

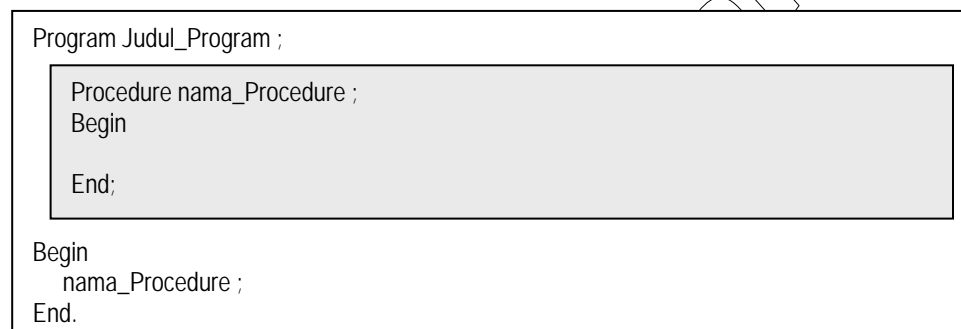
BAB IX PROSEDUR

Prosedur adalah suatu program terpisah dalam blok tersendiri yang berfungsi sebagai subprogram(program bagian). Prosedure diawali dengan kata cadangan **Procedure** di dalam bagian deklarasi prosedur. Prosedur di panggil dan digunakan di dalam blok program yang lainnya dengan menyebutkan judul prosedurnya.

Prosedur banyak digunakan pada program yang terstruktur, karena :

1. Merupakan penerapan kosenp program modular, yaitu memecah-mecah program yang rumit menjadi program-program bagian yang lebih sederhana dalam bentuk prosedur-prosedur
2. 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 DAN CARA PEMANGGILANNYA



PARAMETER DALAM PROSEDUR

Nilai di dalam suatu modul program sifatnya adalah lokal, artinya hanya dapat digunakan pada modul atau unit program yang bersangkutan saja, tidak dapat digunakan pada modul atau unit program yang lainnya.

Example 1 :

```

Procedure Tanya_Hitung ;
Var
  x,y : real ;
Begin
  Write('Nilai X ?') ;
  Readln(x) ;
  y := x * x
  Writeln('Nilai Y = ', Y:6:2 ) ;
End; {Akhir dari prosedur Tanya_Hitung}
  
```

```

Begin
  Tanya_Hitung ;
End.
  
```

Contoh 9.1 Variabel Lokal dalam Prosedur

Pada contoh diatas, variabel x dan y sifatnya adalah lokal untuk prosedur Tanya_Hitung, yang artinya hanya dapat digunakan pada modul tersebut saja . Pada modul yang lain, variabel tersebut tidak dapat digunakan. Bila digunakan maka salah.

Example 2 :**Procedure Tanya_Hitung ;****Var**

x,y : real ;

Begin

Write('Nilai X ?') ;

Readln(x) ;

y := x * x

End; {Akhir dari prosedur Tanya_Hitung}**Begin**

Tanya_Hitung ;

Writeln('Nilai Y = ', Y:6:2) ;

End.Contoh 9.2 Contoh penggunaan variabel lokal yang salah

Bila program ini akan dikompilasi, akan dideteksi kesalahan oleh kompiler, karena variabel Y yang sifatnya lokal di prosedur Tanya_Hitung digunakan pada program utama. Kesalah yang ditunjukkan oleh kompilasi akan berbentuk pemberitahuan sebagai berikut :

Error 31 : Unknown IdentifierExample 3 :**Procedure CetakNama ;****Begin**

Writeln>NamaKu) ;

End;

{Main Program}

Var Namaku : String[20] ;**Begin**

Namaku := 'Mesran' ;

CetakNama ;

End.Contoh 9.3 Contoh penggunaan variabel lokal yang salah kedua.

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

PARAMETER GLOBAL

Supaya suatu nilai dapat bersifat global, maka harus dideklarasikan diatas modul yang akan menggunakannya.

Procedure KeSatu ;**Begin****End;** {Akhir dari procedure KeSatu}**Var** A, B : Integer ;**Procedure KeDua ;****Begin****End;** {Akhir dari procedure KeDua}**Procedure KeTiga ;****Begin****End;** {Akhir dari procedure KeTiga}**Begin****End.** {Akhir dari Program Utama}

Pada contoh diatas, variabel A & B bersifat global untuk prosedur KeDua, KeTiga dan Modul Utama (Program Utama), tetapi tidak untuk prosedur kesatu. Sehingga prosedur kesatu tidak bisa menggunakan variabel tersebut.

Kenapa ?

Bandingkan dengan contoh berikut ini.

```

Var      A, B : Integer ;

Procedure KeSatu ;
Begin
End; {Akhir dari procedure KeSatu}

Procedure KeDua ;
Begin
End; {Akhir dari procedure KeDua}

Procedure KeTiga ;
Begin
End; {Akhir dari procedure KeTiga}

Begin
End. {Akhir dari Program Utama}

```

PARAMETER YANG DIKIRIMKAN

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 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. Pada Pascal, parameter dapat dikirimkan secara nilai (by value) atau secara acuan (by reference).

a. PARAMETER SECARA NILAI (by value)

Pengiriman secara nilai ini merupakan pengiriman searah, yaitu dari parameter formal ke parameter nyata. Parameter-parameter yang digunakan dengan pengiriman secara nilai ini disebut dengan parameter nilai (value parameter).

Example 4 :

```

Procedure Hitung(A, B : Integer ) ;
Var
  c : Integer ;
begin
  c := A + B ;
  Writeln(#10, ' Nilai C = ', c ) ;
End;

```

```

Var      x, y : Integer ;
Begin
  Write('Nilai X ?') ; Readln(x) ;
  Write('Nilai Y ?') ; Readln(y) ;
  Hitung (x, y) ;
End.

```

Contoh 9.4 pengiriman parameter secara nilai (by value)

Example 5 :

```

Procedure Hitung(A, B, C : Integer ) ;
begin
  c := A + B ;
  Writeln
  Writeln('A =', A , 'B =', B , 'C =', C )
End;

```

```

Var    x, y, z : Integer ;
Begin
  X := 2 ; Y := 3 ; Z := 0 ;
  Hitung(X, Y, Z) ;
  Writeln('X =', X , 'Y =', Y , 'Z =', Z ) ;
End.

```

Contoh 9.5 Pengiriman parameter secara nilai (by value)

Perhatian hasil dari program pada saat di jalankan. Akan tampak jelas bahwa variabel a, b, c tidak akan merubah nilai dari variabel x, y, z.

b. PARAMETER SECARA ACUAN (by reference)

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

```

PROCEDURE Hitung(Var A, B, C : Integer)

```

Menunjukkan pengiriman parameter secara acuan

Example 6 :

```

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.

```

Contoh 9.6 Pengiriman parameter secara acuan (by reference)

Program berikut digunakan untuk menghitung besarnya nilai suatu faktorial, dengan rumus :

$$N! = 1 \times 2 \times 3 \times 4 \times \dots \times N$$

Penyelesaiannya adalah :

```

Procedure Faktorial(Var Fak, Hasil : Integer) ;
Var    I : Integer
Begin
  Hasil := 1 ;
  For I = 2 To Fak Do Hasil := Hasil * I ;
End; {Akhir dari prosedur Faktorial }

```

```

{Main Program}
Var N, F : Integer ;
Begin
    Write ('Berapa Faktorial ? ');
    Readln (N) ;
    Faktorial(N, F) ;
    Writeln; Writeln(N, ' Faktorial = ', F);
End

```

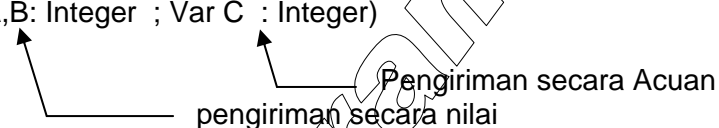
c. PARAMETER SECARA NILAI DAN ACUAN

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.

```

PROCEDURE Hitung(A,B: Integer ; Var C : Integer)

```



Example 7 :

```

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.

```

Contoh 9.7 Pengiriman parameter sebagian secara nilai dan sebagian secara acuan.

Hubungan antara pengiriman parameter tersebut akan nampak sebagai berikut .

```

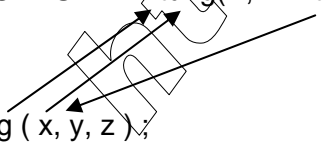
PROCEDURE Hitung(A,B: Integer ; Var C : Integer)

```

```

Hitung ( x, y, z ) ;

```



PROSEDUR MEMANGGIL PROSEDUR YANG LAIN

Di dalam suatu prosedur dapat melakukan pemanggilan terhadap suatu prosedur yang lain. Tetapi prosedur yang dipanggil harus berada diatas prosedur yang memanggil.

Example 8 :

```

Procedure Pro1(x1 : Integer) ;
Begin
    Writeln('Nilai x =', x1, ' ada di prosedur PRO1') ;
End;

Procedure Pro2(x2 : Integer) ;

```

```

Begin
    Writeln('Nilai x =', x2, ' ada di prosedur PRO2') ;
    Pro1(X2)
End;

Var
    X : Integer ;
Begin
    X := 5 ;
    Pro2(x) ;
End.

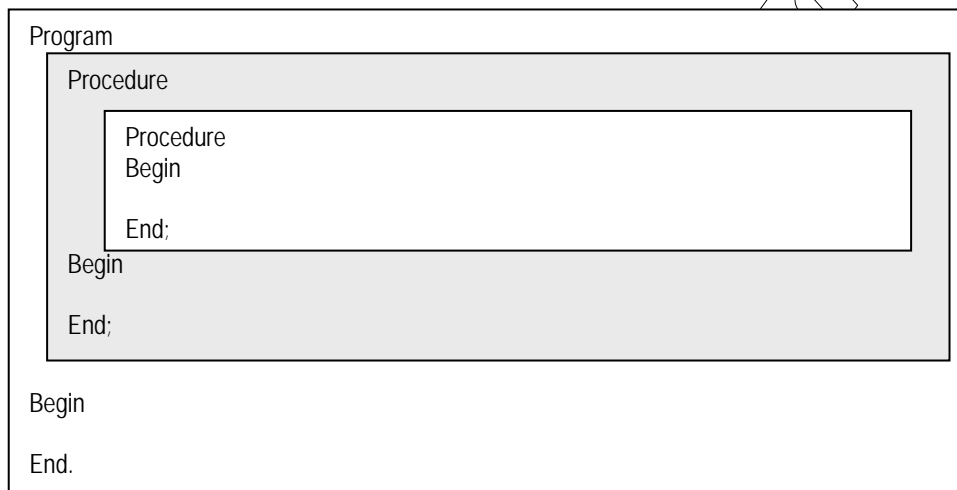
```

Contoh 9.8 Prosedur memanggil prosedur yang lain.

NESTED PROCEDURE (Prosedur tersarang)

Adalah prosedur yang berada di dalam prosedur yang lain.

Syntax



Example 9 :

```

Procedure Kesatu ;
    Procedure Kedua ;
        Begin
            Writeln('Prosedure KEDUA ini ada di dalam prosedure ke KESATU') ;
        End;
    Procedure Ketiga ;
        Begin
            Writeln('Prosedure KETIGA ini ada di dalam prosedure ke KESATU') ;
        End;
    Begin
        Writeln('Ini Prosedur KESATU');
        Writeln;
        Kedua ;
        Ketiga
    End;
Begin
    Writeln('Ini di Modul Utama')
    Writeln;
    Kesatu ;
End.

```

Contoh 9.9 Program Prosedur tersarang-Nested Procedure

PROSEDUR MEMANGGIL DIRINYA SENDIRI

Merupakan prosedur memanggil dirinya sendiri merupakan suatu 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. Walaupun proses ini merupakan algorithm yang baik, tetapi membutuhkan banyak memori karena setiap kali program bagian dipanggil oleh dirinya sendiri, sejumlah ruang memori tambahan dibutuhkan.

