

# **CONTOH PROGRAM KECIL**

**Dalam Bahasa Pascal**

**Oleh:**

**Inggriani Liem**

Program Studi Teknik Informatika  
STEI - Institut Teknologi Bandung  
Versi: April 2007

## PRAKATA

Diktat program kecil dalam bahasa Pascal ini adalah salah satu dari serial “program kecil” yang sejak tahun 1999 dipakai untuk perkuliahan pemrograman di lingkungan program studi Teknik Informatika (dahulu Jurusan/Departemen Teknik Informatika). Oleh karena itu, diktat ini tidak berdiri sendiri, melainkan terintegrasi dengan konsep pemrograman yang diberikan dalam perkuliahan.

Penyajian diktat ini sengaja dilakukan hanya berupa program (source code) yang utuh, dapat dieksekusi, dan hanya diberi komentar secukupnya, dalam bahasa terkait. Program-program kecil dalam diktat ini dimaksudkan sebagai pola program dalam bahasa Pascal sebagai langkah awal bagi pemula dalam berkenalan dan memprogram dalam bahasa Pascal. Setelah mampu memprogram pola program kecil, diharapkan siswa dapat melakukan kombinasi instruksi dan memakainya dalam memprogram. Jadi, program kecil merupakan tools, semacam “baby walker” ketika seorang bayi mulai belajar berjalan. Setelah itu dapat ditinggalkan. Walaupun diktat ini dipublikasi di web site perkuliahan, sangat tidak disarankan bahwa mahasiswa melakukan copy/paste teks program. Tujuan dari adanya program kecil ini adalah agar mahasiswa belajar mengetik dari awal dan memahami aspek eksekusi di lab.

Contoh-contoh yang sama akan dapat dijumpai dalam seri program kecil dalam bahasa lain yang separadigma. Ini akan membantu mahasiswa dalam mempelajari suatu bahasa baru, berdasarkan bahasa yang sudah dikuasainya.

Selamat mencoba memprogram kecil, sebagai langkah awal dalam menghasilkan program yang lebih besar.

Bandung, April 2007

## DAFTAR ISI

PROGRAM SEDERHANA .....	6
program hello; .....	6
program hellodos; .....	6
INPUT/OUTPUT .....	6
program baca; .....	6
ASSIGNMENT .....	6
program asign; .....	6
program asign1; .....	6
TIPE DASAR .....	7
program TDasar; .....	7
EKSPRESI DAN OPERATOR .....	7
Program oprator; .....	7
STRING .....	8
program manipstr; .....	8
program BACASTR; .....	8
KONSTANTA .....	8
program KONSTANTA; .....	8
program KONSTAN2; .....	9
HIMPUNAN (SET) .....	9
program Himpunan; .....	9
RECORD, Type Komposisi, Type Terstruktur .....	10
program tipe; .....	10
program bacarec; .....	11
ANALISA KASUS, KONDISIONAL .....	11
Program IF1; .....	11
program IF2; .....	12
program IF3; .....	12
program KASUS; .....	12
program wujudair; .....	13
program MAX2; .....	13
SUBPROGRAM .....	14
program subprg; .....	14
LINGKUP (SCOPE) .....	15
program Lingkup; .....	15
PENGULANGAN .....	16
program PRIFOR; .....	16
program PRIW; .....	17
program PRIREP; .....	17
program PRITER; .....	17
program KASUSREP; .....	18
TABEL (ARRAY) .....	19
program TABEL; .....	19
program TABSTRU; .....	19
TABEL MULTI DIMENSI .....	20
program Tab2dim; .....	20
program Tab3dim; .....	21
RECORD VARIANT .....	22
program RecVar; .....	22
program RecVarx; .....	22
POINTER .....	23
program Ptint; .....	23
program PTab; .....	24
program PRec; .....	24
LIST LINIER SEDERHANA .....	24
program list; .....	24

FILE EKSTERNAL .....	25
program BacaText; .....	25
program RekamText;.....	25
program TulisInt;.....	26
program Frec; .....	27
Unit dalam Turbo Pascal (dan FreePascal).....	28
unit upoint; .....	28
program mainpoint; .....	31

## DAFTAR PROGRAM KECIL PASCAL

KELOMPOK	NAMA	DESKRIPSI
Struktur Program pascal	Hello.pas	Menuliskan hello ke layar (Pascal standart)
	hellodos.pas	Menuliskan hello ke layar (Lingkungan DOS)
Assignment	assign.pas	Deklarasi integer, Assignment
	assignl.pas	Assignment
Type dasar	Tdasar.pas	Deklarasi dan penulisan nilai type dasar
	oprator.pas	Assignment dan ekspresi sederhana
String	manipstr.pas	Manipulasi string sederhana
	bacastr.pas	Pembacaan string
Konstanta	konstant.pas	Deklarasi konstanta, ekspresi
	konstan2.pas	Deklarasi konstanta, ekspresi
Type SET	Himpunan.pas	Deklarasi dan pemanfaatan SET dalam Pascal
Type bentukan	tipe.pas	Deklarasi type bentukan
Input/Output	Bacarec.pas	Pembacaan data
Analisa Kasus	if1.pas	Kasus: satu kasus
	if2.pas	Kasus: dua kasus komplementer
	if3.pas	Kasus: 3 kasus
	kasus.pas	Kasus: banyak kasus
	wujudair.pas	Terjemahan wujud air dari diktat
	max2.pas	Maksimum dua nilai
	subprg.pas	Prosedur dan fungsi : pendefinisian dan pemanggilan
Subprogram	Lingkup.pas	Scope & Life time dari variabel
	Funcrec.pas	Realisasi Fungsi yang hrs menghasilkan type record
Pengulangan	prifor.pas	Pengulangan FOR
	priw.pas	Pengulangan WHILE
	prirep.pas	Pengulangan REPEAT
	Priter.pas	Pengulangan ITERATE
	kasusrep.pas	Kasus digabung dengan REPEAT
Tabel	tabel.pas	Deklarasi tabel integer, pengisian dan penulisan
	tabstru.pas	Deklarasi tabel dengan elemen type bentukan
	Tab2dim.pas	Tabel dua dimensi
	Tab3dim.pas	Tabel tiga dimensi
Record varian	recvar.pas	Deklarasi, mengacu field record varian
	recvarx.pas	Record varian dengan komponen type bentukan
Pointer	ptint.pas	Pointer ke integer
	ptab.pas	Pointer ke tabel integer
	prec.pas	Pointer ke record
	list.pas	Deklarasi list linier sederhana
File Eksternal	Bacatext.pas	Membaca text file dan menuliskan isinya ke layar. File diakhiri titik dan EOF
	Rekamtext.pas	Rekam file teks
	Rwint.pas	Merekam dan menuliskan isi File of integer
	frec.pas	Merekam dan menulis isi File of integer
Unit dalam Turbo Pascal dan FreePascal	upoint.pas maintpoint.pas	Unit untuk manipulasi type point, dan tabel yang elemennya point

