

UNIVERSITAS NEGERI PADANG  
FACULTY OF SCIENCE

**BAHAN AJAR**  
**DASAR-DASAR PEMROGRAMAN KOMPUTER**

**FIS006 3 SKS**



**PAKHRUR RAZI, M.SI**

UNIVERSITAS NEGERI PADANG	AGUSTUS 2009
TANGGAL	2-5-2009
KELOMPOK	Hafiah
KOLEKSI	K1
NO. INVENTARIS	157/Hd/2009-d.1
LOKASI	005.3 RA2 d.1

**JURUSAN FISIKA FMIPA**  
**UNIVERSITAS NEGERI PADANG**

© 2008

# **BAB 1**

## **KONSEP DASAR**

### **PEMROGRAMAN PASCAL**

#### **1.1 Perkembangan Pascal**

Pascal adalah bahasa tingkat tinggi yang orientasinya pada segala tujuan, dirancang oleh Prof. Niklaus Wirth dari Technical University di Zurich, Switzerland. Nama pascal diambil sebagai penghargaan terhadap Blaise Pascal, ahli matematik dan filosofi terkenal abad 17 dari Perancis.

#### **1.2 Struktur Program Pascal**

Struktur dari suatu program pascal terdiri dari sebuah judul program dan suatu blok program atau badan program. Blok program dibagi lagi menjadi dua bagian, yaitu : bagian deklarasi dan bagian pernyataan. Secara ringkas, struktur suatu program pascal dapat terdiri dari :

1. Judul program
2. Blok program
  - a. Bagian deklarasi
    - deklarasi label
    - deklarasi konstanta
    - deklarasi tipe
    - deklarasi variable
    - deklarasi prosedur
    - deklarasi fungsi
  - b. Bagian pernyataan

Program pascal yang paling sederhana adalah program yang hanya terdiri dari sebuah bagian pernyataan saja.

*Begin*  
*End.*

Bentuk umum dari bagian pernyataan adalah :

*Begin*  
*Statement;*  
*End .*

<p>Contoh :</p> <pre> Begin   Writeln ( 'Saya pascal' );   Writeln ( '-----' ) End . </pre>	<p>Outputnya : Saya pascal</p> <p style="text-align: center;">-----</p>
---	---

Program pascal tidak mengenal aturan penulisan di kolom tertentu, jadi boleh ditulis mulai kolom berapapun. Penulisan *statement-statement* pada contoh program yang menjorok masuk beberapa kolom tidak mempunyai pengaruh diproses, hanya dimaksudkan supaya mempermudah pembacaan program, sehingga akan lebih terlihat bagian-bagiannya, dan baik untuk dokumentasi.

*Judul program* sifatnya *optional* dan tidak signifikan didalam program. Jika ditulis dapat digunakan untuk memberi nama program dan daftar dari parameter tentang komunikasi program dengan lingkungannya yang sifatnya sebagai dokumentasi saja. *Judul program* bila ditulis harus terletak pada awal dari program dan diakhiri dengan titik koma.

<p>Contoh :</p> <pre> Program contoh ; Begin Writeln ( 'Fisika FMIPA UNP' ); Writeln( '-----' ); End . </pre>
---

### 1.3 Deklarasi

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

#### 1.3.1 Deklarasi Konstanta

Definisi konstanta diawali dengan kata cadangan **Const** diikuti oleh kumpulan identifier yang diberi suatu nilai konstanta. *Data konstanta nilainya sudah ditentukan dan pasti, tidak dapat dirubah didalam program.*

Contoh :

```
Program contoh_konstanta;
Const
  Potongan = 0.2 ; Gaji = 25000 ;
  Namaperusahaan = 'PT ABC' ;
Begin
  Writeln ( 'Potongan =', Potongan ) ;
  Writeln ( 'Gaji   =', Gaji ) ;
  Writeln ( 'Nama   =', Namaperusahaan) ;
End.
```

### 1.3.2 Deklarasi Variabel

Variabel adalah identifier yang berisi data yang dapat diubah-ubah nilainya didalam program. Menggunakan kata cadangan **Var** sebagai judul didalam bagian deklarasi variable dan diikuti oleh satu atau lebih identifier yang dipisahkan koma, diikuti dengan titik dua dan tipe dari datanya diakhiri dengan titik koma.

Contoh :

```
Var
  Total, Gaji, Tunjangan : real ;
  Jumlahanak : integer ; Keterangan : string [25] ;
Begin
  Gaji := 500000 ; Jumlahanak := 3 ;
  Tunjangan := 0.25 * Gaji + Jumlahanak * 30000 ;
  Total := Gaji + Tunjangan ;
  Keterangan := 'Karyawan Teladan' ;
  Writeln ( ' Gaji bulanan = Rp. ', Gaji ) ;
  Writeln ( ' Tunjangan   = Rp. ', Tunjangan ) ;
  Writeln ( ' Total gaji   = Rp. ', Total ) ;
  Writeln ( ' Keterangan   = Rp. ', Keterangan ) ;
End .
```

### 1.3.3 Deklarasi Tipe

Pascal menyediakan beberapa macam tipe data, yaitu :

1. Tipe data sederhana, terdiri dari :

- a. tipe data standar : integer, real, char, string, Boolean.

- b. Tipe data didefinisikan pemakai : enumerated atau scalar, subrange
- 2. Tipe data terstruktur : array, record, file, set
- 3. Tipe data penunjuk

```

Contoh :

Type
  Pecahan = real ;
  Bulat = integer ;
  Huruf = string [25] ;
Begin
  .....
  .....
  .....
End .
    
```

**1.3.4 Deklarasi Label**

Jika program menggunakan statement Goto untuk meloncat ke suatu statement yang tertentu, maka dibutuhkan suatu label pada statement yang dituju dan label tersebut harus di deklarasikan terlebih dahulu pada bagian deklarasi. Menggunakan kata cadangan Label diikuti oleh kumpulan identifier label dengan dipisahkan oleh koma dan diakhiri dengan titik koma.

<pre> Contoh Label   100, selesai ; Begin   Writeln ( 'Bahasa' ) ;   Goto 100 ;   Writeln ( 'Cobol' ) ; 100 :   Writeln ( 'Pascal' ) ;   Goto selesai ;   Writeln ( 'Fortran' ) ; Selesai : End .         </pre>	<pre> Output :   Bahasa            Pascal         </pre>
--	--

**1.3.5 Deklarasi Prosedur**

Prosedur merupakan bagian yang terpisah dari program dan dapat diaktifkan dimanapun didalam program. Prosedur dibuat sendiri bilamana program akan dibagi-bagi menjadi beberapa blok-blok modul. Prosedur dibuat di dalam program dengan cara mendeklarasinya dibagian deklarasi prosedur menggunakan kata cadang Procedure

<p>Contoh :</p> <pre> Procedure tambah ( x, y : integer ; var hasil : integer ) ; •Begin   Hasil := x+y End; {program utama} var   z : integer ; Begin   Tambah ( 2, 3, z ) ;   Writeln ( ' 2 + 3 = ', z ); End </pre>	<p>Output</p> <p>2 + 3 = 5</p>
--	--------------------------------

### 1.3.6 Deklarasi Fungsi

Fungsi juga merupakan bagian program yang terpisah mirip dengan prosedur, tetapi ada beberapa perbedaannya. Kata cadangan yang digunakan **Function**.

<p>Contoh :</p> <pre> Function Tambah ( x, y : integer ) : integer ; Begin   Tambah := x + y ; End ; {program utama} Begin   Writeln ( ' 2 + 3 = ', Tambah ( 2, 3 ) ) ; End. </pre>
---

## 1.4 Unit

Suatu unit adalah kumpulan dari konstanta, tipe-tipe data, variable, prosedur dan fungsi- fungsi. Tiap-tiap unit tampak seperti suatu program Pascal yang terpisah. Unit standar sudah merupakan kode mesin (sudah dikompilasi), bukan kode sumber Pascal lagi dan sudah diletakkan di memori pada waktu menggunakan pascal. Untuk menggunakan suatu unit, harus diletakkan suatu anak kalimat **Uses** diawal blok program, diikuti oleh daftar nama unit yang digunakan.

### 1.4.1 Unit System

Sebenarnya adalah pustaka dari runtime Turbo Pascal yang mendukung semua proses yang dibutuhkan pada waktu runtime. Unit system berisi semua prosedur dan fungsi standar turbo pascal. Unit system ini akan secara otomatis digunakan didalam program, sehingga *boleh tidak disebutkan didalam anak kalimat Uses*.

### 1.4.2 Unit Crt

Digunakan untuk memanipulasi layer teks ( windowing, peletakkan cursor dilayar, color untuk teks, kode extended keyboard dan lainnya). Unit standar crt hanya dapat digunakan oleh program yang digunakan dikomputer IBM PC, IBM AT, IBM PS/2 atau yang kompatibel dengannya.

```

Contoh :
Uses crt ;
Begin
  Clrscr ;
  Writeln ( 'Hallo' ) ; Writeln ( '-----' ) ;
End.
    
```

Prosedur standar yang menggunakan unit crt adalah :

AssignCrt	HighVideo	Sound
ClrEol	InsLine	TextBackgroun
ClrScr	LowVideo	TextColor
Delay	NormVideo	TextMode
DellLine	NoSound	Window
GotoXY		

Fungsi standar yang menggunakan unit Crt adalah :  
 KeyPressed, ReadKey, WhereX, WhereY

### 1.4.3 Unit Printer

Merupakan unit yang sangat kecil dirancang untuk penggunaan printer didalam program.

```

Contoh :

Uses printer ;
Begin
  Writeln ( Lst, 'Hallo' ) ;
  Writeln ( Lst, '-----' ) ;
End
    
```

#### 1.4.4 Unit Dos

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

Contoh :

```
Uses Dos ;  
Begin  
  Writeln ( DiskFree(0), ' byte sisa isi disk ' ) ;  
End.
```

#### 1.4.5 Unit Graph

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

Contoh :

```
Uses graph ;  
Var  
  DriveGrafik, ModeGrafik : integer ; l, x, y : integer ;  
Begin  
  DriveGrafik := detect ;  
  .....  
End .
```



## BAB 2

# JENIS DATA SEDERHANA & INPUT/OUTPUT DATA

### 2.1 Tipe Data

#### A. Data Tipe Sederhana

Dihubungkan dengan sebuah identifier untuk sebuah data. Data tipe ini digolongkan menjadi tipe data standar dan tipe data yang didefinisikan oleh pemakai.

- Tipe data standar : Integer, real, char, string, Boolean
- Tipe data yang didefinisikan pemakai Enumerated atau scalar type, subrange type

#### B. Data Tipe Terstruktur

Terdiri dari beberapa data item yang dihubungkan satu dengan lainnya. Masing-masing grup dari data item dihubungkan dengan suatu identifier tertentu. Ada 4 macam yang termasuk dalam data ini : array, record, file dan set

#### C. Data Tipe Penunjuk

Digunakan untuk membuat data terstruktur tipe dinamik.

#### 1.1.1 Integer

Merupakan nilai bilangan bulat baik dalam bentuk desimal maupun hexadecimal. Ada 5 macam tipe data integer berdasarkan jangkauan nilai, yaitu :

Tipe	Ukuran Memori (dalam byte)	Jangkauan Nilai
Byte	1	0 ... 255
Shortint	1	- 128 ... 127
Integer	2	- 32768 ... 32767
Word	2	0 ... 65535
Longint	4	- 2147483648 ... 2147483647

```

Contoh (1) :
Tipe data byte digunakan jika nilai tidak lebih dari 255
Var
Jumlah : byte ;
Begin
Jumlah := 200 ;
Writeln (' Nilai jumlah = ', Jumlah);
End.
    
```

```

Contoh (2) :
Bila nilai yang digunakan lebih besar dari 255, tetapi
tidak lebih besar dari 32767, maka dapat digunakan
tipe integer

Var
Jumlah ; integer ;
Begin
Jumlah := 25000 ;
Writeln (' nilai jumlah = ', Jumlah) ;
End
    
```

### 1.1.2 Real

Nilai konstanta numeric real berkisar dari 1E-38 sampai 1E+38. E menunjukkan nilai 10 pangkat, dan tipe data ini menempati memori sebesar 6 byte.

Contoh :	Output dari contoh program
<pre> Var Nilai1, Nilai2 : real ; Begin Nilai1 := 12345678901.2345 ; Nilai2 := 12345 ; Writeln (' Nilai 1 = ', Nilai1) ; Writeln (' Nilai 2 = ', Nilai2) ; End                     </pre>	<pre> Nilai1 = 1.2345678901E+10 Nilai2 = 1.2345000000E+04                     </pre>

### 1.1.3 Karakter

Nilai data karakter berupa sebuah karakter yang ditulis diantara tanda petik tunggal, seperti : ' A '. penggunaan variable untuk menyimpan tipe data karakter ini harus dideklarasikan dengan tipe Char.