Example 10 :

```

Var
  I : Integer ;
  Procedure Rekursi ;
  Begin
    Writeln('Halo Saya Pascal') ;
    Inc(I);
    If I < 10 Then Rekursi ;
  End.
Begin
  I := 1 ;
  Rekursi ;
End.

```

Contoh 9.10 Program Prosedur memanggil dirinya sendiri (rekursif):

PROSEDUR STANDAR EXIT & HALT▪ **EXIT**

Prosedur standar ini 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.

Example 11 :

```

Program ContohFaktorial ;
  Procedure Faktorial(Fak : Integer ; Var Hasil : Integer) ;
  Var
    I : Integer ;
  Begin
    Hasil := 1 ;
    If Fak = 0 Then
      Exit ;
    For I := 2 To Fak DO
      Hasil := Hasil * I ;
    End;
  Var
    N, F : Integer ;
  Begin
    Write(' Berapa Faktorial ? ');
    Readln(N) ;
    F := 1 ;
    Faktorial(N, F) ;
    Writeln(N, ' Faktorial = ', F)
  End.

```

Contoh 9.11 Penggunaan Prosedur standar Exit.

▪ HALT

Prosedur standar ini digunakan untuk menghentikan proses program baik di program bagian maupun di program utama.

Example 12 :

```

Procedure TampilkanTerus ;
Var
    Tekan : Char ;
Begin
    Tekan := 'T' ;
    While (Tekan = 'T') Or (Tekan = 't') DO
        Begin
            Write('Haloo saya MESRAN  ') ;
            Writeln('Tekan <T> untuk Terus <B> untuk berhenti - - ->');
            Tekan := Readkey();
            If (Tekan = 'B') Or (Tekan = 'b') Then
                Halt
            Else
                Begin
                    Writeln('Ngawur....')
                    TampilkanTerus ;
                End;
        End;
    End ;
End;

Begin
    TampilanTerus ;
End.

```

Contoh 9.12 Penggunaan prosedur standar Halt

LATIHAN

1. Nama Program : **Latih91.Pas**

```

PROGRAM Mencari_Gaji ;
USES CRT ;
TYPE
    Pegawai = Record
        NIP : String[8] ;
        Nama : String[20] ;
        GOL : String[3] ;
    END;

VAR
    rPeg : Array[1..20] of Pegawai ;
    i, j : byte ;
    Gaji, TGaji : Longint ;
    lg : Char ;

PROCEDURE InputPEGAWAI ;
PROCEDURE FormPEGAWAI ;
BEGIN
    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] :');
END;

BEGIN
  REPEAT
    FORMPegawai ;
    WITH Rpeg[ i ] do
      Begin
        Gotoxy(27,4); Readln( NIP );
        Gotoxy(27,5); Readln( NAMA );
        Gotoxy(27,6); Readln( GOL );
        Gotoxy(33,8); Readln( Lg );
      END ;
      inc(i);
    UNTIL Upcase(Lg)='T';
  END;
  PROCEDURE TabelPegawai ;
  BEGIN
    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('_____');
  END;

  PROCEDURE HITGAJI(cGol : String; Var nGaji: Longint );
  BEGIN
    If cGol='I' then nGaji := 750000
    Else If cGol='II' then nGaji := 1000000
    Else If cGol='III' then nGaji := 1500000
    Else nGaji :=0;
  END;

  BEGIN
    i := 1 ;
    InputPEGAWAI;
    TabelPegawai ;
    FOR j:=1 TO I-1 DO
      BEGIN
        With RPeg[j] Do
          Begin
            HitGaji (Gol, Gaji) ;

            Gotoxy(5 ,6+j); Write( j:2,'    ',NIP: 8,'    ', NAMA);
            Gotoxy(44,6+j); write( Gol );
            Gotoxy(49,6+j); write( Gaji:8 );
          End
        End
      End
    End
  End

```

```

    TGaji := TGaji + Gaji ;
  End;
END;
Gotoxy(3,7+j); Write('_____');
Gotoxy(3,8+j); Write('  TOTAL GAJI          RP.', tgaji:10);
Repeat Until Keypressed ;
END.

```

2. Nama Program : **Latih92.Pas**

```

PROGRAM Mencari_Gaji ;
USES CRT ;
TYPE
  Pegawai = Record
    NIP      : String[8] ;
    Nama     : String[20] ;
    GOL      : String[3] ;
  END;
VAR
  rPeg      : Array[1..20] of Pegawai ;
  i, j      : byte ;
  Gaji, TGaji : Longint ;
  Pil,
  lg        : Char ;

PROCEDURE InputPEGAWAI ;
PROCEDURE FormPEGAWAI ;
BEGIN
  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] :');
END;

BEGIN
  REPEAT
    FORMPegawai ;
    WITH Rpeg[i] do
      Begin
        Gotoxy(27,4); Readln( NIP );
        Gotoxy(27,5); Readln( NAMA );
        Gotoxy(27,6); Readln( GOL );
        Gotoxy(33,8); Readln( Lg );
      END ;
      inc(i);
    UNTIL Upcase(lg)='T';
  END;

PROCEDURE OutputPegawai ;

```

PROCEDURE TabelPegawai;

```

BEGIN
  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('_____');
END;
```

PROCEDURE HITGAJI(cGol : String ; Var nGaji: Longint);

```

BEGIN
  If cGol='I' then nGaji := 750000
  Else If cGol='II' then nGaji := 1000000
  Else If cGol='III' then nGaji := 1500000
  Else nGaji :=0;
END;
```

Begin

```

TabelPegawai ;
FOR j:=1 TO I-1 DO
BEGIN
  With RPeg[j] Do
  Begin
    HitGaji (Gol, Gaji) ;

    Gotoxy(5 ,6+j); Write( j:2,' ',NIP: 8,' ',NAMA);
    Gotoxy(44,6+j); write( Gol );
    Gotoxy(49,6+j); write( Gaji:8 );

    TGaji := TGaji + Gaji ;
  End;
END;
Gotoxy(3,7+j); Write('_____');
Gotoxy(3,8+j); Write(' TOTAL GAJI           RP.', tgaji:10);
Repeat Until Keypressed ;
End;
```

PROCEDURE MENU ;

```

BEGIN
  Clrscr;
  Gotoxy(20,2); Write('MENU PEGAWAI');
  Gotoxy(10,3); Write('_____');
  Gotoxy(10,4); Write('[1]. Input Data Pegawai');
  Gotoxy(10,5); Write('[2]. Output Data Pegawai');
  Gotoxy(10,6); Write('[3]. Keluar Program');
  Gotoxy(10,7); Write('_____');

  Gotoxy(10,8); Write('Pilihan  : ');
END;
```

BEGIN { Program Utama }

```

i := 1 ; Pil :='1' ;
While Pil<>'3' Do
```

```
Begin
  Menu ;
  Pil := Readkey ;
  Case PIL of
    '1' : InputPEGAWAI ;
    '2' : OutputPegawai;
  End;
End;
END.
```

Bandingkanlah kedua program tersebut diatas. Mana yang lebih fleksibel?

TUGAS PRAKTIKUM

1. Modifikasikan Program **TUGAS81.PAS** menjadi bentuk Latih92.Pas dengan menggunakan menu & procedure.

MENU MAHASISWA

```
-----
[ 1 ]. Input Data Nilai Mahasiswa
[ 2 ]. Output Data Nilai mahasiswa
[ 3 ]. Keluar Program
-----
```

Pilihan : x

Nama File : **TUGAS91.PAS**

BAB X FUNGSI

Pada prinsipnya penggunaan fungsi sama dengan prosedur, tetapi fungsi jauh lebih fleksibel dalam pemakaiannya. Hal ini disebabkan fungsi bisa menampung suatu nilai.

Fungsi juga bisa mengirimkan dan mengembalikan suatu nilai layaknya suatu prosedur. Dalam penggunaan fungsi lebih banyak membantu dibandingkan dengan prosedur.

Bentuk umum :

```
Function nmFungsi( parameter ) : typefungsi ;
Begin
    ...
    ...
End ;
```

Example 1 :

```
Function Hitung(x,y : Real ) : Real ;
Begin
    Hitung := x * y
End;
```

```
Var a, b, c : Real ;
Begin
    Write('Nilai A ?') ; Readln(a) ;
    Write('Nilai B ?') ; Readln(b) ;
    c := Hitung(a, b) ;
    Writeln('Nilai C = ', C:6:2) ;
End.
```

Contoh 10.1 Penggunaan fungsi dengan variable Lokal.

Contoh fungsi diatas merupakan fungsi dengan pengiriman parameter secara nilai (by value). Hasil dari **perkalian x & y** akan dimasukkan ke dalam **HITUNG**. Mengapa ?

Hal ini disebabkan **HITUNG** juga memiliki type data yaitu : REAL. Sehingga hasilnya dimasukkan ke Variabel C. Variabel a, b, c merupakan variabel Lokal karena dideklarasikan di bawah fungsi.

Program di atas juga bisa dituliskan seperti berikut ini :

Example 2 :

```
Function Hitung(x,y : Real ) : Real ;
Begin
    Hitung := x * y
End;
```

```
Var a, b : Real ;
Begin
    Write('Nilai A ?') ; Readln(a) ;
    Write('Nilai B ?') ; Readln(b) ;
    Writeln('Nilai C = ', Hitung(a, b) ) ;
End.
```

Contoh 10.2 Penggunaan fungsi

Example 3 :

VAR

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

Procedure INPUTData ;

Begin

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

End;

Function CariNilai(NA : Byte) : Char ;

Var NH : Char ;

Begin

CASE N OF

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

END;

CariNilai := NH ;

End;

BEGIN

InputData

Clrscr;

Writeln ('Nama : ', Nama);

Writeln ('Matakuliah : ', MTK);

Writeln ('Nilai Angka : ', N);

Writeln ('Nilai Huruf : ', CariNilai(N));

END.

Contoh 10.3 Penggunaan fungsi

LATIHAN

1. Ketiklah program dibawah ini : **Latih101.PAS**

```

PROGRAM MencariPEMBELIAN;
USES CRT ;
TYPE
  Pembeli = Record
    Kd      : String[4] ;
    Nama    : String[25] ;
    Jumlah  : Byte      ;
  END;

VAR
  rPbl      : Array[1..20] of Pembeli ;
  Jenis     : String [25] ;
  Harga,
  TH,tTot   : Longint ;
  i, j      : byte ;
  Pil,
  Lg        : Char ;

PROCEDURE InputPEMBELI ;
  PROCEDURE FormPEMBELI ;
  BEGIN
    Clrscr;
    Gotoxy(20,2); Write('INPUT DATA PEMBELIAN RUMAH');
    Gotoxy(10,3); Write('_____');
    Gotoxy(10,4); Write('Kode Rumah      :');
    Gotoxy(10,5); Write('Nama Pembeli      :');
    Gotoxy(10,6); Write('Jumlah Beli       :');
    Gotoxy(10,7); Write('_____');
    Gotoxy(10,8); Write('Masih Ada Data [Y/T] : ');
  END;
BEGIN
  REPEAT
    FORMPembeli;
    WITH RPbl[i] do
      Begin
        Gotoxy(27,4); Readln( Kd );
        Gotoxy(27,5); Readln( NAMA );
        Gotoxy(27,6); Readln( Jumlah );
        Gotoxy(33,8); Readln( Lg );
      END ;
      inc( i );
    UNTIL Upcase(Lg)='T';
  END;

PROCEDURE TabelPembeli;
BEGIN
  Clrscr ;
  Gotoxy(3,1);write('Informasi Pembelian Rumah');
  Gotoxy(3,2);write('CV. CITRA ANUGRAH PERMAI');

```

```

Gotoxy(3,3);write('_____');
_____');
Gotoxy(3,4);write(' No Kode      Jenis Rumah      Nama      Harga      Jumlah
Total ');
Gotoxy(3,5);write('      Rumah      Pembeli      Beli
Harga ');
Gotoxy(3,6);write('_____');
_____');
END;

