2013

MODUL

PEMROGRAMAN PASCAL

Sejarah Pascal – Struktur – Branching – Pengulangan – Himpunan – Record – File - Array

Teknologi Informasi dan Komunikasi
2013

Pembahasan 1
Pengenalan Pascal

Bahasa Pascal merupakan bahasa yang sangat populer di dunia, terlebih karena digunakan sebagai bahasa untuk mengajarkan pemrograman terstruktur bagi para SISWA
di pelbagai perguruan tinggi, baik di Indonesia maupun negara lainnya. Sebagai bahasa pemrograman yang terstruktur, bahasa ini menyediakan sejumlah kontrol (berupa REPEAT, WHILE dan FOR) serta mendukung pembuatan program yang tersusun atas sejumlah blokblok
kecil (yang dikenal dengan sebutan rutin pada berbagai bahasa pemrograman). Blok-blok kecil ini dapat dipakai untuk membentuk blok yang lebih besar, dan secara keseluruhan membentuk program kerja. Keuntungan dari pemrograman terstruktur, program mudah dibuat, dipahami dan dikembangkan.
Pencipta bahasa Pascal adalah Niklaus Wirth (EidgenOssische Technishe Hochshule, Zurich, Swiss). Nama Pascal sendiri diambil untuk mengenang dan mengabadikan seorang matematikawan Perancis bernama Blaise Pascal (1623-1662), penemu mesin penjumlah mekanis yang pertama kali.
Bahasa Pascal lahir sebagai hasil pengembangan dari bahasa ALGOL-60. Versi pertama Pascal muncul pada tahun 1970 dan laporan pertama mengenai bahasa ini baru dipublikasikan pada tahun 1971. Pada tahun 1974, buku berjudul *Pascal User Manual and Report* dipublikasikan. Buku ini ditulis oleh Niklaus Wirth dan Kathleen Jensen.
Kepopuleran Pascal berkembang seiring dengan berkembangnya PC. Pada platform ini terdapat sejumlah kompiler Pascal. Salah satunya Turbo Pascal, yang dikeluarkan oleh *Borland International*. Turbo Pascal telah mengalami perubahan semenjak versinya yang
pertama. Tercatat pada tahun 1992, Turbo Pascal For Windows telah dirilis.
Yang menarik, Turbo Pascal bukanlah sekedar kompiler. Turbo Pascal menyediakan suatu lingkungan terpadu (dikenal dengan sebutan IDE, *Integrated Development Environment)*, yang diantaranya berisi :

- Kompiler (untuk mengkompilasi program menjadi suatu objek yang dipahami oleh komputer)
- Debugger (untuk mencari kesalahan program)
- Linker (untuk menggabungkan file objek dan file pustaka serta membentuk file *executable*)
- Editor (untuk menulis program sumber)

Kompiler (+Linker) yang tersedia dapat dipakai untuk mengkompilasi program menjadi file yang *executable* (dapat dieksekusi secara langsung dari prompt DOS). Namun demikian pemrogram juga dapat memilih agar kompilasi dilakukan ke memori, sehingga pembuatan dan pengujian program dapat dilakukan dengan cepat. Hal ini sangat bermanfaat pada masa pembuatan/pemodifikasian program.
Sebagai perluasan dari pengembangan program secara modular (tersusun atas sejumlah modul), Turbo Pascal juga memungkinkan pemrogram untuk membuat suatu pustaka yang berisi sejumlah prosedur, fungsi ataupun elemen-elemen yang lain, yang dikompilasi secara
terpisah. Kemudian pustaka-pustaka tersebut dapat dipergunakan oleh suatu program, tanpa perlu melibatkan kode/program sumbernya. Modul seperti ini biasa disebut unit.

Semenjak versi 5.5, Turbo Pascal menyediakan kemampuan pemrograman berorientasiobjek (PBO). Pemrograman berorientasi objek yang terkenal dengan sebutan OOP *(Object-Oriented Programming)* merupakan suatu metode pemrograman yang akhir-akhir ini sangat populer. OOP atau PBO

mengkombinasikan data dan fungsi atau prosedur yang mengakses data menjadi suatu wadah yang disebut objek. Metode pemrograman ini dirasakan dapat mengatasi kesulitan yang timbul pada pemrograman yang besar dan kompleks, sehingga dapat meningkatkan produktivitas pemrogram.
Turbo Pascal merupakan bahasa pemrograman yang serbaguna. Anda dapat menggunakannya untuk tujuan komputasi, pembuatan permainan *(game)* ataupun masalah masalah

Struktur Bahasa Pascal

- Judul
  Program <judulnya>;
  Uses <unitnya>;
- Bagian Deklarasi
  Label <namalabelnya>;
  Const <nama=nilai constantanya>;
  Type <namatype:typedatanya>;
  Var <nama_variabelnya>;
  Procedure <nama_prosedurnya>;
  Function <nama_fungsinya>;
- Bagian Pernyataan
  Begin
  <pernyataan-pernyataan>;
  End.