<pre> Contoh : Var Huruf : char ; Begin Huruf := ' D ' ; Writeln ( ' Hurufnya adalah : ', Huruf ) End         </pre>	<pre> Output program :  Hurufnya adalah : D         </pre>
--	--

### 1.1.4 String

Merupakan urutan dari karakter yang terletak diantara tanda petik tunggal. Nilai data string akan menempati memori sebesar banyaknya karakter string ditambah dengan 1 byte. Bila panjang dari suatu string didalam deklarasi variable tidak disebutkan, maka dianggap panjangnya adalah 255 karakter.

```

Contoh :
Var
Tanggal : string [10] ;
Hari : string [6] ;
Jarak : char ;

Begin
Tanggal := ' 19 Februari 2007 ' ;
Hari := ' Senin ' ;
Jarak := ' ' ;
Writeln (Tanggal, Jarak, Hari) ;
End.
    
```

### 1.1.5 Boolean

Mempunyai dua buah nilai, yaitu true dan false.

```

Contoh :
Var
Lagi : Boolean ;
Begin
Lagi := True ;
Writeln ( ' nilai lagi adalah ', Lagi ) ;
End.
    
```

## 1.2 Binary Operator

Disebut dengan binary operator karena operator ini digunakan untuk mengoperasikan dua buah operand. Operand dapat berbentuk konstanta ataupun variable. Digunakan untuk operasi

aritmatika yang berhubungan dengan tipe data integer dan real.

<b>Operator</b>	<b>Operasi</b>	<b>Tipe Operand</b>	<b>Tipe</b>
*	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

### 1.3 Relational Operator

Operator hubungan digunakan untuk membandingkan hubungan antara dua buah operand dan akan didapatkan hasil tipe Boolean, yaitu, true atau 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

### 1.4 Input Dan Output

#### 1.4.1 Memasukkan Data

Cara yang paling banyak digunakan untuk memasukkan data adalah dengan mengetikkannya melalui keyboard. Prosedur yang digunakan untuk melakukannya dalam pascal adalah *Read* dan *Readln*.

Perbedaan perintah *read* dan *readln* adalah :

**Readln** : untuk memasukkan data per baris, setelah menekan tombol enter, maka akan ganti baris

**Read** : untuk memasukkan data dan tidak berganti baris, masih

---

---

dalam baris yang sama

Contoh : Var Sisi, p, l : integer ;  Begin Write ('Masukkan sisi persegi panjang ='); Readln (sisi) ; Write ('Masukkan panjang persegi panjang ='); Readln (p) ; Luas := sisi x panjang ; Writeln ('Luas persegi panjang adalah', Luas) ; End.
Output program : Masukkan sisi persegi panjang = <u>15</u> { data yg diinput} Masukkan panjang persegi panjang = <u>10</u> Luas persegi panjang adalah 10

#### 1.4.2 Menampilkan Hasil

Untuk menampilkan hasil digunakan prosedur standar *Write* atau *Writeln*.

Perbedaan perintah *write* dan *writeln* adalah :

**Write** : menampilkan hasil tanpa ganti baris dan tampilan berikutnya akan disambung dalam baris yang sama

**Writeln**: Untuk menampilkan tampilan per baris, akan ganti baris untuk tampilan berikutnya.

(untuk contoh program, dapat dilihat contoh di sub bab 1.4.1 Memasukkan Data)

## BAB III

# MANIPULASI STRING

### 1.1 Merangkai String

Pengerjaan suatu string hanya mempunyai sebuah operator, yaitu ' + '.

Bila operator ini digunakan untuk penambahan nilai numeric, maka akan berfungsi menjumlahkan dua buah nilai elemen numeric, tetapi pada string operator ini digunakan untuk merangkai dua buah elemen string menjadi sebuah string. Panjang maksimum suatu string yang diijinkan oleh Pascal adalah 255 karakter.

Contoh :

Const

```
Jenis = ' Bahasa ' ;  
Nama = ' Pascal ' ;
```

Var

```
Sifat : string[10] ;  
Kalimat : string[80] ;
```

Begin

```
Sifat := ' terstruktur ' ;  
Kalimat := Jenis + ' ' + Nama + ' ' + 'merupakan' + Jenis + ' ' + sifat;  
Writeln (kalimat) ;
```

End.

Output program :

Bahasa Pascal merupakan bahasa terstruktur

### 1.2 Penyeleksian String

Dua buah string dikatakan sama bila nilai string yang pertama sama isi dan panjangnya dengan nilai string yang kedua. Blank dianggap nilai yang signifikan.

### 1.3 Prosedur Standar untuk Operasi String

#### 1.3.1 Prosedur Standar Delete

Bentuk Umum :

Delete (var s : string, index : integer, count : integer);

---

Digunakan untuk membuang sejumlah karakter tertentu (jumlah karakter yang dibuang ditunjukkan oleh nilai integer *count*) mulai dari posisi tertentu (ditunjukkan oleh nilai integer *index*) dari suatu nilai string ditunjukkan oleh pengenal variable string *var s*). Jika posisi *index* lebih besar dari panjang string *s*, maka tidak ada karakter yang akan terhapus.

Contoh :

```
Var
    Bahasa : string [12] ;
    Urutan , Posisi : integer ;
Begin
    Bahasa := 'Turbo Pascal' ;
    For Urutan := 1 to 12 do
        Begin
            Posisi := 14 - Urutan ;
            Delete (Bahasa, Posisi, 1);
            Writeln (Bahasa) ;
        End ;
    End.
End.
```

### 1.3.2 Prosedur Standar Insert

Bentuk Umum :

```
Insert ( source : string , var s : string , index : integer );
```

Digunakan untuk menyisipkan suatu string (ditunjukkan oleh nilai string *source*), ke nilai string yang lainnya (ditunjukkan oleh nilai string *var s*) mulai dari posisi yang ditunjukkan oleh nilai integer *index*.

Contoh :

```
Var
    Nama : string [30] ;
Begin
    Nama := 'Pal' ;
    Insert ('sca', Nama, 3) ;
End.
```

### 1.3.3 Prosedur Standar Str

Bentuk Umum :

Str ( x [:width [: decimals]] , var s : string ) ;

- Digunakan untuk merubah nilai numeric (ditunjukkan oleh nilai x) menjadi nilai string (ditunjukkan oleh pengenal variable string var s). nilai x dapat berupa nilai numeric integer maupun nilai numeric real. Nilai width menunjukkan format panjang dari nilai utuh dan nilai decimals menunjukkan format dari panjang nilai dibelakang koma.

Contoh :

Var

    Nilai1, Nilai2 : integer ;

    X1, x2 : string[5] ;

Begin

    Nilai1 := 1234 ;

    Nilai2 := 567 ;

    Writeln (Nilai1 + Nilai2) ;

    Str (Nilai1 : 4, x1) ;

    Str (Nilai2 : 4, x2) ;

    Writeln (x1+ x2) ;

End.

Output program :

1801

1234 567

### 1.3.4 Prosedur Standar Val

Bentuk Umum :

Val ( s : string , v , var code ) :

Digunakan untuk mengkonversi suatu nilai string (ditunjukkan oleh nilai s) menjadi nilai numeric (ditunjukkan oleh nilai variable numeric v). S walaupun merupakan nilai string, tetapi harus berisi angka atau tanda plus / minus, bila tidak berarti salah dan letak kesalahannya ditunjukkan oleh nilai variable code. Nilai variable ini akan nol bila tidak mengandung kesalahan.



Contoh :

```
Var
    NilaiString : string[6] ;
    NilaiReal : real ;
    PosisiSalah : integer ;
Begin
    NilaiString := '123.A5' ;
    Val (NilaiString, NilaiReal, PosisiSalah) ;
    Writeln ('Nilai String =', NilaiString) ; Writeln
    ('Nilai Real =', NilaiReal) ;
    Writeln ('Posisi Salah =', PosisiSalah);
End.
```

Output program : Nilai  
string = 123.A5  
Nilai real = 1.2300000000E+02  
Posisi Salah = 5

## 1.4 Fungsi Standar Operasi String

### 1.4.1 Fungsi Standar Copy

Bentuk Umum :

Copy (s : string ; index : integer ; count : integer ) : string

Digunakan untuk menyalin sejumlah karakter (jumlah karakter yang disalin ditunjukkan oleh nilai *count*) mulai dari posisi yang ditunjukkan oleh nilai integer *index* dari nilai string yang ditunjukkan oleh *s*.

Contoh program :

```
Var
    Karakter : string[15];
Begin
    Karakter := 'ABCDEFGHIJKLMNO' ;
    Writeln (Karakter);
    Writeln (copy(Karakter, 5, 3));
    Writeln (copy(Karakter, 3, 10));
End.
```

Output program :

```
ABCDEFGHIJKLMNO
EFG
CDEFGHIJKL
```

#### 1.4.2 Fungsi Standar ConCat

Bentuk Umum :

ConCat (s1[,s2,..., sn] : string) : string ;

Mempunyai operasi yang sama dengan operator string "+", yaitu merangkai beberapa nilai string yang ditunjukkan oleh nilai string s1, s2, ..., sn.

Contoh program :

```
Const
    Nama = 'Pascal' ;
    Lain = 'dan keluarga' ;
Begin
    Writeln (Concat('Bahasa', Nama, 'untuk Anda', lain)) ;
End.
```

Output program :

Bahasa Pascal untuk Anda dan keluarga

#### 1.4.3 Fungsi Standar Pos

Bentuk Umum :

Pos (Substr : string ; s : string) : byte ;

Digunakan untuk mencari posisi letak dari suatu nilai string (ditunjukkan oleh Substr) yang ada didalam nilai string yang lain (ditunjukkan oleh s). Nilai yang dihasilkan adalah berupa nilai byte yang menunjukkan letaknya. Bila bernilai nol berarti nilai string yang dicari tidak ada.

Contoh program :

Const

```
Huruf = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ' ;  
Kar1 = 'FGHI' ;  
Kar2 = 'F' ;  
Kar3 = 'JUGA' ;  
Kar4 = 'DCB' ;  
Kar5 = 'PQRST' ;
```

Begin

```
Writeln (Huruf) ;  
Writeln ;  
Writeln (Kar1, 'ada diposisi ke', Pos(Kar1,Huruf)) ;  
Writeln (Kar2, 'ada diposisi ke', Pos(Kar2,Huruf)) ;  
Writeln (Kar3, 'ada diposisi ke', Pos(Kar3,Huruf)) ;  
Writeln (Kar4, 'ada diposisi ke', Pos(Kar4,Huruf)) ;  
Writeln (Kar5, 'ada diposisi ke', Pos(Kar5,Huruf)) ;
```

End.

Output program :

ABCDEFGHIJKLMNOPQRSTUVWXYZ

```
FGHI ada diposisi ke 6  
F ada diposisi ke 6  
JUGA ada diposisi ke 0  
DCB ada diposisi ke 0  
PQRST ada diposisi ke 16
```

#### 1.4.4 Fungsi Standar Length

Bentuk Umum :

Length (s : string) : integer ;

Digunakan untuk menghitung panjang atau jumlah karakter yang ada dinilai string yang ditunjukkan oleh s. hasil dari fungsi ini adalah nilai numeric integer positif.

Contoh program :

```
Var
    Nama : string[30] ;
Begin
    Write ( ' Nama Anda : ' ) ;
    Readln (Nama) ;
    Writeln ( ' Panjang nama Anda adalah ', Length(Nama),' karakter');
End.
```

Output program :

```
Nama Anda : Dita
Panjang nama Anda adalah 4 karakter
```

## 1.5 Tampilan Terformat

### 1.5.1 Parameter CH : N

Digunakan untuk membentuk format tampilan karakter selebar n karakter dengan blank dimuka sebanyak n-1

Contoh program :

```
Var
    Ket1, Ket2 : char ;
Begin
    Ket1 := 'A' ;
    Ket2 := 'B' ;
    Writeln (Ket1 : 5, Ket2 : 3);
End.
```

Output program :

```
bbbbAbbB
```

\* ket : b = blank

### 1.5.2 Parameter S : N

Digunakan untuk membentuk format tampilan string dengan lebar n karakter.

Contoh program :

```
Var
    Kata : string[6] ;
Begin
    Kata := 'Pascal' ;
    Writeln ('Bahasa :', Kata : 10);
End.
```

Output program :

Bahasa : bbbbPascal

### 1.5.3 Parameter I : N

Digunakan untuk membentuk format tampilan integer dengan lebar n digit.  
(sama dengan format karakter)

### 1.5.4 Parameter R : N

Digunakan untuk membentuk format tampilan nilai numeric real dengan lebar n digit rata sebelah kanan.

Contoh program :

```
Var
    Nilai : real ;
Begin
    Nilai := 123.45 ;
    Writeln ('Nilai real =', Nilai : 12);
End.
```

Output program :

Nilai real = 1.23450E+02

### 1.5.5 Parameter R : N : M

Digunakan untuk membentuk format tampilan nilai numeric real dengan lebar n digit rata sebelah kanan, dengan m digit angka dibelakang koma tidak dalam bentuk eksponensial.