## PROGRAM SEDERHANA

```
program hello;
(* File : HELLO.PAS *)
  (* menuliskan Hello ke layar *)
begin
  writeln ( 'hello ' ) ;
end.
```

```
program hellodos;
(* File : HELLODOS.PAS *)
(* menuliskan Hello ke layar *)
uses crt;
begin
  clrscr;
  writeln ( 'hello ' ) ;
end.
```

## INPUT/OUTPUT

```
program baca;
(* File : BACA.PAS *)
(* contoh membaca integer*)
(* kemudian menuliskan nilai yang dibaca *)
(* Kamus *)
var
  a : integer;
begin
(* Program *)
  writeln ('Contoh membaca dan menulis, ketik nilai integer: ') ;
  readln (a) ;
  writeln ('nilai yang dibaca : ', a) ;
end.
```

## ASSIGNMENT

```
program asign;
(* File : ASIGN.PAS *)
(* Assigntment dan print *)
var
(* Kamus *)
  i : integer;
begin (*Algoritma *)
writeln ( 'hello' );
  i := 5;
  writeln ( 'Ini nilai i : ',i);
end.
```

```
program asign1;
(* File : ASIGN1.pas *)
(* Assignment dan print *)
(* Kamus *)
var
  i : integer;
  ii : longint;
begin (* Algoritma *)
writeln ( 'hello');
```

```

i := 1234;
ii := 123456 ;
writeln ( 'Ini nilai i=1234 = : ',i);
writeln ( 'Ini nilai ii=123456 : ',ii);
writeln ( 'Ini nilai max integer: ',maxint);
writeln ( 'Ini nilai max longint: ',maxlongint);
end.

```

## TIPE DASAR

```

program TDasar;
(* File : TDasar.pas *)
(* Deklarasi, assignment dan penulisan type dasar *)
(* Kamus *)
var
    i : integer ;
    x,y : real ;
    found : boolean;
(* Algoritma *)
begin
    i:= 5; writeln ('i = ', i);
    x := 0.5; writeln ('x = ', x);
    y := 1.0e + 3; writeln ('y = ', y);
    found:= true; writeln ('Found = ', found);
end.

```

## EKSPRESI DAN OPERATOR

```

Program oprator;
(* File : oprator.pas *)
(* Contoh pengoperasian variabel bertipe dasar *)
(* Kamus *)
VAR
    Bool1, Bool2, TF : Boolean;
    i, j, hsl : Integer;
    x,y,res : real;
Begin
(* algoritma *)
    writeln ('Utk program ini, baca teksnya dan tambahkan output');
    Bool1 := True; Bool2 := False;
(** contoh-contoh ekspresi: bukan untuk assignment berulang-ulang **)
    TF := Bool1 And Bool2 ;
    TF := Bool1 or Bool2 ;
    TF := Not Bool1 ;
    TF := Bool1 Xor Bool2 ;
(* operasi numerik *)
    i := 5; j := 2 ;
    hsl := i+j; hsl := i - j; hsl := i div j; hsl := i * j;
    hsl := i div j ; (* pembagian bulat *)
    hsl := i Mod j ; (* sisa *)

(* operasi numerik *)
    x := 5.0 ; y := 2.0 ;
    res := x + y; res := x - y; res := x / y; res := x * y;

(* operasional relasional numerik *)
    TF := i < j; TF := i > j; TF := i <= j;

```

```

    TF := i >= j; TF := i <> y;
(* operasional relasional numerik *)
    TF := x < y; TF := x > y; TF := x <= y;
    TF := x >= y; TF := x <> y;

end.

```

## STRING

```

program manipstr;
(* manipulasi string sederhana *)

var
    str1, str2 : string;
    strr : string;
    stri : string;
    i, kode: integer;
begin
    str1 := 'Saya ';
    str2 := ' Belajar di ITB';
    strr := str1 + str2;
    readln (strr);
(* koneversi string numerik ke nilai integer *)
    stri := '123';
    val (stri, i, kode ) ; (* amatilah nilai kode setelah eksekusi *)
    writeln(stri, '-', i, '-', kode);
end.

```

```

program BACASTR;
(* File : BACASTR.pas *)
(* alokasi string, kemudian mengisinya dengan membaca *)
(* Kamus *)
VAR
    str : string;
    str1 : string;
begin
(* Program *)
    writeln ( 'Baca string, maks 256 karakter: ');
    readln ( str);
    writeln ( 'String yang dibaca : ' ,str);
    str1 := str;
    writeln ( 'String yang disalin : ',str1);
end.

```

## KONSTANTA

```

program KONSTANTA;
(* File : konstant.pas *)
(* Membaca jari-jari, menghitung luas lingkaran *)
(* Latihan pemakaian konstanta *)
(* Kamus *)
CONST
    pi=3.1415;
VAR
    r : real;
begin
(* program *)
    write ('Jari-jari lingkaran =');
    readln (r) ;

```



```

        writeln ('Luas lingkaran = ', pi*r*r);
        writeln (' Akhir program ');
(* Kompilasi, amatilah apa yang terjadi jika komentar sbb. dibuang *)
(* perhatikan option pada kompilator anda *)
        (* pi := 10.0; *)
end.

```

```

program KONSTAN2;
(* File : KONSTAN2.PAS *)
(* Menghitung luas lingkaran, dari jari-jari yang dibaca *)
(* Latihan pemakaian konstanta *)
(* Kamus *)
CONST
    pi = 3.1415;
    dua = 2.0;
VAR
    r : real;
    luas : real;
    kel : real;
begin
(* program *)
    writeln ( 'Jari-jari lingkaran =');
    readln (r) ;
    luas := pi * r * r;
    writeln ( 'Luas lingkaran = ', luas:6:2);
    kel := dua* pi * r ;
    writeln ('keliling lingkaran= ',luas:6:2);
    writeln ('akhir program ');
end.