Contoh Program

Listing Programnya

```
Uses WinCrt;
Begin
        WriteLn( 'Saya' );
        Write( 'Belajar' );
        WriteLn( 'Turbo' );
        Write( 'Pascal' );
        Write( 'For Windows' );
End.
```

Tekan Ctrl F9 untuk ekseskusi program
Tekan F5 untuk melihat hasilnya

Hasilnya :
Saya
Belajar Turbo
Pascal For Windows

Tugas:
Buatlah Program cetak dilayar Biodata diri anda

Pembahasan 2
PERINTAH DASAR – IDENTIFIER - DEKLARASI

Perintah Dasar/Sederhana Pascal

- Write ('Text/tulisan',Variabel);
  Perintah untuk menampilkan atau cetak dilayar monitor tanpa pindah baris
- WriteLn('Text/Tulisan',Variabel);
  Perintah untuk menampilkan/cetak dilayar monitor lalu pindah baris kebawah
- Read(Variabel);
  Perintah untuk menginput/mengisi data tanpa pindah baris
- ReadLn(Variabel);
  Perintah untuk menginput/mengisi data lalu pindah baris

Identifier/Pengenal Data

Digunakan untuk penamaan elemen-elemen deklarasi seperti Label, Constanta, type, variabel, procedure, function

Syarat Identifier

- Diawali huruf
- Tidak boleh ada spasi/blank
- Tidak boleh menggunakan reserved word
- Tidak boleh menggunakan simbol khusus, kecuali underscore(tanda bawah)
- Panjang maximal 63 character

Contoh :
Luas_Segi_Tiga
LuasSegiTiga
LuasS3
LS3

Deklarasi Variabel
Adalah elemen data yang belum memiliki nilai dan nilainya dapat berubah ubah
Contoh :

```
USES WINCRT;
VAR Nama : String[25];
      Nilai   : Integer;
BEGIN
    CLRSCR;
    WRITE( 'Nama Siswa : ' );
    READLN(Nama);
    WRITE( 'Nilai Ujian    : ' );
    READLN(Nilai);
END.
```

Ctrl F9 lalu isi datanya
Nama Siswa : ...
Nilai Ujian : ...
Ctrl F5 lihat hasilnya

Deklarasi Constanta
Adalah suatu elemen data yang sudah ada nilainya dan nilainya tetap
Contoh :

```
USES WINCRT;
CONST Namasis= 'Ujang Wahyudi';
        IPK     = 3.35;
        Mutu    = 'A'
BEGIN
   CLRSCR;
   WRITE( 'Nama Siswa : ' , Namasis);
   WRITE( 'Nilai IPK      : ' , IPK);
   WRITE( 'Mutu Ujian   : ' , Mutu);
END.
```

Ctrl F9 lalu Alt F5
Nama Siswa : Ujang Wahyudi;
Nilai IPK : 3.350000E+0
Mutu Ujian : A

Pembahasan 3
PENGUNAAN USES WINCRT – TIPE SEDERHANA - OPERATOR

Penggunaan Unit WINCRT (Cathoda Ray Tube)

- CLRSCR ;
  Untuk membersihkan layar
- Delay (nilainya);
  Untuk menghentikan program dalam satuan milisecond
- Gotoxy(x,y);
  Untuk menempatkan kursor diposisi x dan y
- Window(x1,y1,x2,y2)
  Untuk membuat batasan jendela pada layar

Tipe Data Bahasa Pascal

- Bilangan Bulat [integer] Terdiri dari

Tipe	Jangkauan nilainya
Shortint	-128 .. 127
Byte	0 .. 255
Integer	-32768 .. 32767
Word	0 .. 65535
Longint	-21747483648 .. 21747483647

- Bilangan Pecahan [real] terdiri dari single, real, double, extended
- Character → satu huruf/simbol saja
- String → lebih dari satu huruf/simbol
- Boolean → data berisi True/False

Operator-Operator Pascal

- Assignment Operator (Operator Pengerjaan) → :=
  Contoh A:=5 Nama:=Budi
- Binari Operator → * / + - DIV MOD
- Unary Operator (Operator Tunggal) →
  Contoh -5 +2.5
- Bitwise Operator → NOT, AND, OR, XOR, SHL, SHR
- Relasi Operator (Operator Relasi) → = <> > < <= >= IN
- Logical Operator (Operator Logika) → NOT, AND, OR, XOR
- Address Operator (Operator Alamat) → @ ^
- Set Operator (Operator Himpunan) → + - *
- String Operator → +
  Contoh Nama1:= ‘Syaiful Anwar’; Nama2:= ‘Anwar’;
  Nama3 := Nama1 + Nama2 ;
  Hasilnya Nama3 → Ujang Wahyudi

Contoh Program Tipe Data Sederhana

Nama Barang : ...
Jumlah Beli : ...
Harga Satuan : Rp. ...
Harga Barang : Rp. ___
Tipe A/B/C/D : ...
Discount : Rp. ___
Harga Bayar : Rp. ___
Uangnya : Rp. ...
Kembalinya : Rp. ___

Catatan :
Tanda ... berarti program diisi/input
Tanda ___ berarti program tampil/cetak

Listing Program Tipe Data Sederhana

```
Uses WinCrt;
Var Nb          : String[20];
    Hbr, U,Hs   : Longint;
    Jb          : Byte;
    Dis, Hby,K  : Real;
    Tipe        : Char;
Begin Clrscr;
  Write('Nama Barang     : '); Readln(Nb);
  Write('Jumlah Beli     : '); Readln(Jb);
  Write('Harga Satuan    : Rp. ');
Readln(Hs);
  Hbr:=Jb*Hs;
  Writeln('Harga Barang  : Rp. ',Hbr);
  Write  ('Tipe A/B      : '); Readln(Tipe);
  If Tipe='A' Then Dis:=5/100*Hbr
    Else Dis:=0;
  Writeln('Discount      : Rp. ',Dis:8:1);
  Hby:=Hbr-Dis;
  Writeln('Harga Bayar   : Rp. ',Hby:10:1);
  Write('Uangya          : Rp. '); Readln(U);
```