```

PROCEDURE OutputPembeli;

FUNCTION CariJenis(cKd: String ; VAR Hrg: Longint) : String ;

VAR cJns : String ;

BEGIN

IF cKd = 'RS1' Then

Begin

cJns := 'Sederhana' ; Hrg := 40000000 ;

End

Else IF cKd = 'RS2' Then

Begin

cJns := 'Sangat Sederhana' ; Hrg := 30000000 ;

End

Else IF cKd = 'RS3' Then

Begin

cJns := 'Sangat Sederhana Sekali' ; Hrg := 20000000 ;

End ;

CariJenis := cJns ;

END;

FUNCTION CariTH(Hrg: Longint ; Jumlah : Byte) : Longint ;

BEGIN

CariTH := Hrg * Jumlah ;

END;

Begin

TabelPembeli;

FOR j:=1 TO k-1 DO

BEGIN

With rPBL[j] Do

Begin

Jenis := CariJenis(Kd, Harga) ;

TH := CariTH(Harga, Jumlah) ;

Gotoxy(5,6+j); Write(j:2, ' ', Kd: 4, ' ', Jenis);

Gotoxy(39,6+j); write(Nama);

Gotoxy(54,6+j); write(Harga:8, ' ', Jumlah:2, ' ', TH:8);

TTot := TTot + TH ;

End;

END;


```

Gotoxy(3,7+j);
write('_____');
_____');
Gotoxy(3,8+j); Write('  TOTAL Harga Penjualan          RP.', tTot:10);
Repeat Until Keypressed ;
End;

```

PROCEDURE MENU ;

BEGIN

```

Clrscr;
Gotoxy(20,2); Write('MENU PEMBELIAN RUMAH');
Gotoxy(10,3); Write('_____');
Gotoxy(10,4); Write('[ 1 ]. Input Data Pembelian');
Gotoxy(10,5); Write('[ 2 ]. Informasi ');
Gotoxy(10,6); Write('[ 3 ]. Keluar Program');
Gotoxy(10,7); Write('_____');
Gotoxy(10,8); Write(' Pilihan   : ');

```

END;

BEGIN

```

i := 1 ; Pil := '1' ;
While Pil <> '3' Do
Begin
  Menu ;
  Pil := Readkey ;
  Case PIL of
    '1' : InputPembeli;
    '2' : OutputPembeli;
  End;
End;

```

```

Clrscr ;

```

END.

2. Buatlah program berikut ini : **Latih102.PAS**

```

PROGRAM PenjualanKomputer;
USES CRT ;
TYPE
  Pembeli = Record
    Kd       : String[4] ;
    Nama     : String[25] ;
    Jumlah   : Byte      ;
  END;
VAR
  rPbl      : Array[1..20] of Pembeli ;
  Jenis     : String [25] ;
  Harga,
  HJ        : Longint      ;
  Disc,
  HB, tTot  : Real         ;
  i, j      : byte         ;
  Pil,
  lg        : Char         ;

```

```

PROCEDURE InputPEMBELI ;
PROCEDURE FormPEMBELI ;
BEGIN
  Clrscr;
  Gotoxy(20,2); Write('INPUT DATA PEMBELIAN KOMPUTER');
  Gotoxy(10,3); Write('_____');
  Gotoxy(10,4); Write('Kode :');
  Gotoxy(10,5); Write('Nama Pembeli :');
  Gotoxy(10,6); Write('Jumlah :');
  Gotoxy(10,7); Write('_____');
  Gotoxy(10,8); Write('Masih Ada Data [Y/T] :');
END;
BEGIN
  REPEAT
    FORMPembeli ;
    WITH RPbl[ i ] do
      Begin
        Gotoxy(27,4); Readln( Kd);
        Gotoxy(27,5); Readln( NAMA );
        Gotoxy(27,6); Readln( Jumlah);
        Gotoxy(33,8); Readln( Lg );
      END ;
      inc( i );
    UNTIL Upcase(Lg)='T';
  END;

```

```

PROCEDURE TabelPembeli;
BEGIN
  Clrscr ;
  Gotoxy(1,1);write("SUMATERA KOMPUTER");
  Gotoxy(1,2);write('Jln. HM. YAMIN 120 C');
  Gotoxy(1,3);write('          Informasi Penjualan Komputer');

  Gotoxy(1,4);write('_____')
  Gotoxy(1,5);write('No  Kode          Jenis          Harga  Jumlah  Nama          Harga')
  Discount  Harga ');
  Gotoxy(1,6);write('          Komputer  Komputer          Jual          Pembeli')
  Jual      Bayar ');
  Gotoxy(1,7);write('_____')
  END;

```

```

PROCEDURE OutputPembeli;
FUNCTION CariJenis(cKd: String ; VAR Hrg: Longint) : String ;
VAR cJns : String ;
BEGIN
  IF cKd = 'PI' Then
    Begin
      cJns := 'Pentium I' ; Hrg := 1500000 ;
    End
  Else IF cKd = 'PII' Then

```

```

        Begin
            cJns := 'Pentium II' ; Hrg := 2500000 ;
        End
    Else IF cKd = 'PIII' Then
        Begin
            cJns := 'Pentium III' ; Hrg := 3500000 ;
        End
    Else IF cKd = 'PIV' Then
        Begin
            cJns := 'Pentium IV' ; Hrg := 4500000 ;
        End ;
    CariJenis := cJns ;
END;

FUNCTION CariHargaJual(Hrg: Longint ; Jumlah : Byte ) : Longint ;
BEGIN
    CariHargaJual:= Hrg * Jumlah ;
END;

FUNCTION CariDiscount(Hrg: Longint ) : Real;
BEGIN
    If Hrg>=25000000 Then CariDiscount := 0.1 * Hrg
    Else If Hrg>=15000000 Then CariDiscount := 0.07 * Hrg
    Else If Hrg>=10000000 Then CariDiscount := 0.05 * Hrg
    Else CariDiscount := 0;
END;

Begin
    TabelPembeli;
    FOR j:=1 TO I-1 DO
        BEGIN
            With rPBL[j] Do
                Begin
                    Jenis := CariJenis(Kd, Harga ) ;
                    HJ := CariHargaJual(Harga, Jumlah) ;
                    Disc := CariDiscount(HJ);

                    HB := HJ - Disc ;
                    Gotoxy(2,7+j); Write( j:2,' ',Kd:4,' ',Jenis);
                    Gotoxy(21,7+j); write( Harga:8,' ',Jumlah:3,' ',Nama);
                    Gotoxy(47,7+j); write( HJ:8,' ', Disc:7:0,' ',HB:8:0);
                    TTot := TTot + HB;
                End;
            END;
            Gotoxy(1,8+j);
            write('
            _____');
            Gotoxy(1,9+j); Write(' TOTAL Harga Penjualan RP.', tTot:9:0);
            Repeat Until Keypressed ;
        End;

PROCEDURE MENU ;
BEGIN

```

```
Clrscr;
Gotoxy(20,2); Write('MENU PENJUALAN KOMPUTER');
Gotoxy(10,3); Write('_____');
Gotoxy(10,4); Write('[ 1 ]. Input Data Penjualan');
Gotoxy(10,5); Write('[ 2 ]. Informasi ');
Gotoxy(10,6); Write('[ 3 ]. Keluar Program');
Gotoxy(10,7); Write('_____');
Gotoxy(10,8); Write('Pilihan  : ');
END;

BEGIN
i := 1 ; Pil := '1' ;
While Pil <> '3' Do
Begin
Menu ;
Pil := Readkey ;
Case PIL of
'1' : InputPembeli;
'2' : OutputPembeli;
End;
End;
Clrscr ;
END.
```

BAB XI TURBO PASCAL UNIT

Yaitu sebuah program pembantu yang terpisah dari program pemanggil yang bertujuan untuk hal-hal tertentu mis pembuatan prosedur atau fungsi yang biasa digunakan dalam pemrograman.

Bentuk Umum

```

UNIT nmUnit ;
INTERFACE
Uses .....
Var
    nmVar : TypeData ;
    ....

    NmProcedure ;
    NmFunction ;
    ...
    ...
    ...

IMPLEMENTATION
    NmProcedure ;
    NmFunction ;
    ...
    ...
    ...

Begin
End.
  
```

SYARAT TPU

1. Nama File dan Nama Unit harus Sama
2. Pada bagian akhir program bisa dituliskan **Begin End.** atau cukup **End.** saja.
3. NmProcedure, NmFunction pada bagian Interface & Implementation harus sama.
4. TPU hanya bisa di **Compile** tetapi tidak bisa di jalankan.

PEMBUATAN UNIT

Nama FILE : MyUNIT.Pas;

Unit MyUnit ;

INTERFACE

Uses crt;

type

_Str = string[80];

_Str2 = string[80];

PROCEDURE Warna(d, b : byte);

PROCEDURE WriteTO(c, b : byte ; txt : _str);

PROCEDURE WriteTOL(c, b: byte ; txt : longint);

PROCEDURE WriteTOR (c, b: byte ; txt : real);

FUNCTION _GetSTR(c, b: byte) : _str;

FUNCTION _GetL(c, b : byte) : longint;

FUNCTION _GetCH(c, b: Byte):Char ;

```
PROCEDURE Cleararea(c, b, c1, b1 : byte);  
PROCEDURE Box(c, b, c1, b1 : byte);  
PROCEDURE AREA(c, b, n : byte);
```

IMPLEMENTATION

```
PROCEDURE Warna;  
Begin  
    Textcolor(d); Textbackground(b);  
End;
```

```
PROCEDURE WriteTO ;  
Begin  
    Gotoxy(c, b); Write(Txt);  
End;
```

```
PROCEDURE WriteTOL;  
Begin  
    Gotoxy(c, b) ; Write(txt);  
End;
```

```
PROCEDURE WriteTOR;  
begin  
    Gotoxy(c, b); Write(Txt:9:0);  
end;
```

```
FUNCTION _GetSTR ;  
Var Field : _str;  
Begin  
    gotoxy(c, b) ; Readln(field);  
    _GetSTR:=field;  
End;
```

```
FUNCTION _GetL;  
Var Field : Longint;  
Begin  
    Gotoxy(c, b); Readln(field);  
    _GetL := Field;  
End;
```

```
FUNCTION _GetCH;  
Var Field : Char ;  
Begin  
    Gotoxy(c, b); Readln(field);  
    _GetCH := Field;  
End;
```

```
PROCEDURE Cleararea;  
Var K, L : byte;  
Begin  
    For K:= b to b1 do  
        For L:= c to c1 do  
            WriteTO(L, K, ' ');  
End ;
```

```

PROCEDURE Box;
Var K, L : byte;
Begin
  Cleararea(c, b, c1, b1);
  WriteTO(c, b, 'U'); { Alt 218 }
  WriteTO(c1, b, 'ç'); { Alt 191 }
  WriteTO(c, b1, 'A'); { Alt 192 }
  WriteTO(c1, b1, 'Ù'); { Alt 217 }
  For K := c + 1 to c1 - 1 do
  Begin
    WriteTO(k, b, 'Ä'); { Alt 196 }
    WriteTO(k, b1, 'Ä'); { Alt 196 }
  End;
  For L := b + 1 to b1 - 1 do
  Begin
    WriteTO(c, L, '3'); { Alt 179 }
    WriteTO(c1, L, '3'); { Alt 179 }
  End;
End;

Procedure AREA(c, b, n : byte);
Var L : byte;
Begin
  for L:= c to c+n do
    WriteTO(L, b, ' ');
end;

```

```

Begin
End. { boleh dituliskan end saja tanpa begin }

```

Contoh 11.1 Pembuatan unit pada pascal

PEMANGGILAN SUATU UNIT

Unit yang telah kita buat sebelumnya dapat di panggil melalui program yang terpisah dengan program Unit tadi. Pemanggilan unit tersebut melalui Statemen USES.