```

## HIMPUNAN (SET)

```

program Himpunan;
(* File : Himpunan.pas *)
(* Pendefinisian dan pemanfaatan himpunan : SET *)
(* Kamus *)
type
    Hari = (senin,selasa,rabu,kamis,jumat,sabtu,minggu);
    weekday = SET of Hari;
var
    H : Hari;
    H0, H1, H2 : Hari;
    W : weekday;

(* Algoritma *)
begin
    (* Instruksi berikut salah :type set tidak dapat ditulis/baca*)
    (* writeln (' Hari = ', H); *)
(* Assignment : boleh *)
    H1 := selasa;
    (* prosedur terdefenisi *)
    H2 := succ (H1);
    H0 := pred (H1);
    (* pemanfaatan untk mengontrol pengulangan *)
    (* Akan dibahas pada pengulangan *)
    for H := senin to minggu do
    begin
        writeln ( 'Selamat Pagi ...');
        writeln ( 'Ordinal : ', ord (H) );
    end;
end;

```

```

(* intruksi CASE : akan dibahas pada analisa kasus*)
case H1 of
  senin :
    writeln ( 'senin' );
  selasa :
    writeln ( 'selasa' );
  rabu :
    writeln ( 'rabu' );
  kamis :
    writeln ( 'kamis' );
  jumat :
    writeln ( 'jumat' );
  sabtu :
    writeln ( 'sabtu' );
  minggu :
    writeln ( 'minggu' );
  else
    writeln ( 'tidak terdefinisi ' );
end;
end.

```

## RECORD, Type Komposisi, Type Terstruktur

```

program tipe;
(* File : tipe.pas *)
(* contoh pendefinisian dan pengisian struktur *)
TYPE
  Point = record
    X : integer; (* absis *)
    Y : integer; (* ordinat*)
  end; (* type Point *)
  MAHASISWA = record
    NIM : integer;
    Nama : string;
    Nilai : real;
  end; (* type mahasiswa *)
VAR
  P1 : Point;
  P2 : Point;
  Mhs : MAHASISWA;
begin
  writeln ( 'Contoh mengisi struktur dengan assignment : ');
  writeln ( 'Titik P1, dengan P1.x dan P1.y: ');
  P1.x := 1;
  P1.y := 2;
  writeln ( 'P1.X= ', P1.X);
  writeln ( 'P1.Y= ', P1.Y);

  writeln ( 'Baca Titik P2');
  write ( 'Absis : ');
  write ( 'Ordinat : ');
  readln (P2.Y) ;
  writeln ( 'Koordinat : ', P2.X, ', ', P2.Y);

  mhs.NIM := 7473;
  mhs.Nama := 'Juliette';
  mhs.Nilai := 80;
  writeln ( 'Hasil assignment thd Mhs ');
  writeln ( 'Nama = ',Mhs.Nama);
  writeln ( 'Nim = ',Mhs.NIM);

```

```

        writeln ( 'Nilai = ',Mhs.Nilai:6:2);
(* pemakaian WITH untuk record *)
        writeln ( 'Hasil assignment thd Mhs ');
        with Mhs do
            begin
                writeln ( 'Nama = ',Nama);
                writeln ( 'Nim   = ', NIM);
                writeln ( 'Nilai = ',Nilai:6:2) ;
            end;
end.

```

```

program bacarec;
(* File : Bacarec.PAS *)
(* contoh membaca record*)
(* kemudian menuliskan nilai yang dibaca *)
(* Kamus *)
TYPE Point= record
            x : integer; (* absis *)
            y : integer; (* ordinat *)
        end; (* type Point *)
var
    P1, P2 : Point;
begin
(* Program *)
    writeln ( 'Contoh membaca dan menulis titik ' ) ;
    write ( 'Absis = '); readln(P1.x);
    write ( 'Ordinat = '); readln(P1.y) ;
    writeln ( 'Titik yang dibaca : (', P1.x, ', ', P1.y , ')');

    with P2 do
        begin
            write ( 'Absis = '); readln (x) ;
            write ( 'Ordinat = '); readln (y);
            writeln ( 'Titik yang dibaca : ( ', x, ', ', y, ')');
        end;
end.

```

## ANALISA KASUS, KONDISIONAL

```

(* File : IF1.PAS *)
(* contoh pemakaian IF satu kasus *)
(* membaca nilai integer, menuliskan nilainya jika positif *)
Program IF1;
(* Kamus *)
var
    a : integer;
begin
(* Program *)
    writeln ( 'Contoh IF satu kasus ');
    write ( 'Ketikkan satu nilai integer : ');
    readln (a);
    if (a >= 0) then
        begin
            writeln ( 'Nilai a positif... ', a);
        end;
end.

```

```

(* File :IF2.PAS *)
(* contoh pemakaian IF dua kasus komplementer *)
(* Membaca sebuah nilai, *)
(* menuliskan 'Nilai a positif , nilai a', jika a >=0 *)
(*      'Nilai a negatif , nilai a', jika a <0 *)
program IF2;
(* Kamus *)
var
    a : integer;
begin
    (* Program *)
    writeln ('Contoh IF dua kasus ');
    write ( 'Ketikan suatu nilai integer :');
    readln (a);
    if (a >= 0) then
        begin
            writeln ( 'Nilai a positif ', a);
        end else (* a<0 *)
        begin
            writeln ( 'Nilai a negatif ', a);
        end;
end.

```

```

(* File : IF3.PAS *)
(* contoh pemakaian IF dua kasus komplementer *)
(* Membaca sebuah nilai, *)
(* menuliskan 'Nilai a positif , nilai a', jika a>0 *)
(*      'Nilai a sama dengan nol , nilai a', jika a =0 *)
(*      'Nilai a negatif , nilai a', jika a <0 *)
program IF3;
(* Kamus *)
var
    a : integer;
begin
    (* Program *)
    writeln ( 'Contoh IF tiga kasus');
    write ( 'Ketikkan suatu nilai integer :');
    readln (a);
    if (a > 0) then
        begin
            writeln ( 'Nilai a positif ', a);
        end else
        if (a=0) then
            begin
                writeln ( 'Nilai a sama dengan nol ', a);
            end else if (a<0) then
                begin
                    writeln ( 'Nilai a negatif ', a);
                end;
        end;
end.

