarang;

```
Program Perulangan_Bersarang;  
Uses Crt;  
Var i,j,k:byte;  
begin  
  clrscr;  
  for i:= 1 to 5 do begin  
    for j:= 1 to 3 do begin  
      k:=i*j;  
      writeln(i,'      ',j,'      ',k);  
    end;  
  end;  
  readln;  
end.
```

Perulangan diatas jika dijalankan akan menghasilkan deretan angka sebagai berikut :

<i>i</i>	<i>j</i>	<i>k</i>
1	1	1
1	2	2
1	3	3
2	1	2
2	2	4
2	3	6
3	1	3
3	2	6
3	3	9
4	1	4
4	2	8
4	3	12
5	1	5
5	2	10
5	3	15

Latihan :

1. Buatlah program untuk peluncuran roket dengan menghitung waktu mundur selama sepuluh detik, setelah detik ke 0 keluar tulisan ditengah layar "DUARRRRRRRRR"
2. Buatlah program untuk membuat program perpangkatan bilangan. Jika dimasukkan bilangan 3 dan pangkatnya 3 maka hasil pangkatnya adalah 27.
3. Buat program untuk menampilkan bilangan ganjil antara 1-20
4. Buatlah program untuk menjumlahkan bilangan ganjil antara 1-20
5. Buatlah program untuk membalik sebuah kalimat yang dimasukkan oleh user.

Contoh :

Kalimat Yang Dimasukkan : STMIK

Kalimat Yang Dibalik : KIMTS

6. Buatlah program yang apabila dimasukkan 2 bilangan yaitu 5 dan 10 maka akan tampil :

```
*      *      *      *      *      *      *      *      *
*
*
*
*
*      *      *      *      *      *      *      *      *
```

7. Buatlah program jika dimasukkan bilangan 5 maka akan memerintahkan user untuk menginput bilangan lima kali secara bebas dan menghitung nilai rata-rata dari bilangan yang dimasukkan
8. Buatlah program untuk menampilkan tulisan :

```
S
SU
SUR
SURA
SURAK
SURAKA
SURAKAR
SURAKART
SURAKARTA
```

2. Struktur While ... Do

Struktur While ... Do mengandung arti bahwa perintah-perintah yang ada dalam begin ... end akan dikerjakan selama kondisi bernilai benar (true).

Contoh Program While...Do :

Program Contoh_While;

```
uses Crt;  
Var i,h,a:integer;  
begin  
  ClrScr;  
  i:=1;  
  a:=5;  
  While a < 8 do  
  begin  
    h:=h+a;  
    i:=i+a;  
    Inc(a);  
    writeln (h);  
    writeln (a);  
  end;  
  readln;  
end.
```

3. Struktur Repeat ... Until

Struktur Repeat ... Until mengandung arti bahwa perintah-perintah yang ada akan dikerjakan selama kondisi bernilai benar (true).

Program Contoh_Repeat;

```
uses Crt;  
Var i,h,a:integer;  
begin  
  ClrScr;
```



```

i:=1;
a:=5;
Repeat
    h:=h+a;
    i:=i*a;
    a:=a+2;
Until a < 10;
writeln (i);
writeln (h);
writeln (a);
readln;
end.

```

Catatan : *pengurangan dan penambahan variabel dalam struktur perulangan while dan repeat harus dilakukan secara manual.*

Daftar Pustaka

- Abdul Kadir, 2004, *Pemrograman Pascal*, Edisi I, Penerbit Andi, Yogyakarta
Aji Supriyanto, 2005, *Pengantar Teknologi Informasi*, , Salemba Infotek, Jakarta
Budi Raharjo, 2005, *Teknik Pemrograman Pascal*, Informatika, Bandung
<http://www.boxpascal.110mb.com/Contoh%20Program%20Pascal.html> (Diakses Tanggal 14 Januari 2009)
<http://www.hu.freepascal.org/faq.html.id> (Diakses Tanggal 14 Januari 2009)
http://www.dikmentidki.go.id/downlaod/latihan_fpc.pdf (Diakses Tanggal 14 Januari 2009)

Latihan :

1. Buatlah program untuk peletusan 5 balon satu per satu dengan tampilan program sebagai berikut :
Meletus balon ke 5 balonku tinggal 4
Meletus balon ke 4 balonku tinggal 3
Meletus balon ke 3 balonku tinggal 2
Meletus balon ke 2 balonku tinggal 1
Meletus balon ke 1 balonku tinggal 0
Catatan : tulisan berwarna beda untuk setiap barisnya
2. Amir berlatih berenang setiap 4 hari sekali sedangkan Budi berlatih berenang setiap 7 hari sekali. Hitunglah berapa kali mereka berlatih bersama dalam waktu 6 bulan (1 bulan = 30 hari), dan pada hari keberapa saja?
3. Sebuah bus berangkat dari kota A dengan membawa 5 orang penumpang. Bus menempuh rute 10 kota, disetiap kota ganjil bus akan menaikkan 3 orang penumpang dan disetiap kota genap bus akan menaikkan 2 orang penumpang. Berapakah jumlah total penumpang bus ketika telah melalui 10 kota.
4. Sebuah bola seberat 100 gr, dilempar dengan ketinggian 5 m, bola tersebut memantul lurus vertikal. Tinggi pantulan berikutnya adalah $\frac{2}{3}$ dari pantulan sebelumnya. Selain itu setiap pantulan genap berat bola berkurang 2,3% dari berat sebelumnya, dan setiap pantulan ganjil beratnya bertambah 1,5% dari berat sebelumnya. Hitunglah berapa kali bola memantul? Dan berapa berat bola setelah pantulan selesai ?

Kompetensi Dasar :

1. Mahasiswa dapat memahami konsep pemrograman modular dengan menggunakan procedure dan function
2. Mahasiswa dapat membedakan procedure dan function dengan menggunakan parameter maupun non parameter
3. Mahasiswa dapat menyelesaikan soal dengan menggunakan procedure maupun function

Sebuah program yang baik adalah program yang membagi permasalahan utama menjadi bagian-bagian kecil dimana setiap bagian kecil ditangani oleh sebuah subprogram, cara ini disebut dengan *modular programming* (pemrograman terbagi/terpecah). Cara ini termasuk pemrograman terstruktur dan sangat didukung oleh bahasa Pascal. Untuk itu, Pascal telah menyediakan dua jenis subprogram, yaitu *procedure* dan *function* (prosedur dan fungsi). Dengan *modular programming*, program lebih mudah dibaca dan dimengerti. Selain itu, pembenahan program dan penelusuran jalannya program (*debugging*) menjadi lebih mudah sebab dapat langsung diketahui subprogram mana yang berjalan tidak sesuai dengan yang diharapkan

A. Procedure

Prosedur adalah subprogram yang menerima masukan tetapi tidak mempunyai keluaran secara langsung. Dalam pascal dikenal dua macam procedure yaitu non parameter dan parameter.