Contoh program :

```

Var
    Nilai : real ;
Begin
    Nilai := 123.45 ;
    Writeln ('Nilai real =', Nilai : 12 : 3 );
End.

```

Output program :

Nilai real = bbbbb123.450

## 1.6 Pengaturan Letak Layar Menggunakan Unit CRT

### 1.6.1 Prosedur CLRSCR

Digunakan untuk membersihkan layar dari tampilan-tampilan sebelumnya dan meletakkan cursor di posisi ujung kiri atas dari layar.

### 1.6.2 Prosedur GOTOXY

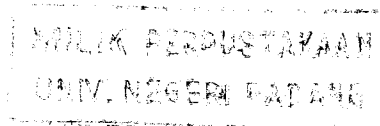
Digunakan untuk meletakkan cursor diposisi layer yang ditunjukkan oleh nilai X dan Y. Posisi horizontal / kolom ( X ) mempunyai lebar sampai dengan 80 kolom, dan posisi vertical / baris ( Y ) mempunyai baris sampai 25 baris.

Contoh Program :

```

Uses Crt ;
Begin
    Clrscr ;
    Goto (10, 5) ;
    Writeln ('Belajar') ;
End.

```



Keterangan program :

Kata "Belajar" akan tercetak pada posisi kolom 10 dan baris 5.

### 1.6.3 Prosedur CLREOL

Digunakan untuk menghapus semua karakter dalam satu baris disebelah kanan posisi cursor tanpa merubah posisi dari cursor.

### 1.6.4 Prosedur DELLINE

Digunakan untuk menghapus sebuah baris diposisi cursor dan menggeser naik tampilan-tampilan baris dibawahnya.

### 1.6.5 Prosedur INSLINE

Digunakan untuk menyisipkan sebuah baris pada posisi cursor dan menggeser kebawah tampilan baris dibawahnya.

### 1.6.6 Prosedur DELAY

Digunakan untuk menghentikan sejenak proses program selama nilai argument tempo. Dalam ukuran waktu milli second (1/1000 detik ), dan dalam bentuk numeric integer.

Contoh :

Delay (5000); artinya : proses berhenti sejenak selama 5 detik

## BAB IV

# STRUKTUR PERULANGAN

### 1.1 Perulangan FOR

Digunakan untuk mengulang statemen berulang kali sejumlah yang ditentukan.

#### 1.1.1 Perulangan Positif

Dengan penghitung / counter dari kecil ke besar (penambahan positif).

Bentuk umum :

**FOR** variable control : = nilai awal **TO** nilai akhir **DO** statemen

Contoh program (1) :

<pre> Var I : integer ; Begin   For I : = 1 to 5 do     Write (i); Writeln ('Pascal');   End.         </pre>	<p>Output program (1) :</p> <p>12345Pascal</p>
<pre> Var I : integer ; Begin   For I : = 1 to 5 do     Begin       Write ( I ) ;       Writeln ('Pascal');     End;   End.         </pre>	<p>Output program (2) :</p> <p>1Pascal 2Pascal 3Pascal 4Pascal 5Pascal</p>

#### 1.1.2 Perulangan Negatif

Dengan penghitung / counter dari besar ke kecil (pertambahannya negatif).

Bentuk umum :

**FOR** variable control : = nilai awal **DOWN TO** nilai akhir **DO** statemen



Contoh program :

<pre> Var I : integer ; Begin For I := 5 down to 1 do   Begin     Write ( I ) ;     Writeln ('Pascal');   End; End.                 </pre>	<p>Output program :</p> <pre> 5Pascal 4Pascal 3Pascal 2Pascal 1Pascal                 </pre>
--	--

### 1.1.3 Perulangan Tersarang

Perulangan yang berada didalam perulangan yang lainnya. Perulangan yang lebih dalam akan diproses lebih dulu sampai habis, kemudian perulangan yang lebih luar baru akan bertambah mengerjakan perulangan yang lebih dalam lagi mulai dari nilai awalnya dan seterusnya.

Contoh program :

<pre> Var I, J : integer; Begin For I := 1 to 5 do   Begin     For J := 1 to 3 do       Write ( I : 8, J : 3);       Writeln ;     End;   End; End.                 </pre>
--

### 1.2 Perulangan WHILE DO

Bentuk umum :

**WHILE** ungkapan logika **DO** statemen

Digunakan untuk melakukan proses perulangan suatu statemen terus menerus selama kondisi ungkapan logika pada while masih bernilai logika benar.

<pre> Var i: integer; Begin; I := 0; While I &lt; 5 do   Begin     Writeln (I); I := I + 1 ;   End; End.                 </pre>	<p>Output program : 0</p> <pre> 1 2 3 4                 </pre>
---	--

### 1.3 Perulangan REPEAT UNTIL

Digunakan untuk mengulang statemen sampai kondisi yang diseleksi di *Until* tidak terpenuhi. Bentuk umum :

**REPEAT** statemen **UNTIL** ungkapan

Contoh program :

<pre> Var i: integer; Begin; I := 0;   Repeat     I := I + 1 ;     Writeln (I) ;   Until I = 5; End.                 </pre>	<p>Output program :</p> <pre> 1 2 3 4                 </pre>
---	--

Perbedaan antara struktur " repeat until " dengan " while do " adalah :

1. Paling sedikit statemen-statemen dalam repeat until diproses
2. sekali, karena seleksi kondisi ada pada statemen until yang terletak dibawah.
3. Pada while do paling sedikit dikerjakan nol kali, karena seleksi kondisi ada pada statemen while yang terletak diatas, sehingga apabila kondisi tidak terpenuhi maka tidak akan masuk ke dalam lingkungan perulangannya.
4. Pada repeat until dapat tidak menggunakan blok statemen ( BEGIN dan END ) untuk menunjukan batas perulangannya, karena batas perulangannya sudah ditunjukkan oleh repeat sampai dengan until.

# BAB V

## SELEKSI KONDISI DAN GOTO

### 1.1 Statemen IF

#### 1.1.1 IF THEN

Bentuk umum :

IF ungkapan THEN statemen

Contoh program :

<pre> Var   Nilai : real ;   Ket : string[11] ; Begin   Ket := 'Tidak Lulus' ;   Write ('Nilai yang didapat ?') ;   Readln (Nilai) ;   If Nilai &gt; 60 then ket := 'Lulus' ;   Writeln (Ket); End.                 </pre>	<p>Output program :</p> <p>Nilai yang didapat ? 70 Lulus</p>
--	--

#### 1.1.2 IF THEN ELSE

Bentuk umum :

```

IF kondisi THEN Statemen1
ELSE
Statemen2 ;
                
```

Statemen1 /dapat berupa blok statemen akan diproses jika kondisi yang diseleksi benar (terpenuhi), dan statemen2 akan diproses jika kondisi yang diseleksi tidak terpenuhi.

Contoh program :

<pre> Var Nilai : real ;  Begin   Write ('Nilai yang didapat ?') ;   Readln (Nilai) ;   If Nilai &gt; 60 then   Writeln ('Lulus')   Else   Writeln ('Tidak Lulus'); End.</pre>	<p>Output program :</p> <p>Nilai yang didapat ? 55 Tidak Lulus</p>
--	--

1.1.3 IF Tersarang

Bentuk umum :

```

IF kondisi1 THEN
Begin
• IF kondisi2 THEN Statemen1
  ELSE Statemen2
End ;
```

Bisa juga bentuknya :

```

IF kondisi1 THEN
  IF kondisi2 THEN Statemen1
  Else Statemen2
  Else Statemen3;
```

**1.2 Statemen CASE**

1.2.1 CASE OF

Bentuk umum :

```

CASE ungkapan OF
  Daftar case label1 : statemen1 ;
  Daftar case label2 : statemen2 ;
  Daftar case label3 : statemen3 ;
  ...
  ...
  ...
```

```

Daftar case label n : statemen n
;
END ;

```

Contoh program :

```

Var
Nilai : char ;

Begin
Write ('Nilai huruf yang didapat ?'); readln (Nilai);
Case Nilai of
'A' : writeln ('Sangat baik') ;
'B' : writeln ('baik') ;
'C' : writeln ('Cukup') ;
'D' : writeln ('Kurang') ;
'E' : writeln ('Gagal') ;
End;
End.

```

### 1.2.2 CASE OF ELSE

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

Contoh program :

```

Uses Crt ;
Var
Pilihan : byte ;
R, L, T, Luas : real ;
Begin

Clrscr ;
GotoXY (10,2) ; writeln ('Pilihan') ;
GotoXY (10,4) ; writeln ('1. Menghitung Luas Lingkaran') ;
GotoXY (10,6) ; writeln ('2. Menghitung Luas Segitiga') ;
GotoXY (10,8) ; writeln ('3. Menghitung Luas Bujur Sangkar') ; GotoXY
(10,20) ; writeln ('Pilih nomer[1-3] ='); readln ( Pilihan ) ; Clrscr;

```

## Case Pilihan of

```
1 : Begin
  write ('Jari lingkaran ='); readln ( R ); Luas := Pi * R * R ;
  Writeln ;
  Writeln ('Luas lingkaran =', Luas : 9 : 2 ) ;
  End;
2 : Begin
  write ('Panjang sisi alas ='); readln ( L ); write ('Tinggi
  segitiga ='); readln ( T ); Luas := 0.5 * L * T ;
  Writeln ;
  Writeln ('Luas segitiga =', Luas : 9 : 2 ) ;
  End;
3 : Begin
  write ('Panjang bujur sangkar ='); readln ( T ); write ('Lebar
  bujur sangkar ='); readln ( L ); Luas := T * L ;
  Writeln ;
  Writeln ('Luas bujur sangkar =', Luas : 9 : 2 ) ;
  End;
Else
  Begin
  Writeln ('Pilihannya hanya 1, 2, dan 3 !!!');
  Writeln ('silahkan ulangi lagi');
  End;
End;
End.
```

## BAB VI

# PROCEDURE

Prosedur adalah suatu program terpisah dalam blok sendiri yang berfungsi sebagai subprogram (program bagian). Diawali dengan kata cadangan "Procedure" didalam bagian deklarasi prosedur. Prosedur banyak digunakan pada program yang terstruktur karena :

- merupakan penerapan konsep program modular, yaitu memecah-mecah program yang rumit menjadi program-program bagian yang lebih sederhana dalam bentuk prosedur-prosedur.
- Untuk hal-hal yang sering dilakukan berulang-ulang, cukup dituliskan sekali saja dalam prosedur dan dapat dipanggil atau dipergunakan sewaktu-waktu bila diperlukan.

Bentuk Umum :

```
PROGRAM judul_program ;  
  
PROCEDURE judul_prosedur ;  
    Begin  
        Statement prosedur ;  
        ...  
        ...  
        ...  
    End ;  
Begin  
    Statement program utama ;  
    ...  
    ...  
    ...  
end.
```

### 1.1 Parameter Dalam Prosedur

Nilai didalam modul program Pascal sifatnya adalah local, artinya hanya dapat digunakan pada modul atau unit program yang bersangkutan saja, tidak dapat digunakan pada modul atau unit program yang lainnya.

---

Contoh program :

```
Procedure Tanya_hitung ;
Var
    X, Y : real ;
Begin
    Write ('Nilai X =') ;
    Readln (X) ;
    Y := X * X ;
End ;
Begin
    Tanya_hitung ;
    Writeln ('Nilai Y =', Y : 6 : 2 ) ;
End.
```

Keterangan contoh program :

Bila program dikompilasi, akan di deteksi kesalahan oleh compiler, karena variable Y yang sifatnya local di procedure Tanya\_hitung digunakan pada program utama.

Agar nilai-nilai variable dapat digunakan di modul lainnya yang membutuhkan, maka dapat dilakukan dengan beberapa cara, yaitu :

- Dibuat bersifat global.

Harus di deklarasikan diatas modul yang akan menggunakannya.

Contoh :

```
Procedure kesatu ;
Begin
    ...
End ;
Var
    A, B : word ;
Procedure kedua ;
Begin
    ...
End ;
Procedure ketiga ;
Begin
    ...
End ;
Begin { program utama }
    ...
End.
```



Keterangan contoh :

Variabel A dan B dapat digunakan untuk modul Procedure kedua dan ketiga serta modul/program utama, tetapi tidak bersifat global untuk procedure kesatu.

- Dikirimkan sebagai parameter ke modul yang membutuhkannya. Parameter yang dikirimkan dari modul utama ke modul prosedur disebut dengan parameter nyata (actual parameter) dan parameter yang ada dan dituliskan pada judul prosedur disebut dengan parameter formal (formal parameter). Proses pengiriman data lewat parameter nyata ke parameter formal disebut dengan parameter passing. Parameter nyata dan parameter formal harus dengan tipe yang sama. Didalam pascal, parameter dapat dikirimkan secara nilai (by value) atau secara acuan (by reference).