```

```

program KASUS;
(* File : KASUS.PAS *)
(* Contoh kasus dengan intruksi CASE *)
VAR
(* Kamus *)
    cc : char;
begin
    (* Program *)

```

```

writeln ( 'Ketikkan sebuah huruf, akhiri dengan RETURN ');
readln (cc);
case cc of
  'a' : begin
        writeln ( ' Yang anda ketik adalah a ' );
      end;
  'u' : begin
        writeln ( ' Yang anda ketik adalah u ' );
      end;
  'e' : begin
        writeln ( ' Yang anda ketik adalah e ' );
      end;
  'o' : begin
        writeln ( ' Yang anda ketik adalah o ' );
      end;
  'i' : begin
        writeln ( ' Yang anda ketik adalah i ' );
      end
  else writeln ( ' Yang anda ketik adalah huruf mati atau angka' );
end;
end.

```

```

(* File : wujudair.PAS *)
(* contoh pemakaian IF tiga kasus : wujud air *)
program wujudair;
(* Kamus : *)
var
  T: integer;
begin
  (* program *)
  writeln ( 'Contoh IF tiga kasus ' );
  write ( 'Temperatur (der. C) = ' );
  readln (T);
  if (T < 0) then
  begin
    writeln ( 'Wujud air beku ');
  end else
  begin
    if ( (0<=T) and (T<=100) ) then
    begin
      writeln ( 'Wujud air cair ');
    end else
    begin
      if (T>100) then
      begin
        writeln ( 'Wuju air uap/gas ');
      end;
    end;
  end;
end;
end.

```

```

program MAX2;
(* File :MAX2.PAS *)
(* Maksimum dua bilangan yang dibaca *)
VAR
(* Kamus *)
  a,b : integer;
begin
  (* Program *)

```

```

writeln ( 'Maksimum dua bilangan : ' );
write ( 'Ketikan bilangan pertama : ' );
readln (a) ;
write ( 'Ketikan bilangan kedua : ' );
readln (a) ;
if (a >= b) then
  begin
    writeln ( 'Nilai a yang maksimum ', a);
  end
else
  begin
    writeln ( 'Nilai b yang maksimum ', b) ;
  end ;
end.

```

## SUBPROGRAM

```

program subprg;
(* File : subprg.PAS *)
(* contoh pendefinisian dan pemanggilan FUNGSI dan PROSEDUR *)
(* deklarasi dan badan procedure/fungsi LOKAL *)
(* Konsep yang harus dijelaskan :*)
(* - perbedaan fungsi dan prosedur *)
(* - parameter formal dan aktual *)
(* - passing parameter by value dan by ref *)
var
  a, b : integer;

function maxab (a,b : integer) : integer;
begin (* mencari maksimum dua bilangan bulat *)
  if a>=b then
    begin
      maxab := a;
    end else
    begin
      maxab := b;
    end;
end;

procedure tukar (var a, b : integer);
(* menukar nilai dua buah variabel a dan b *)
(* parameter input/output *)
var
  temp : integer;
begin (* menukar dua bilangan bulat *)
  temp := a;
  a := b;
  b := temp;
end;

begin (* program utama *)
(* Membaca dua bilangan integer *)
(* Menuliskan maksimum dua bilangan yang dibaca dg memanggil fungsi*)
(* Menukar kedua bilangan dengan 'prosedur' *)
writeln ( 'Maksimum dua bilangan : ');
writeln ( 'Ketikkan bilangan pertama : ');
readln (a) ;
writeln ( 'Ketikkan bilangan kedua : ');
readln (b) ;
writeln ( 'Ke dua bilangan : a = ',a) ;
writeln ( '                               b = ',b) ;

```

```

    writeln ( 'Maksimum =      ', maxab (a,b) ) ;
    writeln ( 'Tukar kedua bilangan... ' ) ;
    tukar ( a, b ) ;
    writeln ( 'Ke dua bilangan setelah tukar: a = ',a ) ;
    writeln ( '                                b = ',b) ;
end.

```

## LINGKUP (SCOPE)

```

program Lingkup;
(* File : Lingkup.pas *)
(* arti nama : Lingkup dan masa hidup variabel; parameter by value,
by ref *)
(* Kamus *)
var
a, b : integer;

procedure Plus1 (var x:integer);
(* x adalah parameter input/output *)
(* Menambah nilai thd Prosedur Plus1 *)
(* Fungsi lokal thd Prosedur Plus1 *)
function Incr (i:integer) : integer;
(* mengirimkan nilai i ditambah 1 *)
begin
    Incr := i+1;
end;
begin
    x := Incr(x) ; (* nilai parameter input x ditambah 1 *)
end;

Procedure Swap (var a,b : integer );
(* perhatikan bada a dan b disini dengan deklarasi a dan b global *)
var
    temp : integer; (*variabel lokal *)
begin
    temp := a;
    a:= b;
    b:= temp;
end;

function Plus2 (i:integer) : integer;
(* Mengirimkan nilai i ditambah 2 *)
begin
    Plus2 := i+2;
end;

(* Algoritma program utama*)
begin
    (* berikut tidak dikenal *)
    (* writeln ("Nilai i= ',i); *)
    a := 2;
    (* Berikut ini salah *)
    (* b := Incr (a) ; *)
    b := a;
    Plus2 (a);
    swap (a,b);
    (* Berapa hasilnya ?? *)

```

```

    writeln ('1. a= ',a, 'b= ', b );
    b := Plus2 (b);
    (* Berapa hasilnya ?? *)
    writeln ('2. a= ',a, 'b= ', b );
end.

program FuncRec;
(* File : FuncRec.pas *)
(* Fungsi yang harus mengembalikan type bentukan : tidak mungkin *)
(* Kamus *)
type
    Point = record
        x : integer; (* absis*)
        y : integer; (* ordinat*)
    end;
var
    T, T1, T2 : Point;
    procedure Tulis (T:Point);
    (* menuliskan sebuah titik T *)
begin
    writeln ( 'Titik T ( ',T.x, ', ',T.y, ' ) ');
end;

(* function MidPoint (T1,T2: Point) : Point; *)
(* Menghasilkan Titik tengah T1,T2 berupa titik *)
(* karena fungsi dalam bahasa Pascal tidak bisa mengembalikan *)
(* type record *)
(* berikut ini transformasi untuk mendapatkan efek yang dimaksud *)
procedure TtkTengah (T1,T2 : Point;
                    var MidPoint : Point) ; (* titik
hasil *)
(* Menerima T1 dan T2 dua buah Point *)
(* Menghasilkan Midpoint : sebuah VARIABEL bertype Point *)
begin
    MidPoint.x := ((T1.x + T2.x) div 2 );
    MidPoint.y := ((T1.y + T2.y) div 2 );
end;

(* Algoritma *)
begin
    T1.x := 0; T1.y := 0;
    T2.x := 10; T2.y := 10;
    Tulis (T1);
    Tulis (T2);
    TtkTengah (T1,T2,T);
    Tulis (T);
end.