Example 2 : Nama Program **Latih112.Pas**

```

Program PanggilTPU;
Uses Crt, MyUNIT; {Pemanggilan Unit MyUNIT di bagian Uses }
Begin
  Warna(4,15); Clrscr;
  WriteTO (20,10, 'Test menggunakan File TPU');
  Warna(13, 2); BOX(10, 5, 50, 20)
  Warna(14, 1);
  WriteTO (20,10, 'Oke Dong ');
End.

```

Contoh 11.2 Program untuk memanggil unit yang telah kita ciptakan.

LATIHAN

1. Buatlah Program berikut ini : **Latih111.Pas**

```

Uses Crt, MyUNIT;
Var
  Nama : String[30];

```

```

Harga      : longint  ;
Jual       : Integer  ;
THarga,
Disc,
TBayar    : Real      ;

PROCEDURE Form ;
BEGIN
    BOX(15,4,70,15) ;
    Warna(15,1) ; WriteTO (17,4,'Entry Data Penjualan Barang');
    Warna(14,1) ;
    WriteTO (20, 6,'Nama Barang      :[                      ]');
    WriteTO (20, 7,'Harga Rp.       :[                      ]');
    WriteTO (20, 8,'Jumlah Jual      :[                      ]');
    WriteTO (20, 9,'Total Harga Rp.  :[                      ]');
    WriteTO (20,10,'Discount Rp.     :[                      ]');
    WriteTO (20,11,'Total Bayar Rp.  :[                      ]');
    WriteTO (20,13,'Ingin Menghitung Lagi [Y/T] :');
END;

PROCEDURE InputData ;
Begin
    Nama := _GetS(35,6) ;
    Harga:= _GetL(35,7) ;
    Jual := _GetL(35,8) ;
End ;

FUNCTION CariDISC(TH : Longint ) : Real ;
Var   Disc : Real ;
BEGIN
    IF TH > 1000000 Then Disc := 0.1 * TH
    Else If TH > 500000 Then Disc := 0.05 * TH
    Else Disc := 0 ;
    CariDISC := Disc ;
END;

BEGIN
    Warna(2,1) ; Clrscr ; Warna(15,1) ;
    REPEAT
        Form ;
        InputData ;
        THarga := Harga * Jual ;

        warna(11,1) ; WriteTOR (35, 9,THarga) ;

        Disc := CariDisc(THarga)
        WriteTOR (35,10, Disc ) ;

        TBayar := THarga - Disc ;
        WriteTOR (35,11,TBayar ) ;
    UNTIL  UPCASE(_GetC(50,13)) = 'T' ;
END.

```


TUGAS PRAKTIKUM

Buatlah Program berikut ini dengan menggunakan UNIT yang telah ada.

Nama Program : **Tugas111.Pas**

Tampilan Program

Entry Data Penjualan Komputer		
Type Komputer	:	[]
Jenis	:	[]
Harga Rp.	:	[]
Jumlah Penjualan	:	[] Unit
Total Harga Rp	:	[]
Discount Rp	:	[]
Total Bayar Rp	:	[]

Ingin menghitung Lagi [Y/T] : []

Ketentuan Proses :

- Jenis & Harga diperoleh dari
 - Jika Type = 'PI' maka Jenis = 'Komputer Pentium I'
 - Harga = 1500000
 - Jika Type = 'PII' maka Jenis = 'Komputer Pentium II'
 - Harga = 2500000
 - Jika Type = 'PIII' maka Jenis = 'Komputer Pentium III'
 - Harga = 3500000
 - Jika Type = 'PIV' maka Jenis = 'Komputer Pentium IV'
 - Harga = 4500000
- Total Harga = Harga * Jumlah penjualan
- Disc diperoleh dari
 - Jika Total Harga > 20000000 maka Disc = 0.2 * Total Harga
 - Jika Total Harga > 15000000 maka Disc = 0.1 * Total Harga
 - Jika Total Harga > 10000000 maka Disc = 0.05 * Total Harga
 - Jika Total Harga < 10000000 maka Disc = 0
- Total Bayar = Total Harga – Disc

Untuk mencari Jenis & Harga Komputer, Discount gunakan suatu procedure/Function.

BAB XII FILE

File merupakan kumpulan dari record. File digunakan agar data yang telah terkumpul dapat disimpan kedalam disk sehingga pemeliharaan data menjadi lebih mudah dilakukan.

Pascal menyediakan 3 jenis file :

1. File Bertipe
2. File Teks
3. File Tak Bertipe

PROSEDUR & FUNGSI STANDAR

Pascal menyediakan beberapa perintah yang akan digunakan untuk pemeliharaan file. Perintah-perintah tersebut ada yang dalam prosedur standar dan ada juga dalam fungsi standar. Diantaranya :

❖ **ASSIGN**

Digunakan untuk menghubungkan nama dari external file ke dalam suatu file variabel.

Syntax : Assign (VarFile, namaFile)

Ex : Assign (fMhs , 'Mhs.Dat')

❖ **REWRITE**

Digunakan untuk membuka dan menciptakan file. Seandainya dalam disk sudah ada file data yang sama dengan nama file yang berkaitan dengan varFile, maka isi file akan menjadi kosong.

Syntax : ReWrite (VarFile) ;

Ex : Assign (fMhs , 'Mhs.Dat')
ReWrite (fMhs)

❖ **RESET**

Digunakan untuk membuka file yang telah ada di disk, dan jika file yang dibuka tidak ada di disk, maka akan terjadi runtime error.

Syntax : Reset (VarFile) ;

Ex : Assign (fMhs , 'Mhs.Dat')
Reset (fMhs)

❖ **CLOSE**

Digunakan untuk menutup file yang sedang aktif, bila file tersebut tidak dibutuhkan lagi.

Syntax : Close (VarFile) ;

Ex : Close (fMhs)

❖ **ERASE**

Digunakan untuk menghapus file, Erase hanya bisa digunakan jika file dalam keadaan tertutup, jika file sedang terbuka, pemanggilan Erase akan menyebabkan kesalahan.

- Syntax : Erase (VarFile) ;
- Ex : Erase (fMhs)
- ❖ **RENAME**
Digunakan untuk menggantikan nama file yang ada di disk dengan nama lain. Rename hanya bisa dipakai terhadap file yang tertutup.
- Syntax : Rename (VarFile, namaBARU) ;
- Ex : Rename (fMhs , 'DBMHS.Dat')
- ❖ **EOF (End of FILE)**
Digunakan untuk mengetahui status dari suatu file, apakah suatu proses telah mencapai posisi terakhir atau tidak.
- Syntax : EOF(VarFile) : Boolean ;
- Ex : IF Not EOF(fMhs) Then

FILE BERTIPE

File bertipe merupakan file yang diakses secara urut (sequential access) dan secara acak (random access), maka dengan demikian data dalam file dapat dibaca dan direkam dimanapun (acak) didalam file tersebut.

Pengolahan data dengan file bertipe mempunyai urutan proses sebagai berikut :

1. Menyebutkan variabel file dan nama file
2. Membuka FILE
3. Melaksanakan pengaksesan FILE
4. Menutup FILE

DEKLARASI FILE BERTIPE

Deklarasi file bertipe dilakukan seperti contoh berikut :

```

Type
    Mahasiswa = record
        NPM      : String[9] ;
        Nama     : String[30] ;
        Jenjang  : String[5] ;
        Jurusan  : String[25] ;
    End;
Var
    fMhs : FILE of Mahasiswa ;
    rMhs : Mahasiswa ;
  
```

PROSEDUR DAN FUNGSI STANDAR FILE BERTIPE

1. WRITE

Digunakan untuk merekam data dari variabel record ke dalam file di disk.

Syntax : Write (VarFILE, VarRec) ;

2. READ

Digunakan untuk membaca record yang tersimpan dalam file data.

Syntax : Read (VarFILE, VarRec);

3. SEEK

Digunakan untuk menempatkan petunjuk record (pointer) ke record yang diinginkan.

Syntax : Seek (VarFILE, nomorRecord);

4. FILESIZE

Digunakan untuk memperoleh jumlah Record dalam suatu File Data

Syntax : FileSize (VarFILE);

PROGRAM TAMBAH PADA FILE BERTIPE

Berikut adalah contoh program untuk menambahkan data pada File bertipe.

Nama Program : **UnitKU.Pas**

UNIT UNITKU;

INTERFACE

Uses Crt ;

Procedure Warna(d, b : byte);

Procedure Clear(c, b, cl, bl :byte);

Procedure Garis(c, cl, b: Byte) ;

Procedure Pesan(txt: String);

IMPLEMENTATION

Procedure Warna;

Begin

Textcolor(d); Textbackground(b);

End;

Procedure Clear;

var K, L :byte;

begin

for K:= b to bl do

for L:= c to cl do

begin

GotoXY(L, K) ; Write(' ');

end;

end;

Procedure Garis;

var K : byte;

Begin

For K := c to cl do

begin

GotoXY(k, b) ; Write('-');

end;

End;

Procedure Pesan;

Begin

```

Warna(14,4) ; Clear(1,24,80,24) ;
GotoXY(10, 24) ; Write( txt );
end;

```

```

Begin
end.

```

Nama Program : entUK.Pas

```

USES Crt, UnitKU ;
TYPE
    UangKuliah = record
        Kode      : String[3] ;
        Jurusan: String[30] ;
        UKuliah,
        BayarA,
        BesarC : Longint      ;
        JumlahC: Byte         ;
    end;
VAR
    fUK      : file of uangkuliah ;
    rUK      : uangKuliah          ;
    Mad, Dsb : Char               ;
    POS      : Byte               ;
    cKode     : String[3]         ;

    FUNCTION SeekUK(Kd : String ; VAR rPos : Byte ) : Boolean ;
    VAR
        i      : Byte ;
        Ketemu : Boolean ;
    BEGIN
        Ketemu := False ;
        rPos := 0 ;
        SeekUK:= False ;
        Seek(fUK, 0) ; i := 1 ;
        WHILE not eof(fUK) And Not KETEMU Do
            BEGIN
                Seek(fUK, i-1) ;
                Read(fUK, rUK) ;
                if rUK.Kode = Kd Then
                    Begin
                        Ketemu := True ; SeekUK:= True ; rPos := i ;
                    End;
                inc(i) ;
            END;
        END;

    PROCEDURE OpenDB ;
    BEGIN
        Assign(fUK, 'UangK.Dat') ;
        {$I-} Reset (fUK) ;
        {$I+} If IoResult <> 0 Then Rewrite(fUK) ;
    END ;

```

```

PROCEDURE FormUK ;
BEGIN
    Warna(14,4)    ; Clear(19,5,70,5) ;
    GotoXY(20, 5) ; Write('ENTRI DATA UANG KULIAH');
    Warna(14,1)    ; Clear(19,6,70,13) ;
    GotoXY(20, 7) ; Write(' Kode           : ');
    GotoXY(20, 8) ; Write(' Jurusan        : ');
    GotoXY(20, 9) ; Write(' Uang Kuliah Rp.   : ');
    GotoXY(20,10) ; Write(' Pembayaran Awal Rp.: ');
    GotoXY(20,11) ; Write(' Besar Cicilan Rp.  : ');
    GotoXY(20,12) ; Write(' Jumlah Cicilan     : ');
END;

```

```

FUNCTION Tanya(txt : String ) : Boolean ;
VAR x : Char ;
BEGIN
    Tanya := False;
    Warna(14,4)    ; Clear(19,14,70,14) ;
    GotoXY(21,14) ; Write(txt);
    repeat
        GotoXY(45,14) ; x := Readkey ;
    Until x in['Y','y','t','T'] ;
    if Uppcase(x) = 'Y' Then Tanya := True ;
END ;

```

```

PROCEDURE ShowUK ;
BEGIN
    Warna(15,2) ;
    With rUK Do
    Begin
        GotoXY(42, 8) ; Write(Jurusan) ;
        GotoXY(42, 9) ; Write(UKuliah) ;
        GotoXY(42,10) ; Write(BayarA) ;
        GotoXY(42,11) ; Write(BesarC) ;
        GotoXY(42,12) ; Write(JumlahC) ;
    End;
END;

```

```

PROCEDURE InputUK(cKode : String) ;
BEGIN
    Warna(15,3) ;
    With rUK Do
    Begin
        Kode := cKode ;
        GotoXY(42, 8) ; Readln(Jurusan) ;
        GotoXY(42, 9) ; Readln(UKuliah) ;
        GotoXY(42,10) ; Readln(BayarA) ;
        GotoXY(42,11) ; Readln(BesarC) ;
        GotoXY(42,12) ; Readln(JumlahC) ;
    End;
END;

```

```

BEGIN
    OpenDB ;

```

```

REPEAT
    FormUK;
    GotoXY(42, 7) ; Readln(cKode) ;
    if cKode='' Then Exit ;
    if SeekUK(cKode, POS) Then
    begin
        ShowUK ;
        Pesan('DATA INI SUDAH ADA...MAS ! ');
    end
    Else
    Begin
        REPEAT
            InputUK(ckode) ;
            UNTIL Tanya('Data Sudah Benar [Y/T] : ');
            Seek(fUK, FileSize(fUK)) ;
            Write(fUK, rUK);
        End;
    UNTIL Not Tanya('Masih Ada Data [Y/T] : ');
    Close(fUK);
END.

