

BAB I

SEJARAH / PERKEMBANGAN PASCAL

Merupakan pengembangan dari bahasa ALGOL 60, bahasa pemrograman untuk sains komputasi. Tahun 1960, beberapa ahli komputer bekerja untuk mengembangkan bahasa ALGOL, salah satunya adalah Dr. Niklaus Wirth dari Swiss Federal Institute of Technology (ETH-Zurich), yang merupakan anggota grup yang membuat ALGOL.

Tahun 1971, dia menerbitkan suatu spesifikasi untuk highly-structured language (bahasa tinggi yang terstruktur) yang menyerupai ALGOL. Dia menamainya dengan PASCAL (seorang filsuf dan ahli matematika dari Perancis) Pascal bersifat data oriented, yaitu programmer diberi keleluasaan untuk mendefinisikan data sendiri. Pascal juga merupakan teaching language (banyak dipakai untuk pengajaran tentang konsep pemrograman).

Jenis jenis pascal antara lain yaitu :

- Pascal yang didefinisikan oleh K. Jensen & Niklaus Wirth
- Pascal di Eropa didefinisikan oleh ISO (International Standards Organization)
- Di Amerika oleh kerjasama antara ANSI (America National Standard Institute)

Saat ini pascal yang beredar di pasaran dan diminati oleh para programmer yaitu pascal yang didefinisikan oleh K. Jensen dan Prof. Niklaus Wirth.

PASCAL SEBAGAI BAHASA TERSTRUKTUR

Sebagai bahasa terstruktur, PASCAL mempunyai ciri-ciri sebagai berikut :

▪ Berurutan

Susunan dari kode-kode dalam teks Pascal harus ditulis secara urut dari atas, pernyataan-pernyataan yang ditulis lebih awal akan dieksekusi lebih dahulu. Oleh karena itu, suatu pernyataan yang menyangkut suatu variabel di dalam program, maka variable itu harus terdefinisi dahulu sebelumnya. Hal ini terutama menyangkut pada pemanggilan sub-program oleh sub-program yang lain. Bisa dibaca lebih lanjut pada bagian sub-program.

▪ Blok dengan batas-batas yang jelas.

Pascal memberikan pembatas yang jelas pada tiap-tiap blok, seperti pada blok program utama, sub-program, struktur kontrol (pengulangan/pemilihan), dll. Pemakaian kata kunci begin untuk mengawali operasi pada blok dan end untuk menutupnya memudahkan programmer menyusun programnya dengan mudah.

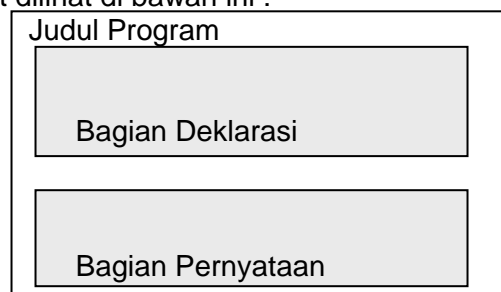
▪ Satu pintu masuk dan satu pintu keluar pada blok pemilihan dan pengulangan.

Contoh di atas juga mengilustrasikan pintu masuk tunggal pada suatu blok pemilihan dengan pintu keluaran yang satu pula (satu disini maksudnya bukan dua baris perintah output tapi suatu paket perintah yang dirangkai dengan **begin .. end**).

STRUKTUR PROGRAM

Struktur dari program pascal dapat dilihat di bawah ini :

- Judul Program
- Blok Program
 - Bagian deklarasi
 - deklarasi label
 - definisi konstanta
 - definisi tipe
 - deklarasi variable
 - deklarasi prosedur



- deklarasi fungsi
- b. Bagian pernyataan

PROGRAM PASCAL YANG PALING SEDERHANA

Suatu program pascal yang paling sederhana adalah program yang hanya terdiri dari sebuah bagian pernyataan saja. Bagian pernyataan (statement part) merupakan bagian yang terakhir dari suatu blok. Bagian ini diawali dengan kata cadangan (reserved word) *Begin* dan diakhiri dengan kata cadangan *End*.

Begin

End.

Contoh 2.1 Program pascal sederhana

Begin

Writeln('saya pascal')

End.

Contoh 2.2 Program pascal sederhana

Jika program contoh 2.2 dijalankan maka akan tampak hasil di layar sebagai berikut :

saya pascal

PENULISAN PROGRAM PASCAL

Program Pascal tidak mengenal aturan penulisan di kolom tertentu, jadi boleh ditulis mulai kolom ke berapapun. Contoh :

Begin

Writeln('Saya Pascal') ;
Writeln('-----')

End.

Bisa juga ditulis

Begin Writeln('Saya Pascal') ; Writeln('-----') End.

JUDUL PROGRAM

Pada Turbo Pascal, judul program sifatnya optional dan tidak signifikan di dalam program. Jika ditulis dapat digunakan untuk memberikan nama program dan daftar dari parameter tentang komunikasi program dengan lingkungannya yang sifatnya sebagai dokumentasi saja.

Judul program bila harus ditulis terletak pada awal program dan diakhiri dengan titik koma. Contoh :

Program **Contoh1** ; → diakhiri dengan titik koma

Begin

Writeln('Saya Pascal') ;
Writeln('-----')

End.

BAGIAN DEKLARASI

Bagian deklarasi digunakan bila di dalam program menggunakan pengenal (identifier) Identifier dapat berupa label, konstanta, tipe, variable, prosedur dan fungsi.

DEKLARASI KONSTANTA

Bila kita ingin menggunakan identifier yang berisi nilai-nilai konstanta yang bernilai tetap (tidak berubah selama program berjalan), maka diawali dengan kata cadangan *Const* diikuti oleh kumpulan Identifier yang diberi suatu nilai konstanta.

```

Program Contoh_Konstanta;
Const NamaPerusahaan      = 'STMIK Budidarma' ;
        Alamat              = 'Jln. Djamin No 133' ;
Begin
    Writeln('Nama Perusahaan : ', NamaPerusahaan);
    Writeln('Alamat          : ', Alamat);
End.

```

Contoh 2.3 Program pascal sederhana

Bila program dijalankan, maka akan didapatkan hasil :

```

Nama Perusahaan      : STMIK BUDIDARMA
Alamat              : Jln. Djamin No 133

```

DEKLARASI VARIABEL

Jika konstanta merupakan identifier berisi data konstanta yang nilai sudah ditentukan dan pasti, tidak dapat dirubah di dalam program, maka variable adalah identifier yang berisi data yang dapat berubah-ubah nilainya di dalam program.

Kata cadangan *Var* digunakan sebagai judul di dalam bagian deklarasi variable dan diikuti oleh satu atau lebih identifier yang dipisahkan koma, diikuti dengan titik dua dan tipe dari datanya serta diakhiri dengan titik koma.

```

Program Contoh_Variabel ;
Var
    Nama   : String[15] ;
    Harga,
    Jumlah : Longint
    Nilai   : Real ;

BEGIN
    { Memasukkan Data }
    Nama   := 'Semen' ; { pengisian nilai Semen ke Nama }
    Harga  := 25000   ; { pengisian nilai 25000 ke Harga }
    Jumlah := 10      ;

    { Memproses Data }
    Nilai  := Harga * Jumlah ;

    { Menampilkan Data yang telah diproses }
    Writeln('Nama Barang      : ', Nama) ;
    Writeln('Harga           : ', Harga) ;
    Writeln('Jumlah          : ', Jumlah) ;
    Writeln('Nilai Barang     : ', Nilai) ;
END.

```

Contoh 2.4 Contoh pendeklarasian Variabel

Bila program dijalankan maka akan didapat hasil seperti berikut :

```

Nama Barang      : Semen

```

Harga : 25000
Jumlah : 10
Nilai Barang : 250000

DEKLARASI TYPE

Suatu identifier yang akan dipergunakan dalam program Pascal harus disebutkan tipenya terlebih dahulu. Pascal menyediakan beberapa macam tipe data, terdiri dari :

- a. Tipe data sederhana (simple-type data), terdiri dari
 1. Tipe data standar (standard data type)
 - Integer
 - Real
 - Char
 - Boolean
 2. Tipe data didefinisikan pemakai (user defined data type)
 - Enumerated atau scalar type
 - Subrange type
- b. Tipe data terstruktur (structured type), terdiri dari
 - Array (termasuk didalamnya string)
 - Record
 - File
 - Set
- c. Tipe data penunjuk (pointer type data)

Program Contoh_Type_Variabel ;

Type

```
Pecahan= Real ;
Bulat =Longint ;
Huruf =String[15];
```

menggunakan '='

Var

```
Nama : Huruf ;
Harga,
Jumlah : Bulat ;
Nilai : Pecahan ;
```

menggunakan ':'

Begin

```
Nama := 'Semen' ;
Harga := 25000 ;
Jumlah := 10 ;
Nilai := Harga * Jumlah ;
Writeln('Nama Barang :', Nama) ;
Writeln('Harga :', Harga) ;
Writeln('Jumlah :', Jumlah) ;
Writeln('Nilai Barang :', Nilai) ;
```

End.

DEKLARASI LABEL

Digunakan pada saat kita menggunakan statement *GoTo* untuk meloncat ke suatu statemen yang tertentu. Mendeklarasikan label diawali dengan kata cadangan *Label* diikuti oleh kumpulan identifier label dengan dipisahkan oleh koma dan diakhiri dengan titik koma.

Program Contoh_Label;

Label Start, Selesai ;

Identifier Label

Begin

```
Writeln('Bahasa') ;
Goto Start ;
```

```

      Writeln('Basic') ;
      Writeln('Cobol') ;
Start : _____→
      Writeln('Pascal') ;
      Goto Selesai ;
      Writeln('Fortran') ;
Selesai : _____↑
End.

```

Label yang dituju ;

Contoh 2.5 Contoh penggunaan Label

Jika program ini di jalankan, akan didapatkan hasil :

**Bahasa
Pascal**

DEKLARASI PROCEDURE

Merupakan bagian yang terpisah dari program dan dapat diaktifkan dimanapun di dala program. Prosedur dapat berupa prosedur standar (sudah disediakan Pascal) maupun prosedur yang akan dibuat sendiri oleh pemakai.