#### 1.1.1 Pengiriman Parameter Secara Nilai

Bila parameter dikirimkan secara nilai, parameter formal di prosedur akan berisi nilai yang dikirimkan yang kemudian bersifat local di prosedur. Bila nilai parameter formal di prosedur berubah, tidak akan mempengaruhi nilai parameter nyata ( nilai parameter nyata tetap, tidak berubah). Pengiriman secara nilai ini merupakan pengiriman searah, yaitu dari parameter nyata ke parameter formal, yang tidak dikirimkan balik dari parameter formal ke parameter nyata.

Parameter – parameter yang digunakan dengan pengiriman secara nilai ini disebut dengan parameter nilai ( value parameter ).

Contoh program :

```

Procedure Hitung(A, B : integer ) ;
Var
    C : integer ;
Begin
    C := A + B ;
    Writeln ('Nilai C =', C ) ;
End ;
Var
    X, Y : integer ;
Begin
    Write ( 'Nilai X =' ) ; readln ( X ) ;
    Write ( 'Nilai Y =' ) ; readln ( Y ) ;
    Hitung ( X, Y ) ;
End.

```

---

Output program :

Nilai X = 2  
 Nilai Y = 3  
 Nilai C = 5

Penjelasan program :

- prosedur dimulai dengan deklarasi prosedur dengan judul prosedur hitung. Variable A dan B adalah parameter formal dan integer adalah tipe parameternya.
- Variable local yang hanya dipergunakan di prosedur dan tidak termasuk parameter formal (parameter nilai), harus didefinisikan sendiri, yaitu : variable C
- Hubungan antara parameter formal di prosedur dengan parameter nyata di modul utama adalah : nilai parameter nyata X dan Y di modul utama dikirimkan ke parameter formal A dan B di prosedur. Dengan demikian nilai parameter A dan B diprosedur akan berisi nilai yang sama dengan parameter X dan Y di modul utama.

### 1.1.2 Pengiriman Parameter Secara Acuan

Bila pengiriman parameter secara acuan, maka perubahan-perubahan yang terjadi pada nilai parameter formal di prosedur akan mempengaruhi nilai parameter nyata. Parameter-parameter ini disebut dengan variable parameter serta dideklarasikan di deklarasi prosedur dengan menggunakan kata cadangan Var, sebagai berikut :

```
PROCEDURE hitung (VAR A, B, C : integer) ;
```

Contoh program :

```
Procedure Hitung ( var A, B, C : integer ) ;
Begin
  C := A + B ;
End ;
Var
  X, Y, Z : integer ;
Begin
  X := 2 ; Y := 3 ;
  Hitung ( X, Y, Z ) ;
  Writeln ( ' X = ', X , ' Y = ', Y, ' Z = ', Z ) ;
End.
```

---



---

Output program :

X = 2 Y = 3 Z = 5

Penjelasan program :

- pengiriman parameter secara acuan merupakan pengiriman dua arah, bolak-balik, sehingga perubahan nilai di parameter formal akan mempengaruhi nilai parameter nyata juga. Pada contoh, nilai parameter nyata Z akan mengikuti perubahan nilai dari parameter formal C.

### 1.1.3 Pengiriman Parameter Sebagian Secara Nilai, Sebagian Acuan

Pengiriman parameter dapat dicampur sebagian secara nilai dan sebagian secara acuan dalam suatu prosedur. Parameter yang hanya dibutuhkan pada prosedur saja dapat dikirimkan secara nilai dan yang ingin dikirimkan balik dapat dilakukan secara acuan, sebagai berikut :

```
PROCEDURE Hitung ( A, B : integer ; Var C : integer ) ;
```

Contoh program :

```
Procedure Hitung ( A, B : integer ; Var C ; integer ) ;
Begin
    C := A + B ;
End ;
Var
    X, Y, Z : integer ;
Begin
    X := 2 ; Y := 3 ;
    Hitung ( X, Y, Z ) ;
    Writeln ( 'X = ', X, ' Y = ', Y, ' Z = ', Z ) ;
End.
```

Output program :

X = 2 Y = 3 Z = 5

---

## 1.2 Prosedur Memanggil Prosedur Yang Lain

Prosedur dapat memanggil prosedur yang lainnya.

Contoh program :

```
Procedure Pro1 ( X1 : integer ) ;  
Begin  
    Writeln ( 'Nilai X = ', X1, 'ada di prosedur Pro1' ) ;  
End ;
```

```
Procedure Pro2 (X2 : integer ) ;  
Begin  
    Writeln ( 'Nilai X = ', X2, 'ada diprosedur Pro2' ) ;  
    Pro1 (X2) ;  
End ;
```

```
Var  
    X : integer ;  
Begin  
    X := 5 ;  
    Pro2 ( X ) ;  
End.
```

Output program :

```
Nilai X = 5 ada diprosedur Pro2  
Nilai X = 5 ada diprosedur Pro1
```

### 1.3 Prosedur Tersarang

Adalah prosedur yang berada didalam prosedur yang lainnya.

Bentuk Umum :

```

Program
  Procedure
    Procedure
      Begin
        ...
        ...
        ...
      End ;
    Begin
      ...
      ...
      ...
    End ;
  Begin
    ...
    ...
    ...
  end .

```

### 1.4 Prosedur Memanggil Dirinya Sendiri

Merupakan prosedur yang memanggil atau menggunakan prosedur itu juga. Proses dari suatu program bagian yang memanggil dirinya sendiri dikenal dengan istilah recursion. Tidak semua bahasa menyediakan kemampuan untuk melakukan proses recursion, tetapi pascal dapat. Walaupun proses ini merupakan algoritma yang baik, tetapi membutuhkan banyak memori, karena setiap kali program bagian dipanggil oleh dirinya sendiri, sejumlah ruang memori tambahan dibutuhkannya.

Contoh program :

```

Var
  I : integer ;
Procedure Rekursi ;
Begin
  Writeln ( 'Hallo saya Pascal ' ) ;
  I := I + 1 ;
  If I < 10 Then
    Rekursi ;
End ;

```

---

```
Begin
  I := 1 ;
  Rekursi ;
End.
```

### 1.5 Acuan Forward

Digunakan untuk mendeklarasikan dimuka judul prosedur terpisah dari bloknnya. Judul prosedur yang berisi parameter terpisah ini merupakan judul yang semestinya dan di deklarasikan dengan menambahkan kata cadangan " Forward ". Blok prosedur ini letaknya terpisah dari judulnya, juga diawali dengan judul prosedur, tetapi hanya mencantumkan judul atau nama prosedurnya saja, tanpa mencantumkan parameter formalnya.

Contoh program :

```
Procedure Pro1 (Var I : integer ) ; Forward ;
Procedure Pro2 (Var I : integer ) ;
Begin
  Writeln ( ' Prosedur PRO', I ) ;
End ;

Procedure Pro1 ;
Begin
  Writeln ( ' Prosedur PRO', I ) ;
End ;

Var
  I : integer ;
Begin
  I := 1 ;
  Pro1 ( I ) ;
  I := 2 ;
  Pro2 ( I ) ;
End.
```

Output :

```
Prosedur PRO1
Prosedur PRO2
```

## 1.6 Prosedur Standar

### 1.6.1 Prosedur Standar Exit

Digunakan untuk keluar dari suatu blok. Bila diletakkan pada program bagian, prosedur standar exit akan menyebabkan proses di program bagian berhenti dan proses kembali ke blok program yang memanggilnya. Bila berada di program utama, menyebabkan proses program berhenti.

### 1.6.2 Prosedur Standar Halt

Digunakan untuk menghentikan proses program baik di program bagian maupun program utama.

### 1.6.3 Prosedur Standar Move

Bentuk umum :

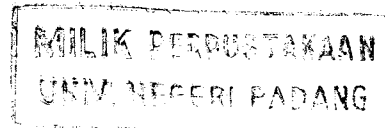
Move ( var source, dest; count : word )

Digunakan untuk menyalinkan suatu blok sebanyak count byte memori dari blok dimulai byte pertama source dan disalinkan ke byte pertama dest.

Contoh program :

```

Var
    Sumber, Tujuan : string [5] ;
    Sejumlah : word ;
Begin
    Sumber := 'ABCDE' ;
    Tujuan := 'FGHIJ' ;
    Sejumlah := 4 ;
    Move (Sumber, Tujuan, Sejumlah) ;
    Writeln (Sumber : 7, Tujuan : 7) ;
End.
```



Output program :

ABCDE ABCIJ

---

#### 1.6.4 Prosedur Standar FillChar

Digunakan untuk mengisi sejumlah byte nilai ke dalam suatu variable, sbb :

FillChar ( x ; count : word ; ch )

X adalah variable yang dapat bertipe apapun yang akan diisi dengan nilai tipe ordinal ch sebanyak count byte.



## BAB VII

# FUNGSI

Blok fungsi hampir sama dengan blok prosedur, hanya fungsi harus dideklarasikan dengan tipenya. Tipe deklarasi ini menunjukkan tipe hasil dari fungsi.

Bentuk umum :

FUNCTION identifier (daftar parameter) : type ;

Blok fungsi juga diawali dengan kata cadangan Begin dan di akhiri dengan kata cadangan End dan titik koma.

Perbedaan fungsi dengan prosedur adalah :

- Pada fungsi, nilai yang dikirimkan balik terdapat pada nama fungsinya ( kalau pada prosedur pada parameter yang dikirimkan secara acuan).
- Karena nilai balik berada di nama fungsi tersebut, maka fungsi tersebut dapat langsung digunakan untuk dicetak hasilnya. Atau nilai fungsi tersebut dapat juga langsung dipindahkan ke pengenal variable yang lainnya.
- Pada prosedur, nama prosedur tidak dapat digunakan langsung, yang dapat langsung digunakan adalah parameternya yang mengandung nilai balik.

### 1.1 Fungsi Tanpa Parameter

Fungsi yang tanpa parameter berarti nilai balik yang akan dihasilkan merupakan nilai yang sudah pasti. Karena tidak menggunakan parameter, maka hasil fungsi tersebut tidak dapat diatur dari modul yang menggunakannya, karena tidak ada parameter yang dikirimkan. Fungsi tanpa parameter jarang dipergunakan.

Contoh program :

```
Type Huruf = string [6] ;
Function Garis : Huruf ;
Begin
    Garis := '=====' ;
End ;
Begin
    Writeln (Garis) ; Writeln ('Pascal') ; Writeln (Garis) ;
End.
```

Output program :

```
=====
Pascal
=====
```

Keterangan program :

Fungsi "Garis" hanya menghasilkan nilai yang sudah pasti yaitu :  
"=====".

### 1.2 Parameter Dalam Fungsi

Parameter dalam fungsi dapat dikirimkan secara nilai atau secara acuan.

Contoh penulisan parameter dengan pengiriman secara nilai :

```
Function Hitung (A, B : integer) : integer ;
```

Contoh program :

```
Function Terbesar(X, Y : real) : real ;
Begin
    If X > Y Then
        Terbesar := X
    Else
        Terbesar := Y ;
End ;
Var
    Nilai1 , Nilai2 : real ;
Begin
    Write ('Nilai pertama =') ; readln (Nilai1) ;
    Write ('Nilai kedua = ') ; readln (Nilai2) ;
    Writeln ('Nilai terbesar adalah =', Terbesar (Nilai1, Nilai2) : 9 : 3) ;
End.
```

Output program :

```
Nilai pertama = 12.356
Nilai kedua = 55.182
Nilai terbesar adalah = 55.182
```

Penulisan judul fungsi yang menggunakan parameter dengan pengiriman secara acuan adalah dengan menambahkan kata cadangan var.

Contoh penulisan :

```
Function Hitung ( Var A, B : integer ) : integer ;
```

Pengiriman parameter secara acuan akan mengakibatkan perubahan nilai parameter di fungsi juga merubah nilai parameter di modul yang mengirimkannya. Fungsi yang menggunakan pengiriman parameter secara acuan ini mirip dengan prosedur, yaitu parameter yang dikirimkan secara acuan tersebut dapat di manfaatkan sebagai hasil balik.

### 1.3 Fungsi Pangkat

Pascal tidak menyediakan fungsi untuk perpangkatan tinggi, yang ada hanya fungsi standar Sqr, yaitu pemngkatan kuadrat saja. Bila akan melakukan perpangkatan lebih dari pangkat dua, maka harus dibuat program tersendiri.

### 1.4 Fungsi Memanggil Dirinya Sendiri

- Proses fungsi memanggil dirinya sendiri juga merupakan proses recursion.