1. Procedure Non Parameter

Procedure non parameter maksudnya tidak ada pengiriman parameter atau variabel tertentu ke program utama. Cara mendefinisikan procedure non parameter adalah sebagai berikut :

```

    Procedure NamaProcedure;
    Begin
        {Statement};
    End;

```

Program Contoh_Procedure1;

```

Uses Crt;

Procedure Tampil_Menu; {Nama Procedure}
begin
    textcolor(12);
    gotoxy(15,10);writeln('1. Nasi Goreng');
    gotoxy(15,11);writeln('2. Bakmi Goreng');
    gotoxy(15,12);writeln('3. Bakmi Godok');
    gotoxy(15,13);writeln('4. Es teh');
    gotoxy(15,14);writeln('5. Es Jeruk');
end;

{**** Program Utama **** }
begin
    ClrScr;
    textcolor(14);
    gotoxy(15,6);writeln('Selamat Datang di Warung G4UL');
    gotoxy(15,8);writeln('Tekan Enter untuk melihat daftar menu .....');
    readln;

    Tampil_Menu;{Cara Memanggil Procedure Non Parameter}
    textcolor(14);
    gotoxy(15,16);writeln('Selamat Mencoba');
    readln;
end.

```

Program Contoh_Procedure2;

```
Uses crt;
Uses Crt;
Var p,l,luas : integer;
Procedure Hitung_Luas;
begin
    luas:=p*l;
end;
{**** Program Utama ****}
begin
    clrscr;
    Write('Panjang :');readln(p);
    write('Lebar    :');readln(l);
    Hitung_Luas; {Memanggil Procedure}
    writeln;
    write('Luas Persegi Panjang :',' ',luas);
    readln;
end.
```

Catatan : Dari contoh diatas dapat dilihat tidak ada pengiriman variabel atau paramater tertentu dari procedure ke program utama.

2. Procedure Parameter

Parameter adalah data masukan untuk sub program yang nantinya akan diproses lebih lanjut pada sub program tersebut. Dalam pascal dikenal 2 macam parameter yaitu :

- a. Parameter Nilai
- b. Parameter Referensi

Cara mendeklarasikan procedure dengan parameter adalah :

```
Procedure B(X : integer; var Y : integer);
```

begin

{ statement }

End;

Pada deklarasi prosedur di atas, parameter X adalah parameter nilai sedang parameter Y adalah parameter referensi. Jadi, pendeklarasian parameter referensi didahului oleh *reserved word* **var**. Parameter referensi ini nantinya dapat dijadikan sebagai variable keluaran dari prosedur.

Berikut ini adalah contoh program dengan procedure parameter :

Program Prosedur_Parameter1;

Uses Crt;

Procedure Tambah (A,B:byte; Var C:integer);

begin

C:=A+B;

{C digunakan untuk menampilkan hasil}

end;

{*** Program Utama*****}**

Var hasil:integer;

begin

Tambah(4,5,hasil);

write('4',' ','+', '5',' ','=',' ',hasil);

readln;

end.

Penjelasan : dapat dilihat pada program diatas bahwa A dan B merupakan parameter nilai sedangkan C merupakan parameter referensi. Variabel C akan menampung hasil penjumlahan dari A dan B. dapat dilihat terjadi pengiriman parameter A, B, C ke 4,5 dan hasil.

ProgramProsedur_Parameter2;

Uses Crt;

```

Var  luas: real;
      alas,tinggi:byte;
Procedure Hitung_Luas(a,t:byte; var l:real);
begin
  l:=(a*t)/2;
end;
{**** Program Utama ****}
begin
  clrscr;
  Write('Alas      :');readln(alas);
  write('Tinggi    :');readln(tinggi);
  Hitung_Luas(alas,tinggi,luas);
  writeln;
  write('Luas Segitiga adalah :',' ',luas:3:0);
  readln;
end.

```

Penjelasan : dapat dilihat pada program diatas bahwa a dan t merupakan parameter nilai sedangkan l merupakan parameter referensi. Variabel l akan menampung hasil perhitungan dari a dan t sehingga terjadi pengiriman parameter **a, t, l** ke alas, tinggi dan luas.

Contoh program berikut ini akan menuntu anda untuk menggunakan procedure parameter dan non parameter secara bersama-sama.

Program Procedure_lengkap;

Uses Crt;

{Membuat Tampilan Menu Utama}

Procedure TampilMenu;

Begin

textcolor(12);gotoxy(25,8);write('1.Konversi Suhu');

textcolor(12);gotoxy(25,10);write('2.Konversi Waktu');

```

    textcolor(12);gotoxy(25,12);write('3.Konversi Berat');
    textcolor(12);gotoxy(25,14);write('4.Keluar');
End;
{Membuat Program Konversi Suhu}
Procedure suhu(c:integer; var r,f,k : real);
begin
    r:=(4/5)*c;
    f:=((9/5)*c)+32;
    k:=c+273;
end;
{membuat Program Konvesi Waktu}
Procedure Waktu(a1:integer; var j,m,d:integer);
begin
    j:=a1 div 3600;
    m:=(a1 mod 3600) div 60;
    d:=(a1 mod 3600) mod 60;
end;
{membuat Program Konvesi Berat}
Procedure Berat(a2:integer; var k,o,g:integer);
begin
    k:=a2 div 1000;
    o:=(a2 mod 1000) div 100;
    g:=(a2 mod 1000) mod 100;
end;
{Definisi Variabel}
Var
    jawab:char;
    Celcius,angka1,angka2 :integer;
    jam,menit,detik,kg,ons,gram:integer;
    reamur,fahrenheit,kelvin : real;
{***** Program Utama *****}
begin
    clrscr;
    repeat
        TampilMenu;
        textcolor(14);gotoxy(25,16);
        write('Pilihan anda :');readln(jawab);
        if jawab='1' then