Kata cadangan *Procedure* digunakan sebagai judul dari bagian deklarasi prosedur, diikuti oleh identifier yang akan merupakan nama dari prosedurnya.

```

Program Contoh_prosedur ;
Procedure Tambah(x,y : integer ; var Hasil : integer) ;
Begin
  Hasil := x + y ;
End;

```

```

Var z : Integer ;
Begin
  Tambah(5, 6, z) ;
  Writeln('5 + 6 =' , z);
End.

```

Contoh 2.6 Contoh Program procedure

Jika program ini di jalankan, akan didapatkan hasil :

5 + 6 = 11

DEKLARASI FUNCTION

Fungsi juga merupakan bagian program yang terpisah mirip dengan prosedur, tetapi ada beberapa perbedaannya. Fungsi dapat berupa fungsi standar (sudah disediakan oleh pascal) atau fungsi yang akan dibuat sendiri oleh pemakai.

Kata cadangan *Function* mengawali bagian deklarasi fungsi diikuti oleh identifier yang merupakan nama dari fungsinya dan secara optional dapat diikuti dapat diikuti oleh kumpulan parameter, tipe dari fungsinya dan diakhiri dengan titik koma.

```

Program Contoh_Function ;
Function Hasil(x,y : integer ) : integer ;
Begin
  Hasil := x + y ;
End;

```

```

Begin

```

```
Writeln('5 + 6 =' , Hasil(5, 6) );  
End.
```

Contoh 2.7 Contoh Program Function

Jika program ini di jalankan, akan didapatkan hasil : **5 + 6 = 11**

<http://mesran.net>

BAB II ELEMEN-ELEMEN PROGRAM PASCAL

Pascal mempunyai elemen-elemen sebagai berikut :

- Simbol dasar (*basic words*)
- Kata-kata cadangan (*reserved words*)
- Pengenal didefinisikan oleh pemakai (*user define identifier*)
- Tipe data
- Karakter kontrol (*control character*)
- Tanda operasi (*operator*)
- Komentar program (*comment*)
- Statemen (*statement*)
- Prosedur (*procedure*) Bab tersendiri
- Fungsi (*function*) Bab tersendiri

SIMBOL-SIMBOL DASAR

Program pascal dibentuk dari simbol-simbol yang terdiri dari :

- Huruf : a..z , A..Z
- Angka : 0..9
- Simbol Khusus : + - * / = ^ () [] { } . , ; ' # \$

Ex. `WriteLn('Jln. Djamin No 133 Telp. 7222702');`

KATA-KATA CADANGAN

Kata cadangan atau *reserved words* adalah kata-kata yang sudah didefinisikan oleh Pascal yang mempunyai maksud tertentu. Kata-kata cadangan tidak boleh didefinisikan ulang oleh pemakai, sehingga tidak dapat dipergunakan sebagai pengenal (identifier).

Adapun kata-kata cadangan itu antara lain yaitu :

* Absolute	End	* Inline	Procedure	Type
And	* External	* Interface	Program	* Unit
Array	File	* Interrupt	Record	Until
Begin	For	Label	Repeat	* Uses
Case	Forward	Mod	Set	Var
Const	Function	Nil	* Shl	While
Div	Goto	Not	* Shr	With
Do	If	Of	* String	* Xor
Downto	* Implementation	Or	Then	
Else	In	Packed	To	

Kata cadangan yang diberi tanda * menunjukkan kata-kata cadangan yang tidak ada di standar Pascal.

PENGENAL DIDEFINISIKAN PEMAKAI.

Nama yang dipergunakan dalam program pascal disebut dengan pengenal (identifier). Pemakai dapat mendefinisikan sendiri suatu nama sebagai pengenal, tetapi dengan ketentuan sebagai berikut :

- Gabungan huruf dan angka dengan karakter pertama harus berupa huruf. Huruf besar dan kecil dianggap sama.
- Tidak boleh ada spasi
- Tidak boleh ada simbol-simbol khusus, kecuali garis bawah.

- Panjang bebas, tetapi hanya 63 karakter pertama yang dianggap signifikan.

Example :

GajiKaryawan : Benar
 No_Mhs : Benar
 P3K : Benar
 1X : Salah
 A&B : Salah
 A B : Salah

Contoh Mendeklarasikan Variabel diatas :

Var

P3K : Integer ;
 GajiKaryawan : Real ;
 No_Mhs : String[8] ;

TIPE DATA

Tipe data yang disediakan oleh PASCAL meliputi:

▪ 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 yang menunjukkan jenis data, ukuran dalam memori dan rentang nilainya.

Tipe Data	Ukuran Tempat	Jangkauan
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

Contoh bilangan integer adalah: 34 6458 -90 0 1112

Penggolongan tipe data integer tersebut dimaksudkan untuk membatasi alokasi memori yang dibutuhkan misalkan untuk suatu perhitungan dari suatu variabel bilangan diperkirakan nilai maksimumnya 32767 kita cukup mendeklarasikan variabel bilangan sebagai integer (2 byte), daripada sebagai longint (4 byte).

Di dalam kompilernya, Pascal menyediakan konstanta untuk bilangan Integer yaitu: MaxInt and MaxLongInt, pemrogram bisa menggunakannya di dalam programnya tanpa harus terlebih dahulu mendefinisikannya.

- MaxInt bernilai 32.767
- MaxLongint bernilai 2.147.483.647.

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 dapat dilihat pada tabel berikut.

Tipe Data	Ukuran Tempat	Jangkauan
real	6 bytes	2.9 x 10 ⁻³⁹ s/d 1.7 x 10 ³⁸
single	4 bytes	1.5 x 10 ⁴⁵ s/d 3.4 x 10 ³⁸
double	8 bytes	5.0 x 10 ⁻³²⁴ s/d 1.7 x 10 ³⁰⁸
extended	10 bytes	3.4 x 10 ⁻⁴⁹³² s/d 1.1 x 10 ⁴⁹³²
comp	8 bytes	-9.2x 10 ¹⁸ s/d 9.2x 10 ¹⁸

c. Bilangan Char

Tipe data ini menyimpan karakter yang diketikkan dari keyboard, memiliki 255 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. Bilangan 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.

Tipe Variabel	Ukuran Tempat
Boolean	1 byte
WordBool	2 byte
Longbool	3 byte

▪ 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. Sedangkan untuk array, record, dan file perlu dijelaskan dalam bab yang lain karena agak banyak hal-hal yang perlu dibahas.

a. Type Data STRING (ARRAY)

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.

b. Type Data SET

Sebuah set merupakan suatu himpunan yang berisi nilai (anggota). set merupakan Tipe data yang khusus untuk Pascal. Set dalam pemrograman sangat mirip dengan himpunan dalam ilmu matematika,

Contoh: A = { 1, 2, 3, 4, 5 }

Syntax : set of contoh:

Type Digits = set of 0..9 ;
Letters = set of 'A'..'Z';

Type Day = (Sun, Mon, Tue, Wed, Thu, Fri, Sat);
CharSet = set of Char;
Digits = set of 0..9;
Days = set of Day;

Const EvenDigits : Digits = [0, 2, 4, 6, 8];
Vowels : Letters = ['A', 'E', 'I', 'O', 'U', 'Y'];

Untuk type data Record, Pointer, File akan dibahas pada bab khusus

Example 1:

Program Masuk1(layar) ;

Var

A, B : **Integer** ;
C : **Real** ;
D : **String[10]** ;
E : **Char** ;

Begin

{ Memasukkan data untuk masing-masing variabel }

Readln(A) ;
Readln(B) ;
Readln(C) ;
Readln(D) ;
Readln(E) ;

{ Menampilkan data untuk masing-masing variabel }

Writeln(A) ;
Writeln(B) ;
Writeln(C) ;
Writeln(D) ;
Writeln(E) ;

End.

Contoh 3.1 Contoh penggunaan tipe Variabel

KARAKTER KONTROL

Turbo Pascal memungkinkan karakter-karakter kontrol untuk dilekatkan di dalam suatu string. Karakter

- Simbol # : yang diikuti nilai interger antara 0 s.d 255 untuk menunjukkan suatu karakter ASCII

#7 : ASCII 7 adalah Bel
#65 : ASCII 65 adalah karakter A
#10 : ASCII 10 adalah Line Feed
#13 : ASCII 13 adalah Carriage Return

- Simbol ^ : diikuti oleh suatu karakter, menunjukkan hubungannya dengan karakter kontrol.

^G : Control **G** berarti Bel, sama dengan ASCII 7

^M : Control M berarti Carriage Return, sama dengan ASCII 13

^[: Control [berarti Escape

TANDA OPERASI

1. Assignment Operator

Merupakan operator pengerjaan menggunakan simbol titik dua diikuti oleh tanda sama dengan (:=").

Ex : A := B

2. Binary Operator

Merupakan tanda operasi Biner karena operator ini digunakan untuk mengoperasikan dua buah operand. Dapat berbentuk konstanta atau variabel, yang digunakan untuk operasi aritmatika yang berhubungan dengan nilai tipe data integer dan real.

Operator	Operasi	Tipe Operand	Tipe Hasil
*	Perkalian	Real, real Integer, integer Real, integer	Real Integer real
DIV	Pembagian Bulat	Integer, integer	integer
/	Pembagian Real	Real, real Integer, integer Real, integer	Real Real Real
MOD	Modulus (sisa Pembagian)	Integer, integer	Integer
+	Pertambahan	Real, real Integer, integer Real, integer	Real Integer real
-	Pengurangan	Real, real Integer, integer Real, integer	Real Integer Real

3. Unary Operator

Dikatakan Unary karena hanya menggunakan sebuah operand saja. Dapat berupa unary **minus** dan unary **plus**. Berguna untuk menunjukkan variabel tersebut bernilai positif atau negatif

4. Bitwise Operator

Digunakan untuk operasi bit pada nilai integer, yaitu terdiri dari :

NOT AND OR XOR Shl Shr

5. Relational Operator

Merupakan operator hubungan yang digunakan untuk membandingkan hubungan antara dua buah operand dan akan didapatkan hasil tipe *Boolean* yaitu *True* dan *False*.

Operator	Operasi
=	Sama dengan
<>	Tidak sama dengan
>	Lebih besar dari
>=	Lebih besar sama dengan dari
<	Lebih kecil dari
<=	Lebih kecil sama dengan dari
IN	Seleksi dari anggota himpunan

6. Logical Operator

Terdapat 4 buah operator logika yaitu **Not, And, Or, Xor**. Bentuk operator ini sama dengan bitwise operator, tetapi penggunaannya lain. Logical operator bekerja dengan nilai-nilai logika, yaitu True dan False

7. Address Operator

Berguna untuk pengaksesan alamat yang terdapat di memory yaitu address-of operator (@) dan indirection operator (^). Operator @ akan menghasilkan alamat dari suatu bilai variabel dan operator ^ akan memberikan nilai di alamat yang ditunjukkan.