Cth. Program Operator

Listing Program Operator

```
USES WINCRT;
BEGIN
    CLRSCR;
    WRITELN(15=10);
    WRITELN(15>10);
    WRITELN(15 DIV 2);
    WRITELN(15 MOD 2);
    WRITELN(15/2);
    WRITELN(12 AND 23);
    WRITELN(12 OR 23);
    READLN;
END.
```

Hasilnya

```
False
True
7
1
7.5000000E+00
4
31
```

Tugas Program

1. Buatlah Program Luas Segi Tiga
2. Buatlah Program Luas dan Keliling Lingkaran
3. Buatlah Program Persegi Panjang

Pembahasan 4
DEKLARASI TIPE - DEKLARASI LABEL

Deklarasi Type

Pengenal (identifier) dari data yang digunakan harus diperkenalkan Tipenya. Jika ingin menggunakan tipe data dengan nama yang dibuat oleh pemakai, maka harus disebutkan tipe data standarnya.

Contoh tanpa tipe	Contoh dengan tipe
Var Nama : String[25]; Kelas : String[7]; Nilai : Integer; Grade : Char;	Type Nm : String[25]; Kls : String[7]; Nil : Integer; Gr : Char; Var Nama : Nm; Kelas : Kls; Nilai : Nil; Grade : Gr;

Deklarasi Label

Deklarasi label digunakan untuk melompati beberapa perintah program. Memiliki arah dan tujuan yaitu dengan menggunakan GOTO

Tugas :

```
PROGRAM DGN_LABEL;
USES WINCRT;
LABEL A,B,C;
BEGIN
  CLRSCR;
  WRITELN('TEXT1');  GOTO A;
  WRITELN('TEXT2');
A:WRITELN('TEXT3'); GOTO B;
  WRITELN('TEXT4');
B:WRITELN('TEXT5'); GOTO C;
  WRITELN('TEXT6');
C:READLN;
END.

Hasilnya
TEXT1
```

Contoh Tipe Dalam Program

```
USES WINCRT;
TYPE Pecahan = Real;
     Bulat   = Integer;
VAR  A,T : Bulat;
     LS3 : Pecahan;
BEGIN
  CLRSCR;
  WRITE('Masukkan Nilai Alas   : ');
  READLN(A);
  WRITE('Masukkan Nilai Tinggi : ');
  READLN(T);
  LS3:=(A*T)/2;
  WRITELN('Luas Segitiga:',LS3:8:1);
  READLN;
END.
```

Tugas Program

1. Rubah Program Labelnya dgn Layout sbb
   TEXT2 TEXT4 TEXT6
2. Buat luas dan Keliling lingkaran dengan menggunakan deklarasi type dan Var

Pembahasan 5
FORMAT TIPE DATA– LOKASI CURSOR

Tampilan Terformat

Untuk mengatur bentuk tampilan dari tampilan default ke bentuk yang diinginkan atau Batasan cetak

- Parameter Char:n
  Batasan cetak/ bentuk tampilan/ jarak tampil data char
- Parameter String:n
  Batasan cetak/ bentuk tampilan/ jarak tampil data string
- Parameter Boolean:n
  Batasan cetak/ bentuk tampilan/ jarak tampil data Boolean
- Parameter Integer:n
  Batasan cetak/ bentuk tampilan/ jarak tampil data Integer
- Parameter Real:n:m
  Batasan cetak/ bentuk tampilan/ jarak tampil data Real

```
Uses WinCrt;
Const Kampus='PPKM';      {Tipe String}
      Nrata=2.5;          {Tipe Real}
      Ntotal=55;          {Tipe Integer}
      Cek   =True;        {Tipe Boolean}
      Huruf ='A';         {Tipe Char}
BEGIN
  Clrscr;
  Writeln('Nama Kampus :',Kampus);
  Writeln('Nama Kampus :',Kampus:3);
  Writeln('Nama Kampus :',Kampus:6);
  Writeln('Nilai Rata  :',Nrata);
  Writeln('Nilai Rata  :',Nrata:3:0);
  Writeln('Nilai Rata  :',Nrata:6:1);
  Writeln('Nilai Total :',Ntotal:2);
  Writeln('Nilai Total :',Ntotal:5);
  Writeln('Cek Boolean :',Cek);
  Writeln('Cek Boolean :',Cek:7);
  Writeln('Huruf Abjad :',Huruf);
  Writeln('Huruf Abjad :',Huruf:3);
  Readln;
End.
```

Hasilnya :

```
Nama Kampus :PPKM
Nama Kampus :PPKM
Nama Kampus :   PPKM
Nilai Rata      :2.500000e+00
Nilai Rata      :3
Nilai Rata      :   2.5
Nilai Total     :55
Nilai Total     :   55
Cek Boolean :True
Cek Boolean :   True
Huruf Abjad :A
Huruf Abjad :  A
```

Contoh Program Lokasi Cursor

```
BEGIN
  CLRSCR;
  GOTOXY(10,2);
  WRITE('JUDUL INI TERCETAK DI KOLOM 10 BARIS 2');
  READLN;
END.
```

Pembahasan 6
BRANCHING IF TUNGGAL/MAJEMUK – CASE OF

Branching/ Penyeleksian Kondisi

Untuk menyeleksi kondisi-kondisi suatu proses makan diperlukan perintah IF. Adapun bentuk-bentuk IF sbb:

- Statement IF

  Bentuk Umum : If <Kondisi> Then <Statement>;

  contoh : IF N>=60 Then Ket:='Lulus';