```



```

begin
    clrscr;
    write('Masukkan suhu [dalam celcius] :');
    readln(celcius);
    Suhu(celcius,reamur,fahrenheit,kelvin);
    writeln('Celcius      : ',celcius);
    writeln('Reamur       : ',reamur:4:2);
    writeln('Fahrenheit: ',fahrenheit:4:2);
    writeln('Kelvin        : ',kelvin:4:2);
end;
if jawab = '2' then
begin
    clrscr;
    write('Masukkan angka [dalam detik] :');
    readln(angka1);
    Waktu(angka1,jam,menit,detik);
    writeln('Angka : ',angka1);
    writeln('Hasil : ',jam,' jam ',menit,' menit ',detik,' detik');
end;
if jawab = '3' then
begin
    clrscr;
    write('Masukkan angka [dalam gram] :');
    readln(angka2);
    Berat(angka2,kg,ons,gram);
    writeln('Angka : ',angka2);
    writeln('Hasil : ',kg,' Kg ',ons,' Ons ',gram,' Gram');
end;
until jawab='4';
end.

```

B. Function

Seperti juga procedure, function adalah suatu program terpisah dalam blok sendiri yang dapat diaktifkan dimanapun dalam program utama. Perbedaan utama antara prosedur dan fungsi adalah dalam menghasilkan keluaran. Walaupun prosedur bisa menghasilkan nilai keluaran, tetapi nilai tersebut tidak

dapat diambil secara langsung, melainkan harus diambil melalui parameter referensi. Sedangkan keluaran dari fungsi dapat diambil langsung dari fungsi tersebut. Seperti juga procedure, function juga ada dua macam yaitu function non parameter dan parameter. Cara mendefinisikan function non parameter adalah sebagai berikut :

```
Function Nama : Tipe_Data;  
Begin  
    Statement;  
End;
```

Contoh function non parameter adalah sebagai berikut :

```
Program Function_NonParameter1;  
Uses Crt;  
Function angka:integer;  
Begin  
    angka:=10;  
End;  
Var  
    a:integer;  
Begin  
    Clrscr;  
    a:=angka;  
    write(a);  
    Readln;  
End.
```

Penjelasan : Program function diatas bernama angka dengan tipe data integer. Function angka dapat dipanggil secara langsung pada program utama tanpa harus menggunakan parameter referensi.

Sedangkan jenis function yang kedua yaitu function parameter dapat dideklarasikan sebagai berikut :

```
Function Nama (a,b:tipe_data): Tipe_Data;  
Begin  
    Statement; {a dan b adalah parameter}  
End;
```

Kode program berikut adalah contoh program function dengan menggunakan parameter :

```
Program FunctionParameter1;  
Uses Crt;  
Function Tambah (a,b:integer):integer;  
Begin  
    Tambah:=a+b;  
End;  
Var  
    C : integer;  
{**** Program Utama****}  
Begin  
    Clrscr;  
    C:=Tambah(4,5);  
    Write( C );  
    Readln;  
End.
```

```
Program FunctionParameter2;  
Uses Crt;  
Function Tambah(A,B:byte):integer;  
begin  
    Tambah:=A+B;  
end;  
{***** Program Utama*****}  
Var hasil:integer;
```

```

        bill1,bil2:byte;
begin
    Clrscr;
    write('Bilangan 1:');readln(bil1);
    write('Bilangan 2:');readln(bil2);
    writeln;

    Hasil:=Tambah(bil1,bil2);
    write(bil1,' ','+',bil2,' ','=',' ',hasil);
    readln;
end.

```

Contoh program berikut akan memberikan gambaran kepada kita tentang perbedaan penggunaan procedure dan function :

program Fungsi;

uses CRT;

var

Bil_1, Bil_2, Hasil : integer;

procedure Awal;

begin

Writeln('Latihan Pascal : Prosedur dan Fungsi');

Writeln('-----');

Writeln;

Writeln('Nama : ');

Writeln('NIM : ');

Writeln;

end;

procedure Baca_Data;

begin

Write('Masukkan bilangan pertama : ');

Readln(Bil_1);

Write('Masukkan bilangan kedua : ');

Readln(Bil_2);

```

        Writeln;
    end;
function Kali(A,B : integer) : integer;
var
    I,J : integer;
begin
    J := 0;
    for I := 1 to B do J := J + A;
    Kali := J;
end;
procedure Kalikan(A,B : integer; var C : integer);
var
    I : integer;
begin
    C := 0;
    for I := 1 to B do C := C + A;
end;
{**** Program Utama ****}
begin
    ClrScr;
    Awal;
    Baca_Data;
    Writeln(Bil_1:3, ' x ', Bil_2:3, ' = ', Kali(Bil_1, Bil_2):5);
    Kalikan(Bil_1, Bil_2, Hasil);
    Writeln(Bil_1:3, ' x ', Bil_2:3, ' = ', Hasil:5);
    Writeln;
    Write('Tekan Enter...');
    Readln;
end.

```

Penjelasan: Perhatikan program di atas. Prosedur Kalikan dan fungsi Kali mempunyai keluaran yang sama, tetapi cara mengambil keluarannya berbeda. Perhatikan dan jelaskan apa yang terjadi jika

baris keempat dalam program utama yang semula perintah :

```
Writeln(Bil_1:3, ' x ', Bil_2:3, ' = ', Kali(Bil_1, Bil_2):5);
```

diubah menjadi :

```
Writeln(Bil_1:3, ' x ', Bil_2:3, ' = ', Kalikan(Bil_1, Bil_2, Hasil):5);
```

Apa yang terjadi ?? mengapa demikian ??