```

Jika Program Entri Data Uang Kuliah telah selesai maka isikanlah data berikut ini :

Kode	Jurusan	Uang Kuliah	Pembayaran Awal	Besar Cicilan	Jumlah Cicilan
SI4	Sistem Informasi	4000000	2000000	250000	8
TI4	Teknik Informatika	4500000	2100000	300000	8
MI3	Manajemen Informatika	3500000	1500000	250000	8
KA3	Komputer Akuntansi	3250000	1250000	250000	8

Bila data diatas diisi untuk yang kedua kalinya maka program diatas akan menampilkan data yang telah diinputkan sebelumnya.

PROGRAM LIHAT(INFORMASI) DATA FILE BERTIPE

Data-data yang telah tersimpan dalam disk dapat dilihat atau ditampilkan kembali pada saat diperlukan. Untuk menampilkan data-data tersebut buatlah program dibawah ini :

```

Nama Program : entUK.Pas
USES Crt, UnitUK ;
TYPE
    UangKuliah = record
        Kode    : String[3] ;
        Jurusan: String[30] ;
        UKuliah,
        BayarA,
        BesarC : Longint ;
        JumlahC: Byte ;
    End;

VAR
    fUK    : file of uangkuliah ;
    rUK    : uangKuliah ;
    br,
    i      : Byte ;

```

```

Ket      : string[3]      ;

PROCEDURE OpenDB ;
Begin
    Assign(fUK, 'UangK.Dat');
    {$I-} Reset (fUK) ;
    {$I+} If IoResult <> 0 Then ReWrite(fUK) ;
end;

PROCEDURE FormInfo ;
Begin
    Warna(15,4) ;Clear(1,1,80,25);
    GotoXY(3, 2);Write('SEKOLAH TINGGI MANAJEMEN INFORMATIKA &
KOMPUTER');
    GotoXY(3, 3);Write('Jln. Yos Sudarso');
    GotoXY(3, 5);Write('Informasi Data Uang KULIAH per JURUSAN');
    Warna(15,1) ;Clear(2,6,79,9);
    Garis(2,79,6);
    GotoXY(3, 7);Write( ' No.   Kode Jurusan
Uang   Pembayaran   Besar       Jumlah ');
    GotoXY(3, 8);Write( '
Kuliah Rp   Awal Rp. Cicilan Rp. Cicil ');
    Garis(2,79,9);
    Warna(15,1) ; Clear(2,10,79,23);
    Garis(2,79,23) ;
end;

BEGIN
    OpenDB ;
    FormInfo;
    Seek(fUK, 0) ; i := 1 ; br := 10;
    WHILE not Eof(fUK) do
    BEGIN
        Seek(fUK, I-1) ;
        Read(fUK, rUK) ;
        With rUK do
        Begin
            Warna(14,1) ;
            Gotoxy(5 ,br) ; Write(i,' ', Kode) ;

            if Copy(kode,length(Kode),1)='4' Then Ket :='S1'
            Else if Copy(kode,length(Kode),1)='3' Then Ket :='D3'
            Else if Copy(kode,length(Kode),1)='1' Then Ket :='D1';

            Gotoxy(14,br) ; Write(Ket, '-',Jurusan) ;
            Gotoxy(39,br) ; Write(UKuliah:9) ;
            Gotoxy(51,br) ; Write(BayarA:9) ;
            Gotoxy(60,br) ; Write(BesarC:9) ;
            Gotoxy(73,br) ; Write(JumlahC:3) ;

            if br=22 Then
            begin
                Pesan('Tekan Enter.....') ;
                Readln;
            end;
        end;
        i := i + 1 ;
        br := br + 1 ;
    end;
end;

```



```

        Warna(15,1) ; Clear(2,10,79,22);
        br:= 9 ;
    end;
End;
inc(i);
inc(br) ;
END ;
Pesn('Tekan Enter untuk Melanjutkan.....') ;
Readln;
close(fUK);
END.

```

PROGRAM KOREKSI DATA FILE BERTIPE

Data yang telah diinputkan ada kalanya ingin kita perbaiki untuk menjaga updatenya data, program dibawah ini untuk mengedit data-data yang telah diinputkan.

Nama Program : **korUK.Pas**

```

USES Crt, UnitKU ;
TYPE
    UangKuliah = record
        Kode      : String[3] ;
        Jurusan: String[30] ;
        UKuliah,
        BayarA,
        BesarC : Longint ;
        JumlahC: Byte ;
    end;
VAR
    fUK      : file of uangkuliah ;
    rUK      : uangkuliah ;
    Mad, Dsb : Char ;
    POS      : Byte ;
    cKode : String[3] ;

    FUNCITON SeekUK(Kd : String ; VAR rPos : Byte ) : Boolean ;
    VAR
        i : Byte ;
        Ketemu : Boolean ;
    BEGIN
        Ketemu := False ;
        rPos := 0 ;
        SeekUK:= False ;
        Seek(fUK, 0) ; i := 1 ;
        While not eof(fUK) And Not KETEMU Do
            Begin
                Seek(fUK, i-1) ;
                Read(fUK, rUK) ;
                if rUK.Kode = Kd Then
                    Begin
                        Ketemu := True ; SeekUK:= True ; rPos := i ;
                    End;
                inc(i) ;
            End;
        End;
    END;

```

```

        End;
    END ;

    PROCEDURE OpenDB ;
    BEGIN
        Assign(fUK, 'UangK.Dat');
        {$I-} Reset (fUK) ;
        {$I+} If IoResult <> 0 Then ReWrite(fUK) ;
    END ;

    PROCEDURE FormUK ;
    BEGIN
        Warna(14,4) ; Clear(19,5,70,5) ;
        GotoXY(20, 5) ; Write('EDIT DATA UANG KULIAH');
        Warna(14,1) ; Clear(19,6,70,13) ;
        GotoXY(20, 7) ; Write(' Kode : ');
        GotoXY(20, 8) ; Write(' Jurusan : ');
        GotoXY(20, 9) ; Write(' Uang Kuliah Rp. : ');
        GotoXY(20,10) ; Write(' Pembayaran Awal Rp.: ');
        GotoXY(20,11) ; Write(' Besar Cicilan Rp. : ');
        GotoXY(20,12) ; Write(' Jumlah Cicilan : ');
    END ;

    FUNCTION Tanya(txt : String ): Boolean ;
    VAR x : Char ;
    BEGIN
        Tanya := False;
        Warna(14,4) ; Clear(19,14,70,14) ;
        GotoXY(21,14) ; Write(txt);
        repeat
            GotoXY(45,14) ; x := Readkey ;
        Until x in ['Y', 'y', 't', 'T'] ;
        if Uppcase(x) = 'Y' Then Tanya := True ;
    END ;

    PROCEDURE ShowUK ;
    BEGIN
        Warna(15,2) ;
        With rUK Do
            Begin
                GotoXY(42, 8) ; Write(Jurusan) ;
                GotoXY(42, 9) ; Write(UKuliah) ;
                GotoXY(42,10) ; Write(BayarA) ;
                GotoXY(42,11) ; Write(BesarC) ;
                GotoXY(42,12) ; Write(JumlahC) ;
            End;
    END ;

    PROCEDURE InputUK(cCode : String) ;
    BEGIN
        Warna(15,3) ;
        With rUK Do
            Begin
                Kode := cCode ;
            End;
    END ;

```

```

        GotoXY(42, 8) ; Readln(Jurusan) ;
        GotoXY(42, 9) ; Readln(UKuliah) ;
        GotoXY(42,10) ; Readln(BayarA) ;
        GotoXY(42,11) ; Readln(BesarC) ;
        GotoXY(42,12) ; Readln(JumlahC) ;
    End;
END;

BEGIN
    OpenDB ;
    REPEAT
        warna(10,3);
        clrscr ;
        FormUK;
        GotoXY(42, 7) ; Readln(cKode) ;
        if cKode='' Then Exit ;
        if SeekUK(cKode, POS) Then
            begin
                ShowUK ;
                IF Tanya('Data ini di edit [Y/T]: ') Then
                    begin
                        FormUK ;
                        GotoXY(42, 7) ; Write (cKode) ;
                        REPEAT
                            InputUK(cKode) ;
                        UNTIL Tanya('Data Sudah Benar [Y/T] : ');
                        Seek(fUK, POS - 1) ;
                        Write(fUK, rUK);
                    end;
                end
            end
            Else Pesan('DATA TIDAK SUDAH ADA...MAS ! ');
        UNTIL Not Tanya('Masih Ada Data [Y/T] : ');
        Close(fUK);
    END.

```

PROGRAM HAPUS DATA FILE BERTIPE

Data yang tidak diperlukan lagi, sebaiknya dihapus agar tempat penyimpanan tidak penuh.

Nama Program : **hapUK.Pas**

```

USES Crt, UnitKU ;
TYPE
    UangKuliah = record
        Kode    : String[3] ;
        Jurusan: String[30] ;
        UKuliah,
        BayarA,
        BesarC : Longint    ;
        JumlahC: Byte       ;
    End;

VAR
    fDummy,

```

```

fUK   : file of uangkuliah ;
rUK   : uangKuliah          ;
Mad, Dsb : Char             ;
i,
POS   : Byte                ;
cKode : String[3]           ;

FUNCTION SeekUK(Kd : String ; VAR rPos : Byte ) : Boolean ;
VAR
    i      : Byte      ;
    Ketemu : Boolean    ;
BEGIN
    Ketemu := False ;
    rPos := 0 ;
    SeekUK:= False ;
    Seek(fUK, 0) ; i := 1 ;
    While not eof(fUK) And Not KETEMU DO
    Begin
        Seek(fUK, i-1) ;
        Read(fUK, rUK) ;
        if rUK.Kode = Kd Then
        Begin
            Ketemu := True ;
            SeekUK:= True ;
            rPos := i ;
        End;
        inc(i) ;
    End;
END;

PROCEDURE OpenDB ;
BEGIN
    Assign(fUK, 'UangK.Dat');
    {$I-} Reset (fUK) ;
    {$I+} If IoResult <> 0 Then ReWrite(fUK) ;
END;

PROCEDURE OpenDummy ;
BEGIN
    Assign(fDummy, 'Dummy.Dat');
    ReWrite(fDummy) ;
END;

PROCEDURE FormUK ;
BEGIN
    Warna(14,4) ; Clear(19,5,70,5) ;
    GotoXY(20, 5) ; Write('HAPUS DATA UANG KULIAH');
    Warna(14,1) ; Clear(19,6,70,13) ;
    GotoXY(20, 7) ; Write(' Kode                : ');
    GotoXY(20, 8) ; Write(' Jurusan                : ');
    GotoXY(20, 9) ; Write(' Uang Kuliah Rp.          : ');
    GotoXY(20,10) ; Write(' Pembayaran Awal Rp.: ');
    GotoXY(20,11) ; Write(' Besar Cicilan Rp.   : ');
    GotoXY(20,12) ; Write(' Jumlah Cicilan      : ');