8. Set Operator

Digunakan untuk operasi himpunan.

- +** : Union
- : Perbedaan Himpunan
- *** : Perkalian Himpunan

9. String Operator

Digunakan untuk operasi string. Hanya ada sebuah operator string saja, yaitu operator + yang digunakan untuk menggabung dua buah nilai string.

Example 2 :

```
Var
    Nama1, Nama2, Nama3 : String[15];
Begin
    Nama1 := 'Arief ' ;
    Nama2 := 'Kurniawan' ;
    Nama3 := Nama1 + Nama2 ;
    Writeln(Nama3) ;
End.
```

Contoh 3.2 Contoh String Operator

Arief Kurniawan

KOMENTAR PROGRAM

Digunakan hanya sebagai dokumentasi saja, agar program mudah dibaca. Suatu komenter program ditulis dengan pembatas { dan } atau pembatas (* dan *)

Example 3 :

```
{      Ini adalah komentar program
      untuk memberi keterangan atau penjelasan program
      sehingga baik untuk dokumentasi
}
Begin { awal program utama }
    writeln('Just Komentar')
End. (* akhir program utama *)
```

Contoh 3.3 Komentar Program

Just Komentar

STATEMEN

Merupakan perintah pengerjaan program. Kumpulan dari statement membentuk suatu program. Bila suatu program pascal tidak mengandung suatu statement maka tidak akan ada yang dikerjakan oleh program.

Statement terletak dibagian deklarasi yang diawali oleh kata cadangan **Begin** dan ditutup dengan kata cadangan **End**.

Masing-masing statement diakhiri dengan titik koma (;). Didalam program pascal statement terdiri dari statement sederhana (simple statement) dan statement terstruktur (Structured statement).

LATIHAN

1. Perhatikan listing program di bawah ini, carilah mana dari penggunaan variable dan statemen yang salah.

```

Uses Crt;
Var
    Nama      : String[20];
    Kd        : String[5];
    Jm l      : Integer
    1Hrg, Disc,
    Byr       : Real;
Begin
    Clrscr;
    Write (' Kode Barang      : '); Readln (Kd);
    Write (' Nama              : '); Readln (Nam);
    Write ('Jumlah             : '); Readln (Jml);
    Write ('Harga              : '); Readln (1Hrg);

    Byr := Jm l * 1Hrg;
    Disc := 0,15 * Byr

    Clrscr;
    Writeln ('Daftar Penjualan TOKO LARIS ');
    Writeln ('SEMARANG ');
    Writeln ('===== ');
    Writeln ('Nama                : ', Nm );
    Writeln ('Kode                : ', Kd );
    Writeln ('Jumlah              : ', Jml );
    Writeln ('Harga Rp           : ', Hrg : 8:2);
    Writeln ('Total Harga   Rp   : ', Byr : 8:2);
    Writeln ('Discount Rp      : ', Disc : 8:2);
    Writeln ('Total Pembayaran Rp. : ', ( Byr + Disc) : 8:2);
End.

```

BAB III SEKILAS EDITOR PASCAL

Untuk menggunakan Pascal kita harus menyediakan File-file pendukung pascal itu sendiri diantaranya :

- **TURBO.***

Turbo.Exe, Turbo.TP, Turbo.TPL, Turbo.HLP

Turbo.Exe → Merupakan file Editor pascal. Sehingga bila kita ingin menggunakan Pascal cukup Ketikkan :

C:\Pascal5\Turbo [Enter]

Turbo.TP → file yang berguna untuk menyimpan seting dari Turbo Pascal kita.

Turbo.TPL → merupakan file yang sangat berguna untuk meLINK program atau mengcompile program sebelum di jalankan. Jika file ini tidak ada maka akan berakibat file tidak akan bisa di compile.

Turbo.HLP → merupakan file yang membantu kita mengenai penjelasan perintah-perintah tertentu (sebagai HELP)

- ***.CHR**

Merupakan file yang berekstensi .CHR yang berguna untuk pengaturan Character di dalam pemrograman yang berorientasi dengan grafik. Pascal menyediakan 10 jenis huruf dan tiap-tiap huruf tersebut terletak pada File-file CHR tersebut.

- ***.BGI**

Borland Grafik Interface → berguna untuk mengaktifkan mode berbentuk Grafik. Biasanya cukup satu file saja yang di butuhkan yaitu EGAVGA.BGI

- ***.TPU**

yaitu Turbo Pascal Unit yang berguna untuk unit pembantu misal pada saat kita menggunakan Mode Grafik maka kita memerlukan File Graph.TPU

MENGUNAKAN PASCAL

Pascal yang kita gunakan disini ada beberapa jenis bisa Pascal versi 5, atau Pascal versi 7 atau yang lainnya. Tetapi setiap versi tersebut nama file Executable tetap sama yaitu TURBO. Jadi bila kita menggunakan Pascal cukup cari File Turbo.EXE dan Double Click file tersebut (Windows). Jika kita menggunakan Dos maka ketikkan :

C:\TURBO [Enter]

Jika ada terdapat direktori maka anda harus masuk dahulu ke direktori tersebut.

Contoh C:\Pascal5\TURBO [Enter]

C:\TP\BIN\TURBO [Enter]

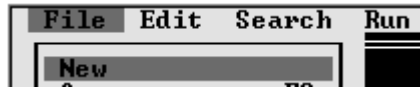
Setelah anda ketikkan perintah tersebut atau double click file Turbo.Exe maka akan muncul suatu Editor Pascal, seperti gambar di bawah ini.



Editor diatas merupakan tampilan dari Pascal Versi 7.0 untuk menghilangkan kotak About cukup tekan **ESC**. Untuk pascal versi 5 perintah-perintah menunya hampir sama. Tetapi akan lebih menguntungkan jika kita menggunakan versi yang lebih tinggi, dikarenakan banyak menawarkan fasilitas-fasilitas yang sangat baik.

MEMBUAT PROGRAM BARU

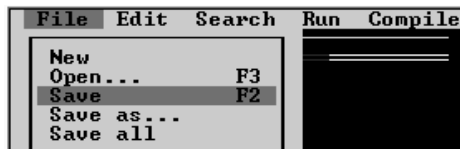
Pada memasuki pascal secara otomatis akan terbentuk file baru dengan Default Name yaitu : NONAME00.Pas. Untuk membuat program baru cukup pilih menu FILE dengan cara tekan **Alt + F**, pilih New.



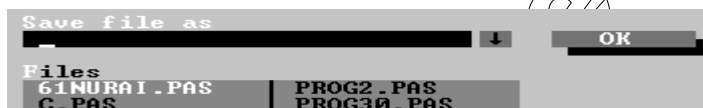
Sehingga muncul kembali editor Noname01.Pas (jika kita tidak menutup Noname00.Pas)

MENYIMPAN PROGRAM

Setelah program selesai diketik, kita bisa menyimpan program kita ke Disk atau Hardisk dengan memilih menu FILE dengan cara tekan **Alt + F**, pilih Save **F2**.



Atau kita bisa langsung dengan menekan **F2** tanpa memilih menu FILE. Setelah kita pilih Save atau tekan F2 maka akan keluar dialog Save.



Untuk menyimpan ke Hardisk cukup ketikkan saja nama filenya Seperti : **Test** kemudian tekan [Enter]

Jika ingin menyimpan ke Disk cukup ketikkan A: sebelum nama File seperti **A:Test** kemudian tekan [Enter]

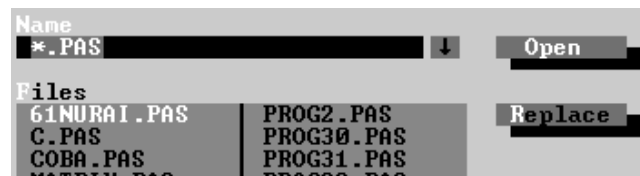
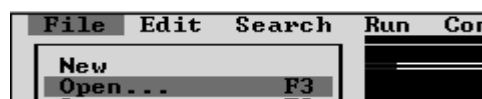
Notes :

Save As... → berguna untuk menyimpan dengan nama yang baru.

Save All → berguna untuk menyimpan seluruh program yang telah terbuka (pada versi 7.0)

MEMBUKA PROGRAM

Untuk membuka program yang telah kita ketikkan sebelumnya yaitu dengan cara memilih menu FILE (**Alt+F**) kemudian pilih **Open...F3** atau cukup dengan menekan **F3** dan akan keluar dialog Open



Untuk memilih file cukup tekan [Enter] pada dialog open dan kemudian kita bisa memilih dengan menekan tombol Panah. Atau kita bisa mengetikkan nama file jika kita mengetahui file yang akan kita buka.

Untuk membuka file yang berada di Disk cukup ketikkan **A: *.Pas** [Enter] (kita tinggal menyebutkan Nama Direktori)

KELUAR KE LAYAR DOS

Untuk keluar ke layar DOS cukup pilih DOS Shell pada menu FILE



Jika kita telah berada pada layar DOS maka untuk kembali ke Pascal cukup ketikkan Exit.

KELUAR DARI PASCAL

Untuk keluar dari pascal cukup tekan **Alt + X**. atau dengan menekan memilih *Exit* pada menu File. Pada Pascal 5 kita memilih Quit dari Menu(Alt + Q) .

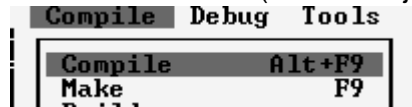
Notes :

- Change Dir → untuk mengganti direktory Aktif.
- Print → untuk mencetak program yang sedang aktif di layar Editor.
- Printer Setup → untuk mengatur setting Printer.

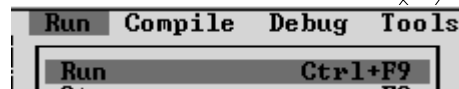
KOMPILASI PROGRAM

Setelah program di ketik, sebelum di jalankan maka kita harus melakukan Compile terhadap Program dengan cara :

Menekan **Alt + F9** (bisa **F9** saja) atau memilih perintah *Compile* pada menu **COMPILE**

**MENJALANKAN PROGRAM**

Jika pada saat kita mengcompile program tidak muncul pesan kesalahan, maka kita bisa menjalankan program dengan perintah *Run* pada menu **RUN**. Atau cukup menekan **Ctrl + F9**.