C. Rekursi

Dalam Pascal, ada satu kelebihan dalam cara pemanggilan subprogram. Pascal mengijinkan pemanggilan suatu subprogram dari dalam subprogram itu sendiri. Tidak semua bahasa pemrograman mengijinkan cara pemanggilan subprogram seperti itu karena akan banyak memakan memori. Untuk lebih jelasnya perhatikan potongan program di bawah ini :

```
procedure Z;  
begin  
{ statement }  
Z;  
end;
```

Pada baris terakhir prosedur Z di atas, terdapat pemanggilan kembali terhadap prosedur Z, sehingga prosedur di atas tidak akan pernah selesai dijalankan sebab begitu sampai pada baris terakhir dari prosedur, program akan kembali lagi ke awal prosedur. Yang terjadi adalah semacam perulangan tanpa perintah perulangan Pascal, dan perulangan dengan cara ini disebut dengan **rekursi**. Rekursi berlaku terhadap semua subprogram dalam Pascal, yaitu prosedur dan fungsi. Dengan adanya rekursi ini, banyak algoritma komputer menjadi lebih mudah dibuat programnya. Berikut ini adalah program menghitung suku banyak Legendre, salah satu contoh perhitungan yang dapat diselesaikan dengan menggunakan rekursi :

```
program Rekursi;  
uses CRT;
```

```

var
Jum_Suku, I : integer;
Bil_X : real;
function Legendre(X : real; N : integer) : real;
var
    Suku_1, Suku_2 : real;
begin
    if N = 0 then Legendre := 1
    else if N = 1 then Legendre := X
    else begin
        Suku_1 := ((2*N - 1) * (X * Legendre(X, N-1))) / N;
        Suku_2 := ((N-1) * Legendre(X, N-2)) / N;
        Legendre := Suku_1 + Suku_2;
    end;
end;

procedure Awal;
begin
    Writeln('Latihan Pascal 2 : Prosedur dan Fungsi');
    Writeln('-----');
    Writeln;
    Writeln('Nama : ');
    Writeln('NIM : ');
    Writeln;
end;

procedure Baca_Data;
begin
    Writeln('Menghitung Suku Banyak Legendre');
    Writeln;
    Write('Sampai suku ke : ');
    Readln(Jum_Suku);
    Write('Masukkan nilai X : ');
    Readln(Bil_X);
    Writeln;

```

```

end;
{*** Program Utama ***}
begin
    ClrScr;
    Awal;
    Baca_Data;
    for I := 0 to Jum_Suku do
    begin
        Writeln('Suku ke-',I:2,', Nilainya = ',Legendre(Bil_X, I):8:3);
    end;
    Writeln;
    Write('Tekan Enter...');
    Readln;
end.

```

Penjelasan : Perhatikan program diatas, terlihat bahwa legendre (nama fungsi) dipanggil pada dirinya sendiri.

Daftar Pustaka

Abdul Kadir, 2004, *Pemrograman Pascal*, Edisi I, Penerbit Andi, Yogyakarta
 Aji Supriyanto, 2005, *Pengantar Teknologi Informasi*, , Salemba Infotek, Jakarta
 Budi Raharjo, 2005, *Teknik Pemrograman Pascal*, Informatika, Bandung
<http://www.boxpascal.110mb.com/Contoh%20Program%20Pascal.html> (Diakses Tanggal 14 Januari 2009)
<http://www.hu.freepascal.org/faq.html.id> (Diakses Tanggal 14 Januari 2009)
http://www.dikmentidki.go.id/downlaod/latihan_fpc.pdf (Diakses Tanggal 14 Januari 2009)

Latihan :

1. Apa perbedaan procedure dan function ?
2. Apa yang anda ketahui tentang procedure parameter dan ada berapa jenis parameter dalam procedure ?
3. Carilah letak kesalahan pada pendeklarasian sub program berikut :

```
Function kali (a,b : integer);  
Begin  
    Kali:=a*b;  
End;
```

4. Buatlah program untuk menghitung factorial suatu bilangan (menggunakan metode rekursi).
5. Buatlah program untuk pembelajaran luas bangun datar dengan tampilan sebagai berikut :

***** AYO BELAJAR LUAS BANGUN DATAR *****

-
1. Luas Persegi Panjang
 2. Luas Persegi
 3. Luas Segi Tiga
 4. Luas Lingkaran
 5. Luas Trapesium
 6. Luas Jajaran Genjang
 7. Luas Belah Ketupat
 8. Luas Layang-Layang
 9. Keluar Program
-

Tentukan Pilihan [1/2/3/4/5/6/7/9] :

Kompetesi Dasar :

1. Mahasiswa dapat memahami konsep dasar tipe data array dan record
2. Mahasiswa dapat menyelesaikan kasus dengan menggunakan array dan record

Telah dijelaskan pada bab 4 (tipe data dan Operator) bahwa dalam bahasa Pascal, secara garis besar dikenal dua macam tipe data yaitu tipe data sederhana (*primitive type*) dan tipe data kompleks (*complex type*). Contoh tipe data sederhana adalah tipe numerik (integer dan real), tipe data karakter, tipe data *boolean* dan tipe data enumerasi. Contoh tipe data kompleks adalah *string*, *array* (larik), *record* dan *object*. Tipe data sederhana adalah tipe data yang hanya mampu menyimpan satu nilai tiap satu variabelnya. Sebaliknya tipe data kompleks adalah tipe data yang mampu menyimpan lebih dari satu nilai dalam tiap satu variabelnya. Dalam bab ini hanya akan dibahas dua tipe data kompleks yaitu *array* dan *record*

A. Array

Array adalah tipe data kompleks yang elemen-elemennya mempunyai tipe data yang sama. Jumlah elemen array bersifat tetap dan tidak bisa ditambah atau dikurangi setelah pendeklarasiannya. Tiap elemen mempunyai nomer indeks sendiri dan pengaksesan terhadap elemen array dilakukan dengan menunjukkan nomer indeks dari elemen yang akan diakses. Cara pendeklarasian suatu variabel bertipe array adalah sebagai berikut :

Var

NamaArray : array [IndeksAwal..IndeksAkhir] of TipeData;

Contoh :

Var

Angka:array[1..10] of integer;

Penjelasan : contoh penggalan program diatas menunjukkan bahwa variabel angka berupa array dengan tipe data integer dan jumlah elemen sebanyak 10. Untuk mengakses elemen dari angka dapat menunjuk ke indeksinya seperti berikut ini :

Angka [nomor_indeks]

contohnya :

Angka [1] := 10 {nilai elemen 1 =10}

Dalam bahasa pascal, tersedia dua buah fungsi yang dapat digunakan untuk mengambil indeks terendah dan tertinggi dari sebuah array, yaitu fungsi **Low** dan **High**. Adapun parameter dari kedua fungsi tersebut adalah nama array yang akan dicari indeksinya.

Contoh :

Var

A : array [1..100] of integer;

terendah, tertinggi : integer;

begin

terendah := Low(A);

tertinggi:= High(A);

.....