  Bentuk Umum : If <Kondisi> Then <Statement1>
  Else <Statement2>;

  contoh : IF N>=60 Then Ket:='Lulus'
  Else Ket:='Gagal';

Nb : apabila dalam satu statement lebih dari 1 baris perintah maka statemen tsb dimasukkan blok

```
Begin
  Perintah1;
  Perintah2;
End;
```

Branching Majemuk

Bentuk Umum : If <Kondisi1> Then <Statement1>
Else If <Kondisi2> Then <Statement2>
Else <Statement3>;

contoh : IF (N>=84) and (N<100) Then Mutu:='Amat Baik'
Else IF N>74 Then Mutu:='Baik'
Else IF N>60 Then Mutu:='Cukup'
Else Mutu:='Kurang';

Statement Case…OF

Bentuk Umum :	Contohnya :
CASE <Variabel> OF Konstanta1 : <Statement1>; Konstanta2 : <Statement2>; Konstanta3 : <Statement3>; Konstanta4 : <Statement4>; END;	CASE N OF 85..100 : Mutu:='Amat Baik'; 75..84 : Mutu:='Baik' ; 60..74 : Mutu:='Cukup' 0..59 : Mutu:='Kurang'; END;

Pembahasan 7
LOOPING (PERULANGAN)

1. **Statement For....Do (Perulangan Counter)**
   B.U : For *variabel_kontrol* : = *nilai_awal* To *nilai_akhir* Do
   Statement ;
   B.U : For *variabel_kontrol*:= *nilai_awal* DownTo *nilai_akhir* Do
   Statement ;
2. **Statement While ... Do (Perulangan Kondisi Terpenuhi)**
   B.U : While *ungkapan_logika* Do
   Statement ;
3. **Statement Repaet...Until (Perulangan Kondisi Tidak Terpenuhi)**
   B.U : Repeat
   Statement ;
   Until *ungkapan_logika* ;

```
Program For_To ;

Uses WinCrt ;
Var
   i  : byte ;
begin
clrscr  ;
for  i : =  1 to 5 do
  writeln (‘ Pascal ‘) ;
readln ;
end .
```

```
Program For_DownTo;

Uses WinCrt ;
Var
   i  : byte ;
begin
clrscr  ;
for  i : =  5 downto 5 do
   writeln (‘ Pascal ‘) ;
readln ;
end .
```

```
Program While ;

Uses WinCrt ;
Var
   i  : byte ;
begin
clrscr  ;
      I : = 0 ;
 While I <= 5 Do
  begin
       writeln(I);
       I : = I +1 ;
   end ;
readln ;
end .
```

```
Program Repeat;

Uses WinCrt ;
Var
   i  : byte ;
begin
clrscr  ;
      I : = 1 ;
 Repeat
   writeln ( I ) ;
      I : = I +1;
Until I >= 5
readln ;
end .
```

Tugas :

1. Buat deret bilangan ganjil dan deret bilangan genap
2. Buatlah deret bilangan Fibonanci → 1 1 2 3 5 8 13 21 34 55
3. Buatlah deret

```
1 2 3 4 5      b.  1              c.          5      d.  1 2 3 4 5
1 2 3 4            1 2                      4 5            2 3 4 5
1 2 3              1 2 3                  3 4 5              3 4 5
1 2                1 2 3 4              2 3 4 5                4 5
1                  1 2 3 4 5          1 2 3 4 5                  5
```

4. Buatlah deret

```
1 . 2   1 . 4   1 . 6   1 . 8
2 . 2   2 . 4   2 . 6   2 . 8
```

Pembahasan 8
HIMPUNAN / SET

Himpunan termasuk salah satu tipe data terstruktur,yang terdiri dari sejumlah elemen yang bertipe data sama.
Pendefinisian tipe data himpunana diawali dengan kata tercadang **Set** dan **Of,**kemudian didikuti dengan tipe elemen.

Bentuk umum : **SET OF** tipe_elemen
Elemen Himpunan mempunyai bentuk eperti : [daftar_elemen] yang mana etiap elemen harus bertipe sama.

Operator hubungan dalam himpunan

Himp1 = Himp2 ⟶ Dua Himpunan Dikatakan sama jika jumlah elemen sama dan masing-masing elemen sama.Hasilnya True bila Himp1 sama dengan Himp2

Himp1 <> Himp2 ⟶ Dua Himpunan Dikatakan tidak sama jika jumlah elemen tidak sama dan masing-masing elemen tidak sama. Hasilnya True bila Himp1 tidak sama dengan Himp2

Himp1 <= Himp2 ⟶ Hasil bernilai True jika semua elemen dari Himp1 terdapat pada Himp2

Himp1 >= Himp2 ⟶ Hasil bernilai True jika semua elemen dari Himp2 terdapat pada Himp1

Elemen In Himp ⟶ Memeriksa elemen adalah anggota himpunan Himp atau tidak.Hasil true jika elemen adalah anggota dari Himp

Operator Operasi Himpunan

Himp1 * Himp2 ⟶ Operasi Interseksi himpunan.untuk memperoleh himpunan yang terdiri dari elemen yang merupakan anggota himpunan Himp1 dan sekaligus juga merupakan anggota Himp2

Himp1 + Himp2 ⟶ Operasi penggabungan himpunan.Untuk memperoleh himpunan yang terdiri dari gabungan elemen anggota Himp1 dan Himp2

Himp1 - Himp2 ⟶ Operasi selisih himpunan.Untuk memperoleh himpunan yang terdiri dari elemen anggota Himp1 yang bukan anggota himpunan Himp2