```

## PENGULANGAN

```

program PRIFOR;
(* File : PRIFOR.PAS *)
(* Baca N, Print 1 s/d N dengan FOR *)
(* Kamus *)
var
    i : integer;
    N: integer;
begin

```



```

(* program *)
writeln ( 'Baca N, print 1 s/d N ');
write ( 'N = ');
readln (N) ;
for i:=1 to N do
  begin
    writeln (i);
  end; (* FOR *)
  writeln ( 'Akhir program ');
end.

```

```

program PRIW;
(* File : PRIW.PAS *)
(* Baca N, *)
(* Print i = 1 s/d N dengan WHILE *)
VAR
(* Kamus *)
  n : integer;
  i : integer;
begin (* Program loop WHILE *);
  write ( 'Nilai N = ');
  readln (N);
  i := 1 ;
  writeln ( 'Print i dengan WHILE: ');
  while (i<=N) do
    begin
      writeln (i);
      i := i + 1;
    end ; (* i>N *)
end.

```

```

program PRIREP;
(* File : PRIREP.PAS *)
(* contoh baca N, *)
(* print 1 s/d n dengan REPEAT *)
(* Kamus : *)
var
  N : integer;
  i : integer;

begin (* Program *)
  write ( 'Nilai N= ');
  readln (N);
  i := 1 ;
  writeln ( 'Print i dengan REPEAT: ');
  repeat
    writeln (i);
    i := 1;
  until (i > N);
end.

```

```

program PRITER;
(* File : PRITER. Pas *)
(* Baca N, *)
(* Print i = 1 s/d N dengan ITERATE *)
(* Kamus : *)
VAR
  N : integer;
  i : integer;

```

```

    stop : boolean;
begin
(* Program *)
    write ( 'Nilai N = ');
    readln (N);
    i := 1;
    writeln ( 'Print i dengan ITERATE : ');
    stop := false;
    repeat
        writeln (i);
    if (i=N) then stop := true else
        begin i := i + 1; end;
    until stop; (* i= N *)
end.

```

```

program KASUSREP;
(* File : KASUSREP.PAS *)
(* Contoh kasus dengan switch dan pengulangan *)
(* membaca karakter sampai user mengetikkan q *)
VAR
(* Kamus *)
    cc : char;
    quit : boolean;
begin
(* Program *)
    repeat
        quit := false;
        write ( ' Ketikkan sebuah huruf, akhiri dengan q : ');
        readln (cc);

        case cc of
            'a' : begin
                writeln ( ' Yang anda ketik adalah a ');
                end;
            'u' : begin
                writeln ( ' Yang anda ketik adalah u ');
                end;
            'e' : begin
                writeln ( ' Yang anda ketik adalah e ');
                end;
            'i' : begin
                writeln ( ' Yang anda ketik adalah i ');
                end;
            'q' : begin
                quit := true;
                end;
            else writeln ( ' Yang anda ketik adalah huruf mati ');
                end; (* case *)
        until (quit);
        writeln ( ' Akhir program... sebab anda mengetik q ');
    end.

```

## TABEL (ARRAY)

```
program TABEL;
(* File : TABEL.PAS *)
(* latihan array : mengisi dg assignment, menulis *)
VAR
(* Kamus *)
  i : integer;
  tab : array [1..10] of integer;
  N : integer;
begin
(* Program *)
N := 5;
writeln ( 'Isi dan print tabel: ');
(* isi dengan assignment *)
  for i := 1 to N do
    begin tab [i] := i;
      tab [i] := i;
    end;
(* traversal : print *)
  for i := 1 to N do
    begin
      writeln ( 'i= ',i, ' tab[i]= ', tab[i] );
    end;
end.
```

```
program TABSTRU;
(* File : TABSTRU.PAS *)
(* latihan array yang isinya struktur : mengisi dg assignment,
menulis *)
TYPE Point = record
      X : integer; (* absis *)
      Y : integer; (* ordinat *)
    end;
VAR
(* Kamus *)
  i : integer (* indeks tabel *);
  tabpoint : array [1..10] of Point;
  N : integer;
begin
(* Program *)
N := 5;
writeln ( ' Isi dan print tabel struktur: ');
(* isi dengan assignment *)
  for i := 1 to N do
    begin
      tabpoint [i].X := i;
      tabpoint [i].Y := tabpoint [i].X
    end;
(* traversal : print *)
  writeln ( '-----' );
  writeln ( '      I      X      Y      ' );
  writeln ( '-----' );
  for i := 1 to N do
    begin
      writeln ( i:5, tabpoint [i].X:5, tabpoint [i].Y:5);
    end;
  writeln ( '-----' );
end.
```