.....

end.

Jika anda seorang pemrogram pemula, mungkin saja timbul pertanyaan : “mengapa harus menggunakan array?”. Sekarang coba bayangkan jika anda harus menginputkan nilai sebanyak seratus atau bahkan seribu. Tentu saja kita harus mendefinisikan variabel sebanyak seratus atau seribu buah. Tetapi

dengan array kita cukup mendefinisikan 1 variabel dengan tipe array. Untuk lebih jelasnya perhatikan contoh program berikut ini :

Program Metode_Konvensional;

```
Uses Crt;
Var
    a1,a2,a3,a4,a5,a6,a7,a8,a9,a10 : integer;
begin
    Clrscr;
    {Proses Pemasukan Bilangan}
    Write('Angka ke 1 : ');readln (a1);
    Write('Angka ke 2 : ');readln (a2);
    Write('Angka ke 3 : ');readln (a3);
    Write('Angka ke 4 : ');readln (a4);
    Write('Angka ke 5 : ');readln (a5);
    Write('Angka ke 6 : ');readln (a6);
    Write('Angka ke 7 : ');readln (a7);
    Write('Angka ke 8 : ');readln (a8);
    Write('Angka ke 9 : ');readln (a9);
    Write('Angka ke 10 : ');readln (a10);
    {Proses pencetakan bilangan}
    Clrscr;
    Writeln('angka ke 2 :',a1);
    Writeln('angka ke 3 :',a2);
    Writeln('angka ke 4 :',a3);
    Writeln('angka ke 4 :',a4);
    Writeln('angka ke 5 :',a5);
    Writeln('angka ke 6 :',a6);
    Writeln('angka ke 7 :',a7);
    Writeln('angka ke 8 :',a8);
    Writeln('angka ke 9 :',a9);
    Writeln('angka ke 10 :',a10);
    Readln;
```

end.

Mungkin metode diatas masih bisa diterima karena jumlah data hanya 10 buah tetapi jika seratus buah, akan lebih mudah jika kita menggunakan tipe data array. Kode program diatas, apabila diubah dengan menggunakan tipe data array akan terlihat sebagai berikut :

Program Contoh_Array;

```
Uses crt;
var
    a:array[1..10] of integer;
    i:integer;
Begin
    clrscr;
    {Menginput Data}
    for i:=1 to 10 do
    begin
        write('Angka ke ',i,' : ');
        readln(a[i]);
    end;
    {Mencetak Data}
    for i:=1 to 10 do writeln('angka ke ',i,' : ',a[i]);
    readln;
end.
```

Berdasarkan penggalan program diatas dapat dilihat dengan bahwa betapa merepotkan bila kita harus menampung data yang sama dengan jumlah yang sangat banyak. Bisa dibayangkan berapa variabel yang harus kita definisikan untuk menampung data tersebut jika data lebih dari 100. Itulah alasan mengapa kita menggunakan array.

Nilai yang terkandung dalam sebuah array dapat bernilai konstan, artinya nilai tersebut tidak bisa diubah selama program dijalankan. Untuk mendeklarasikan array yang bernilai konstan kita harus menambahkan kata kunci `Const`. Berikut ini cara pendeklarasian array konstan :

Const

NamaArray=array[IndeksAwal..IndeksAkhir] of TipeData = (Nil1,Nil2...);

Catatan : perlu diingat bahwa banyaknya nilai konstan yang dituliskan diatas, harus sesuai dengan banyaknya elemen array yang didefinisikan. Selain itu nilai yang telah didefinisikan, tidak bisa diubah selama program dijalankan.

Contoh :

Program Array_Konstan;

Uses Crt;

Const

Huruf:array[1..5] of char =('A','B','C','D','E');

Var

i:integer;

Begin

Clrscr;

For i:=1 to 5 do writeln(huruf[i]);

Readln;

End.

Hasil dari program diatas adalah :

A

B

C

D

E

Dimana masing-masing elemen tidak bisa diganti dengan elemen yang lain misal :

E
F
G
H
I

Berikut ini adalah contoh program menghitung nilai rata-rata dari beberapa komponen penilaian :

Program array_Rata;

uses crt;

var

angka:array[1..100] of integer;

n,i,total:integer;

rata:real;

begin

clrscr;

write('Jumlah komponen nilai [max 100] : ');

readln(n);

total:=0;

{menginput data dan mengitung total nilai}

for i:= 1 to n do

begin

write('Komponen Nilai ke ',i,' = ');

readln(angka[i]);

total:=total+angka[i];

end;

{mengitung Nilai Rata}

rata:= total/n;

clrscr;

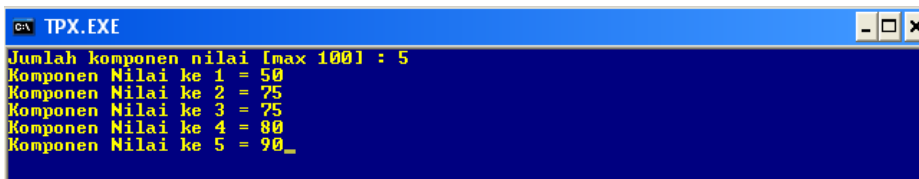
```

{Mencetak nilai yang di inputkan dan juga nilai rata-rata}
for i:= 1 to n do
    writeln('Komponen Nilai : ',i,'      ',angka[i]);
    write('Nilai Rata-rata : ',rata:4:2);
readln;
end.

```

Hasil output program diatas adalah sebagai berikut :

Proses input komponen nilai dan besarnya nilai untuk masing-masing komponen :

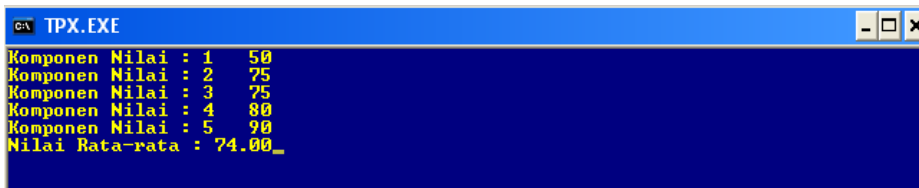


```

C:\> TPX.EXE
Jumlah komponen nilai [max 100] : 5
Komponen Nilai ke 1 = 50
Komponen Nilai ke 2 = 75
Komponen Nilai ke 3 = 75
Komponen Nilai ke 4 = 80
Komponen Nilai ke 5 = 90_