Contoh Ungkapan	Hasil
[1,2,3,4,5] * [1,3,5,7]	[1,3,5]
[1,2,3,4,5] * [6,7,8,9,10]	[]
[1,2,3] * []	[]
[1,2,3,4,5] + [6,7]	[1,2,3,4,5,6,7]
[1,2,4] + [1,2,3]	[1,2,3,4]
[1,2,3,4,5] - [1,2]	[3,4,5]
[1,2,3,4,5] - [4,5,6,7]	[1,2,3]
[4,5,6,7] - [1,2,3,4,5]	[4,6,7]

Contoh Program Himpunan

```
Program Himpunan;
Uses WinCrt;
Type    Hs = Set Of 1..9;
Var     H1,H2,H3 : Hs;
BEGIN
    CLRSCR;
    H1:=[1,2,3,4,5];
    H2:=[3,5,6,7,8];
    H3:=H1+H2;
    IF H3=[1..8] Then Writeln('H1+H2 Adalah [1..8]') Else Writeln('Salah');
    Readln;
    H3:=H1-H2;
    IF H3=[1,2,4] Then Writeln('H1-H2 Adalah [1,2,4]') Else Writeln('Salah');
    Readln;
    H3:=H1*H2;
    IF H3=[3,5] Then Writeln('H1*H2 Adalah [3,5]') Else Writeln('Salah');
    Readln;
END.
```

Pembahasan 9
ARRAY / LARIK

Array (larik) adalah tipe terstruktur yang terdiri dari sejumlah komponen yang mempuyai tipe sama.

B.U : Array [1 . . 20] of tipe_data ; (Dimensi 1)
B.U : Array [1 . . 3 , 1 . . 2] of tipe_data; (Dimensi 2)

Contoh Program Array :

```
Uses WinCrt ;
Var
    Nama                    : array [1..30 ] of string ;
    Tugas, uts, uas, absen  : array [1..30] of integer ;
    Rata                    : array [1..30] of real ;
    x , jd                  : byte;
BEGIN
    CLRSCR ;
    Write ( ' Masukan Jumlah data  : ' ) ; readln ( jd ) ;
    For x : = 1 to jd do
    begin
        Writeln ( ' ***************************** ' ) ;
        Writeln ( '  PENILAIAN UJIAN SISWA PPKM  ' ) ;
        Writeln ( ' ***************************** ' ) ;
        Write ( ' Input Nama                : ' ) ;  readln (nama[x] ) ;
        Write ( ' Input Nilai Tugas         : ' ) ;  readln (tugas[x] ) ;
        Write ( ' Input Nilai Absen         : ' ) ; readln (absen[x] ) ;
        Write ( ' Input Nilai UTS           : ' ) ; readln (uts[x] ) ;
        Write ( ' Input Nilai UAS           : ' ) ; readln (uas[x]) ;
        Rata[x]:=(0.1*tugas[x])+(0.2*absen[x])+(0.3*uts[x])+(0.4*uas[x]);
    End ;
    CLRSCR ;
    For i : = 1 to j do
    Begin
        Writeln ( ' ***************************** ' ) ;
        Writeln ( '  PENILAIAN UJIAN SISWA PPKM  ' ) ;
        Writeln ( ' ***************************** ' ) ;
        Writeln ( ' Nama            : ' , nama[i] ) ;
        Writeln ( ' Nilai Tugas     : ' , tugas[i] ) ;
        Writeln ( ' Nilai Absen     : ' , absen[i] ) ;
        Writeln ( ' Nilai UTS       : ' , uts[i] ) ;
        Writeln ( ' Nilai UAS       : ' , uas[i]) ;
        Writeln ( ' ***************************** ' ) ;
        Writeln ( ' Rata-rata : ' , rata : 1: 0 ) ;
        Readln ;
    End;
END.
```

Pembahasan 10
PROCEDURE

Prosedur adalah suatu program terpisah dalam blok sendiri yang berfungsi sebagai subprogram (program bagian). Parameter pada procedure ada 2 :

Bentuk Umum :

```
PROGRAM Judul_Program ;
    PROCEDURE judul_prosedur ;
    Begin
    .............
    End ;
BEGIN
.............
END.
```

Parameter didalam prosedur terdiri dari 2 yaitu :

a. *Parameter bersifat Lokal* artinya bahwa nilai yang terdapat didalam suatu modul program hanya dapat digunakan pada modul yang bersangkutan sehingga tidak dapat digunakan pada modul lain.
b. *Parameter bersifat Global* artinya bahwa nilai dapat digunakan pada semua modul maka nilai tersebut harus dideklarasikan di atas modul yang akan menggunakannya.

Contoh : Parameter Lokal

```
Procedure Hitung ;
Var
   X, Y : real ;
Begin
   Write ( '  Nilai X  ? ' ) ;
   readln ( X ) ;
   Y : = X * X ;
   Writeln ( ' Nilai Y ? ' , Y :1: 0 ) ;
End ;

BEGIN
    Hitung  ;
END.
```

Contoh : Paramater Global

```
Var
    X, Y : Byte ;
Procedure Tambah ;
Begin
   Write ( ' Input Nilai : ' ) ;
   readln ( X ) ;
    Y : = A + A ;
End ;

BEGIN
    Tambah ;
    Writeln ( ' Nilai Y = ', Y ) ;
Readln;
END.
```

Pemanggilan parameter pada prosedur terdiri dari 2 yaitu :

a. Parameter Secara Nilai (By Value)
   Bersifat satu arah, jika terjadi perubahan pada subprogram, maka tidak akan mempengaruhi program utama.
b. Parameter Secara Acuan (By Reference)
   Bersifat dua arah, jika terjadi perubahan pada subprogram, maka program utamanya akan ikut berubah.