Contoh program :

```
Function Faktorial (Nilai : integer ) : real ;
Begin
    If Nilai = 0 Then
        Faktorial := 1
    Else
        Faktorial := Nilai * Faktorial (Nilai - 1 ) ;
End ;
Var
    N : integer ;
Begin
    Write ('Berapa factorial ?') ;
    Readln (N) ;
    Writeln (N, 'faktorial =', Faktorial(N) : 9 : 0 ) ;
End.
```

Output program :

Berapa factorial ? 5  
5 faktorial = bbbbbb120

ket : b = blank

### 1.5 Fungsi Memanggil Fungsi Yang Lain

Fungsi yang di panggil letaknya harus berada di atas fungsi yang memanggilnya.

Contoh program :

```
Function Fungsi2 (Y : integer) : integer ;  
Begin  
    Fungsi2 := Y * 2 ;  
End ;  
Function Fungsi1 ( X : integer ) : integer ;  
Begin  
    Fungsi1 := Fungsi2 (X) + 5 ;  
End ;  
Begin  
    Writeln (Fungsi1(3)) ;  
End.
```

### 1.6 Acuan Forward Pada Fungsi

Fungsi dapat dideklarasikan terpisah dengan bloknnya. Deklarasi fungsi yang terpisah dengan blok fungsinya dipergunakan kata cadangan Forward dan blok fungsinya juga di deklarasikan ulang, tetapi tidak perlu menyebutkan parameternya.

### 1.7 Fungsi Tersarang

Merupakan fungsi yang letaknya berada di dalam fungsi yang lainnya.

Contoh program :

```
Function Fungsi1 ( X : integer ) : integer ;  
    Function Fungsi2 ( Y : integer ) : integer ;  
        Begin  
            Fungsi2 := Y * 2 ;  
        End ;  
    Begin  
        Fungsi1 := Fungsi2 ( X ) + 5 ;  
    End ;  
Begin  
    Writeln ( Fungsi1 ( 3 ) ) ;  
End.
```

## 1.8 Fungsi Standar

### 1.8.1 Fungsi Standar Aritmatika

Ada Abs, ArcTan, Cos, Exp, Franc, Int, Ln, Pi, Sin, Sqr dan Sqrt

Penjelasan :

- Abs ( x ) ;  
Memutlakkan suatu nilai yang ditunjukkan oleh argument x. dapat berupa tipe real atau integer.
- Exp ( x : real ) : real ;  
Untuk menghitung nilai pangkat dari bilangan e. dapat berupa real atau integer dan hasil dari fungsinya adalah real.
- Ln ( x : real ) : real ;  
Untuk menghitung nilai logaritma alam dari nilai x. tipe datanya real atau integer dan hasil fungsinya adalah real.
- Int ( x : real ) : real ;  
Untuk menghasilkan nilai integer dari x. hasil dari fungsi adalah tipe real dengan nilai yang berupa pembulatan ke bawah dari nilai x.
- Franc ( x : real ) : real ;  
Untuk mendapatkan nilai pecahan dari argument x. Argumen x dapat berbentuk real atau integer dan hasilnya adalah real.
- Sqr ( x ) ;  
Untuk menghitung nilai pangkat kuadrat dari argument x.
- Sqrt ( x : real ) : real ;  
Untuk menghitung nilai akar dari argument x. dapat real dan integer dan hasil dari fungsinya adalah real.
- Pi  
Akan menghasilkan nilai PI sebesar 3, 14.

### 1.8.2 Fungsi Standar Transfer

Digunakan untuk merubah suatu nilai ke bentuk nilai yang lain. Yang tersedia adalah fungsi standar Chr, Ord, Round dan trunc.

Penjelasan :

- Chr (x : byte) : char ;  
Untuk merubah nilai dari byte x ke bentuk karakter yang sesuai dengan kode ASCII.
- Ord (x) : longint ;  
Untuk merubah nilai x ke bentuk nilai longint yang sesuai dengan kode ASCII. Fungsi ini kebalikan dari fungsi standar Chr.
- Round (x : real) : longint ;  
Untuk membulatkan nilai dari real x ke nilai longint yang terdekat. Bila nilai pecahan sama dengan atau lebih besar dari 0,5 akan dibulatkan ke atas, sedangkan bila nilai pecahan lebih kecil dari 0,5 akan dibulatkan ke bawah.
- Trunc (x : real) : longint ;  
Untuk membulatkan nilai dari real x ke nilai longint terkecil, atau dengan kata lain membulatkan ke bawah.

### 1.8.3 Fungsi Standar Lainnya

Terdiri dari : Hi, Lo, Random, SizeOf, Swap, UpCase, ParamCount dan ParamStr.

Penjelasan :

- Hi (x) : byte ;  
Untuk mengisi low order byte dari hasil fungsi dengan high order byte dari ungkapan integer x. high order byte dari hasil fungsi akan bernilai nol. Tipe hasil dari fungsi ini adalah byte.
- Lo (x) : byte ;  
Untuk mengisi low order byte dari hasil fungsi dengan low order byte dari ungkapan integer x. high order byte dari hasil fungsi akan bernilai nol. Tipe hasil dari fungsi ini adalah byte.
- Swap (x) ;  
Untuk membalik bit-bit di low order byte menjadi high order byte dan sebaliknya dari ungkapan x. ungkapan x dapat berupa tipe integer atau word.
- Random [ (range : word ) ] ;  
Untuk menghasilkan angka random berkisar dari nilai lebih besar atau sama dengan nol dan lebih kecil dari satu. Bila range tidak disebutkan, hasil dari fungsi ini adalah real. Bila range disebutkan, hasilnya adalah word.
- SizeOf (x) : word ;  
Untuk menunjukkan besarnya byte yang digunakan oleh suatu variable x. hasilnya adalah word.

- 
- UpCase (Ch : char) : char ;  
Untuk merubah argument suatu karakter yang ditunjukkan oleh Ch menjadi bentuk karakter huruf besar ( upper case ).
  - ParamCount: word ;  
Untuk mengetahui jumlah parameter yang dikirimkan lewat prompt DOS
  - ParamStr (index) : string ;  
Untuk menerima parameter yang dikirmkan lewat prompt DOS, index adalah ungkapan dengan tipe word, dan akan menghasilkan parameter ke index yang dikirmkan tersebut.

## BAB VIII

# ARRAY

Larik / array adalah tipe terstruktur yang terdiri dari sejumlah komponen-komponen yang mempunyai tipe yang sama.

### 1.1 Deklarasi Array

Array yang akan dipergunakan harus di deklarasikan terlebih dahulu. Deklarasi dari Array diawali dengan kata cadangan Array diikuti oleh tipe index yang diletakkan diantara tanda " [ ] " diikuti lagi oleh kata cadangan of dan tipe arraynya.

Array dapat bertipe sederhana byte, word, integer, real, boolean, char, string dan tipe scalar atau subrange.

Contoh:

Var

```
X : array [ 1..100 ] of integer ;
```

Larik X dideklarasikan sebagai larik tipe integer dengan jumlah elemen maksimum 100 elemen, dimana nilai elemen larik ini harus berisi nilai integer.

Misalkan :

```
X [1] := 18 ;
```

```
X [2] := 5 ;
```

Bila nilai elemen ke-1 dari larik X akan ditampilkan, dapat dipergunakan perintah :

```
Writeln ( x[1] ) ;
```

### 1.2 Deklarasi Tipe Index Subrange Integer

Tipe index subrange integer sebagai berikut :

```
Var NilaiHuruf : array [1..5] of char ;
```

Nilai [1..5] inilah yang dimaksud dengan index subrange integer.

Index dari larik diatas dapat dideklarasikan terlebih dahulu di bagian deklarasi tipe :



```

Type
    Jangkauan = 1..5 ;    {tipe subrange integer}
Var
    NilaiHuruf : array [Jangkauan] of char ;
Begin
    ...
    ...
end.
    
```

### 1.3 Deklarasi Tipe Index Subrange Byte

Kalau index dari larik tidak sampai dengan 255, maka index dari larik ini dapat dideklarasikan dengan tipe byte.

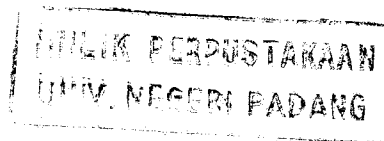
```

Var
    X : array [0..255] of real ;
    
```

Karena nilai 0 sampai 255 merupakan nilai subrange byte, maka deklarasi ini dapat ditulis :

```

Var
    X : array [byte] of real ;
    
```



### 1.4 Deklarasi Tipe Index Subrange Word

Jika jangkauan index dari 0 sampai 65525, maka index dari larik dapat dideklarasikan dengan tipe word.

```

Var
    X : array [byte] of word ;
    
```

### 1.5 Deklarasi Tipe Index Subrange Boolean

Index larik yang bertipe Boolean, hanya mempunyai maksimum 2 buah elemen saja.

```

Type
    Keterangan = string [11] ;
Var
    X : array [Boolean] of keterangan ;
    
```

### 1.6 Deklarasi Tipe Index Subrange Char

Tipe char adalah tipe subrange yang mempunyai nilai sebanyak 256 buah ( 0 – 255 ) sesuai dengan urutan kode ASCII.

Var

X : array [char] of integer ;

### 1.7 Deklarasi Tipe Index Skalar

Index dari larik dapat berupa tipe scalar atau enumerated.

Contoh program:

Var

Jumlah : array [(Jan, Peb, Mar)] of integer ;

Begin

Jumlah[Jan] := 125 ;

Jumlah{Peb} := 75 ;

Jumlah[Mar] := 18 ;

Writeln ( 'Jumlah untuk bulan maret = ', Jumlah{Mar});

End.

Output program :

Jumlah untuk bulan Maret = 18

Contoh diatas dapat ditulis :

Type

Bulan = (Jan, Peb, Mar) ;

Var

Jumlah : array [Bulan] of integer ;

### 1.8 Deklarasi konstanta Larik

Larik tidak hanya dapat berupa suatu variable yang dideklarasikan di bagian deklarasi variable, tetapi juga dapat berupa suatu konstanta yang dideklarasikan dibagian deklarasi konstanta.

Contoh program :

```

Const
    X : array [1..5] of integer = (6, 25, 375, 5, 2) ;
Var
    I : word ;
Begin
    For I := 1 to 5 do
        Writeln ( 'Nilai konstanta larik ke ', I, '=', X[I] ) ;
    End.

```

Output program :

```

Nilai Konstanta larik ke 1 = 6
Nilai Konstanta larik ke 2 = 25
Nilai Konstanta larik ke 3 = 375
Nilai Konstanta larik ke 4 = 5
Nilai Konstanta larik ke 5 = 2

```

### 1.9 String Sebagai Larik Tipe Char

String dapat dianggap sebagai suatu larik tipe char dengan index dari 0 sampai dengan panjang dari string yang di definisikan.

Contoh program :

```

Var
    I : word ;
    Nama : string [255] ;
Begin
    Write ( 'Nama Anda ?' ) ; readln ( Nama ) ;
    Writeln ;
    Writeln ( 'Nama Anda kalau dibaca terbalik adalah : ' ) ;
    For I := ord ( Nama [0] ) down to 1 do
        Write ( Nama [I] ) ;
End.

```

Output program :

Nama Anda ? Dita

Nama Anda kalau dibaca terbalik adalah :  
atid

### 1.10 Larik Dimensi Banyak

Larik dapat juga berdimensi lebih dari satu yang disebut dengan larik dimensi banyak, yang dapat berdimensi dua, dimensi tiga dan seterusnya.

Larik dimensi 2 mewakili suatu bentuk table atau matrik, yaitu index yang pertama dapat menunjukkan baris dan index kedua dapat menunjukkan kolom dari table atau metric.

Bentuk Umum :

Nama\_larik = array [tipe index1] of array [tipe index2] of tipe larik

Atau

Nama\_larik = array [tipe index1, tipe index2] of tipe larik

Contoh program :

Var

Tabel : array [1..3, 1..2] of byte ;  
I, J : byte ;

Begin

Tabel[1,1] := 5 ;  
Tabel[1,2] := 25 ;  
Tabel[2,1] := 200 ;  
Tabel[2,2] := 22 ;  
Tabel[3,1] := 75 ;  
Tabel[3,2] := 50 ;  
For I := 1 to 3 do  
  Begin  
    For J := 1 to 2 do  
      Write ( table [I,J] : 10 ) ;  
    Writeln ;  
  End ;

End.

Output program :

```
5      25
200    22
75     50
```

### 1.11 Parameter Larik

Larik dapat digunakan sebagai parameter yang dikirimkan baik secara nilai atau secara acuan ke prosedur atau ke fungsi. Prosedur yang menggunakan parameter berupa larik harus dideklarasikan didalam judul prosedur yang menyebutkan parameternya bertipe larik.