**MEMBUAT EXECUTABLE**

Untuk bisa menggunakan program pada media DOS atau Windows, maka kita perlu untuk membuat Executable program. Dengan cara.

1. Rubah Destination Memory menjadi Destination Disk
Caranya cukup pilih menu **COMPILE** kemudian pilih Destination Memory. Dengan sendirinya akan berubah menjadi **DISK**.
2. Kemudian anda tinggal menekan **F9**. Perhatikan proses Compile Destination menjadi Disk.

BAB IV STATEMEN INPUT- OUTPUT

MEMASUKKAN DATA

Cara yang paling banyak digunakan untuk memasukkan data adalah dengan cara mengetikkannya lewat keyboard. Turbo Pascal menyediakan prosedur untuk maksud tersebut, yaitu prosedur **Read**, **Readln** dan **Readkey**.

Prosedur standar **Readln** bekerja dengan membaca data yang diketikkan melalui keyboard dan diakhiri dengan penekanan enter. **Readln** juga untuk memasukkan data perbaris, artinya setelah anda menekan tombol Enter, maka akan ganti baris, sedang prosedur standar **Read** tidak ganti baris, masih dalam baris yang sama.

Sedangkah prosedur standar **Readkey** membaca data yang ditekan dari keyboard tetapi hanya satu karakter saja.

Example 1 :

```
Var
    C, F      : Real ;
Begin
    Write('Berapa Celcius ? '); Readln(C);
    F := 1.8 * C + 32 ;
    Writeln ;
    Writeln(C, ' Celcius adalah : ', F, ' Fahrenheit');
End.
```

Contoh 5.1 Contoh memasukkan data

Berapa Celcius ?

Example 2 :

```
Var
    Jawab_C : Char ;
    Jawab_B : Boolean ;
Begin
    Write(' Benar atau Salah (B/S) ? '); Readln(Jawab_C);
    Jawab_B := (Jawab_C = 'B');
    Writeln('yang anda maksud adalah : ', Jawab_B);
End.
```

Contoh 5.2 Contoh memasukkan data

Benar atau Salah (B/S) ?

MENAMPILKAN OUTPUT

Untuk menampilkan hasil dengan bahasa Pascal digunakan prosedur standar **Write** dan **Writeln**. Perbedaannya adalah **Write** menampilkan hasil tanpa ganti baris dan tampilan berikutnya akan disambung dalam baris yang sama, sedangkan prosedur standar **Writeln** digunakan untuk menampilkan tampilan perbaris, akan ganti baris untuk tampilan berikutnya.

Example 3 :

```
Var
    Nama : String[15];
Begin
    Nama := 'Dewi';
    Write('Nama : ');
    Write(Nama);
```

End.
Contoh 5.3 Contoh memasukkan data

Nama : Dewi

Example 4 :

```
Var
  Nama : String[15] ;
Begin
  Nama := 'Dewi' ;
  Writeln('Nama : ') ;
  Writeln(Nama) ;
End.
```

Contoh 5.4 Contoh memasukkan data

**Nama :
Dewi**

Bandingkanlah hasil dari program 5.3 dan program 5.4. Akan tampak sangat jelas sekali perbedaan antara kedua program tersebut dikarenakan penggunaan dari statement *Write* & *Writeln*.

PENGATURAN LETAK DI LAYAR PROSEDUR CLRSCR

Digunakan untuk membersihkan layar dari tampilan-tampilan sebelumnya dan meletakkan cursor di posisi ujung kiri atas dari layar. Prosedur standar ini bila digunakan harus menyebutkan terlebih dahulu unit standar *Crt*.

Example 5:

```
Uses Crt ;
Begin
  ClrScr ;
  Writeln('Bahasa Pascal');
End.
```

Contoh 5.5 Membersihkan layar.

Coba Anda Hilangkan statemen **Uses Crt ;** dan jalankan program ! Jelaskan ?

PROSEDUR GOTOXY

Digunakan untuk meletakkan cursor diposisi layar yang ditunjukkan oleh nilai *xPos* dan *yPos*. Nilai *xPos* menunjukkan posisi sumbu X (horisontal, kolom 1 sampai 80) sedangkan nilai *yPos* menunjukkan sumbuY (vertikal, posisi baris 1 sampai 25). Prosedur standar ini bila digunakan harus menyebutkan terlebih dahulu unit standar *Crt*.

Example 6:

```
Uses Crt ;
Begin
  ClrScr ;
  GotoXY(10,15) ; Writeln('Bahasa Pascal') ;
End.
```

Contoh 5.6. menggunakan prosedur GotoXY

Akan tampak di layar monitor pada kolom 10 dan baris 15 tulisan **Bahasa Pascal**

PENGATURAN WARNA

Pascal menyediakan 5 buah prosedur standar yang dapat digunakan untuk mengatur warna tampilan teks di layar, yaitu :

TextColor, TextBackground, LowVideo, NormVideo, Highvideo.

Penggunaan prosedur diatas melibatkan unit Crt.

Nilai	Warna	Nilai	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
		128	Blink

Example 7 :

Uses Crt ;

Begin

TextColor(4) ; TextBackGround(15) ;

ClrScr ;

TextColor(Blue) ; GotoXY(10,6) ; Writeln('Tulisan ini berwarna Biru) ;

TextColor(0) ; GotoXY(20,7) ; Writeln('Tulisan ini berwarna Hitam) ;

TextColor(14) ; GotoXY(30,8) ; Writeln('Tulisan ini berwarna Kuning) ;

TextColor(1+128) ; GotoXY(40,9) ; Writeln(' dan ini berwarna Biru berkedip) ;

End.

Contoh 5.7. menggunakan prosedur TextColor & TextBackGround

LATIHAN

1. Simpan dengan nama:
- Latih51.Pas**

Program Penjualan; { *Heading program* }

Uses Crt;

Var { *Bagian Deklarasi* }

Nama : String[20] ;

Kode : String[5] ;

Jumlah : Integer ;

Harga, Bayar : Real ;

Begin { *Bagian Pernyataan* }{ *===== Input =====* }Clrscr; { *Menghapus layar* }

Write ('Masukkan Nama Barang = '); Readln (Nama);

Write ('Masukkan Kode Barang = '); Readln (Kode);

Write ('Masukkan Jumlah Barang = '); Readln (Jumlah);

Write ('Masukkan Harga Barang = '); Readln (Harga);

{ *===== Proses =====* }

Bayar := Jumlah * Harga;

{ *===== Keluaran =====* }

Clrscr;

Writeln ('Daftar Penjualan TOKO BUDIDARMA');

Writeln ('MEDAN');

Writeln ('=====');

Writeln ('Nama Barang = ', Nama);

Writeln ('Kode Barang = ', Kode);

Writeln ('Jumlah Barang = ', Jumlah);

Writeln ('Harga Barang = ', Harga : 8:2);

Writeln ('Jumlah Pembayaran = ', Bayar : 8:2);

Readln;

End.

Jelaskan Kegunaan dari **Readln** diatas Statemen End.

2. Simpan dengan nama:
- Latih52.Pas**

Uses Crt;

Var Warna : byte;

Begin

Warna :=3;

Clrscr;

Window(40,5,70,25);

Textcolor(Warna);

Write('Informatika oh yess.....');

End.

Jelaskan Arti dari Statemen **Window**

3. Simpan dengan nama:
- Latih53.Pas**

Program LuasLingkaran ;

Uses Crt;

Var

Jari : Real ;

```

    Luas : integer ;
Begin
    Clrscr;
    Write ('Masukkan Jari-jari ke Lingkaran : '); Readln (Jari);
    Luas := 3.14 * jari *jari ;
    Writeln ('Luas Lingkaran adalah = ', Luas : 8:2);
    Readln;
End.

```

4. Simpan dengan nama: **Latih54.Pas**

```

Uses Crt;
Var
    Nm           : String[20];
    Kd           : String[5];
    Jml          : integer;
    Hrg, Disc,
    Byr          : Real;
Begin
    Clrscr;
    Write (' Kode Barang      : '); Readln (Kd);
    Write (' Nama              : '); Readln (Nm);
    Write ('Jumlah            : '); Readln (Jml);
    Write ('Harga              : '); Readln (Hrg);

    Byr := Jml * Hrg;
    Disc := 0.15 * Byr ;

    Clrscr;
    Writeln ('Daftar Penjualan TOKO LARIS ');
    Writeln ('SEMARANG ');
    Writeln ('===== ');
    Writeln ('Nama          : ', Nm );
    Writeln ('Kode          : ', Kd );
    Writeln ('Jumlah        : ', Jml );
    Writeln ('Harga Rp      : ', Hrg : 8:2);
    Writeln ('Total Harga   Rp : ', Byr : 8:2);
    Writeln ('Discount Rp   : ', Disc : 8:2);
    Writeln ('Total Pembayaran Rp. : ', ( Byr – Disc) : 8:2);
    Readln;
End.

```

TUGAS PRAKTIKUM

Semua latihan dibawah ini anda gunakan prosedur GotoXy, Clrscr & pengaturan Warna.

1. Buatlah program untuk menghitung Luas Persegi Panjang. (**Tugas51.Pas**)
2. Buatlah program untuk mencari Luas trapesium ? (**Tugas52.Pas**)
Rumus : $\text{Luas} = (\text{sisi pendek} + \text{sisi panjang}) / 2 \times \text{tinggi}$
3. Buatlah program untuk menghitung Volume Bola. (**Tugas53.Pas**)
Rumus : $\text{Volume} = 4/3 \times 3.14 \times \text{jari}^3$
4. Buatlah Program dengan Layout di bawah ini.

**DAFTAR NILAI MAHASISWA
JURUSAN INFORMATIKA**

```
=====
Nama Mahasiswa      : .....
NIM Mahasiswa       : .....
Nilai Tugas         : .....
Nilai Mid Test      : .....
Nilai Final Test    : .....
Rata – Rata         : .....
=====
```

Ketentuan Program :

- a. Nama, NIM, Nilai Tugas, Nilai Mid Test dan Nilai Final Test diperoleh dari input
- b. Sedangkan rata-rata diperoleh dari rumus:
 $\text{Rata-rata} = 0,5 \times \text{Nilai Tugas} + 0,25 \times \text{Nilai Mid Test} + 0,25 \times \text{Nilai Final Test}$
- c. Text Judul Warna Merah, Background warna biru

Simpan dengan nama : **Tugas54.Pas**

5. Buatlah program untuk menghitung jumlah harga photo Copy. (**Tugas55.Pas**)

BAB V STATEMEN KENDALI

Di dalam tehnik pemrograman kita mengenal adanya '**structure programming**'. Pada dasarnya struktur programming disusun berdasarkan statement yang terstruktur pula.