Contoh : By Value

```
Uses WinCrt ;
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 ) ;
     Readln ;
END.
```

Contoh : By Reference

```
Uses WinCrt ;
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 ) ;
        Readln ;
END.
```

Contoh Program Prosedure Standar

```
Uses WinCrt;
Const PPKM='PuspikomM';
Begin
  Clrscr;
  Writeln(Copy(PPKM,6,4));                  → mengambil sebagian dari string
  Writeln(Pos('Sarana',PPKM));              → mencari posisi string
  Writeln(Length(PPKM));                    → panjang string
  Writeln(Concat(PPKM,' Informatika'));     → menggabungkan string
  Readln;
End.
```

Pembahasan 11
FUNGSI

Fungsi secara garis besar sama dengan prosedure baik parameter maupun pemanggilan parameternya hanya yang membedakannya adalah nama fungsi harus dideklarasikan dengan type datanya.

Bentuk Umum :

Function Identifier (daftar_ parameter) : t*ype data* ;

Contoh :

```
Uses WinCrt ;
Function Hitung ( Var A, B : integer ) : integer ;
Begin
    Hitung : = A+B ;
End ;
Var
    X, Y : integer ;
BEGIN
    Write ( ‘ Nilai X ? ‘ ) ; readln ( X ) ;
    Write ( ‘ Nilai Y ? ‘ ) ; readln ( Y ) ;
    Writeln ;
    Writeln ( X, ‘ + ‘, Y , ‘ =’, Hitung ( X, Y )) ;
    Readln ;
END.
```

Bagian Fungsi

Program Utama

Parameter pada fungsi sama dengan parameter yang ada pada prosedure yaitu :

a. Pemanggilan secara Nilai (By Value)

Contoh :

```
Uses WinCrt ;
 Function Hitung ( X, Y, Z : integer ) : integer ;
 Begin
     Z : = X+Y ;
      Writeln ( ‘ Nilai X  = ‘, X ) ;
     Writeln ( ‘ Nilai Y  = ‘, Y ) ;
     Writeln ( ‘ Nilai Z  = ‘, Z ) ;
End ;
Var
     A, B, C : integer ;
BEGIN
     A : = 5 ; B : = 7 ; C : = 3 ;
      Hitung ( A, B, C ) ;
     Writeln ( ‘ Nilai A =’, A, ‘ Nilai B = ‘, B, ‘ Nilai C = ‘, C ) ;
      Readln ;
END.
```

c. Pemanggilan secara Reference (Acuan)

Contoh :

```
Uses WinCrt ;
Function Hitung ( var A, B, C : integer ) : integer ;
 Begin
     Hitung : = A+B ;
      C : = A * B ;
End ;
```

```
    Var
        X, Y, Z : integer ;
    BEGIN
        Write ( ‘ Nilai X : ‘ ) ; readln ( X ) ;
        Write ( ‘ Nilai Y :  ) ; readln ( Y ) ;
         Writeln ;
        Writeln ( X , ‘ + ‘ , Y, ‘=’, Hitung ( X, Y, Z )) ;
        Writeln ( X, ‘ * ’,Y,  ‘=’, Z ) ;
        Readln ;
    END.
```

Contoh Function Standar Aritmatika & Transfer

```
Uses WinCrt;
Const     A=2;      B=-2;
          C=4;      D=2.5;
          E=2.3;    F=2.7;      G='A';
Begin
  Clrscr;
  Writeln('Absnya         : ',A, '=' ,Abs(A));     → Argumen Mutlak
  Writeln('Absnya         : ',B, '=' ,Abs(-B));
  Writeln('Expnya         : ',A, '=' ,Exp(A));     → Eksponensial Argumen
  Writeln('Lnnya          : ',A, '=' ,Ln(A));              → Logaritma Natural Argumen
  Writeln('Sqrnya         : ',A, '=' ,Sqr(A));     → Kuadrat Argumen
  Writeln('Sqrtnya        : ',C,'=' ,Sqrt(C));     → Akar Argumen
  Writeln('Pinya          : ',Pi);                 → angka Pi=3.14
  Writeln('Intnya         : ',D, '=' ,Int(D));     → bagian bulat angka real
  Writeln('Truncnya       : ',D, '=' ,Trunc(D));   → bagian bulat angka real
  Writeln('Fracnya        : ',D, '=' ,Frac(D));    → bagian pecahan real
  Writeln('Roundnya       : ',E, '=' ,Round(E));   → bulatkan angka real
  Writeln('Roundnya       : ',F, '=' ,Round(F));
  Writeln('Ordnya         : ',G, '=' ,Ord(G));     → Character ke ordinal
  Writeln('Chrnya         : ',A, '=' ,Chr(67));    → ordinal ke character
  Readln;
End.
```