```

Proses Pencetakan nilai dan perhitungan nilai rata-rata :



```

C:\> TPX.EXE
Komponen Nilai : 1 50
Komponen Nilai : 2 75
Komponen Nilai : 3 75
Komponen Nilai : 4 80
Komponen Nilai : 5 90
Nilai Rata-rata : 74.00_

```

B. Array Dua Dimensi

Pendeklarasian array yang telah kita bahas di subbab sebelumnya adalah array satu dimensi, sedangkan pada subbab ini akan kita bahas tentang array dengan dua dimensi. Pendeklarasian array dua dimensi adalah sebagai berikut :

Var

nama : array [nilai_awal1..nilai_akhir1, nilai_awal2..nilai_akhir2] of tipe_data;

Atau jika array dijadikan type maka pendeklarasian array dua dimensi menjadi:

Type ***nama1 = array [nilai_awal..nilai_akhir, nilai_awal..nilai_akhir] of tipe_data;***

Var

nama:nama1;

Untuk mengisi data dalam array dua dimensi atau lebih, harus menggunakan bentuk pengulangan bersarang (pengulangan di dalam pengulangan). Contoh Berikut ini akan memberikan gambaran kode program untuk pengisian array dua dimensi :

.

Contoh program berikut adalah program untuk mengisi matrik yang memiliki ordo 3 x 2 (3 Baris 2 Kolom) :

Program Contoh_Isi_Matrik;

Uses Crt;

Var nilai:array [1..3,1..2] of integer;

i,j :integer;

Begin

ClrScr;

textcolor(14);

{Input Data Matrik}

for i:= 1 to 3 do begin

for j:= 1 to 2 do begin

write('Nilai baris Ke',' ',i,' ','Kolom',' ',j,' ',' ':' ',' ');

readln(nilai[i,j]);

end;

end;

writeln;

{Cetak Isi Matrik}

writeln('Isi Matrik :');

for i:= 1 to 3 do begin

```

    for j:= 1 to 2 do begin
        write(nilai[i,j], '      ');
    end;
    writeln;
end;
readln;
end.

```

Contoh kode-kode program diatas adalah contoh kode program dengan jumlah baris dan kolom yang sudah ditentukan sebelumnya. Untuk kode program berikut ini, jumlah kolom dan baris di inputkan oleh user :

```

Program _inputArray;
Uses Crt;
{Definisi jumlah baris dan kolom max 100}
Var nilai:array [1..100,1..100] of integer;
    i,j,b,k :integer;
Begin
    ClrScr;
    textcolor(14);
{Input Baris Dan Kolom}
    write('Masukkan Dimensi Baris :');readln(b);
    write('Masukkan Dimensi Kolom :');readln(k);
    writeln;
{Input Isi Matrik}
    for i:= 1 to b do begin
        for j:= 1 to k do begin
            write('Nilai baris Ke',' ',i,' ','Kolom',' ',j,' ',':',',' ');
            readln(nilai[i,j]);
        end;
    end;
    writeln;
{Cetak Isi Matrik}

```

```

writeln('Isi Matrik :');
for i:= 1 to b do begin
  for j:= 1 to k do begin
    write(nilai[i,j], '      ');
  end;
  writeln;
end;
readln;
end.

```

C. Record

Record adalah tipe data kompleks yang elemen-elemennya boleh mempunyai tipe data yang berbeda. Record lebih kompleks daripada array karena record merupakan kumpulan beberapa variabel dengan tipe data yang berbeda. Berbeda dengan array yang tiap elemennya ditandai dengan nomer indeks maka record ditandai dengan nama variabel anggotanya. Cara mengakses elemen dari record dilakukan dengan menyebutkan nama variabel anggota setelah menyebutkan nama record yang akan diakses. Di antara nama record dan nama variabel anggota dipisahkan tanda titik (.). Cara pendeklarasian record adalah sebagai berikut :

var

B : record

X : integer;

Y : real;

end;

atau

Type nama_record=record

Field1 : tipe_data1

Field2 : tipe_data2

Field3 : tipe_data3

.....

Field_n : tipe_data_n

End;

Var variabel : nama_record;

Pada pendeklarasian di atas, maksudnya adalah sebagai berikut : variabel B berupa record dengan dua elemen yaitu X bertipe integer dan Y bertipe real. Untuk mengakses elemen dari variabel B seperti berikut :

B.nama_variabel

contoh :

B.X := 10;

Untuk lebih jelasnya perhatikan program berikut :

Program Contoh_Record;

Uses Crt;

Type mahasiswa=record

nim:string[9];

nama:string[15];

n1,n2,n3:integer;

end;

Var a :mahasiswa;

rata :real;

huruf:char;

ket :string[20];

Begin

ClrScr;

textcolor(14);

write('NIM : ');readln(a.nim);

write('Nama : ');readln(a.nama);

```

write('Nilai 1:');readln(a.n1);
write('Nilai 2:');readln(a.n2);
write('Nilai 3:');readln(a.n3);
rata:=(a.n1+a.n2+a.n3)/3;
if ((rata >= 0) and (rata<= 40)) then
begin
huruf:='E';
ket  :='Sangat Tidak Baik';
end;
if ((rata > 40) and (rata<= 60)) then
begin
huruf:='D';
ket  :='Tidak Baik';
end;
if ((rata > 60) and (rata<= 75)) then
begin
huruf:='C';
ket  :='Baik';
end;
if ((rata > 75) and (rata<= 85)) then
begin
huruf:='B';
ket  :='Sangat Baik';
end;
if ((rata > 85) and (rata<= 100)) then
begin
huruf:='A';
ket  :='Sangat Baik Sekali';
end;
Clrscr;
textcolor(11);
writeln('Data Mahasiswa');
writeln('NIM          :',' ',a.nim);

```

```
writeln('Nama Mahasiswa   :',' ',a.nama);  
writeln('Nilai Rata-rata  :',' ',rata:3:2);  
writeln('Nilai Huruf      :',' ',huruf);  
writeln('Keterangan        :',' ',ket);  
readln;  
end.
```

Daftar Pustaka :

<http://beeography.modblog.com>

Latihan

1. Buatlah Program untuk menentukan nilai maksimum dan nilai minimum dari suatu deretan angka yang anda masukkan
2. Buatlah program untuk menghitung Gaji Karyawan dengan ketentuan sebagai berikut :

Tampilan Program Sebagai berikut :

Procedure Input_Data

Banyak Data : ...
Data Ke : ...
NIP : ...
Nama Pegawai : ...
Jabatan : ...
Jenis Kelamin : ...
Status Perkawinan : ...
Jumlah anak : ...
Jumlah Hari Kerja : ...