## TABEL MULTI DIMENSI

```
program Tab2dim;
(* File : Tab2dim.pas *)
(* Tabel integer dua dimensi (matriks) *)
(* Kamus *)
type
(* Cara 1 : sebagai array dua dimensi *)
  MatInt = array [1..3, 1..3] of integer;
(* Cara 2 : sebagai array of array *)
  MatArr = array [1..3] of array [1..3] of integer;
var
  M1 : MatInt;
  MA1 : MatArr;
  i , j : Integer ;
(* Algoritma *)
begin
  writeln ( ' Array dua dimensi : ');
  (* Mengisi Matrik dua dimensi *)
  for i := 1 to 3 do
    begin
      for j := 1 to 3 do
        begin
          M1 [i,j] := i * j;
        end; (* for j *)
      end;
    (* Menulis hasil isian di atas *)
    for i := 1 to 3 do
      begin
        for j:= 1 to 3 do
          begin
            write ( ' (i, j) = ', i, ', ', j, ' => M1 [i, j] = ', M1 [i, j]);
            end;
          writeln;
        end;

        writeln ( ' Array of array : ');
      (* Mengisi array of array : perhatikan cara mengacu elemen *)
      for i := 1 to 3 do
        begin
          for j := 1 to 3 do
            begin
              MA1 [i] [j] := i*j;
            end;
          end;
        (* Menulis hasil isian di atas *)
        for i := 1 to 3 do
          begin
            for j := 1 to 3 do
              begin
                (* Cobalah dua instruksi write sbb *)
                (* write ( ' (i,j) = ', i, ', ', j, ' => MA1 [i,j] = ',MA1 [i,j] ); *)
                write ( ' (i,j) = ', i, ', ', j, ' => MA1 [i,j] = ',MA1 [i][j] );
                end;
              writeln;
            end;
          end;
        end.
end.
```

```

program Tab3dim;
(* File : Tab3dim.pas *)
(* Tabel integer tiga dimensi *)
(* Kamus *)
type
(* Cara I : sebagai array tiga dimensi *)
  MatInt = array [1..2, 1..2, 1..2] of integer;
(* Cara 2 : sebagai array of array of array *)
  MatArr = array [1..2] of array [1..2] of array [1..2] of integer;
var
  M1: MatInt;
  MA1: MatArr;
  i,j,k : Integer;
(* Algoritma *)
begin
  writeln (' Array tiga dimensi : ');
  (* Mengisi Matriks tiga dimensi *)
  for i := 1 to 2 do
  begin
    for j := 1 to 2 do
    begin
      for k := 1 to 2 do
        M1 [i,j,k] := i*j;
      end;
    end;
  end;
  (* Menulis hasil isian di atas *)
  for i := 1 to 2 do
  begin
    for j := 1 to 2 do
    begin
      for k := 1 to 2 do
        writeln ('(i,j,k) = ', i, ', ', j, ', ', k, ' => M1 [i,j,k] = ', M1 [i,j,k]);
      end;
    end;
  end;
  writeln ( 'Array of array of array : ' );
  (* Mengisikan array of array : perhatikan cara mengacu elemen *)
  for i := 1 to 2 do
  begin
    for j := 1 to 2 do
    begin
      for k := 1 to 2 do
        begin MA1 [i] [j] [k] := i*j; end;
      end;
    end;
  end;
  (* Menulis hasil isian di atas *)
  for i := 1 to 2 do
  begin
    for j := 1 to 2 do
    begin
      for k := 1 to 2 do
        begin
          writeln ('(i,j,k =', i, ', ', j, ', ', k, ', =>MA1[i,j,k]=' MA1[i][j][k]);
        end;
      end;
    end;
  end;
end.

```

## RECORD VARIANT

```
program RecVar;
(* File : RecVar.pas *)
(* Record Varian : dengan komponen yang variabel *)
(* Konsep : representasi data yang tidak "fix" type-nya*)
(* Kamus *)
(* Cell adalah sebuah sel spread sheet, yang mungkin isinya :*)
(* Formula : string; integer atau real *)
type
  trec = (rumus, int, float);

  cell = record
    adrbrs : char;
    adrkol : integer;
    case Tsel : trec of
      rumus : ( form : string [5]);
      int    : ( nili : integer);
      float  : ( nilf : real ) ;
    end; (* cell *)
var
  Fcell, ICell, RCell : Cell;
begin
  (* Algoritma *)
  (* Cara mengisi nilai *)
  (* Type cell adalah formula *)
  Fcell.adrbrs := 'A' ;
  Fcell.adrkol := 1;
  Fcell.Tsel := rumus;
  Fcell.form := 'XYZ12';

  (* Type cell adalah integer *)
  Icell.adrbrs := 'A';
  Icell.adrkol := 2;
  Icell.Tsel := int;
  Icell.form := '10';
  (* Type cell adalah bilangan *)
  Rcell.adrbrs := 'A';
  Rcell.adrkol := 3;
  Rcell.Tsel := float;
  Rcell.nilf := 10.55;
end.
```

```
program RecVarx;
(* File : RecVarx.pas *)
(* Record Varian dengan type bentukan *)
(* Kamus *)
(* Gambar adalah bentuk ayang dapat berupa garis, segi empat *)
type
  trec = ( garis, segi4 );
  Point = record
    x: integer;
    y: integer;
  end;
  TGrS = record
    Pawal : Point; (* titik awal *)
    PAKhir : Point (* titik akhir *)
  end;
```

```

    TS4 = record (* Segi empat *)
        TopLeft : Point; (* Kiri atas *)
        BottRight : Point (* Kanan bawah *)
    end;

    Gambar = record
        id : integer; (* identitas gambar *)
        case TBentuk : trec of
            garis : (G : TGrS);
            segi4 : (S4 : TS4);
        end;
end;

var
    G1, G2 : Gambar;
    G3 : Gambar;

begin (* Algoritma *)
    (* Cara mengisi nilai *)
    (* Gambar adalah garis *)
    G1.id := 1;
    G1.TBentuk := garis;
    G1.G.PAwal.x := 10;
    G1.G.PAwal.y :=10;
    G1.G.PAkhir.x :=10;
    G1.G.PAkhir.y :=10;

    (* Gambar adalah segiempat *)
    G2.id := 99;
    G2.TBentuk := segi4;
    G2.S4.TopLeft.x := 0;
    G2.S4.TopLeft.y :=0;
    G2.S4.BottRight.x :=10;
    G2.S4.BottRight.y :=10;
    (***** HATI - HATI *****)
    (* Perhatikan apa yang terjadi saat kompilasi *)
    (* dengan assignment berikut *)
    G3.id :=99;
    G3.TBentuk :=garis;
    G3.S4.TopLeft.x := 0;
    G3.S4.TopLeft.y :=0;
    G3.S4.BottRight.x :=10;
    G3.S4.BottRight.y :=10;
    (* Komentar anda ???*)
end.

```

## POINTER

```

program Ptint;
(* File : Ptint.pas *)
(* Pointer ke integer *)
(* Kamus *)
var
    i : integer;
    Pti : ^integer ;
(* Algoritma *)
begin
    i := 5;
    new (Pti); (* alokasi *)
    Pti^ := 10;
    writeln ('i = : ', i );
    writeln ( 'Nilai yang ditunjuk Pti = ', Pti^ );
    dispose (Pti); (* dealokasi *)
end.

```

```

program PTab;
(* File : PTab.pas *)
(* Pointer ke tabel integer *)
(* Kamus *)
    type TabInt = array [1..10] of integer;
var
    T : TabInt;
    i : Integer;
    pt: ^TabInt;
begin (* Algoritma *)
    for i := 1 to 10 do
        begin
            T [i] :=i;
        end;
    (* alokasi Pointer ke tabel integer *)
    new (pt) ; (* alokasi *)
    Pt^ := T; (* pendefinisian isi/nilai yang ditunjuk *)
    (* akses elemen *)
    for i := 1 to 10 do
        begin
            writeln ( 'i= ', i, ' pt&[i] = ', pt^[i] );
        end;
    dispose (pt);
end.

program PRec;
(* File : PRec.pas *)
(* Pointer ke record *)
(* Kamus *)
type
    Point = record
        x: integer;
        y: integer;
    end;
var
    T : Point ;
    Pt : ^Point;
begin (* Algoritma *)
    new (Pt); (* alokasi *)
    Pt^.x := 10; Pt^.y := 5; (* akses komponen titik *)
    with Pt^ do
        begin (* dengan WITH *)
            writeln ( 'Absis P : ', x);
            writeln ( 'Ordinat P : ', y);
        end; (* kenapa tidak dilakukan dispose ? *)
end.