Terdapat 3 proses eksekusi utama di dalam *structure programming*, yaitu :

- o Proses secara 'Sequential' (berurutan)
- o Proses secara 'Selection/Decision' (menggunakan statement kendali, yang merupakan inti pembahasan bab ini)
- o Proses secara 'Repetition' (perulangan / Bab VII)

PROSES SEQUENTIAL (Berurutan)

Proses secara sequential terjadi bila proses berjalan dari atas ke bawah dan dikerjakan dari baris-perbaris. Dengan kata lain proses dikerjakan urut kebawah. Proses sequential dibagi dalam dua macam yaitu :

▪ STATEMEN TUNGGAL

Yaitu jika dalam sebuah lokasi hanya terdapat sebuah statement.

Example 1 :

```
Uses Crt ;
Var A : Longint ;
Begin
    Writeln('Nilai A : ' ) ; Readln (A) ;
    If A > 10 then
        Write ('baris ke I')      { Statement Tunggal }
    Else
        Write('Nilai A,Kecil' ) ;
End.
```

Contoh 6.1 : Program dengan Statemen Tunggal

▪ STATEMEN MAJEMUK

Yaitu Statement jenis ini sering juga disebut 'compound statement'. Pengertiannya adalah statement yang terdiri dari sejumlah statement (sekelompok statement) yang akan dieksekusi dengan urutan yang sama. Sekelompok statement ini dapat dianggap sebagai satu statement. Statement Majemuk ditandai dengan **BEGIN** dan di akhiri dengan **END**; .

Example 2 :

```
Uses Crt
Var A : Longint ;
Begin
    Writeln('Nilai A : ' ) ; Readln (A) ;
    If A > 10 then
        Begin
            Write('AKU ..' )
            Write('Senang Makan ....');
            A:=A+1
        End;
    End.
```

Contoh 6.2 : Program dengan Statemen Majemuk

PROSES DECISION (Kendali)

Statement kendali digunakan untuk proses pengambilan keputusan. Dimana proses akan dikerjakan bila kondisi yang disyaratkan sesuai (bernilai true/benar). Pada Pascal terdapat dua statement kendali yaitu :

- IF
- CASE

STATEMEN IF

Struktur dari statemen If dapat berupa If-Then, If-Then-Else, If Kondisi Jamak

- **KONDISI TUNGGAL** yaitu Struktur If-Then
Syntax : If ungkapan Then Statemen

Example 3 :

```
Var
    NilaiUjian : Real ;
    Ket       : String[11] ;
Begin
    Ket := 'Tidak Lulus' ;
    Write('Nilai yang didapat  ?') ; Readln(NilaiUjian) ;

    If NilaiUjian > 60 Then Ket := 'Lulus';
    Writeln(Ket) ;
End.
```

Contoh 6.3 : Program If Kondisi Tunggal

- **KONDISI GANDA** Yaitu Struktur If-Then-Else
Syntax : If *Kondisi* Then
 Statemen1
 Else
 Statemen2 ;

Notes : diatas penulisan Else tidak menggunakan ‘;’

Example 4 :

```
Var
    NilaiUjian : Real ;
Begin
    Write('Nilai yang didapat  ?') ; Readln(NilaiUjian) ;

    If NilaiUjian > 60 Then
        Writeln('Lulus ?')
    Else
        Writeln('Tidak Lulus') ; {Penulisan ; untuk mengakhiri Fungsi dari IF}
End.
```

Contoh 6.4 : Program If Kondisi Ganda

- **KONDISI JAMAK** Yaitu Struktur If-Then Else If-Then-Else If-Then-Else
Syntax : If *Kondisi* Then
 Statemen1
 Else If Kondisi2 Then
 Statemen2
 Else If Kondisi3 Then
 Statemen3 ;

Digunakan untuk mengecek nilai suatu kondisi, jika kondisi bernilai benar atau salah, maka akan mengerjakan statement yang bersesuaian. ‘;’ terakhir untuk menutup if secara keseluruhan.

Example 5 :

```

Var
  BilA,
  BilB   : Longint ;
  Koment :String[15];
Begin
  Write('Bilangan A : ') ; Readln(BilA);
  Write('Bilangan B : ') ; Readln(BilB);

  If BilA > BilB Then
    Koment := 'A Lebih Besar dari B'
  Else If BilA < BilB Then
    Koment := 'A Lebih Kecil dari B'
  Else If BilA = BilB Then
    Koment := 'A Sama dengan B' ;

  Writeln('Koment   :', Koment) ;
End.

```

Contoh 6.5 : Program If Kondisi Jamak

NESTED IF (Tersarang)

Contoh Variasi IF tersarang

```

If <kondisi> then
Begin
  If <kondisi> then
    <statement>
  else
    <statement>
end
Else
Begin
  <statement>
End;
if <kondisi> Then <statement>
Else if <kondisi> Then
  <statement>
Else
  <statement>
....

```

STATEMEN CASE

Statemen Case dapat berbentuk struktur *Case-Of* atau *Case-Of...Else*. Kegunaan dari statemen Case hampir sama dengan statemen IF

A. STATEMEN CASE-OF

Mempunyai suatu ungkapan logika yang disebut dengan selector dan sejumlah statement yang diawali dengan suatu label permasalahan (case label) yang mempunyai tipe data yang sama dengan selector.

Syntax :

Case Ungkapan Of

```
Daftar Case-label1 : Statemen1 ;
Daftar Case-label2 : Statemen2 ;
Daftar Case-label3 : Statemen3 ;
Daftar Case-label4 : Statemen4 ;
```

End;

Example 6 :

```
Var Nilai : Char;
Begin
    Write('Nilai Huruf yang didapat ?') ;
    Readln(Nilai) ;
    Case Nilai Of
        'A' : Writeln('Sangat Baik');
        'B' : Writeln('Sangat Baik');
        'C' : Writeln('Sangat Baik');
        'D' : Writeln('Sangat Baik');
        'E','F' : Writeln('Sangat Baik');

    End;
```

End.

Contoh 6.6 : Program Case-OF

B. STATEMEN CASE-OF...ELSE

Pada struktur case-of bila tidak ada kondisi yang terpenuhi berarti tidak ada statement didalam lingkungan case of yang diproses. Dengan struktur case-of...else, bila tidak ada kondisi yang terpenuhi, maka statement yang akan diproses didalam lingkungan case-of adalah statement yang ada di else.

Syntax :

Case Ungkapan Of

```
Daftar Case-label1 : Statemen1 ;
Daftar Case-label2 : Statemen2 ;
Daftar Case-label3 : Statemen3 ;
Daftar Case-label4 : Statemen4 ;
Else
```

```
Statemen5;
```

End;

Example 7 :

Uses Crt;

Var TEKAN:Char;

Begin

Clrscr;

Write(' Tekan Sembarang tombol..!');

TEKAN:= Readkey;

Writeln;

Case TEKAN of

```
'A'..'Z','a'..'z' : Writeln(' Anda menekan tombol huruf');
```

```
'+', '-', '*', '/' : Writeln(' Anda menekan tombol operator aritmatika');
```

```
Else
```

```
Writeln('Anda menekan Special Char');
```

End;

Repeat Until Keypressed ;

End.

Contoh 6.7 : Program Case-OF...Else

Bagaimana jika kita menekan Angka dari 0 sampai 9 ?

LATIHAN

1. Nama Program : **Latih61.Pas**

Uses Crt;

Var UMUR : Byte;

Begin

Clrscr;

Write('Inputkan Umur : ') ; Readln(UMUR);

If UMUR in [0..5] then writeln(' BALITA')

Elseif UMUR in [6..16] then writeln('REMAJA')

Elseif UMUR in [17..99] then writeln('DEWASA');

End.

Jelaskan Penggunaan In ?

2. Nama Program : **Latih62.Pas**

USES Crt;

VAR

Beli, Bayar, Disc : Real ;

BEGIN

Write ('Jumlah Beli = '); Readln (Beli);

Disc := 0;

IF Beli >= 50000 THEN Disc := Beli * 0.1;

Bayar := Beli - Disc;

Writeln ('Jumlah Bayar =', Bayar:10:2);

END.

3. Nama Program : **Latih63.Pas**

USES Crt ;

VAR

Nm : String [20];

Mtk : String [20];

Ket : String [10];

NI : Byte ;

BEGIN

Writeln (' Entri Nilai Mahasiswa') ;

Writeln ('-----') ;

Write (' NAMA :') ; Readln (Nm) ;

Write (' MATA KULIAH :') ; Readln (Mtk) ;

Write (' NILAI [0..100] :') ; Readln (NI) ;

Writeln ('-----') ;

IF NI >=60 then Ket:='Lulus' Else Ket:='Gagal';

Writeln ('keterangan :', Ket) ;

Writeln ('-----') ;

END.

4. Nama Program : **Latih64.Pas**

```

USES Crt;
VAR
  Np      : String [20] ;
  Kd      : String [3]  ;
  Jp      : String [30] ;
  Hb, Hkwh, Tp : Real   ;
  P       : Byte       ;
BEGIN
  Gotoxy(30, 5); Write ('Entri Data Pembayaran Rekening Listrik') ;
  Gotoxy(20, 6); Write ('_____') ;
  Gotoxy(20, 7); Write ('Nama Pelanggan : '); Readln (np)      ;
  Gotoxy(20, 8); Write ('Kode Rekening   : '); Readln (Kd)      ;
  Gotoxy(20,13); Write ('_____') ;

  IF Kd = 'L01' then
  BEGIN
    Jp := 'Pabrik' ;
    Hb := 40000   ;
    HKwh := 1250   ;
  END
  ELSE IF Kd='L02' then
  BEGIN
    Jp := 'Swalayan & toko';
    Hb := 35000   ;
    Hkwh:= 1000   ;
  END
  ELSE IF Kd = 'L03' then
  BEGIN
    Jp := 'Rumah' ;
    Hb := 10000   ;
    Hkwh:= 250    ;
  END ;

  Gotoxy(45,8) ; Write('Jenis Pelanggan : ', jp);
  Gotoxy(20,9) ; Write('Harga Beban      : ', Hb:10:1);
  Gotoxy(20,10); Write('Harga KWH       : ', Hkwh:10:1);
  Gotoxy(20,11); Write('Pemakaian    : '); Readln(p) ;
  Tp := Hb + Hkwh * P ;
  Gotoxy(20,12); write ('Total Pembayaran Rp :', Tp:8:2 );
END.