Contoh Dengan Format

```
Uses WinCrt;
Const     A=2;      B=-2;
          C=4;      D=2.5;
          E=2.3;    F=2.7;
          G='A';
Begin
  Clrscr;
  Writeln('Absnya         : ',A, '=' ,Abs(A));
  Writeln('Absnya         : ',B, '=' ,Abs(-B));
  Writeln('Expnya         : ',A, '=' ,Exp(A):5:2);
  Writeln('Lnnya          : ',A, '=' ,Ln(A):5:2);
  Writeln('Sqrnya         : ',A, '=' ,Sqr(A));
  Writeln('Sqrtnya        : ',C,'=' ,Sqrt(C):1:0);
  Writeln('Pinya          : ',Pi:4:2);
  Writeln('Intnya         : ',D:3:1, '=' ,Int(D):1:0);
  Writeln('Truncnya       : ',D:3:1, '=' ,Trunc(D));
  Writeln('Fracnya        : ',D:3:1, '=' ,Frac(D):3:1);
  Writeln('Roundnya       : ',E:3:1, '=' ,Round(E));
  Writeln('Roundnya       : ',F:3:1, '=' ,Round(F));
  Writeln('Ordnya         : ',G, '=' ,Ord(G));
  Writeln('Chrnya         : ',A, '=' ,Chr(67));
  Readln;
End.
```

Pembahasan 12
RECORD

Adalah kumpulan item data (field) yang masing-masing dapat mempunyai tipe data yang beda

Contoh
```
Type
        Lgn= Record
                Kode    : integer;
                Nama    : string[35];
                Piutang : real;
        End;
Var Langganan : Lgn;
```

Menggunakan Tipe Data Record
Tiap-tiap komponen field dari record dapat dipergunakan dengan cara menuliskan **Pengenalrecord.pengenalfield**

Contoh
```
Langganan.Nama := 'Maulana' ;
```
PengenalField
PengenalRecord

Penulisan statemen diatas dapat menyebabkan statement menjadi panjang
Contoh lain :
```
        Lingkaran.Keliling := 2 * Pi * JariJari ;
```
Menggunakan Statemen With Do menjadi :
```
With Lingkaran Do
        Begin
                Keliling := 2 * Pi * JariJari ;
        End;
```
Contoh :
```
Uses WinCrt;
Type
        Recsis = Record
                Nama : string[15];
                Nu       : byte;
        End;
Var DataSis : Array[1..10] of RecSis;
     X, JD      : Byte ;
     Pred       : String[6] ;

BEGIN
    CLRSCR ;
    Write( ' Jumlah Data : ' ) ; Readln(JD) ;
    For X := 1 To JD Do
    Begin
        Write( ' Nama Siswa : ' ) ; ReadLn(Datasis[x].Nama) ;
        Write( ' Nilai Ujian     : ' ) ; ReadLn(Datasis[x].Nu) ;
    End;
    For X := 1 To JD Do
    Begin
        With Datasis[X] Do
        Begin
            IF Nu > 60 Then Pred := 'Lulus' ;
            Else Pred := 'Gagal' ;
            WriteLn(X, ' ' , Nama:10, ' ' ,Nu:2, ' ' , Pred:6);
        End;
    End;
    ReadLn;
END.
```

Pembahasan 13
FILE

Suatu File terdiri dari urutn komponen yang mempunyai tipe sama. Berbeda dengan larik yang jumlah komponennya sudah pasti, jumlah komponen dalam file sifatnya luwes, yaitu dapat ditambah dan dikurangi sewaktu-waktu

File Teks

Merupakan file yang berisi kumpulan dari karakter yang dibentuk dalam baris-baris dan masing-masing baris diakhiri dengan tanda akhir dari baris berupa karakter carriage return dan karakter line feed. File teks sebenarnya merupakan file dengan tipe char; tetapi mempunyai perbedaan, yaitu nilai yang bukan tipe char dapat direkam dan dibaca oleh File Teks

Prosedur Standar File Teks

Append : digunakan untuk membuka file yang telah ada untuk keperluan menambah data ke dalam file
Write : digunakan untuk merekam data ke file
Flush : digunakan untuk segera merekam data yang ada di buffer ke file Teks
Read : untuk membaca satu atau lebih nilai dari file ke dalam satu atau lebih variabel
SetTextBuf : digunakan untuk membuat buffer yang baru sebagai pengganti dari internal buffer untuk file variabel

Fungsi Standar File Teks

EoLn : untuk mengetahui apakah posisi dari file berada di end-of-line marker atau tidak
SeekEOF : untuk menghasilkan status akhir dari File
SeekEoLn : untuk menghasilkan status akhir baris