Contoh program :

Type

```
Larik = array [1..50, 1..50] of real ;
```

Var

```
I, J, K : byte ;
```

```
Procedure inverse (Var X : Larik ; N : byte ) ;
```

```
....
```

```
....
```

```
Begin
```

```
....
```

```
...
```

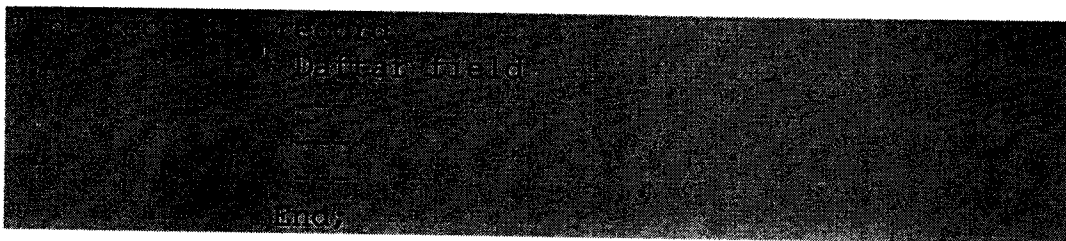
```
end.
```

## BAB IX

# RECORD

Pascal menyediakan tipe data terstruktur yang disebut record. Dengan tipe data record, dapat dikumpulkan beberapa item data yang masing-masing mempunyai tipe data berbeda-beda. Masing-masing item data disebut field. Record terdiri dari kumpulan field yang dapat berbeda tipe.

### Deklarasi Record



Contoh :

Type

```
Lgn = record ;  
Kode : integer ; Nama      : string[35], ; Alamat      :  
string[45] ; Piutang      : real ;  
End ;
```

Langganan : Lgn;

Var

### Menggunakan Tipe Data Record

Tipe-tipe komponen field dari record dapat dipergunakan dengan cara menuliskan :

```
Pengenal-record.pengenal-field[.pengenal-field]
```

Contoh :

```
Langganan Nama := ' Rhea Fatma ' ;
```

### Statemen With

Penulisan pengenal-field seperti contoh diatas dapat ditulis lebih singkat, sehingga penulisannya menjadi :

```
Nama := ' Rhea Fatma ' ;
```

Ini dapat dilakukan dengan menggunakan statemen with yang menyebutkan pengenal-recordnya, sehingga penggunaan field-field selanjutnya sudah tidak perlu menyebutkan pengenal recordnya kembali.

Contoh :

```

Type
    Langganan = record
        Nama : string;
        Alamat : string;
        Kelurahan : string;
        Luas : real;
    end;

Var
    Langganan : array[1..10] of Langganan;

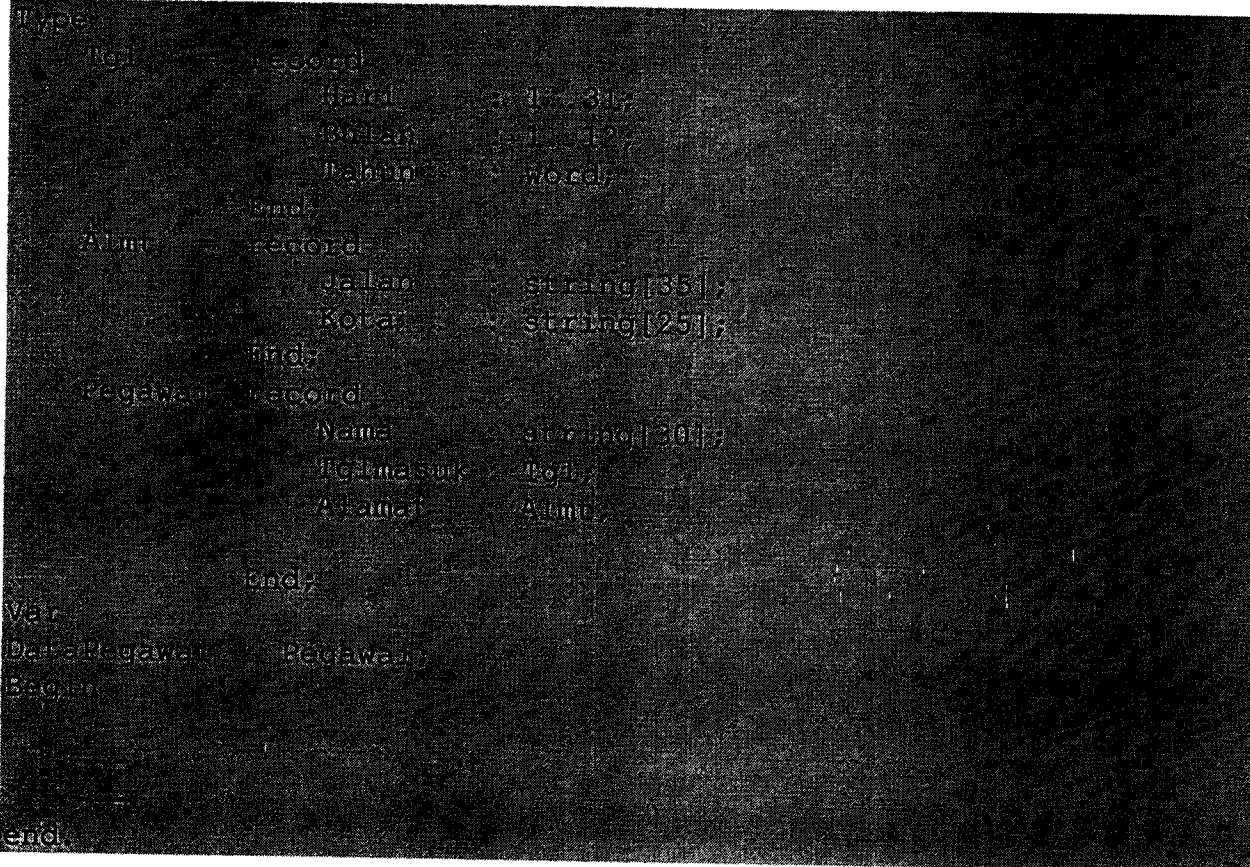
Begin
    WriteLn('Jumlah Langganan = ');
    K := 1;
    While K <= 10 do
        Langganan[K].Nama := 'Rhea Fatma';
        Langganan[K].Alamat := 'Jl. PI';
        Langganan[K].Kelurahan := 'Kec. PI';
        Langganan[K].Luas := 100;
        K := K + 1;
    end;
    WriteLn('Kelurahan Langganan = ', Kelurahan);
    WriteLn('Luas Langganan = ', Luas);
End;
    
```

### Tipe Data Record Dengan Field Tipe Record

Tipe data record dapat juga mempunyai field berupa tipe data record yang lainnya. Misalkan seperti item-item ini :

1. Nama pegawai
2. Tanggal masuk
  - a. hari
  - b. bulan
  - c. tahun
3. Alamat pegawai
  - a. Jalan
  - b. Kota

Contoh :



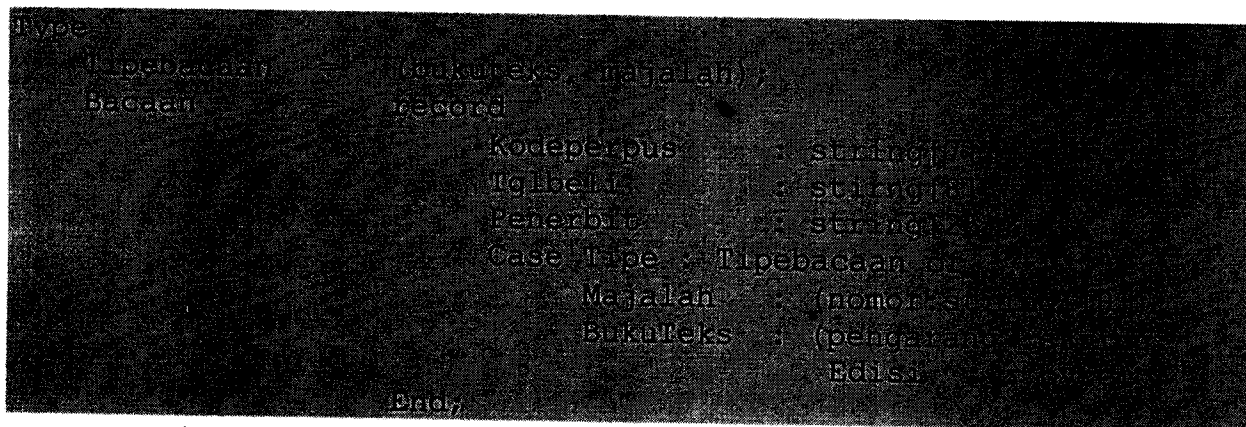
```
#include <iostream>
using namespace std;
struct Date {
    int Hari;
    int Bulan;
    int Tahun;
};
struct Address {
    string Jalan;
    string Kota;
};
struct Pegawai {
    string Nama;
    Date Tanggalmasuk;
    Address Alamat;
};
int main() {
    Pegawai P1;
    P1.Nama = "Daruji";
    P1.Tanggalmasuk.Hari = 31;
    P1.Tanggalmasuk.Bulan = 12;
    P1.Tanggalmasuk.Tahun = 2012;
    P1.Alat Jalan = "Jalan";
    P1.Alat Kota = "Kota";
    cout << endl;
    return 0;
}
Daruji Pegawai
Bantu
```



### Record Bervariasi

Dapat dibuat suatu record yang mempunyai field yang tidak pasti atau bervariasi (variant record). Dalam variant record, dapat mengandung suatu field yang bervariasi tergantung dari suatu kondisi. Bila didalam suatu variant record terdapat field yang tetap, maka field yang bervariasi letaknya harus setelah field yang tetap.

Contoh :



Penjelasan contoh :

Field yang bervariasi dalam record tergantung dari suatu kondisi Case. Tipe field tag yang dipergunakan adalah suatu tipe data scalar Tipebacaan yang mempunyai 2 nilai konstanta, yaitu : Bukuteks dan Majalah. Bila pengenal tipe bernilai Majalah, maka field variant yang akan digunakan adalah nomor. Bila pengenal tipe bernilai Bukuteks, maka field variant yang akan digunakan adalah Pengarang dan Edisi. Jadi field yang digunakan akan bervariasi tergantung dari suatu kondisi.

## Contoh-Contoh Program

```
program genap_ganjil;
uses crt;
var
  n,nilai :integer;
begin
  clrscr;
  writeln('program genap ganjil');
  write('masukkan angka anda :'); readln(n);
  nilai:=n mod 2;
  if nilai=0 then writeln(n,' adalah bilangan genap')
  else writeln(n,' adalah bilangan ganjil');
  readln;
end.
```

```
program jenis_nilai;
uses crt;
var
  n :integer;
begin
  clrscr;
  writeln('program nilai ujian');
  write('masukkan nilai ujian :'); readln(n);

  if (n>=0) and (n<30) then writeln('nilai rendah') else
  if (n>=30) and (n<60) then writeln('nilai sedang') else
  if (n>=60) and (n<=100) then writeln('nilai tinggi');

  readln;
end.
```

```
program akar_PK;
uses crt;
var
  A,B,C,D,X1,X2,Z1,Z2 :real;
begin
  clrscr;
  writeln('akar-akar persamaan kuadrat');
  writeln('masukkan nilai koefisien dari persamaan kuadrat');
  A:=0;
  while A=0 do
  begin
    write('nilai A :');readln(A);
  end;
  write('nilai B :');readln(B);
  write('nilai C :');readln(C);
  writeln('persamaan kuadratnya adalah
: ',A:3:0,'X*X', '+',B:3:0,'X', '+',C:3:0);
  D:=(B*B)-(4*A*C);

  if D>0 then
  begin
    writeln('persamaan kuadrat mempunyai akar berlainan');
    X1:=(-B+sqrt(D))/(2*A);
    X2:=(-B-sqrt(D))/(2*A);
    writeln('X1 =',X1:6:2);
    writeln('X2 =',X2:6:2);
  end
  else
```

```

if D=0 then
begin
  writeln('persamaan kuadrat mempunyai akar kembar');
  X1:=-B/(2*A);
  writeln('X1 = X2 = ',X1:6:2);

end
else

begin
  writeln('persamaan kuadrat mempunyai akar imajiner');
  Z1:=-B/(2*A);
  Z2:=(sqrt(-D))/(2*A);
  writeln('X1 =',Z1:6:2,'+',Z2:6:2,'i');
  writeln('X2 =',Z1:6:2,'-',Z2:6:2,'i');
end;

readln;
end.

```

```

program pilih;
uses crt;
var
  i:integer;
  nama : string[20];
  wn,mn,um : char;
begin
  clrscr;
  i:=0;
  wn:='y';
  mn:='y';
  um:='y';
  writeln;
  writeln('          seleksi calon pilih');
  writeln;
  writeln;

  for i:=1 to 20 do
    begin
      i:=i+1;
      write('          nama pilih : '); readln(nama);
      write('          apakah anda Warga Negara Indonesia (y/t) ? :
');
      readln(wn);

      if (wn = 'y') or (wn='Y') then
        begin
          write('          apakah anda sudah menikah (y/t) ? :
');
          readln(mn);

          if (mn = 'y') or (mn='Y') then
            Begin
              writeln;
              writeln('          PEMILIH')
            end
          else
            begin
              write('          apakah umur anda lebih dari 17
tahun' (y/t) ? : ');
              readln(um);

              if (um = 'y') or (um='Y') then

```

```

                Begin
                    writeln;
                    writeln('      PEMILIH')
                end
            else
                begin
                    writeln;
                    writeln('      BUKAN PEMILIH');
                end;
            end;
        end
    else
        begin
            writeln;
            writeln('      BUKAN PEMILIH');
        end;
        writeln;
        writeln;
    end;

    readln;

end.