```

5. Nama Program : **Latih65.Pas**

```

PROGRAM IF_MAJEMUK_AND;
USES Crt;
VAR
  Nama : String[20];
  Mtk  : String[20];
  N    : Byte ;
  NH   : Char ;
BEGIN
  Write ('NAMA      : '); Readln (Nama) ;
  Write ('MATA KULIAH : '); Readln (Mtk) ;

```

Write ('NILAI [0..100] : '); Readln (N) ;

IF (N >= 80) AND (N <= 100) then NH := 'A'
 ELSE IF (N >= 70) AND (N < 80) then NH := 'B'
 ELSE IF (N >= 60) AND (N < 70) then NH := 'C'
 ELSE IF (N >= 50) AND (N < 60) then NH := 'D'
 ELSE NH := 'E' ;

Clrscr;
 Writeln ('Nama : ', Nama);
 Writeln ('Matakuliah : ', MTK);
 Writeln ('Nilai Angka : ', N);
 Writeln ('Nilai Huruf : ', NH);

END.

6. Nama Program : **Latih66.Pas**

VAR

Nama : String[20];
 Mtk : String[20];
 N : Byte ;
 NH : Char ;

BEGIN

Write ('NAMA : '); Readln (Nama) ;
 Write ('MATA KULIAH : '); Readln (Mtk) ;
 Write ('NILAI [0..100] : '); Readln (N) ;

CASE N OF

80..100 : NH := 'A' ;
 70..79 : NH := 'B' ;
 60..69 : NH := 'C' ;
 46..59 : NH := 'D' ;
 0..45 : NH := 'E' ;

END;

Clrscr;

Writeln ('Nama : ', Nama);
 Writeln ('Matakuliah : ', MTK);
 Writeln ('Nilai Angka : ', N);
 Writeln ('Nilai Huruf : ', NH);

END.

TUGAS PRAKTIKUM

1. Mencari Nilai Huruf Mahasiswa

DAFTAR NILAI MAHASISWA
JURUSAN INFORMATIKA

```

=====
Nama Mahasiswa   : .....
NIM Mahasiswa    : .....
Nilai Tugas      : .....
Nilai Mid Test   : .....
Nilai Final Test : .....
Total Nilai      : .....
Nilai Huruf      : .....
=====
  
```

Ketentuan Program :

- Nama, NIM, Nilai Tugas, Nilai Mid Test dan Nilai Final Test diperoleh dari input
- Sedangkan Total Nilai diperoleh dari rumus:

$$\text{Total Nilai} = (20\% * \text{Nilai Tugas}) + (30\% * \text{Nilai Mid}) + (50\% * \text{Nilai Final})$$
- Range dari Nilai Huruf yaitu :

Total Nilai	Nilai Huruf
80 – 100	A
79 – 70	B
69 – 51	C
50 – 41	D
40 – 0	E

Simpan dengan nama : **TUGAS61.PAS**

2. Mencari Nilai Huruf

Format Input yaituSimpan dengan nama : **TUGAS62.PAS**

DAFTAR NILAI MAHASISWA
JURUSAN INFORMATIKA

```

=====
Nama Mahasiswa   : .....
NIM Mahasiswa    : .....
Nilai Tugas      : .....
Nilai UTS        : .....
Nilai UAS        : .....
=====
  
```

Format Output yaitu

```

NIM : .....
NAMA : .....

Nilai Angka    Nilai Huruf
Tugas         ....      x
UTS            ....      x
UAS            ....      x
AKHIR          ....      x
  
```

Untuk mencari Nilai Akhir yaitu

$$\text{Total Nilai} = (20\% * \text{Nilai Tugas}) + (30\% * \text{UTS}) + (50\% * \text{UAS})$$
Range Masing-Masing tiap nilai yaitu :

Nilai Huruf	Tugas	UTS	UAS	Nilai Akhir
A	65 - 100	90 – 100	85 – 100	80 - 100
B	80 – 94	80 – 89	75 – 84	70 – 79

C	70 – 79	60 – 79	60 – 74	51 – 69
D	< 69	40 – 59	35 – 59	41 – 50
E	< 69	< 39	< 34	<40

3. Mencari Sisa Uang Kuliah

Format Input yaituSimpan dengan nama : **TUGAS63.PAS****PEMBAYARAN UANG KULIAH**

```

=====
Nama Mahasiswa : .....
NPM            : .....
Kode           : .....
Jurusan        : .....
Pembayaran Awal Rp : .....
Jumlah Cicilan : ...
Besar Cicilan Rp. : .....
Pembayaran Ke  : ...
Uang Kuliah Rp  : .....
Uang Kuliah Terbayar Rp : .....
Sisa Uang Kuliah Rp : .....
=====

```

Ketentuan Program :

- Nama, NPM, Kode & Pembayaran Ke diperoleh dari input
- Untuk mencari Jurusan, Pembayaran Awal, Jumlah Cicilan & Besar Cicilan dari Kode yaitu :

Kode	Jurusan	Pembayaran Awal Rp	Jumlah Cicilan	Besar Cicilan Rp.
SI	Sistem Informasi	2100000	7	300000
TI	Teknik Informatika	2500000	7	300000
KA	Komputer Akuntansi	1750000	6	200000
MI	Manajemen Informatika	1500000	6	250000

- Uang Kuliah = Pembayaran Awal + (Jumlah Cicilan & Besar Cicilan)
- Uang Kuliah Terbayar = Pembayaran Awal + (Pembayaran Ke * Besar Cicilan)
- Sisa Uang Kuliah = Uang Kuliah – Uang Kuliah Terbayar

BAB VI STATEMEN PERULANGAN

Di dalam Pascal terdapat tiga macam perulangan yaitu dengan menggunakan statemen **For...Do**, **While...Do** dan **Repeat...Until**.

▪ STRUKTUR PERULANGAN FOR...DO

Jenis perulangan ini digunakan untuk mengulang statemen atau satu blok statemen berulang kali sejumlah yang ditentukan. Syntax For...Do adalah sebagai berikut:

For VarControl := nAwal To/DownTo nAkhir Do Statemen/Blok Statemen

Perulangan FOR dapat berbentuk perulangan positif, perulangan negatif dan perulangan tersarang.

a. FOR ... DO POSITIF

Merupakan perulangan dengan penghitung (counter) dari kecil ke besar atau dengan kata lain pertambahannya positif.

Syntax perulangan FOR

For VarControl := nAwal To nAkhir Do Statemen/Blok Statemen

Syarat Perulangan ini yaitu **nAwal** lebih **KECIL** dari **nAkhir**

Example 1 :

```
Var
  I : Integer ;
Begin
  For I := 1 to 5 Do Writeln('Pascal');
End.
```

Contoh 7.1: Penulisan kata Pascal
(satu statement)

Example 2 :

```
Var
  I : Integer ;
Begin
  For I := 1 to 5 Do
  Begin
    Write(I);
    Writeln(' Pascal');
  End;
End.
```

Contoh 7.2: Penulisan kata Pascal (lebih
dari 1 statement)

Example 3 :

```
Var
  x : Integer ;
  Celcius, Fahrenheit : Real ;
Begin
  Writeln('-----') ;
  Writeln('  Celcius      Fahrenheit  ');
  Writeln('-----') ;
  Celcius := 0 ;
  For x := 1 to 10 Do
  Begin
    Fahrenheit := 1.8 * Celcius + 32 ;
    Writeln(Celcius:8:2, Fahrenheit:14:2);
    Celcius := Celcius + 0.25 ;
  End;
  Writeln('-----') ;
```


End.

Contoh 7.3 : Menghitung nilai fahrenheit

b. FOR ... DO NEGATIF

Merupakan perulangan dengan penghitung (counter) dari besar ke kecil atau dengan kata lain pertambahannya negatif.

Syntax perulangan FOR

For VarControl := nAwal **DownTo** nAkhir **Do** Statemen/Blok Statemen

Syarat Perulangan ini yaitu **nAwal** lebih BESAR dari **nAkhir**

Example 4 :

Var

Celcius, Fahrenheit : Real ;

Begin

Writeln('-----'); ;

Writeln(' Celcius Fahrenheit '); ;

Writeln('-----'); ;

Celcius := 0 ;

For Celcius := 15 DownTo 0 Do

Begin

Fahrenheit := 1.8 * Celcius + 32 ;

Writeln(Celcius:8:2, Fahrenheit:14:2);

End;

Writeln('-----'); ;

End.

Contoh 7.4 : menampilkan Fahrenheit dengan menggunakan DownTo

▪ NESTED FOR ... DO (tersarang)

Perulangan tersarang adalah perulangan yang berada di dalam perulangan yang lainnya. Pada perulangan ini perulangan yang lebih dalam akan diproses terlebih dahulu sampai habis, lalu baru perulangan yang lebih luar.

Example 5 :

Var

I, J : Integer;

Begin

For I := 1 To 5 Do

Begin

For J := 1 To 3 Do

Begin

Write (I:8, J:3) ;

End;

End;

End.