```

## LIST LINIER SEDERHANA

```

program list;
(* File ; list.pas *)
(* contoh deklarasi list dan pengisian nilai Info *)
TYPE
    infotype = integer;
    address = ^ ElmtList; (* alamat elemen *)
    Elmtlist = record (* type elemen *)
        info : infotype;
        next : address;
    end;

```



```

VAR
    First : address;
    P      : address;
begin
    (* Program *)
    (* GetMem (First, Sizeof (ElmtList )) : *)
    First :=nil;
    GetMem (P, Sizeof (Elmtlist));
    P^.Info := 10;
    P^.Next := NIL;
    First := P;
    with First^ do begin
        writeln ( 'Info : ', info );
    end;
    dispose (p);
end.

```

## FILE EKSTERNAL

```

program BacaText;
(* File : Bacatext.pas *)
(* Membaca sebuah text file diakhiri '.' ,dan menuliskan apa adanya
ke layar *)
(* Program ini tidak memanfaatkan EOF. *)
(* Jadi tidak boleh ada file yang hanya mengandung EOF.*)
(* File kosong berisi sebuah karakter '.' *)
(* Kamus *)
var
    f : Text;
    CC : char; (* karalter yang dibaca*)
(* Algoritma *)
begin
    Assign (f,'pitakar.txt');
    Reset (f); (* Buka dengan modus Read Only *)
    if (CC = '.') then
        begin
            writeln ( 'Arsip Kosong ' );
        end else (* CC bukan '.' *)
        begin
            repeat
                read (f,CC);
                write (CC);
            until (CC = '.') ;
        end;
    Close (f) ;
end.

```

```

(* File : Rekamtxt.pas *)
program RekamText;
(* Membaca karakter demi karakter dari keyboard, *)
(* dan menyimpan ke text file. *)
(* Akhiri pembacaan dengan "#" File diakhiri EOF *)
(* Kamus *)
var
    f : text;
    CC : char;
(* Algoritma *)
begin

```

```

Assign (f, 'TextOut.TXT');
Rewrite (f); (* Buka dengan modus rekam *)
write ( 'Masukan karakter, akhiri dengan # '); read (CC);
while (CC <> '#' ) do
  begin
    Write (f,CC);
    read (CC);
    end; (* CC= '#' *)
  Close (f);
end.

(* File : RWInt.pas *)
program TulisInt;
(* Tahap I : Membaca (angka integer) dari keyboard,*)
(* dan menyimpan ke file. Akhiri pembacaan dengan 999 *)
(* Tahap II : membaca dan menulis ke layar, hasil dari penulisan
Tahap I *)
(* Kamus *)
var
  f : file of integer;
  I : integer;
(* Algoritma *)
begin
(* Tahap I : Pembuatan/penulisan file *)
Assign ( f, 'Myint.dat' );
Rewrite (f);
writeln ( 'Input angka yang disimpan, akhiri dengan 999' );
write ( 'Nilai int : ' ); readln (I);
while (I <> 999) do
  begin
    Write (f,I);
    write ( 'Nilai Int : ' ); readln (I);
  end;
Close (f); (* File diakhiri dengan EOF, angka 999 tidak direkam *)

(* Tahap II : Pembacaan file hasil untuk ditulis *)
Reset (f);
if not eof (f) then
  begin
    repeat
      read (f,I);
      Writeln (I:4);
    until eof (f);
  end
else (* eof *)
  begin
    writeln ( 'File kosong' );
  end;
Close (f);
(*Cobalah membuka myint.dat dengan editor text. Apa yang terjadi?*)
end.

```

```

program Frec;
(* File : Frec.pas *)
(* Membaca sebuah file of record, dan menuliskan isinya ke layar *)
(* Kamus *)
type
  Point = record
    x : integer;
    y : integer;
  end;
var
  f : file of Point;
  T : Point; (* karakter yang dibaca *)
(* Algoritma *)
begin
(* Tahap I : membaca data titik untuk direkam *)
  Assign (f, 'mytitik.dat');
  rewrite (f);
  writeln ( 'Input titik, diskhiri dengan x=-999 dan y=-999 ');
(* Baca nilai titik yang akan direkam *)
  write ( 'Absis = ' ); readln (T.x);
  write ( 'Ordinat = ' ); readln (T.y);
  while (T.x <> -999) and (T.y <> -999) do
    begin
      (* rekam ke file *)
      write (f,T);
      (* Baca data berikutnya *)
      write ( 'Absis = ' ); readln (T.x);
      write ( 'Ordinat = ' ); readln (T.y);
    end;
  Close (f);

  (* Tahap II : menuliskan hasil rekaman ke layar *)
  Reset (f);
  if eof (f) then
    begin
      writeln ( 'Arsip Kososng ' );
    end else
    begin
      repeat
        read (f,T);
        (* Tulis data titik *)
        writeln ( 'T (x,y) = ( ', T.x, ', ', T.y, ' ) ');
      until eof (f) ;
    end;
  Close (f);
end.