```

```

end;

FUNCTION Tanya(txt : String ): Boolean ;
VAR x : Char ;
BEGIN
    Tanya := False;
    Warna(14,4) ; Clear(19,14,70,14) ;
    GotoXY(21,14) ; Write(txt);
    repeat
        GotoXY(45,14) ; x := Readkey ;
    Until x in['Y','y','t','T'] ;
    if Ucase(x) = 'Y' Then Tanya := True ;
END ;

PROCEDURE ShowUK ;
BEGIN
    Warna(15,2) ;
    With rUK Do
    Begin
        GotoXY(42, 8) ; Write(Jurusan) ;
        GotoXY(42, 9) ; Write(UKulliah) ;
        GotoXY(42,10) ; Write(BayarA) ;
        GotoXY(42,11) ; Write(BesarC) ;
        GotoXY(42,12) ; Write(JumlahC) ;
    End;
END;

BEGIN
    OpenDB ;
    OpenDummy ;
    REPEAT
        warna(10,3);
        clrscr ;
        FormUK;
        GotoXY(42, 7) ; Readln(cKode) ;
        if cKode='' Then Exit ;
        if SeekUK(cKode, POS) Then
            begin
                ShowUK ;
                IF Tanya('Data ini di hapus[Y/T]: ') Then
                    begin
                        Seek(fUK, 0) ; i := 1 ;
                        WHILE not eof(fUK) do
                            BEGIN
                                Seek(fUK, i-1) ; Read(fUK, rUK) ;
                                if rUK.Kode <> cKode Then
                                    begin
                                        seek(fDummy, fileSize(fDummy)) ;
                                        Write(fDummy, rUK ) ;
                                    End;
                                inc(i) ;
                            END ;
                        Close(fUK); Close(fDummy);
                        Erase(fUK) ;
                    end
                end
            end
        end
    UNTIL

```

```

        Rename(fDummy, 'UangK.Dat') ;
        OpenDB      ;
        OpenDummy ;
        Pesan('Hapus Data Berhasil...MAS ! ');
    end;
end
    Else Pesan('DATA TIDAK SUDAH ADA...MAS ! ');
UNTIL Not Tanya('Masih Ada Data [Y/T]  : ');
Close(fUK);
Close(fDummy);
END.

```

PROGRAM MENU SEDERHANA

Agar program-program yang telah kita buat sebelumnya dapat dijalankan secara sekaligus, maka kita membutuhkan program menu untuk memanggil program-program tersebut.

Nama Program : **Menu.Pas**

Uses CRT, UnitKU ;

Var

Pil : Byte ;

Procedure Entril ;
{ \$I entUK.Pas }

Procedure Edit1 ;
{ \$I korUK.Pas }

Procedure Hapus1 ;
{ \$I hapUK.Pas }

Procedure Infol ;
{ \$I infUK.Pas }

Procedure TampilMenu ;
Begin

```

    warna(3,3) ; Clrscr ;
    warna(14,4) ; Clear(4,8,39,21) ;
    warna(14,4) ; Clear(4,7,39,7) ;
    GotoXY(9, 7) ; Write('Menu Utama') ;
    warna(14,3) ;
    GotoXY(5, 8) ; Write('[ 1] Entri Uang Kuliah') ;
    GotoXY(5, 9) ; Write('[ 2] Edit Uang Kuliah') ;
    GotoXY(5, 10) ; Write('[ 3] Hapus Uang Kuliah') ;
    GotoXY(5, 11) ; Write('-----') ;
    GotoXY(5, 12) ; Write('[ 4] Entri Mahasiswa') ;
    GotoXY(5, 13) ; Write('[ 5] Edit Mahasiswa') ;
    GotoXY(5, 14) ; Write('[ 6] Hapus Mahasiswa') ;
    GotoXY(5, 15) ; Write('-----') ;
    GotoXY(5, 16) ; Write('[ 7] Entri Pembayaran Uang Kuliah') ;
    GotoXY(5, 17) ; Write('-----') ;
    GotoXY(5, 18) ; Write('[ 8] Inf. Uang Kuliah') ;
    GotoXY(5, 19) ; Write('[ 9] Inf. Mahasiswa') ;
    GotoXY(5, 20) ; Write('[10] Keluar') ;

```

```
GotoXY(5, 21); Write('-----') ;
warna(14,4) ; Clear(4,22,39,22) ;
GotoXY(5, 22) ; Write('Pilihan : ');
end;

Begin
  Pil := 1 ;
  While Pil <> 10 Do

    Begin
      TampilMenu ;
      GotoXY(17, 22) ; Readln(Pil) ;
      case pil of
        1 : Entril ;
        2 : Edit1 ;
        3 : Hapus1 ;
        8 : Infol ;
        10 : Halt ;
      end ;
    end;
  end.
end.
```

Jika program menu diatas dijalankan maka akan menimbulkan error pada Dua Tempat yaitu :

- Uses Crt, UnitKU;
- End.

Hal ini disebabkan pada struktur pascal hanya mengenal sebuah Uses & End. pada suatu program saja. Sehingga Jika program : entUK, infUK, hapUK, korUK di gabungkan dengan menu akan menjadi satu program, bukan 5 program.

Jadi penyelesaiannya yaitu :

- Uses Crt, UnitKU; → anda hilangkan
- End. → ganti menjadi End ;

Hal diatas dilakukan pada program **entUK, infUK, hapUK, korUK** bukan pada MENU. Setelah anda ganti maka anda jalankan lagi dari Menu.pas .

BAB XIII GRAFIK

Untuk menggunakan grafik kita harus memasukkan unit Graph pada program kita dengan cara :

USES Graph;

Unit Standar Graph menyediakan fungsi standar lebih dari 50 fungsi grafik yang dapat dipergunakan untuk pembuatan grafik. Selain unit standar Graph, kita juga membutuhkan :

- File extension BGI (*.BGI)
- File extension CHR (*.CHR)

File BGI (Borland Graphic Interface) berguna sebagai grafik driver yang menunjukkan graphics adapter untuk monitor anda :

Graphics Adapter	Nama File Graphics Griver
CGA dan MCGA	CGA.BGI
EGA dan VGA	EGAVGA.BGI
Hercules	HERC.BGI
AT&T 400 Line	ATT.BGI
3270 PC	PC3270.BGI

INIALISASI GRAFIK.

Sebelum kita bekerja dengan grafik, kita harus menginialisasikan dahulu ke mode graph. Untuk menginialisasi ke mode grafik kita membutuhkan statemen **InitGraph**.

Bentuk Umum :

InitGraph(var GraphDriver: Integer; var GraphMode: Integer; PathToDriver: string);

Example :

InitGraph(Gd, Gm, '');

Variabel Gd & Gm di deklarasikan dengan type data Integer. GraphDriver merupakan merupakan driver yang dipergunakan pada komputer. Beberapa konstanta mengenai graphics driver didefinisikan pada standar Graph, yaitu :

Detect	= 0
CGA	= 1
MCGA	= 2
EGA	= 3
EGA64	= 4
EGAMono	= 5
RESERVED	= 6
HercMono	= 7
ATT400	= 8
VGA	= 9
PC3270	= 10

Pada prosedur standar InitGraph, GraphMode adalah mode grafik yang dipergunakan untuk driver bersangkutan. Biasanya GraphMode digunakan sebagai auto detection. Beberapa konstanta untuk nilai GraphMode sebagai berikut :

Graphic Mode	Nilai	Graphic Mode	Nilai
--------------	-------	--------------	-------

CGAC0	0 {320 x 200}	EGALo	0 {640 x 200}
CGAC1	1 {320 x 200}	EGAHi	1 {640 x 350}
CGAC2	2 {320 x 200}	EGA64Lo	0 {640 x 200}
CGAC3	3 {320 x 200}	EGA64Hi	1 {640 x 350}
CGAHi	4 {640 x 200}	EGAMonoHi	3 {640 x 350}
MCGAC0	0 {320 x 200}	HercMonoHi	0 {720 x 348}
MCGAC1	1 {320 x 200}	ATT400C0	0 {320 x 200}
MCGAC2	2 {320 x 200}	ATT400C1	1 {320 x 200}
MCGAC3	3 {320 x 200}	ATT400C2	2 {320 x 200}
MCGAMed	4 {640 x 200}	ATT400C3	3 {320 x 200}
MCGAHi	5 {640 x 480}	ATT400Med	4 {640 x 200}
EGALo	0 {640 x 200}	ATT400Hi	5 {640 x 400}
EGAHi	1 {640 x 350}	VGALo	0 {640 x 200}
EGA64Lo	0 {640 x 200}	MCGAMed	4 {640 x 200}
EGA64Hi	1 {640 x 350}	MCGAHi	5 {640 x 480}
EGAMonoHi	3 {640 x 350}	ATT400C0	0 {320 x 200}
HercMonoHi	0 {720 x 348}	ATT400C1	1 {320 x 200}
VGALo	0 {640 x 200}	ATT400C2	2 {320 x 200}
VGAMed	1 {640 x 350}	ATT400C3	3 {320 x 200}
VGAHi	2 {640 x 480}	ATT400Med	4 {640 x 200}
PC3270Hi	0 {720 x 350}	ATT400Hi	5 {640 x 400}

Example 1:

Uses Graph;

Var

Gd : Integer;

Gm : Integer;

Begin

Gd := Detect;

InitGraph(Gd , Gm);

SetFillStyle(1,9)

Bar(0,0,640,480);

OutTextXY(200,200,'Selamat Anda memasuki Mode GRAFIK');

Readln;

CloseGraph;

End.

Contoh 13.1 Menginialisasi Grafik.**ERROR PADA GRAFIK**

Pada saat kita masuk ke layar grafik bisa saja akan terjadi error. Untuk mendeteksi error yang terjadi maka kita menggunakan statemen **GraphResult**. Adapun bentuk penggunaannya yaitu :

Example 2:

Uses GRAPH ;

Var

Gd,

Gm,

ErrCode: Integer;

Begin

Gd := Detect;

InitGraph(Gd , gM , ' ');

```

    ErrCode := GraphResult;
    If ErrCode = grOK then
    Begin { Do graphics }
        SetFillStyle(1,7) ;
        Bar(0,0,640,480) ;
        Readln;
        CloseGraph;
    End
    Else
        Writeln('Graphics error  :', GraphErrorMsg(ErrCode));
    end.

```

Contoh 13.2 Mendeteksi Error pada Grafik.

GraphResult digunakan untuk mendeteksi kesalahan yang terjadi pada saat menginialisasi grafik. **grOK** digunakan jika Grafik tidak terjadi kesalahan, jika terjadi kesalahan maka koleksi (jenis-jenis kesalahan) akan ditampilkan dengan statemen **GraphErrorMsg**.

MENUTUP GRAFIK

Jika kita ingin kembali ke layar mode TEXT maka sebelum mode Text dibangkitkan kita harus menulis perintah **CloseGraph**.

BERPINDAH KE MODE GRAFIK KE MODE TEKS & SEBALIKNYA

Untuk berpindah ke mode grafik ke mode teks ataupun dari mode teks ke mode grafik kita menggunakan statemen

```

    ▪ RestoreCrtMode
    ▪ SetGraphMode
USES GRAPH;
VAR
    Gd, Gm      : Integer ;
BEGIN
    Gd := Detect ;
    InitGraph (Gd, Gm, '');
    { Program untuk menampilkan Grafik.....}
    ...
    ...
    { Kembali ke Mode Teks ...}
    RestoreCrtMode ;
    ...
    ...
    { Kembali lagi ke mode Grafik ...}
    SetGraphMode (Gm) ;
    ...
END.