Contoh :

```
Uses WinCrt ;
Label a,b ;
Var
    Vfbarang    : text ;
    Nabrg       : string [20 ];
    Hrg         : longint ;
    Jml         : integer ;
    X, y        : byte ;
    Ttl, gth    : longint ;
    Lagi        : char ;
    Pil         : byte ;
BEGIN
    A:
    Clrscr ;
    Writeln ( ' MENU PILIHAN FILE TEXT ' ) ;
    Writeln ( ' 1. Rekam Data File Baru ' ) ;
    Writeln ( '  2. Tambah Data File ' ) ;
    Writeln ( '  3. Lihat Data' ) ;
    Writeln ( '  4. Exit' ) ;
    Write ( ' Silakan Pilih Menu [ 1/2/3/4 ] : ' ); readln ( pil ) ;
    Case pil of
        1..2 :   begin
                    Assign (vfbarang,'barang.txt' ) ; { variabel file barang }
                    IF pil =1 then
                    Rewrite ( vfbarang )
                    Else
                    Append (vfbarang ) ;
                        Lagi : = 'Y' ; X : = 0 ;
                        While (lagi ='Y') or (lagi = 'T') do
```

```
                    begin
                        write ( ‘ Nama Barang : ‘ ) ; readln (  nabrg );
                        write ( ’ Harga Barang : ’ ) ; readln ( hrg ) ;
                        write ( ’ Jumlah Barang : ’ ) ; readln ( jml ) ;
                        write ( vfbarang, nabrg:1      5 , hrg : 10,  jml : 4 ) ;
                        write ( ’ Isi Data Lagi [ Y/T ] : ’ ) ; readln ( lagi ) ;
                    end ;
                    close (vfbarang) ;
                    goto a ;
                end ;
            end;
        3   :   begin
                    Assign ( vfbarang , ’ barang.txt ’ ) ;
                    Reset ( vfbarang ) ;
                    Writeln(’           DATA PENJUALAN BARANG           ’ ) ;
                    Writeln(’****************************************** ’) ;
                    Writeln(’No  Nama Barang        Harga    Jumlah    Total  ’ ) ;
                    Writeln(’****************************************** ’ ) ;
                { proses dan cetak data detail }
                y : = 0 ; fth : = 0 ;
                While not eof ( vfbarang ) do
                    Begin
                        Read ( vfbarang, nabrg, hrg, jml ) ;
                        Total : = hrg * jml ;
                    gth : = gth + ttl ;
                    Inc ( y) ;
                    Write (y, ’  ’ ,nabar:15, ’  ’ ,hrg:7,’  ’,jml:3, ’  ’,ttl) ;
                    End ;
                    { garis penutup subtotal }
                    Writeln(’****************************************** ’ ) ;
                    Writeln (’Grand Total:                          Rp. ’, gth ) ;
                    Writeln(’****************************************** ’ ) ;
                    Close(vfbarang ) ;
                    Readln ;
                    goto a  ;
                end ;
        4   :   Goto b ;
    End ;
    Readln ;
    b:
END.
```

File Bertipe

Berbeda dengan file teks yang hanya dapat diakses secara urut, file bertipe disamping dapt pulajuga diakses secara urut(sequential acces, dapat juga diakses secara acak (random access). Dan Tipe dari file bertipe dapat berupa tipe integer, real, char, string array, record

Write	:	digunakan untuk merekam satu atau lebih nilai ke dalam file
Read	:	digunakan untuk membaca atau lebih, nilai dari file
Seek	:	Digunkan untuk mengarahkan penunjuk file ke suatu komponen tertentu di dalam file
Trunccate	:	digunakan untuk menghapus sejumlah komponen atau record mulai dari posisi record tertentu
FileSize	:	digunakan untuk menghasilkan jumlah dari komponen atau record

Contoh Program Bertipe Input Data

```
Uses WinCrt;
Type    MHS = Record
                    NM          : string[15];
                    UTS,UAS     : byte;
        End;

Var     FileMHS     : File of MHS;
        RecMHS      : MHS;
        No, Bts, JD, T, RN : Byte;
```

```
BEGIN
    CLRSCR;
    Write('Jumlah Data : '); ReadLn(JD);
    Assign(FileMHS,'C:\MHS.DAT');
    ReWrite(FileMHS);
    For No := 1 To JD Do
    Begin
        With RecMHS Do
        Begin
            Write('Nama : '); ReadLn(NM);
            Write('UTS  : '); ReadLn(UTS);
            Write('UAS  : '); ReadLn(UAS);
        End;
        Write(FileMHS,RecMHS);
    End;
    Close(FileMHS);
  ReadLn;
END.
```

Contoh Program Bertipe Tampil Data

```
Uses WinCrt;
Procedure Judul;
Begin
    WriteLn('DAFTAR NILAI SISWA');
    WriteLn('-------------------------------------------------------------------');
    WriteLn('No. Nama SISWA UTS UAS Total Rata Predikat Kelulusan');
    WriteLn('-------------------------------------------------------------------');
End;

Type    MHS=Record
            NM              : string[15];
            UTS,UAS         : byte;
        End;
Var FileMHS                 : File of MHS;
        RecMHS              : MHS;
        No,Brs,JD,T,R,RN    : byte;
        P                   : string[10];
        K                   : string[5];
BEGIN
    CLRSCR;
    Judul;
    Assign(FileMHS,'C:\MHS.DAT');
    Reset(FileMHS);
    No:=0;
    While Not EOF (FileMHS) Do
    Begin
        With RecMHS Do
        Begin
        Read(FileMHS,RecMHS);
        T:= UTS+UAS; R:= T div 2; inc(No);
        Case R of
                00..59 : P:='Kurang';
                60..74 : P:='Cukup';
                75..84 : P:='Baik';
                85..100: P:='Amat Baik';
        End;
        IF R<60 Then K:='Gagal' Else K:='Lulus';
        WriteLn(No:3,' ',NM:15,' ',UTS:3,' ',UAS:3,' ',T:3,' ',R:3,' ',P:10,' ',K:5);
        End;
    End;
    ReadLn;
    Close(FileMHS);
END.
```

Pembahasan 14
PEMBUATAN UNIT BARU

Contoh Program Pembentukan Unitku

```
Unit Unitku;

Interface
Uses WinCrt;
Procedure Bersihkan;
Function Ls3(A,T:Integer):Real;

Implementation
Procedure Bersihkan;
Begin
  Clrscr;
End;

Function Ls3(A,T:Integer):Real;
Begin
    Ls3:=A*T/2;
End;
End
```

Lalu simpan

Contoh Program Pemakaian Unit Baru

```
Uses Unitku;
Var A,T :Integer;
Begin
  Bersihkan;
  Write('Nilai Alas   :');Readln(A);
  Write('Nilai Tinggi :');Readln(T);
  Write('Luas Segitiga :',Ls3(A,T):5:1);
  Readln;
End..
```