Contoh 7.5 : Terdapat dua Perulangan yaitu I & J

▪ STRUKTUR PERULANGAN WHILE...DO

Jenis perulangan ini digunakan untuk mengulang statemen atau satu blok statemen berulang kali yang jumlahnya belum bisa ditentukan, tergantung nilai kondisi yang terletak antara While...do. Perulangan akan dikerjakan bila nilai kondisinya bernilai benar dan akan berhenti jika kondisi bernilai salah. Syntax While...Do adalah sebagai berikut:

While *ungkapan-logika* **Do** Statemen/Blok Statemen

Example 6 :

```
Var
    I : Integer ;
Begin
    I := 0 ;
    While I < 5 Do
    Begin
        Writeln(I) ;
        Inc(I) ;
    End;
End.
```

Contoh 7.6 : Penggunaan While...Do

Example 7 :

```
Uses Crt ;
Var
    Nilai, Total, RataRata : Real ;
    N, I                    : Integer ;
Begin
    Clrscr ;
    {**** memasukkan jumlah data****}
    Write('Jumlah Data ? ') ; Readln(N);
    I := 0 ; Total := 0 ;
    While I < N Do
    Begin
        I := I + 1 ;
        Write('Nilai data Ke ', I, ' ? ') ;
        Readln(Nilai)
        Total := Total + Nilai ;
    End;
    {**** menghitung nilai rata-ratanya****}
    RataRata := Total/N;
    {**** menampilkan hasil****}
    Writeln ;
    Writeln('Total Nilai      = ', Total :8:2) ;
    Writeln('Jumlah Data = ', N :8:2) ;
    Writeln('Rata - Rata = ', RataRata :8:2) ;
End.
```

Contoh 7.7 : Mencari Total Nilai/Rata-rata sesuai dengan jumlah data

▪ STRUKTUR PERULANGAN REPEAT...UNTIL

Jenis perulangan ini digunakan untuk mengulang statemen atau satu blok statemen berulang kali, yang jumlahnya belum bisa ditentukan, tergantung nilai kondisi yang terletak setelah Until. Perulangan akan dikerjakan bila nilai kondisinya bernilai salah dan akan berhenti jika kondisi bernilai benar. Syntax perulangan Repeat...Until sebagai berikut :

Repeat Statemen/Blok Statemen **Until** *ungkapan*

Example 8 :

```
Var
    I : Integer ;
```

```

Begin
    I := 0 ;
    Repeat
        Inc(I) ;
        Writeln(I) ;
    Until I = 5 ;
End.

```

Contoh 7.8 : bentuk penggunaan Repeat...Until

LATIHAN

1. Nama Program : **Latih71.Pas**

```

Var
    i, j : integer;
Begin
    For i:= 1 to 5 do
        For j := 3 downto 1 do
            Writeln ('Reformasi
                total');
        Writeln ('Indonesia');
    End.

```
2. Nama Program : **Latih72.Pas**

```

Var
    i : integer;
Begin
    i:= 1;
    While i<=5 do
        Begin
            Writeln('Gatot');
            i:= i + 1;
        End;
    End.

```
3. Buatlah program dengan Output seperti berikut : Nama Program **Latih73.Pas**

1 STMIK Budidarma	2 STMIK Budidarma
3 STMIK Budidarma	4 STMIK Budidarma
5 STMIK Budidarma	6 STMIK Budidarma
7 STMIK Budidarma	8 STMIK Budidarma
4. Buatlah program di bawah ini : Nama Program **Latih74.Pas**

```

PROGRAM LoopPositif ;
USES Crt;
VAR
    I : Integer;
BEGIN
    Writeln(' I      I*2      I*3      I*4 ');
    Writeln('-----');
    FOR I:=1 To 10 do
        Writeln(I:4, I*2:5, I*3:5, I*4:5);
    Writeln('-----');
END.

```
5. Buatlah program dengan Output seperti berikut : Nama Program **Latih75.Pas**

5	10	15	20
25	30	35	40
45	50	55	60

2. Buatlah program berikut ini . Nama Program : **Latih76.Pas**

```

VAR i : Byte;
BEGIN
  Writeln(' I      Keterangan ');
  Writeln('-----');
  FOR i:= 1 To 10 Do
  BEGIN
    Write (i);
    IF(i Mod 2)=0 then Writeln ( ' Genap') else Writeln ( ' Ganjil');
  END;
  Writeln('-----');
END.

```

6. Buatlah program dengan Output seperti berikut : Nama Program **Latih77.Pas**

```

USES Crt;
VAR
  Nim      :String[8];
  Nama     :String[30];
  Mtk      :String[31];
  NT,NM,NS:Byte;
  NA       :Real;
  NH,
  LG       :Char;

BEGIN
  lg:='y' ;
  WHILE Upcase (lg)='Y' Do
  BEGIN
    Clrscr;
    Writeln('INPUT DATA NILAI MAHASISWA STMIK Budidarma ');
    Writeln('_____');
    Write ('NIM      :'); Readln( NIM );
    Write ('NAMA     :'); Readln( NM );
    Write ('MATA KULIAH :'); Readln( MTK );
    Write ('NILAI TUGAS  :'); Readln( NT );
    Write ('NILAI MID    :'); Readln( NM );
    Write ('NILAI SEMESTER :'); Readln( NS );

    NA := (0.20*NT) + (0.30*NM) + (0.5*NS);
    CASE ROUND(NA) OF
      80..100 : NH :='A';
      70..79  : NH :='B';
      60..69  : NH :='C';
      45..59  : NH :='D';
      0..45   : NH :='E';
    END;

    Writeln('NILAI AKHIR      :', NA:3:1 );
    Writeln('NILAI HURUF       :', NH );
    Writeln('_____');
    Writeln('Masih Ingin Menghitung [Y/T] :') ; Readln(Lg) ;
  END ;
END.

```

TUGAS PRAKTIKUM

1. Buatlah program berikut ini
 Apabila diketahui harga beras per kilogram sebagai berikut:
 Jenis IR 64 = Rp. 2000,-
 Jenis C4 = Rp. 3000,-
 Jenis Delanggu = Rp. 3550,-
 Jenis Rajalele = Rp. 6000,-

Buatlah program menggunakan bahasa Pascal, dengan output sebagai berikut:

**TABEL HARGA BERAS IR 64, C4, DELANGGU DAN RAJALELE
 TOKO "MANTEP" SALATIGA**

No	Jumlah (Kg)	IR64	C4	Delanggu	RajaLele
1	5
2	10
...
50	250

Simpan dengan nama : **TUGAS71.PAS**

3. Ubahlah Program ini sehingga dapat menggunakan Perulangan bila **Ingin Menghitung Lagi [Y/T]** : kita isi dengan **Y** maka dia akan menginputkan kembali dari awal

**DAFTAR NILAI MAHASISWA
 JURUSAN INFORMATIKA**

=====

Nama Mahasiswa :
 NIM Mahasiswa :
 Nilai Tugas :
 Nilai Mid Test :
 Nilai Final Test :
 Total Nilai :
 Nilai Huruf :

=====

Ingin Menghitung Lagi [Y/T] :

Ketentuan Program :

- a Nama, NIM, Nilai Tugas, Nilai Mid Test dan Nilai Final Test diperoleh dari input
 b Sedangkan Total Nilai diperoleh dari rumus:
 $\text{Total Nilai} = (20\% * \text{Nilai Tugas}) + (30\% * \text{Nilai Mid}) + (50\% * \text{Nilai Final})$
 c Range dari Nilai Huruf yaitu :

Total Nilai	Nilai Huruf
80 – 100	A
79 – 70	B
69 – 51	C
50 – 41	D
40 – 0	E

Simpan dengan nama : **TUGAS72.PAS**

4. Dengan menggunakan perulangan For...do dan Gotoxy, buatlah program untuk membuat kata 'STMIK Budidarma' berjalan dari kiri ke kanan dan balik dari kanan ke kiri.

BAB VIII TYPE DATA TERSTRUKTUR

Type Data Terstruktur di dalam bahasa Pascal terdiri dari :

- Array
- Record
- Set
- File

TIPE DATA ARRAY / LARIK

Larik yaitu deretan data yang mempunyai type data sejenis.

Ex : Daftar Nomor Telpn, Tabel Pajak dll.

Tiap rinci data disebut komponen/Elemen Larik.

Tiap larik bisa dioperasikan menggunakan pemilih elemen larik yang disebut INDEX atau Subskrib. Dan Index ini diletakkan dalam tanda '[]', sesudah nama larik.

DEKLARASI ARRAY

Deklarasi perubah selalu diawali dengan kata VAR. Dalam deklarasi perubah biasa, kita hanya perlu menentukan type data perubah tersebut misalnya: real, integer, boolean atau char, sedangkan pada deklarasi larik (array), kita perlu mendefinisikan type larik, juga harus menentukan banyaknya index yang akan mengisi larik tersebut.

Syntax :

VAR

nmVariabel : ARRAY [index] OF typeData

Keterangan :

nmVariabel : Nama larik yang dideklarasikan

index : Batasan/Jumlah Index (cacah elemen)

typeData : Tipe data dari Larik

Larik dapat bertipe data sederhana byte, word, integer, real, boolean, char atau string dan tipe data skalar atau subrange.

Index dari larik menunjukkan maksimum banyaknya elemen-elemen dari larik. Index larik ini dapat berupa tipe subrange atau skalar apapun (integer, byte, word, char) kecuali tipe real.

Suatu larik tidak hanya dapat berupa suatu variabel yang dideklarasikan di bagian variabel, tetapi juga dapat berupa suatu konstanta yang dideklarasikan di bagian deklarasi konstanta.

Example 1 :

Var

Nama : Array[1..10] of String[20] ;

I : Byte ;

Begin

Clrscr

{Bagian Input Data}

For I := 1 To 10 Do

Begin

Writeln('Inputkan Nama Ke ', I, ' : '); Readln(Nama[I]) ;

End;

{Bagian Menampilkan Data yang telah diinputkan }

For I := 1 To 10 Do

Writeln('Nama Ke ', I, ' : ', Nama[I]) ;

Readln;

End.

Contoh 8.1 : Program dengan Penggunaan Array.

Example 2 :

Var

Nama : Array[1..30] of String[20] ;

Gaji : Array[1..30] of Longint ;

i,j : Byte ;

Begin

Clrscr

{Bagian Input Data}

Writeln('Inputkan Jumlah Data : '); Readln(J) ;

For I := 1 To J Do

Begin

Writeln('Inputkan Nama Ke ', I, ':') ; Readln(Nama[I]) ;

Writeln('Inputkan Gaji Ke ', I, ':') ; Readln(Gaji[I]) ;

End;

Clrscr ;

{Bagian Menampilkan Data yang telah diinputkan }

GotoXy(10, 4) ; Write('-----')

GotoXy(10, 4) ; Write('No. Nama Gaji');

GotoXy(10, 6) ; Write('-----')

For I := 1 To J Do

Begin

GotoXy(10, 6+I) ; Write(I) ;

GotoXy(14, 6+I) ; Write(Nama[I]) ;

GotoXy(34, 6+I) ; Write(Gaji[I]) ;

End

GotoX(10, J+7) ; Writeln('-----')

Readln

End.

Contoh 8.2 : Program untuk menginputkan Nama & Gaji menggunakan Array.