```

PROSEDUR STANDAR GRAFIK

Dibawah ini merupakan perintah-perintah standar dari penggunaan Grafik, yaitu :

- **INITGRAPH** → Untuk menginialisasi Grafik
- **SETFILLSTYLE** → untuk membuat arsiran beserta warna kotak. Perintah ini dikombinasikan dengan Statemen BAR.
Bentuk Umum : SetFillStyle (Arsir, Warna)

Arsir bisa bernilai antara 1 s/d 10, setiap nilai akan menghasilkan bentuk arsir yang berbeda.

- **BAR** → untuk membuat Kotak tanpa garis pinggirnya.
Bentuk Umum : BAR (col, bar, col2, bar2)

Col, Bar



Col2, Bar2

Example 3:

```
Uses Graph;
Var
  Gd  : Integer;
  Gm  : Integer;
  i    : Byte  ;

Begin
  Gd  := Detect;
  InitGraph(Gd , Gm , ' ');
  For i := 1 to 10 Do
  Begin
    SetFillStyle( i, i+2) ;
    Bar(0,0,640,480);
    Readln;
  End;

  CloseGraph;
End.
```

Contoh 13.3 Contoh Penggunaan Arsiran pada BAR.

- **BAR3D** → digunakan untuk membuat kotak berbentuk 3 Dimensi .
Bentuk Umum : Bar3D (col, bar, col1, bar1, tebal, True / False)

True / False disini berguna untuk memunculkan bagian atas Bar3D, sedangkan Tebal berguna untuk membuat Ketebalan pada Grafik tersebut .

Example 4:

```
Uses Graph;
Var
  Gd  : Integer;
  Gm  : Integer;

Begin
  Gd  := Detect;
  InitGraph(Gd , Gm , ' ');
  SetFillStyle(1,4) ;
  Bar3D(100,100,200,380, 15, TRUE ) ;
  Readln;
  CloseGraph;
End.
```

Contoh 13.4 Menginialisasi Grafik.

- **SETCOLOR** → untuk pengaturan warna pada Rectangle & Line.
Bentuk Umum : SetColor (warna)
- **RECTANGLE** → untuk membuat suatu media Kotak tanpa Arsiran
Bentuk Umum : Rectangle (col, bar, col1, bar1)
- **LINE** → untuk membuat garis
Bentuk Umum : Line (col, bar, col1, bar1)

Example 5:

```

Uses Graph;
Var
  Gd  : Integer;
  Gm  : Integer;
Begin
  Gd  := Detect;
  InitGraph(Gd , Gm , ' ');
  SetFillStyle(1,4) ;
  Bar( 100,100,200,380 ) ;
  SetColor (15) ;
  Rectangle ( 150,100,300,280 ) ;
  Line ( 150,100,300,280 ) ;
  SetColor (14 ) ;
  Line ( 0,0, 640, 480 ) ;

  Readln;
  CloseGraph;
End.

```

Contoh 13.5 menggunakan perintah Line & Rectangle

- **SETTEXTSTYLE** → Untuk mengatur Tulisan pada media Grafik
Bentuk Umum : SetTextStyle (bentukHuruf, Horiz / Vert , SizeHuruf)

Jumlah bentuk huruf yang dimiliki oleh pascal sejumlah 11 jenis Huruf.
- **OUTTEXTXY** → untuk menampilkan tulisan / karakter pada media Grafik
Bentuk Umum : OutTextXY (col, bar, Tulisan) ;

Example 6:

```

Uses Graph;
Var
  Gd  : Integer;
  Gm  : Integer;
Begin
  Gd  := Detect;
  InitGraph(Gd , Gm , ' ');
  SetFillStyle(1,9) ;
  Bar(0,0,640,480) ;
  SetTextStyle ( 4, 0, 3 ) ;
  OutTextXY(200, 200,'Selamat Anda memasuki Mode GRAFIK') ;
  Readln;
  CloseGraph;
End.

```

End.

Contoh 12.6 Menampilkan Tulisan pada Grafik.

Example 7:

Uses Graph;

Var

Gd : Integer;

Gm : Integer;

i Byte ;

Begin

Gd := Detect;

InitGraph(Gd , Gm , ' ');

SetFillStyle(1,9) ;

Bar(0,0,640,480) ;

For i := 1 to 10 Do

Begin

SetTextStyle (i + 1, 0, 3) ;

OutTextXY(25 * i, 25 * i, ' Mesran, S.Kom ') ;

End;

Readln;

CloseGraph;

End.

Contoh 13.7 Menampilkan Tulisan pada Grafik.

- **CIRCLE** → untuk membuat lingkaran
Bentuk Umum : Circle (x, y , Radius / Tebal Lingkaran) ;

Example 8 :

USES Graph;

VAR

Gd, Gm : Integer;

Radius : Integer;

BEGIN

Gd := Detect;

InitGraph(Gd, Gm, ' ');

If GraphResult <> grOk then Halt(1) ;

For Radius := 1 to 10 do

Begin

SetColor(Radius) ;

Circle(Round(GETMAXX / 2), Round(GETMAXY / 2), Radius * 15);

End;

For Radius := 1 to 5 do

Circle(100, 100, Radius * 10) ;

For Radius := 1 to 5 do

Circle(550, 420, Radius * 10) ;

Readln;

CloseGraph;

END.

Contoh 13.8 Menampilkan lingkaran.

- **ELLIPSE** → untuk membuat lingkaran dan lingkaran yang terbuat bisa diatur misalnya hanya setengah, radius / tebalnya (bulat / bulat telur)
Bentuk Umum : Ellipse (x, y, StartAwal, StartAkhir, RadiusX, RadiusY);

Example 9 :

```

USES Graph;
VAR Gd, Gm : Integer;
BEGIN
  Gd := Detect;
  InitGraph(Gd, Gm, ' ');
  Ellipse(100, 100, 180, 360, 30, 50);
  Ellipse(100, 100, 0, 180, 50, 30);
  Readln;
  CloseGraph;
END.

```

Contoh 13.9 Menampilkan lingkaran.

- **PIESLICE** → untuk membuat lingkaran dengan arsiran dan derajat lingkaran bisa diatur.
Bentuk Umum : PieSlice (x, y, StartAwal, StartAkhir, Radius);

Example 10 :

```

USES Graph;
CONST Radius = 20;
VAR Gd, Gm : Integer;
BEGIN
  Gd := Detect;
  InitGraph(Gd, Gm, ' ');
  PieSlice(100, 50, 5, 355, Radius);
  PieSlice(150, 50, 20, 330, Radius);
  PieSlice(200, 50, 40, 310, Radius);
  PieSlice(200, 50, 60, 280, Radius);
  Readln;
  CloseGraph;
END.

```

Contoh 13.10 Menampilkan lingkaran dengan arsiran.

- **FILLPOLY** → untuk menampilkan suatu bentuk segitiga yang mudah diatur .
Bentuk Umum : FillPoly (NumPoint Div SizeOf(PointType), NumPoint);

Example 11 :

```

USES Graph;
CONST
  Triangle : Array[1..3] of PointType = ((X: 100; Y: 100),
    (X: 150; Y: 100), (X: 125; Y: 150));

  Triangle2 : Array[1..3] of PointType = ((X: 150; Y: 100),
    (X: 175; Y: 150), (X: 125; Y: 150));

  Triangle3 : Array[1..3] of PointType = ((X: 175; Y: 150),
    (X: 150; Y: 200), (X: 125; Y: 150));

  Triangle4 : Array[1..3] of PointType = ((X: 100; Y: 200),
    (X: 150; Y: 200), (X: 125; Y: 150));

```

```
Triangle5 : Array[1..3] of PointType = ((X: 75; Y: 150),
(X: 100; Y: 200), (X: 125; Y: 150));
```

```
VAR Gd, Gm: Integer;
BEGIN
  Gd := Detect;
  InitGraph(Gd, Gm, ' ');
  Setfillstyle (1,9) ; Setcolor(1);
  FillPoly( SizeOf(Triangle) div SizeOf(PointType), Triangle);

  Setfillstyle (1,1) ; Setcolor(9);
  FillPoly( SizeOf(Triangle2) div SizeOf(PointType), Triangle2);

  Setfillstyle (1,15) ; Setcolor(1);
  FillPoly( SizeOf(Triangle3) div SizeOf(PointType), Triangle3);

  Setfillstyle (1,7) ; Setcolor(2);
  FillPoly( SizeOf(Triangle4) div SizeOf(PointType), Triangle4);

  Setfillstyle (1,2) ; Setcolor(1);
  FillPoly( SizeOf(Triangle5) div SizeOf(PointType), Triangle5);

  Readln;
  CloseGraph;
END.
```

Contoh 13.11 Menampilkan suatu bentuk segitiga yang dapat diatur.

- **SECTOR** → sama dengan perintah PieSlice.

Bentuk Umum : Sector (x, y, StartAwal, StartAkhir, Radius);

Example 12 :

```
USES Graph;
VAR
  Gd , Gm : Integer;
  Xasp, Yasp: Word;
BEGIN
  Gd := Detect;
  InitGraph(gd , gm , ' ');
  SetFillStyle (1,9) ; SetColor (2) ;
  Sector (100,100, 10,350, 50, 50 ) ;

  SetFillStyle (1,4) ; SetColor (1) ;
  Sector (200,200, 50,300, 34, 34 ) ;

  SetFillStyle (1,4) ; SetColor (1) ;
  Sector (400,200, 70,290, 40, 40 ) ;

  SetFillStyle (1,15) ; SetColor (3) ;
  Sector (500,100, 50,300, 20, 20 ) ;
  Readln;
  CloseGraph;
end.
```

Contoh 13.12 Menampilkan lingkaran.

- **CLOSEGRAPH** → untuk menutup Grafik

MENYIMPAN & MENAMPILKAN GRAFIK

Grafik yang tampil dilayar adakalanya kita simpan ke memori dan akan ditampilkan kembali ke Layar untuk keperluan tertentu. Adapun perintah untuk menyimpan & menampilkannya yaitu :

- **ImageSize** → mengambil ukuran / size yang akan disimpan ke pointer
- **GetMem** → mengambil penempatan / alokasi ke memori
- **GetImage** → mengambil gambar yang akan disimpan
- **PutImage** → menampilkan kembali gambar yang telah disimpan.

Example 13 :

```

USES Graph;
VAR
  Gd, Gm : Integer;
  P      : Pointer;
  Size   : Word;
BEGIN
  Gd := Detect;
  InitGraph(Gd, Gm, ' ');
  SetFillStyle(1,3) ; Bar(0, 0, GetMaxX, GetMaxY);
  SetFillStyle(1,2) ; Sector (100,100,40,300, 30,30) ;

  Size := ImageSize(70, 70, 130, 130);
  GetMem(P, Size);
  GetImage(70, 70, 130, 130, P^);

  Readln;
  ClearDevice;
  SetFillStyle(1,3) ;
  Bar(0, 0, GetMaxX, GetMaxY);
  PutImage(300, 100, P^, NormalPut);

  Readln;
  CloseGraph;
END.

```

Contoh 13.13 : Menyimpan & Menampilkan Grafik ke layar

MENAMPILKAN TEXT3D

Text yang ditampilkan ke layar masih kurus dan tidak terlalu bagus untuk dilihat. Agar text yang ditampilkan ke layer berkualitas baik, maka penampilan harus menggunakan sebuah procedure.

Example 14 :

```

USES GRAPH ;
Var
  Gd, Gm : Integer ;

  Procedure Print3D(x1,y1:integer ;SizeT:Byte; text:string; clrD,ClrB,Style:byte);
  Begin
    SetTextStyle(Style,0,SizeT) ; SetColor(ClrD) ;

```



```

        OutTextxy(x1,y1,Text) ; SetColor(ClrB) ;
        OutTextxy(x1+1,y1,Text) ; OutTextxy(x1+2,y1,Text) ;
    End;

    Procedure OpenGraph ;
    Begin
        gd := Detect ;
        Initgraph (Gd, Gm, 'c:\TP\BGI\') ;
    End;

    BEGIN
        OpenGraph ;
        SetFillStyle(1,3) ; Bar (0, 0, 640, 480) ;
        SetTextStyle (2,0,6) ; OutTextXY (100,20,'Tanpa Procedure') ;

        Print3D( 100,50, 6, 'Sekolah Tinggi Manajemen Informatika Komputer',7,1, 2 );
        Print3D( 100,70, 1, 'Sekolah Tinggi Manajemen Informatika Komputer',7,0, 7 );

        Readln ; CloseGraph;
    END.