```

```

program piramida_dan_silinder;
uses crt;
var
    n : integer;
    benda : string[20];
    p,l,t,r,ts,vp,vs : real;
    pilih : char;

begin
    clrscr;
    benda:='mulai';
    n:=0;

    while (benda <>'selesai') and (benda<>'SELESAI') do
        begin
            write('benda ke : ');readln(benda);
            if (benda <>'selesai') and (benda<>'SELESAI') then
                begin
                    writeln('masukkan pilihan anda');
                    writeln('A. PIRAMIDA');
                    writeln('B. SILINDER');
                    readln(pilih);
                    case pilih of
                        'A','a' : begin
                            write('panjang piramida :
                                ');readln(p);
                            write('lebar piramida :
                                ');readln(l);
                            write('tinggi piramida :
                                ');readln(t);
                            vp:=p*l*t/3;
                            writeln('volume piramida : ',vp);
                            writeln;
                        end;
                        'B','b' : begin
                            write('jari-jari alas :
                                ');readln(r);

```

```

write('tinggi : ');readln(ts);
vs:=pi*r*r*ts;
writeln('volume silinder : ',vs);
writeln;
end;

'C','c' : EXIT

end;

end;

end;

end.

```

```

program besi_aluminium;
uses crt;
var
  pilih,rho : integer;
  r,s,Vbola,Vkbs,V,mB,mK : real;

procedure tampilan(var pilih:integer);
begin
  while (pilih<>1) and (pilih<>2) do
    begin
      clrscr;
      writeln(' <<< M E N U >>>');
      writeln;
      writeln(' 1. Data Besi');
      writeln(' 2. Data Aluminium');
      writeln;

      write(' pilihan anda !!! : ');readln(pilih);
    end;
end;

procedure masukkan_data(var r,s :real);
begin
  writeln;
  write(' jari-jari bola : ');readln(r);
  write(' panjang sisi kubus : ');readln(s);
  writeln;
end;

procedure vol(r,s :real; var Vbola, Vkbs : real);
begin
  Vbola:=4*pi*r*r*r/3;
  Vkbs:=s*s*s;
end;

function massa(rho, V : real): real;
begin
  massa:=rho*V;
end;

procedure tampilkan_hasil(mB,mK :real);
begin
  writeln;
  writeln(' massa bola = ',mB:8:2);
  writeln(' massa kubus = ',mK:8:2);
  writeln;
  readln;
end;

```

BEGIN

```
clrscr;
pilih:=3;
tampilan (pilih);
```

```
if pilih=1 then
begin
    writeln;
    writeln('  Data Besi');
    rho:=7800;
    masukkan_data(r,s);
    vol(r,s,Vbola,Vkbs);
    mB:=massa(rho,Vbola);
    mK:=massa(rho,Vkbs);
end;
```

```
if pilih=2 then
begin
    writeln;
    writeln('  Data Aluminium');
    rho:=2700;
    masukkan_data(r,s);
    vol(r,s,Vbola,Vkbs);
    mB:=massa(rho,Vbola);
    mK:=massa(rho,Vkbs);
end;
```

```
tampilkan_hasil(mB,mK);
```

END.

```
programurut;
uses crt;
var
```

```
i,j,k,N,cad : integer;
nilai:array [1..60] of integer ;
```

```
procedure masukkan_data;
begin
    write (' banyak data : '); readln (N);
    writeln (' masukkan nilai antara 1 - 100 yang anda sukai ');
    for i:=1 to N do
    begin
        write (' nilai ke ',i,' = ');
        readln (nilai[i]);
    end;
end;
```

```
procedure mengurutkan_data;
begin
    for j:=1 to N do
    begin
        for k := j+1 to N do
        begin
            if (nilai[k]<nilai[j]) then
            begin
                cad := nilai[k];
                nilai[k]:=nilai[j];
                nilai[j]:=cad;
            end;
        end;
    end;
end;
```

```
end;
```

```

procedure tampilkan_hasil;
begin
    writeln;
    writeln (' data setelah diurutkan ');
    writeln;
    for i :=1 to N do
        begin
            writeln (' nilai ke ',i,' = ',nilai[i]);
        end;
    end;

BEGIN
    clrscr;
    masukkan_data;
    mengurutkan_data;
    tampilkan_hasil;
    readln;
END.

program matriks;
uses crt;
type
    angka=array [1..20,1..20] of integer ;
var
    i,j,K,B : integer;
    nilai1,nilai2,jum : angka;

    procedure baca_matriks (var nilai :angka);
    begin
        for i:=1 to B do
            begin
                for j:=1 to K do
                    begin
                        write (' nilai matrik baris ',i,' kolom ',j,' :
');
                        readln (nilai[i,j]);
                    end;
                end;
            end;
        end;

    procedure tulis_matriks (var nilai : angka);
    begin
        writeln;
        for i:=1 to B do
            begin
                writeln;
                for j:=1 to K do
                    begin
                        write (nilai[i,j]:6);
                    end;
                end;
            end;
        end;

    function jumlah_matriks (nilai1, nilai2 : angka; var jum:
angka):integer;
    begin
        for i:=1 to B do
            begin
                for j:=1 to K do
                    begin
                        jum[i,j]:=nilai1[i,j]+nilai2[i,j];
                    end;
                end;
            end;
        end;

```

```

end;

BEGIN
  clrscr;
  write (' banyak kolom matrik : '); readln (K);
  write (' banyak baris matrik : '); readln (B);
  writeln;
  writeln (' masukkan nilai dari matriks !!!');
  writeln;
  writeln(' nilai matriks pertama');
  baca_matriks(nilai1);
  writeln;
  writeln(' nilai matriks kedua');
  baca_matriks(nilai2);
  writeln;
  write(' matriks pertama adalah  :');
  tulis_matriks (nilai1);
  writeln;
  writeln;
  write(' matriks kedua adalah  :');
  tulis_matriks (nilai2);
  writeln;
  writeln;
  writeln(' jumlah matriks pertama dan kedua :');
  jumlah_matriks(nilai1,nilai2,jum);
  writeln;
  tulis_matriks (jum);
  readln;
END.

```

```

program kata;
uses crt;

var
  kal, a : string;
  jum : array [1..1000] of integer;
  jumlah,i : integer;

BEGIN
  clrscr;
  writeln ('masukkan kalimat anda ');readln (kal);
  for i := 1 to length(kal) do
    begin
      if kal[i]=' ' then jum[i]:=1 else jum[i]:=0;
    end;

  jumlah :=0;
  for i := 1 to length(kal) do
    jumlah :=jumlah + jum[i];

    writeln('banyak kata : ',jumlah+1);

  readln;
END.

```

```

program huruf_tabang;
uses crt;

var
  kal : string;
  huruf : string;
  i , j, k, l , panjang, tengah : integer;

```



```

BEGIN
  clrscr;
  write ('masukkan kalimat ');readln (kal);

  panjang := length(kal);
  tengah := 40 - panjang div 2;

  gotoxy (tengah+1,10);writeln (kal);

  for i := 1 to panjang do
    begin
      huruf := copy(kal,i,1);
      for j := 79 downto tengah+i do
        begin
          gotoXY (j,12);writeln (huruf,' ');
          delay (1000);
        end;
      end;

  for i := panjang downto 1 do
    begin
      huruf := copy(kal,i,1);
      for j := 1 to tengah+i-1 do
        begin
          gotoXY (j,14);writeln (' ',huruf);
          delay (1000);
        end;
      end;

  end;

  readln;

```

END.

```

program huruf_muncul;
uses crt;

```

```

var
  kal : string;
  huruf : string;
  i , j ,panjang , tengah : integer;

```

```

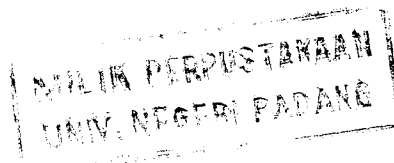
BEGIN
  clrscr;
  write ('masukkan kalimat ');readln (kal);

  panjang := length(kal);
  tengah := 40 - panjang div 2;

  for i := 1 to panjang do
    begin
      huruf := copy(kal,i,1);
      gotoXY (tengah+i,10);writeln (huruf,' ');
      delay (15000);
    end;
  readln;

```

END.



```

program recorddata;
uses crt;
type
  data = record
    nama : string[25];
    alamat : string[100];
    nim : longint;
  end;
var
  format : data;

BEGIN
  clrscr;
  write (' nama mahasiswa   : '); readln (format.nama);
  write (' nim mahasiswa     : '); readln (format.nim);
  write (' alamat mahasiswa   : '); readln (format.alamat);
  writeln;
  writeln (' data mahasiswa ');
  writeln (' nama       : ',format.nama);
  writeln (' nim        : ',format.nim);
  writeln (' alamat    : ',format.alamat);
  readln;

END.

```

```

program recorddata;
uses crt;
type
  data = record
    nama : string[25];
    kelas : string[5];
    alamat : string[100];
    nim : longint;
  end;
  Tfile = file of data;
var
  nfile : string[15];
  Vfile : Tfile;
  format : data;

BEGIN
  clrscr;
  write (' nama file : '); readln (nfile);

  assign (Vfile, nfile);
  rewrite (Vfile);

  write (' nama siswa   : '); readln (format.nama);
  write (' nim siswa     : '); readln (format.nim);
  write (' kelas         : '); readln (format.kelas);
  write (' alamat siswa  : '); readln (format.alamat);

  write(Vfile,format);
  close (Vfile);

  writeln;
  writeln (' data siswa ');
  writeln;
  writeln (' nama       : ',format.nama);
  writeln (' nim        : ',format.nim);
  writeln (' kelas      : ',format.kelas);
  writeln (' alamat    : ',format.alamat);
  readln;

END.

```

```

program recorddata;
uses crt;
type
  data = record
    nama : string[25];
    kelas : string[5];
    alamat : string[100];
    nim : longint;
  end;
  Tfile = file of data;
var
  i, n : integer;
  nfile : string[15];
  Vfile : Tfile;
  DTM : array[1..50] of data;

BEGIN
  clrscr;
  write (' nama file : '); readln (nfile);
  write (' jumlah siswa : '); readln (n);

  assign (Vfile,nfile);
  rewrite (Vfile);

  for i := 1 to n do
    begin
      writeln;
      writeln;
      with DTM[i] do
        begin
          write (' nama siswa   : '); readln (nama);
          write (' nis siswa    : '); readln (nim);
          write (' kelas       : '); readln (kelas);
          write (' alamat siswa : '); readln (alamat);
        end;
      write (Vfile,DTM[i]);
    end;

    writeln;
    writeln;
    writeln ('
=====')
;

    writeln (' no          nama          nim          kelas
alamat');
    writeln ('
=====')
;

    for i := 1 to n do
      begin
        with DTM[i] do
          begin
            writeln (i:4,nama:23,nim:8,kelas:7,'
',alamat);
          end;
        end;
      writeln ('
=====')
;

      readln;
END.