Procedure Cetak_Data

REKAPITULASI GAJI PEGAWAI PT. SEJAHTERA

NIP	Nama Pegawai	Gaji Pokok	Tunjangan Jabatan	Tunjangan Istri	Tunjangan Anak	Uang Makan	Uang transport	Gaji Bersih
-----	-----------------	---------------	----------------------	--------------------	-------------------	---------------	-------------------	----------------

Procedure Tampil_Menu

Program Perhitungan Gaji Karyawan PT. Sejahtera

[1] Input Data [2] Cetak Data [3] Keluar

Masukkan Pilihan Anda : ...

3. Pegawai di PT. Sejahtera dibagi dalam tiga golongan yaitu

Jabatan	Gaji Pokok	Tunjangan Jabatan	Tunjangan Istri	Tunjangan Anak	Uang Makan/hari	Uang transport /hari
Direktur	100000	3%	2.5%	2.5%	15000	12500
Staff	750000	2%	2.5%	2.5%	12500	10000
Karyawan	645000	1%	2.5%	2.5%	10000	7500

Jumlah anak yang mendapat tunjangan *maksimal 2 orang anak*

Tunjangan istri hanya diberikan jika pegawai berjenis kelamin *P*

Tunjangan istri dan anak hanya diberikan bagi pegawai yang sudah menikah

$U_Makan = \text{Uang Makan/hari} * \text{Jumlah hari kerja}$

$U_Transport = \text{Uang Transport/hari} * \text{Jumlah hari kerja}$

$\text{Gaji kotor} = \text{gaji pokok} + T_Anak + T_Istri + T_Jabatan + U_Makan + U_Transport$

$\text{Gaji Bersih} = \text{gaji kotor} - PPh (2.5\% * \text{gaji kotor})$

Jenis Kelamin = P : Pria dan W : Wanita

Status Perkawinan = 1 : Menikah dan 2 : Belum Menikah

Buatlah program aplikasi untuk menghitung gaji masing-masing pegawai diatas

4. Buatlah Program untuk operasi matrik ordo 2×2 :

- Pertambahan Matrik
- Pengurangan Matrik
- Transpose Matrik
- Perkalian Matrik

A. Tipe Data Record

B. Sorting

Ada 3 metode pengurutan data yang digunakan yaitu :

a. Bubble Sort

Teknik mengurutkan data dengan cara membandingkan data yang ada. Metode ini merupakan metode yang paling sederhana tetapi lambat.

Contoh :

Program Contoh_BubbleSort;

```
Uses Crt;
Var nilai:array [1..100] of integer;
    i,n,j,t :integer;
Begin
  ClrScr;
  textcolor(14);
  write('Jumlah Data :');readln(n);
  for i:= 1 to n do begin
    readln(nilai[i]);
  end;
  ClrScr;
  for i:= 1 to n-1 do begin
    for j:= i+1 to n do begin
      if nilai[i] > nilai[j] then begin
        t:=nilai[i];
        nilai[i]:=nilai[j];
        nilai[j]:=t;
      end;
    end;
  end;
```

```

end;
writeln;
Writeln('Nilai Yang Telah Diurutkan Adalah :');
for i:= 1 to n do
write(nilai[i], ' ');
readln;
end.

```

b. Shell Sort

Prinsipnya sama dengan bubble sort, hanya saja dilakukan optimalisasi sehingga lebih cepat. Metode ini dilakukan dengan membandingkan jarak antar data.

Contoh :

Program Contoh_ShellSort;

```

Uses Crt;
Var nilai:array [1..100] of integer;
    i,n,j,t :integer;
Begin
  ClrScr;
  textcolor(14);
  write('Jumlah Data :');readln(n);
  for i:= 1 to n do begin
    readln(nilai[i]);
  end;
  for j:= (n div 2) downto 1 do
    for i:= 1 to n-j do
      if nilai[i] > nilai[i+j] then begin
        t:=nilai[i];
        nilai[i]:=nilai[i+j];
        nilai[i+j]:=t;
      end;
    end;
  end;
  writeln;

```

```
Writeln('Nilai Yang Telah Diurutkan Adalah :');  
for i:= 1 to n do  
  write(nilai[i], ' ');  
  readln;  
end.
```

c. **Quick Sort**

Metode ini dilakukan dengan metode rekursi sampai habis. Dimana data dibagi menjadi dua dan data tengah berfungsi sebagai pivot (pusat operasi). Data dengan nilai rendah akan berada disebelah kiri pivot dan data dengan nilai tinggi akan berada disebelah kanan pivot. Metode ini merupakan metode paling rumit namun paling cepat dalam mengurutkan data.

Contoh :

```

procedure qsort(bawah, atas : integer)
var kiri, kanan, pivot : integer;
begin
pivot:=Data[(bawah+atas) div 2];
kiri:=bawah;
kanan:=atas;
while kiri<=kanan do
begin
while Data[kiri] < pivot do kiri:=kiri+1; { memisahkan ke
kiri }
while Data[kanan]>pivot do kanan:=kanan-1;{Memisahkan ke
kanan}
if kiri<=kanan then { chek perubahan }
begin
swap Data[kiri] with Data[kanan];
kiri:=kiri+1;
kanan:=kanan-1;
end;
end;
if kanan>bawah then qsort(bawah,kanan);{ urutkan bagian
KIRI }
if atas>kiri then qsort(kiri ,atas);{ urutkan bagian KANAN}
end;

```

Pemanggilan dalam program utama : **qsort(1,NumberOfData);**

Bab ini akan mempelajari bagaimana menyimpan suatu data dalam disk. Ada tiga cara untuk menangani file :

a. Text File

Pembacaan file dilakukan secara berurutan.

b. Typed File

Pembacaan dilakukan secara acak, sesuai dengan keinginan user dan tidak harus dari awal file. Ciri khas file ini tipe filenya dinyatakan dengan *file of* yang diikuti oleh tipe komponen dari file.

c. Untyped File

File ini tidak mengansumsi apa-apa tentang tipe datanya. Bila anda ingin membaca atau menulis file jenis *untyped file* diperlukan perintah khusus yaitu *blockwrite* dan *blockread*.