```

Contoh 13.14 : Menampilkan Text berbentuk 3D

MEMBUAT TOMBOL BUTTON, BUTTON MASUK & BUTTON YANG TERSOROT

Untuk membuat suatu Tombol yang berbentuk Visual harus menggunakan suatu procedure dan memakai beberapa statemen sekaligus. Sehingga akan lebih efisien dengan menggunakan procedure

Example 15 :

```

    USES GRAPH ;
    Var
        Gd, Gm : Integer ;

    Procedure OpenGraph ;
    Begin
        gd := Detect ; Initgraph (Gd, Gm, 'c:\TP\BGI\') ;
    End;

    Procedure BoxOn(X1,Y1,X2,Y2 :Integer ; Text: String ; Posisi : Byte) ;
    Var Hor, Ver : Real ;
    Begin
        setlinestyle(0,1,0) ; setfillstyle(1,7) ;
        Bar(X1,Y1,X2,Y2) ;
        SetColor(15) ;
        Line(x1+1,y1+1,x1+1,y2-1) ; Line(x1+1,y1+1,x2-1,y1+1) ;
        SetColor(0) ;
        Line(x2,y1+1,x2,y2) ; Line(x1+1,y2,x2-1,y2) ;
        SetColor(8) ;
        Line(x1+2,y2-1,x2-1,y2-1) ;
        hor := ( x2-x1 ) / 2 + x1 - ( ( length(text) * 8 ) / 2 ) ; ver := ( ( y2-y1 ) / 2 ) + y1-6;
        SetTextStyle(2,0,4) ; SetColor(0) ;
        If length(text) < 5 Then Hor:=Hor-3 ;
        OutTextxy(round(hor)+8,round(ver),Text) ;
        If Posisi <> 0 Then OutTextxy(round(Hor)+1+(6*Posisi),round(Ver)+4,'Ä');
    End;

```

```

Procedure BoxSorot(X1,Y1,X2,Y2:Integer) ;
Begin
  SetColor(0) ; Rectangle( X1,Y1,X2,Y2) ;
  SetColor(8) ; Line(X2-1,Y1+2,X2-1,Y2) ;
  SetLineStyle(1,0,1) ;
  SetColor(0) ; Rectangle(x1+8,y1+3,x2-8,y2-3) ;
  setlinestyle(0,1,0) ;
End;

```

```

Procedure BoxOFF(X1,Y1,X2,Y2 :Integer;Text:String;Kon:Boolean);
var
  hor, ver : real ;
Begin
  dec(y1,2) ;
  setlinestyle(0,0,2) ;
  setfillstyle(1,7) ;
  Bar(X1+1,Y1+2,X2,Y2) ;
  SetColor(0) ;
  Rectangle(X1+1,Y1+2,X2,Y2);
  SetColor(8) ;
  Rectangle(X1,Y1+3,X2,Y2) ;
  SetColor(15) ;
  Line(X2,Y1+2,X2,Y2) ;
  Line(X1+2,Y2,X2,Y2) ;
  hor :=((x2-x1)/2)+x1-(length(text)*8)/2; ver :=((y2-y1)/2)+y1+2;
  SetTextStyle(2,0,4) ;
  OutTextxy(round(hor)+8,round(ver)-6,Text) ;
  If Kon = True Then
  Begin
    SetLineStyle(1,0,1);
    Rectangle(X1+6,Y1+6,X2-5,Y2-3);
  End;
  Sound(1150);Delay(200);NoSound;
End;

```

```

BEGIN
  OpenGraph ;
  SetFillStyle(1,3) ; Bar (0, 0, 640, 480) ;

  BoxON (100,300,160,320,'Simpan',1) ;
  BoxOFF (100,300,160,320,'Simpan',True) ;
  BoxON (162,300,224,320,'Edit',1) ;
  BoxON (226,300,286,320,'Hapus',1) ;
  BoxON (288,300,348,320,'Ok',1) ;
  BoxON (350,300,410,320,'Batal',1) ;

  BoxSorot (350,300,410,320) ;

  Readln ;
  CloseGraph;

```

END.

Contoh 13.15 : Button 3D, Button Tersorot & Button terClick

MEMBUAT TEXTBOX

Procedure di bawah ini berguna untuk menampilkan textbox tempat kita mengisi data pada media grafik.

Example 16 :

```
USES GRAPH ;
```

```
Var
```

```
Gd, Gm : Integer ;
```

```
Procedure OpenGraph ;
```

```
Begin
```

```
gd := Detect ; Initgraph (Gd, Gm, 'c:\TP\BGI\') ;
```

```
End;
```

```
Procedure TextBox(X1,Y1,X2,Y2 :Integer;style1,InClr:Byte);
```

```
Begin
```

```
setlinestyle(0,0,1) ;
```

```
setcolor(15) ;
```

```
Rectangle(X1-1,Y1-1,X2+2,Y2+2);
```

```
Setfillstyle(Style1,InClr) ;
```

```
Bar(X1,Y1,X2,Y2) ;
```

```
setcolor(8) ;
```

```
line(X1-2,Y1-1,X1-2,Y2+1) ;
```

```
line(X1-1,Y1-1,X1-1,Y2+1) ;
```

```
line(X1-2,Y1-2,X2+2,Y1-2) ;
```

```
line(X1-1,Y1-1,X2+1,Y1-1) ;
```

```
Delay(50) ;
```

```
End;
```

```
BEGIN
```

```
OpenGraph ;
```

```
SetFillStyle(1,3) ; Bar(0, 0, 640, 480) ;
```

```
Print3D(30,120,5,'Kode OBAT ',7,0,2) ;
```

```
TextBox(140,121,190,136,1,15);
```

```
Readln ;
```

```
CloseGraph;
```

```
END.
```

Contoh 13.16 : Textbox untuk media penginputan data

MEMBUAT FUNGSI INPUT DATA PADA MEDIA GRAFIK

Prosedur standar Read & Readln tidak bisa digunakan pada media grafik sehingga jika kita ingin menginputkan data di media grafik kita harus membuat prosedur khusus untuk menangani masalah input data.

Pada prinsipnya adalah kita selalu mendeteksi setiap penekanan keyboard sejumlah perulangan tertentu pada saat kita melakukan penginputan. Hasil dari input ini tentunya setiap karakter yang kita tekan akan dirangkai dan dikembalikan ke program utama.

Example 17 :

```
Function _ReadData(yy, xx, Loop:Integer ; Kon : Boolean) : String ;
```

```
Var
```

```
Ch : Char ;
```

```
Posisi : Integer ;
```

```

Awal      : Integer ;
Panjang   : Integer ;
Huruf     : integer ;
Looping   : Integer ;
Xhuruf    : Array[1..35] Of String ;
Kalimat,
kata      : String[80] ;

```

Procedure kursor(Colom,Baris:Integer;Status:Boolean);

```

Begin
  If Status=True Then
    Begin
      Setfillstyle(1,0) ;
      Bar(Colom,Baris,Colom,Baris+11);
      Setfillstyle(1,15) ;
      Bar(Colom+1,Baris,Colom+textwidth('h'),Baris+11);
    End
  Else
    Begin
      Setfillstyle(1,15) ;
      Bar(Colom,Baris,Colom+Textwidth('H'),Baris+11);
    End;
End;

```

Procedure Kedip;

```

Begin
  Repeat
    Kursor(yy, xx, True) ; Delay(250) ;
    Kursor(yy, xx, False) ; Delay(250) ;
  Until Keypressed;
End;

```

```

Begin
  Looping := 0 ; Posisi := 1 ;
  Panjang := 0 ; Kalimat := " ;

  For Awal := 1 To 15 Do Xhuruf[Awal] := " ;

```

```

Repeat
  If Not KeyPressed Then Kedip;
  Ch := Readkey ;
  Case ch Of
    'A'..'Z','a'..'z','0'..'9',' ','-','.',':','\','/',' ','&','@','!','%','*','^':
      Begin
        Settextstyle(Smallfont,0,4) ;
        Setlinestyle(0,0,1) ;
        Kursor(yy, xx, False) ;
        SetColor(0) ;
        If Kon Then Ch :=Ucase(Ch) ;
      End

```

```

        OutTextxy(yy, xx, ch)          ;
        Cursor(yy+TextWidth(Ch), xx, True);
        inc(yy, TextWidth(Ch))         ;
        Inc(Posisi)                    ;
        Xhuruf[Posisi] := Ch           ;
        Inc(Looping)                  ;
    End;

    Begin
    If Posisi > 1 Then
    Begin
        Dec(Posisi) ; Cursor(yy, xx, False) ; SetColor(0) ;
        OutTextxy(yy-TextWidth(Xhuruf[Posisi+1]),xx, Kata[Posisi]) ;
        Cursor(yy-TextWidth(Xhuruf[Posisi+1]),xx, True) ;
        Dec(Huruf,TextWidth(Xhuruf[Posisi+1])) ;
        Dec(yy, TextWidth(Xhuruf[Posisi+1])) ;
        Dec(Looping) ; Kata := ' ' ;

        For Awal := 1 To Posisi Do
            Kata := Kata + Xhuruf[Awal] ;
        End;
    End;
    End;
    Kata := "";

    For Awal := 1 To Posisi Do
        Kata := Kata + XHuruf[Awal] ;

    Until (Loop = Looping) Or (Ch = #13) Or (Ch = #27) ;

    For Awal := 1 To Posisi Do
        Kalimat := Kalimat + XHuruf[Awal] ;
    Cursor(yy, xx, False) ;
    ReadData := Kalimat ;
    End;

```

Contoh 13.17 : Contoh fungsi Input data pada media grafik.

MEMBUAT FORM PENGINPUTAN DATA

Program terakhir ini adalah suatu media form yang diperlukan pada saat kita mau membuat form entri data.

Example 18 :

```

USES GRAPH ;
Var
    Gd, Gm : Integer ;
    Kode   : String[6] ;

```

```

.....
.....
.....

```

```

Procedure FORM(x1, y1, x2, y2 : Integer ; Txt : String) ;
Begin
  BoxON(x1,y1,x2,y2,"0) ;
  SetFillStyle(1,1) ;
  Bar(x1+2,y1+3,x2-2,y1+22) ;
  Print3d(x1+20,y1+2,6, txt ,1,14,2) ;
End;

```

Procedure FormMHS;

```

Begin
  Form(10,90,370,275,'Entri Data OBAT');
  print3d(30,120,5,'Kode',7,0,2) ;
  Print3d(30,145,5,'Nama',7,0,2) ;
  Print3d(30,170,5,'Satuan',7,0,2) ;
  Print3d(30,195,5,'Harga Jual',7,0,2) ;
  Print3d(30,220,5,'Harga Beli',7,0,2) ;

  TextBox(130,121,170,136,1,15);
  TextBox(130,147,340,161,1,15);
  TextBox(130,172,230,186,1,15);
  TextBox(130,197,200,211,1,15);
  TextBox(130,222,200,236,1,15);

  BoxON (25,248,90,268,'New',1) ;
  BoxOFF(25,248,90,268,'New',True) ;
  BoxON (91,248,157,268,'Save',1) ;
  BoxON (158,248,223,268,'Delete',1) ;
  BoxON (224,248,289,268,'Cancel',1) ;
  BoxON (290,248,355,268,'Close',1) ;
End;

```

BEGIN

```

OpenGraph ; SetFillStyle(1,3) ; Bar (0, 0, 640, 480) ;
FormMHS;
Kode := InStr(132,122, 6, False) ;
Readln ;
CloseGraph;

```

END.

Contoh 13.18 : Contoh fungsi Input data pada media grafik.
Hasil output jika program diatas dijalankan yaitu :