```

```

program recorddata;
uses crt;
type
  data = record
    nama : string[25];
    ipk : real;
    alamat : string[100];
    nim : string;
  end;
  Tfile = file of data;
var
  nama,alamat, nm, rmh, nim, no : string;
  ipk, nilai : real;
  i, n , k, pilih, cari, cr : integer;
  nfile : string[15];
  Vfile : Tfile;
  DTM : array[1..50] of data;
  lagi : char;

  procedure masukkan_data (i : integer; var nama, alamat, nim : string ;
var ipk : real);
  begin
    with DTM[i] do
      begin
        write ((' ',i,')', 'nama mahasiswa : '); readln
(nama);
        write (' nim : '); readln (nim);
        write (' ipk : '); readln (ipk);
        write (' alamat : '); readln (alamat);
      end;
    end;

  procedure tampilkan_hasiltotal;
  begin
    writeln ('
=====');
    writeln (' no nama nim ipk
alamat');
    writeln ('
=====');
    for i := 1 to n do
      begin
        with DTM[i] do
          begin
            writeln (i:4,nama:23,nim:8,ipk:9:2,'
,alamat);
          end;
        end;
      end;
    writeln ('
=====');
  end;

  procedure tampilkan_hasil (cr : integer);
  begin
    writeln ('
=====');
    writeln (' nama nim ipk
alamat');
    writeln ('
=====');
    with DTM[cr] do
      begin
        writeln (nama:23,nim:8,ipk:9:2,'
,alamat);
      end;
  end;

```

```

        writeln ('
=====');
    end;

BEGIN
    clrscr;
    write (' nama file : '); readln (nfile);
    write (' jumlah mahasiswa : '); readln (n);

    assign (Vfile,nfile);
    rewrite (Vfile);

    for i := 1 to n do
        begin
            writeln;
            writeln;
            masukkan_data (i, nama, alamat, nim, ipk);
            write (Vfile,DTM[i]);
        end;

    writeln (' pilih tampilan data yang anda inginkan : ');
    writeln (' 1. tampil seluruhnya');
    writeln (' 2. tampil data yang diinginkan');
    readln (pilih);

    case pilih of
    1 : begin
            clrscr;
            tampilkan_hasiltotal;
            readln;
        end;

    2 : begin
            lagi :='y';
            while (lagi ='y') or (lagi ='Y') do
                begin
                    clrscr;
                    cr :=0;
                    writeln (' pilih kategori pencarian data');
                    writeln (' 1.berdasarkan nama');
                    writeln (' 2.berdasarkan nim');
                    writeln (' 3.berdasarkan ipk');
                    writeln (' 4.berdasarkan alamat');
                    write ('pilihan anda : ');readln (cari);

                    if cari = 1 then
                        begin
                            write (' nama mahasiswa : '); readln (nm);
                            for i := 1 to n do ,
                                begin
                                    if (DTM[i].nama = nm) then cr := i;
                                end;
                            tampilkan_hasil(cr);
                        end else

                    if cari = 2 then
                        begin
                            write (' nim mahasiswa : '); readln (no);
                            for i := 1 to n do
                                begin
                                    if (DTM[i].nim = no) then cr := i;
                                end;
                            tampilkan_hasil(cr);
                        end else

```

```

        if cari = 3 then
            begin
                write (' ipk mahasiswa : '); readln (nilai);
                for i := 1 to n do
                    begin
                        if (DTM[i].ipk = nilai) then cr := i;
                    end;
                tampilkan_hasil(cr);
            end else

        if cari = 4 then
            begin
                write (' alamat mahasiswa : '); readln (rmh);
                for i := 1 to n do
                    begin
                        if (DTM[i].alamat = rmh) then cr := i;
                    end;
                tampilkan_hasil(cr);
            end;
        write (' ingin mengulang pencarian (y/t) ? : ');
        readln(lagi);
        end;
    end;
end;

```

END.

```

program gambar_oto;
uses crt, graph;
var

```

```

    gd, gm, i : integer;

```

```

begin

```

```

    gd:=Vga;
    gm:=VgaHi;
    initgraph (gd,gm,'c:\tp\bgi');
    setbkcolor (green);
    setcolor(blue);

```

```

    for i :=0 to 5 do
        begin

```

```

            rectangle (getmaxx div 6-i, getmaxy div 3-i, 3*getmaxx div 5+i,
2*getmaxy div 3+i);
            circle (getmaxx div 6+50,2*getmaxy div 3, 30+i);
            circle (getmaxx div 6+50,2*getmaxy div 3, 30-5*i);
            setfillstyle (2,red);
            floodfill (getmaxx div 6+10, getmaxy div 3+10, blue);
            circle (getmaxx div 6+350,2*getmaxy div 3, 30+i);
            circle (getmaxx div 6+350,2*getmaxy div 3, 30-5*i);
            moveto (3*getmaxx div 5+i, 2*getmaxy div 3+i);
            lineto (3*getmaxx div 5+150, 2*getmaxy div 3+i);
            lineto (3*getmaxx div 5+150+i, getmaxy div 2+10);
            lineto (3*getmaxx div 5+100+i, getmaxy div 2-50);
            lineto (3*getmaxx div 5, getmaxy div 2-50+i);
            moveto (3*getmaxx div 5+140, getmaxy div 2+10+i);
            lineto (3*getmaxx div 5+30, getmaxy div 2+10+i);
            lineto (3*getmaxx div 5+30-i, getmaxy div 2-40);
            lineto (3*getmaxx div 5+90, getmaxy div 2-40-i);
            lineto (3*getmaxx div 5+140-i, getmaxy div 2+10);
            setfillstyle (9,yellow);
            floodfill (3*getmaxx div 5+10, getmaxy div 2-10, blue);
        end;
    end;

```

```

repeat until keypressed;
closegraph;

end.

program gambar_rumah;
uses crt, graph;
var

gd, gm, i, j : integer;

begin
gd:=Vga;
gm:=VgaHi;
initgraph (gd,gm,'c:\tp\bgi');

for i :=0 to 5 do
begin

setcolor(yellow);
rectangle (getmaxx div 5-i, 3*getmaxy div 7-i, 4*getmaxx div 5+i, 6*getmaxy div 7+i);
moveto (getmaxx div 5-50-i, 3*getmaxy div 7-i);
lineto (4*getmaxx div 5+50+i, 3*getmaxy div 7-i);

rectangle (getmaxx div 5+10-i, 5*getmaxy div 7-20-i, 2*getmaxx div 5+i-50, 6*getmaxy
div 7+i);

rectangle (2*getmaxx div 5-i-20, 4*getmaxy div 7-i+50, 3*getmaxx div 5+i-70, 6*getmaxy
div 7-50+i);
setfillstyle (8,brown);
floodfill (2*getmaxx div 5-i-10, 4*getmaxy div 7-i+60, yellow);

rectangle (2*getmaxx div 5-i-20, 3*getmaxy div 7-i+20, 3*getmaxx div 5+i-70, 5*getmaxy
div 7-80+i);
floodfill (2*getmaxx div 5-i-10, 3*getmaxy div 7-i+30, yellow);

rectangle (3*getmaxx div 5-i+10, 3*getmaxy div 7-i+20, 4*getmaxx div 5+i-40, 5*getmaxy
div 7-80+i);
floodfill (3*getmaxx div 5-i+20, 3*getmaxy div 7-i+30, yellow);

rectangle (3*getmaxx div 5-i+10, 4*getmaxy div 7-i+50, 4*getmaxx div 5+i-40, 6*getmaxy
div 7-50+i);
floodfill (3*getmaxx div 5-i+20, 4*getmaxy div 7-i+60, yellow);

setfillstyle (9,blue);
floodfill (getmaxx div 5+30, 3*getmaxy div 7+30, yellow);

circle (getmaxx div 5+25,5*getmaxy div 7+20,10-i);

setfillstyle (1,brown);
floodfill (getmaxx div 5+20-i, 5*getmaxy div 7+10-i, yellow);

end;

for j :=0 to 50 do
begin
setcolor(white);
circle (70, 70, 50-2*j);
setcolor (yellow);
line (getmaxx div 5-50+2*j, 3*getmaxy div 7-2*j,4*getmaxx div 5+50-2*j, 3*getmaxy div
7-2*j);
setcolor (green);
line (getmaxx div 5-5-2*j, 6*getmaxy div 7+5+2*j, 4*getmaxx div 5+5+2*j, 6*getmaxy div
7+5+2*j);
end;

repeat until keypressed;
closegraph;

end.

```

```

program garis_animasi;
uses crt, graph;
var

```

```

    gd, gm, i : integer;

```

```

begin

```

```

    gd:=Vga;
    gm:=VgaHi;
    initgraph (gd,gm,'c:\tp\bgi');
    setbkcolor (green);
    i :=0;

```

```

    repeat
    setcolor (red);
    line (i,0, i, getmaxy);
    delay (3);
    setcolor (blue);
    line (getmaxx-i,0, getmaxx-i, getmaxy);
    delay (3);
    setcolor (green);
    line (getmaxx-i,0, getmaxx-i, getmaxy);
    setcolor (green);
    line (i,0, i, getmaxy);
    i :=i+1;
    until i = getmaxx;
    repeat until keypressed;
    closegraph;

```

```

end.

```

```

program lingkaran_gelombang;
uses crt, graph;
var

```

```

    gd, gm : integer;
    x, y, t, F, ef, A, B : real;

```

```

begin

```

```

    clrscr;
    write (' amplitudo gel 1 : '); readln (A);
    write (' amplitudo gel 2 : '); readln (B);
    write (' frekuensi gel 1 : '); readln (F);
    write (' frekuensi gel 2 : '); readln (ef);

```

```

    gd :=vga;
    gm :=vgahi;
    initgraph (gd,gm,'C:\tp\BGI');
    setbkcolor(green);

```

```

    repeat

```

```

        x:=A*sin(2*pi*F*t/180);
        y:=B*sin(2*pi*ef*t/180);
        setcolor (red);
        circle (round(x)+getmaxx div 2, round(y)+getmaxy div 2, 10);
        delay(5);
        setcolor (green);
        circle (round(x)+getmaxx div 2, round(y)+getmaxy div 2, 10);

```

```

        t:=t + 0.1;

```



```

    until keypressed;
    closegraph;

end.

program lingkaran_animasi;
uses crt, graph;
var
    gd, gm, i : integer;

begin
    .gd:=Vga;
    gm:=VgaHi;
    initgraph (gd,gm,'c:\tp\bgi');
    setbkcolor (green);
    i :=0;

    repeat
        setcolor (red);
        circle (i+10, getmaxy div 2, 10);
        delay (2);
        setcolor (green);
        circle (i+10, getmaxy div 2, 10);
        i :=i+1;
    until i = getmaxx;
    repeat until keypressed;
    closegraph;

```

end.

```

program peluru_animasi;
uses crt, graph;
var
    theta, x , y, r : real;
    gd, gm : integer;

```

```

begin
    clrscr;
    write('    jari-jari :');readln(r);
    theta:=0;
    gd:=Vga;
    gm:=VgaHi;
    initgraph (gd,gm,'c:\tp\bgi');
    setbkcolor (green);

    repeat

        x:= r *cos (theta*pi/180);
        y:= r *sin (theta*pi/180);
        setcolor (blue);
        circle (round(x)+getmaxx div 2,round(y)+getmaxy div 2,5);
        delay(2);
        setcolor (green);
        circle (round(x)+getmaxx div 2,round(y)+getmaxy div 2,5);
        theta := theta +1;

    until keypressed;
    closegraph;
end.

```

```

program peluru_animasi;
uses crt, graph;
var
  Vo,theta,t,tt,x,yy,y,Vx,Vy,V,P,g : real;
  gd, gm : integer;

begin
  clrscr;
  writeln;
  writeln('      gerak peluru');
  writeln;
  writeln('      masukkan nilai berikut :');
  writeln;
  write('      kecepatan awal :');readln(Vo);
  write('      sudut elevasi :');readln(theta);
  writeln;
  theta:=(theta*pi)/180;
  g:=10;
  tt:=(2*Vo*sin(theta))/g;
  t:=0;

  gd:=Vga;
  gm:=VgaHi;
  initgraph (gd,gm,'c:\tp\bgi');
  setbkcolor (green);
  line (50,4*getmaxy div 5, getmaxx-50, 4*getmaxy div 5);
  line (50,10, 50, 4*getmaxy div 5);

  repeat

    x:=Vo*cos(theta)*t;
    y:=(Vo*sin(theta)*t)-((g*t*t)/2);
    setcolor (blue);
    circle (round(x)+50,4*getmaxy div 5-round(y),5);
    delay(10);
    setcolor (green);
    circle (round(x)+50,4*getmaxy div 5-round(y),5);
    putpixel (round(x)+50,4*getmaxy div 5-round(y),blue);
    t:=t+0.05;

  until t>=tt;
  setcolor (blue);
  circle (round(x)+50,4*getmaxy div 5-round(y),5);
  delay(10);
  repeat until keypressed;
  closegraph;
end.

```

```

program lingkaran_animasi;
uses crt, graph;
var
  gd, gm, i, dx : integer;

begin
  gd:=Vga;
  gm:=VgaHi;
  initgraph (gd,gm,'c:\tp\bgi');
  setbkcolor (green);
  i :=0;
  dx:=1;

```

```

repeat
setcolor (red);
circle (i, getmaxy div 2, 10);
delay (2);
setcolor (green);
circle (i, getmaxy div 2, 10);
i :=i+dx;

if i=getmaxx-10 then dx:=-dx;
if i=10 then dx:=1;

until keypressed;
closegraph;

end.

```

```

program lingkaran_animasi;
uses crt, graph;
var

gd, gm : integer;
i, j, dx, dy : real;

begin
gd:=Vga;
gm:=VgaHi;
initgraph (gd, gm, 'c:\tp\bgi');
setbkcolor (green);
i :=0;
dx:=0.5;
dy:=0.5;

repeat
setcolor (red);
circle (round(i), round(j), 10);
delay (2);
setcolor (green);
circle (round(i), round(j), 10);
i :=i+dx;
j :=j+dy;

if i=getmaxx-10 then dx:=-dx;
if i=10 then dx:=0.5;
if j=getmaxy-10 then dy:=-dy;
if j=10 then dy:=0.5;

until keypressed;
closegraph;

end.

```