Berikut ini adalah beberapa perintah yang digunakan untuk pembacaan file :

Perintah	Keterangan	Sintaks Penulisan
Assign	Untuk memberi nama bagi txtfile	Assign (txtfile, 'Contoh.txt');
Rewrite	Digunakan bila dipakai sebagai output baru	Rewrite (txtfile);
Append	Digunakan untuk membuka kembali jika akan dipakai sebagai output baru	Append(txtfile);
Reset	Digunakan jika dibaca sebagai input	Reset(txtfile);
Read	Untuk membaca isi file	Read (txtfile);

	yang telah di-assign sebelumnya	
Close	Digunakan untuk menutup file	<code>Close(txtfile);</code>
EOF (end Of File)	Bernilai true jika penunjuk file berada diakhir file	<code>EOF(txtfile);</code>
EOFLn (End Of Line)	Bernilai true jika penunjuk file berada di akhir baris	<code>EOFLn(txtfile);</code>

Contoh Program Pengelolaan File :

a. Membaca file dan menampilkan dilayar

Program Baca_File_Text;

```
Uses Crt;
var fileku:text;
    nama  :string;
    s      :string;
begin
    clrscr;
    write('Nama File yang akan dibaca :');
    readln(nama);
    assign(fileku,nama);
    reset(fileku);
    while not EOF(fileku) do begin
        readln(fileku,s);
        writeln(s);
    end;
    close(fileku);
    readln;
end.
```

b. Membuat file baru

Program Buat_File_Text;

```
Uses Crt;
var fileku:text;
    nama  :string;
    s      :string;
begin
    clrscr;
    write('Nama File yang akan dibuat :');
    readln(nama);
    assign(fileku,nama);
```

```
rewrite(fileku);  
writeln('Mulai Menulis');  
writeln('Tekan enter & spasi untuk berhenti');  
while s<>' ' do begin  
    readln(s);  
    writeln(fileku,s);  
end;  
close(fileku);  
readln;  
end.
```


c. Menambahkan isi file pada file yang sudah ada

Program Tambah_Isi_File_Text;

```
Uses Crt;
var fileku:text;
    nama  :string;
    s      :string;
begin
    clrscr;
    write('Nama File yang akan dibuat :');
    readln(nama);
    assign(fileku,nama);
    append(fileku);
    writeln('Mulai Menulis');
    while s<>' ' do begin
        readln(s);
        writeln(fileku,s);
    end;
    close(fileku);
    readln;
end.
```

d. Membuat file bertipe atau Typed File

Program File_Typed;

```
Uses Crt;
Type s10=string[10];
var txtfile  :text;
    stringfile:file of s10;
    s          :s10;
begin
    ClrScr;
    writeln('Rewriting output.txt');
    assign(txtfile,'output.txt');
```

```
rewrite(txtfile);  
writeln('Rewriting output.rtf');  
assign(stringfile,'output.rtf');  
rewrite(stringfile);  
s:='PASCAL';  
writeln('Menyimpan data ke  
output.txt');write(txtfile,s);  
writeln('Menyimpan data ke  
output.rtf');write(stringfile,s);  
close(txtfile);  
close(stringfile);  
readln;  
end.
```

DAFTAR PUSTAKA

Hartono, Jogianto, 1989, *Teori dan Aplikasi Program Komputer Bahasa Turbo Pascal*,
Yogyakarta, Andi

Raharjo, Budi, 2005, *Teknik Pemrograman Pascal*, Informatika, Bandung

Pardosi, Mico, 1999, *Bahasa Pemrograman Turbo Pascal 7.0*, Surabaya, Penerbit
Indah.

http://sipil.uns.ac.id/kulol/bahasa_komputer

<http://www.boxpascal.110mb.com/Contoh%20Program%20Pascal.html>

<http://www.hu.freepascal.org/faq.html.id>

<http://yogyafree.net/forum2/viewtopic.php?f=8&t=8240>

<http://rosihanari.net/programming-pascal.php>

(14 Januari 2009 pukul 16.00)

<http://www.a-rang.info>

<http://materi-praktek.blogspot.com/2007/03/bab-1-mengenal-tampilan-turbo-pascal.html>

<http://rosihanari.net/programming-pascal.php>

<http://www.dwisanjaya.com/downloads.htm>

Profil



Eko Purwanto, Lahir di Klaten 27 Juni 1984, menyelesaikan pendidikan SDN 2 Sidoluas (1997), SLTPN 2 Tulung (2000), SMUN 1 Karanfganom (2003) dan menyelesaikan S1 di STIMIK Duta Bangsa Surakarta (2008). Menikah dengan Sutatik dan telah di karuniaai Seorang putra bernama Fawzi Muhammad Afdhal. Saai ini sebagai Dosen di Almamaternya STIMIK Duta Bangsa Surakarta sejak tahun 2008.

Sedang menempuh Strata Dua (S2) di STIMIK AMIKOM Yogyakarta.



Istilah program atau program komputer mungkin sudah tidak asing lagi ditelinga kita, begitu juga istilah pemrograman. **Program atau program komputer** adalah instruksi-instruksi yang diberikan kepada komputer untuk melaksanakan tugas tertentu. sedangkan **pemrograman** adalah upaya untuk membuat kode kode yang dapat dikenal oleh komputer.

Salah satunya Pascal merupakan bahasa pemrograman yang terpopuler di lingkungan akademis dan banyak digunakan di sebagian besar universitas dan institusi pendidikan lainnya, untuk menerangkan dasar-dasar logik pemrograman.

Dalam buku ini, pembahasan akan dimulai dari pemahaman konsep dan esensi yang terdapat dalam Bahasa Pascal sebagai bekal dasar untuk melangkah ke tahap selanjutnya. Untuk menguji ketajaman anda dalam menerima setiap materi yang disampaikan, di setiap bab disertakan juga soal-soal yang disuguhkan dalam bentuk langsung ke pembuatan program.

Buku ini tidak hanya digunakan bagi mereka yang ingin menjadi seorang programmer Pascal tapi juga bagi anda para pengguna Delphi akan memberikan konsep dan dasar yang kuat untuk dikembangkan dan diimplementasikan ke dalam aplikasi visual.