TIPE DATA RECORD (Rekaman)

Seperti halnya array (larik), record juga punya elemen, di sini elemennya disebut "Field". Tiap elemen bisa punya type data yang berbeda. Banyaknya field dapat bervariasi atau bisa juga tetap.

Record yang punya field yang bervariasi disebut 'Variant Record'. Deklarasi type data Record dapat ditetapkan di bagian deklarasi Var atau Type.

Syntax :

TYPE

nmPengenal = RECORD

Field1 : type1;

Field2 : type2;

:

fieldn : typen;

END;

VAR

nmRec: nmPengenal ;

Keterangan :

nmPengenal : Nama Dari Record ;
 Field1, Field2, Fieldn : Nama dari Variabel/Field yang terdapat dalam Record
 Type1, Type2, Typen : Type Dari tiap-tiap Field yang dideklarasikan
 nmRec : Nama Dari Record yang di deklarasikan

Ex.

TYPE

Mahasiswa = **RECORD**

NIM : String[8] ;
 Nama : String[20] ;
 Alamat : String[30] ;
 Sex : Char ;
 Umur : Byte ;

END;

VAR

RecMhs : Mahasiswa ;

MEMBACA & MENULIS FIELD

Dapat dilaksanakan pada seluruh record atau sebagian record.

Syntax :

nmRec.Field1 ;

Tanda titik harus ditulis

Notasi ini disebut Penandaan field ('field designator').

Berbeda dengan statement pemberian (assignment), maka untuk membaca dan menulis record selalu dilakukan menurut Field yang ada, tidak dilakukan secara keseluruhan record.

Contoh : Membaca `Readln(RecMhs.NIM);`
 `Readln(RecMhs>Nama);`

Menulis `Writeln(RecMhs.NIM);`
 `Writeln(RecMhs>Nama);`

Example 3 :

TYPE

Mahasiswa = RECORD

NIM : String[8] ;
 Nama : String[20] ;
 Umur : Byte ;

END;

Var

RecMhs : Mahasiswa ;

Begin

Clrscr

{Bagian Input Data}

Writeln('NIM Anda :'); readln(RecMhs.NIM);

Writeln('Nama :'); readln(RecMhs>Nama);

Writeln('Umur :'); readln(RecMhs.Umur);

{Bagian Menampilkan Data yang telah diinputkan }


```

Writeln('NIM Anda      : ', RecMhs.NIM ) ;
Writeln('Nama         : ', RecMhs.Nama ) ;
Writeln('Umur         : ', RecMhs.Umur ) ;
Readln;

```

End.

Contoh 8.3 : Program dengan Penggunaan Array.

STATEMEN WITH

Digunakan untuk mengurangi penulisan yang berulang-ulang untuk nama recordnya. Maka dengan menggunakan statement WITH hanya cukup sekali saja menyebutkan nama recordnya.

Syntax :

```

WITH nmRecord DO
    {langsung dimasukkan/dicetak fieldnya}

```

Example 4 :

TYPE

```

Mahasiswa = RECORD
    NIM      : String[8] ;
    Nama     : String[20] ;
    Umur     : Byte      ;

```

END;

Var

```

RecMhs : Mahasiswa ;

```

Begin

Clrscr

With RecMhs **Do**

Begin

{Bagian Input Data}

```

Writeln('NIM Anda      : '); readln(NIM ) ;

```

```

Writeln('Nama         : '); readln(Nama ) ;

```

```

Writeln('Umur         : '); readln(Umur ) ;

```

{Bagian Menampilkan Data yang telah diinputkan }

```

Writeln('NIM Anda      : ', NIM ) ;

```

```

Writeln('Nama         : ', Nama ) ;

```

```

Writeln('Umur         : ', Umur ) ;

```

End;

Readln;

End.

Contoh 8.4 : Penggunaan Statemen WITH pada RECORD.

LATIHAN1. Nama Program : **Latih81.Pas**

Program Latih81; {Program input data pada record}

Uses crt;

Type

Barang = record

Kode : String[5] ;

Nama : String[20] ;

Jumlah: Integer ;

Harsat : Longint ;

End;

Var

RecBrg : Array [1..40] of Barang ;

Lagi : Char;

I, J : Integer;

Begin

Clrscr ; I:=0 ; J := 0 ;

{ Untuk Menginputkan Data Barang}

Repeat

Inc(I)

Write('Kode Barang :'); readln(RecBrg[I].Kode);

Write('Nama :'); readln(RecBrg[I].Nama);

Write('Jumlah :'); readln(RecBrg[I].Jumlah);

Write('Harga satuan :'); readln(RecBrg[I].Harsat);

Write('Ada lagi [Y/T] : '); readln(lagi);

Until (lagi<>'Y') or (lagi<>'y') ; { bisa ditulis **Until Uppcase(lagi)='Y' ; }**

{ Menampilkan Data Barang }

Clrscr ;

GotoXy(10, 4) ; Write('-----');

GotoXy(10, 5) ; Write('No. Kode Nama Jumlah Harga');

GotoXy(10, 6) ; Write('-----');

For J := 1 To I Do

Begin

GotoXy(10, 6+J) ; Writeln(I) ;

GotoXy(13, 6+J) ; Write(RecBrg[J].Kode) ;

GotoXy(19, 6+J) ; Write(RecBrg[J].Nama) ;

GotoXy(39, 6+J) ; Write(RecBrg[J].Jumlah) ;

GotoXy(39, 6+J) ; Write(RecBrg[J].Harsat) ;

End

GotoX(10, J+7) ; Write('-----');

Readln;

End.

2. Nama Program : **Latih82.Pas**

USES CRT ;

VAR

NIP : Array[1..20] of string [8] ;

NM : Array[1..20] of string [20];

GOL : Array[1..20] of string [3] ;

i, j : byte ;

Gaji, TGaji : Longint ;

lg : Char ;

```
BEGIN
```

```
  i := 1 ;
```

```
  REPEAT
```

```
    Clrscr;
```

```
    Gotoxy(20,2); Write(' INPUT DATA GAJI PEGAWAI');
```

```
    Gotoxy(10,3); Write('_____');
```

```
    Gotoxy(10,4); Write(' NIP      :');
```

```
    Gotoxy(10,5); Write(' NAMA      :');
```

```
    Gotoxy(10,6); Write(' GOL [I,II,III] :');
```

```
    Gotoxy(10,7); Write('_____');
```

```
    Gotoxy(10,8); Write(' Isikan Data Lagi[Y/T] :');
```

```
    Gotoxy(27,4); Readln( NIP[ i ] );
```

```
    Gotoxy(27,5); Readln( NM[ i ] );
```

```
    Gotoxy(27,6); Readln( GOL[ i ] );
```

```
    Gotoxy(33,8); Readln( Lg );
```

```
    inc( i );
```

```
  UNTIL Upcase(Lg)='T';
```

```
  Clrscr;
```

```
  Gotoxy(3,3);write('LAPORAN DATA GAJI PEGAWAI');
```

```
  Gotoxy(3,4);write('_____');
```

```
  Gotoxy(3,5);write(' No    NIP    NAMA          GOL    GAJI Rp ');
```

```
  Gotoxy(3,6);write('_____');
```

```
  FOR j:=1 TO i -1 DO
  BEGIN
```

```
    If Gol [ j ]='I' then Gaji := 750000
```

```
    Else If Gol [ j ]='II' then Gaji := 1000000
```

```
    Else If Gol [ j ]='III' then Gaji := 1500000
```

```
    Else Gaji :=0;
```

```
    Gotoxy(5 ,6+j); Write( j:2, ' ', NIP[ j ]: 8, ' ', NM[ j ] );
```

```
    Gotoxy(44,6+j); Write( Gol [ j ] );
```

```
    Gotoxy(49,6+j); Write( Gaji:8 );
```

```
    TGaji := TGaji + Gaji ;
```

```
  END;
```

```
  Gotoxy(3,7+j);
```

```
  Write('_____');
```

```
  Gotoxy(3,8+j); Write(' TOTAL GAJI          RP.', tgaji:10);
```

```
END.
```

TUGAS PRAKTIKUM

1. Buatlah Program untuk Menampilkan LayOut seperti berikut ini

Laporan Data Nilai Mahasiswa STMIK Budidarma

No.	NIM	Nama	Mata Kuliah	Nilai Mid	Nilai Semester	Nilai Huruf	Ket
1.	0303001	Aldyan	Pascal I	90	90	A	Lulus
...

Ketentuan Proses

- a. NIM, Nama, Nilai Mid Test dan Nilai Sem diperoleh dari input

- b. Sedangkan Total Nilai diperoleh dari rumus:

$$\text{Total Nilai} = (40\% * \text{Nilai Mid}) + (60\% * \text{Nilai Sem})$$

- c. Range dari Nilai Huruf yaitu :

Total Nilai	Nilai Huruf
80 – 100	A
79 – 70	B
69 – 51	C
50 – 41	D
40 – 0	E

- d. Ket di dapat dari Nilai E yaitu Ket = 'GAGAL' selain itu Ket = 'LULUS'

Simpan dengan nama : **TUGAS81.PAS**

2. Buatlah program berikut ini

CV. "INTAN GANDINI" yang bergerak dalam konsultan teknik akan membuat laporan gaji semua karyawan yang berjumlah 100 orang, dengan data-data sebagai berikut :

- NoPeg (Nomer pegawai)
- Nama (Nama pegawai)
- Gol (Golongan)
- JJJ (Jumlah jam lembur)

Laporan yang diinginkan dari data-data tersebut adalah

LAPORAN GAJI PEGAWAI**CV. "INTAN GANDINI"**

Jl. Waru Doyong No. 69 Semarang

No	Nomor Pegawai	Nama Pegawai	Gol	Gaji Pokok	Tunjangan	Jumlah Jam Lembur	Uang Lembur	Gaji Bersih
99	x(8)	x(25)	x	9(9)	9(9)	9999	9(9)	9(9)
99	x(8)	x(25)	x	9(9)	9(9)	9999	9(9)	9(9)

Ketentuan Proses

1. Mencari Gaji Pokok, Tunjangan, Uang Lembur

Gol	Gaji Pokok	Tunjangan	Uang Lembur/Jam
1	100000	50000	3500
2	200000	100000	4500
3	350000	200000	6000
4	500000	350000	Tidak Ada

2. Uang Lembur = Jumlah Jam Lembur * Uang Lembur/Jam

3. Gaji Bersih = Gaji Pokok + Tunjangan + Uang Lembur

Simpan dengan nama : **TUGAS82.PAS**