```

## Unit dalam Turbo Pascal (dan FreePascal)

### Catatan penting :

1. Unit bukan merupakan Pascal standard, bahkan "berlawanan" dengan konsep bahasa Pascal yang "nested", berstruktur blok.
2. Unit baru diajar ketika siswa sudah memprogram dalam skala "agak besar", dan kompleks, dimana sebuah program utuh dipecah-pecah menjadi beberapa file supaya implementasinya dapat dilakukan oleh lebih dari satu programmer
3. Bagian ini adalah bagian penutup (rangkuman dari berbagai konsep kecil yang pernah dipelajari), yang tidak diajarkan jika tidak ada waktu.
4. Unit ini harus diajarkan jika Pascal dipakai sebagai bahasa untuk memprogram pelajaran struktur data. Unit dipakai misalnya untuk merepresentasi ADT. Atau mesin
5. Setiap unit harus mempunyai "driver", yaitu main program yang dipakai untuk mentest semua primitif yang ditulis dalam unit.
6. Implementasi sebuah program yang dibagi-bagi menjadi unit, dengan banyak primitif, harus dilakukan secara incremental. (Nomor tahapan dalam angka romawi) yang ada pada contoh ini menunjukkan urutan pengerjaan/implementasi kode pada unit

```
(* file           : upoint.pas                               *)
(* pemrogram      : Inggriani                               *)
(* tanggal       : 11 Oktober 2000 *)
(* deskripsi      : unit, isinya definisi type point & *)
(*               : beberapa primitif untuk manipulasi typepoint *)
unit upoint;

interface
Const
    Maxpoint=8;
(* Kamus *)
type Point = record
    x:integer; (* absis*)
    y:integer; (* ordinat*)
end;
var
    TabPoint : array [1..MaxPoint] of Point;
    Npoint : integer ; (* banyaknya elemen array 1 s/d 8*)
    F_point : file of Point;

Procedure MakePoint (x,y:integer; var P:Point);
(* Membentuk Point dari x dan y*)
Procedure TulisPoint (P:Point);
(* Menulis nilai sebuah point dengan format *)
(* (x,y) *)
function EqPoint (P1,P2:Point):boolean;
(* mengirim true jika P1=P2; *)
(* yaitu absisnya sama dan ordinatnya sama *)

(***** prosedur manipulasi array TabPoint *****)
procedure BuatTabKosong;
(* membuat tabel kosong, supaya bisa ditambah dg AddElmt*)

procedure AddElmt(P:Point);
(* menambahkan P sebagai elemen tabPoint, N bertambah 1 *)
```

```

(* I.S. tabel kosong sudah ada *)
procedure TulisTab;
(* menuliskan isi tabel*)

(***** prosedur memindahkan array TabPoint ke file***** )
procedure TabToFile;
(* memindah isi tabel TabPOint ke f_point*)

(***** prosedur memindahkan isi file f_point ke *)
(* array TabPoint ***** )
procedure FileToTab;
(* memindah isi tabel TabPOint ke f_point*)

(***** prosedur untuk menulis isi file *)
procedure TulisFile;
(* menulis isi file f_point*)

Implementation
Procedure MakePoint (x,y:integer; var P:Point);
(* Membentuk Point dari x dan y*)
begin
  P.x:=x;
  P.y := y;
end;

Procedure TulisPoint (P:Point);
(* Menulis nilai sebuah point dengan format *)
(* (x,y) *)
begin
  writeln ('(',P.x, ', ', P.y, ' ');
end;

function EqPoint (P1,P2:Point):boolean;
(* mengirim true jika P1=P2; *)
(* yaitu absisnya sama dan ordinatnya sama *)
begin
  EqPoint:= (P1.x=P2.x) and (P1.y=P2.y)
end;

(***** prosedur manipulasi array TabPoint ***** )
procedure BuatTabKosong;
(* membuat tabel kosong, supaya bisa ditambah dg AddElmt*)
begin
  NPoint:=0;
end;

procedure AddElmt(P:Point);
(* menambahkan P sebagai elemen tabPoint, N bertambah 1 *)
(* I.S. tabel kosong sudah ada, N pasti < MaxPOint *)
begin
  NPoint := NPoint + 1;
  TabPoint[NPoint] := P;
end;

procedure TulisTab;
(* menulis isi tabel *)
(* Jika tabel kosong, menulis pesan: tabel kosong*)
VAR
  i:integer;
begin

```

```

if (NPoint=0) then
begin
  writeln('Tabelnya kosong');
end else
begin
  writeln ('Isi tabel');
  for i:=1 to NPoint do
  begin
    TulisPoint(tabPoint[i]);
  end;
end;

end;

(***** prosedur memindahkan array TabPoint ke file*****)
procedure TabToFile;
(* memindah isi tabel TabPoint ke f_point*)
(* nama filenya fpoint.dat *)

(* kamus lokal *)
VAR
  I:integer;

begin
  assign (F_point, 'fpoint.dat');
  rewrite (F_Point);
  I := 1;
  While (I<=NPoint ) do
  Begin
    Write(F_point, TabPoint[I]);
    I := I + 1;
  End;
  Close (F_point);
end;

(***** prosedur memindahkan isi file f_point ke *)
(* array TabPoint *****)
procedure FileToTab;
(* memindah isi tabel TabPoint ke f_point*)
(* file mungkin kosong, menghasilkan tabel kosong *)
(* kamus lokal *)
VAR
  P:Point;

begin
  NPoint := 0;
  Assign (F_Point, 'fpoint.dat');
  reset (F_Point);
  NPoint := 0;
  if eof(F_Point) then
  begin
    writeln ('File kosong');
  end else
  Begin
    writeln('isi file adalah : ');
    Read(F_Point,P);
    TabPoint[NPoint] := P;
  End;
  Close (F_Point);

end;

```

```

procedure TulisFile;
VAR
P: PPoint;
begin
    assign (f_point, 'fpoint.dat');
    reset(f_point);
    if eof(f_point) then
    begin
        writeln ('file kosong');
    end else
    begin
        writeln ('Isi file ');
        repeat
            read(f_point, P);
            TulisPPoint(P);
        until eof (f_point);
        close (f_point);
    end;
end;

begin

end.

(* file           : mainpoint.pas           *)
(* pemrogram      : Inggriani             *)
(* tanggal       : 11 Oktober 2000      *)
(* deskripsi      : main program untuk mentest upoint *)

program mainpoint;
uses upoint;
(* Kamus *)
var
    P:Point;
(* fase III *)
    P1:Point;
(* fase VI *)
    i: integer ;
begin (* algoritma program utama *)
    writeln('main');
(* fase I*)
    MakePoint (5,5,P);
(* Fase II: kode TulisPPoint *)
    TulisPoint (P);
(* fase III: setelah EqPPoint selesai *)
(* tambahkan deklarasi P1 *)
    MakePPoint(0,0,P);
    if EQPoint(P,P1) then
    begin
        writeln('P=P1');
    end
    else begin
        writeln('P tidak sama dnegan P1');
    end;
(* fase IV : buat tabel kosong *)
    BuatTabKosong;
(* fase V: tulis ke file, hasilnya file kosong *)
    TabToFile;
(* Fase VI : akan muncul tulisan tabel kosong *)

```

```
FiletoTab;
(* Fase VII: isi tab dg 5 elemen. Tidak apa-apa isinya semua sama! *)
  for i:= 1 to 5 do
    begin
      AddElmt(P);
    end;
  TulisTab;
(* fase VIII : simpan ke file *)
  TabToFile;
  (* Fase IX *)
  TulisFile;
end. (* main *)
```