

Tutorial Turbo Pascal 7.0

Terdiri dari Sembilan Modul, lengkap dengan contoh dan latihan.



salim_file

Informasi :

- Created by Salim.
- Sumber : [indoprogrammer](#) : Indonesia Programmer.
- Kritik & saran : salim_file@yahoo.co.id
- <http://groups.yahoo.co./group/salim-org>
- www.freewebs.com/salim_web

Selamat Belajar.

Semoga Bermanfaat, Terimakasih.

Tutorial ini bersifat Gratis, tidak boleh dikomersialkan.

Modul 1

Apa itu Turbo Pascal ?

Pascal merupakan suatu bahasa komputer tingkat tinggi yang dibuat sekitar tahun 1970 oleh Niklaus Wirth dan digunakan untuk pendidikan komputer. Nama Pascal diambil dari seorang ahli matematika yang bernama Blaise Pascal yang menemukan mesin hitung pertama. Bahasa Pascal dirancang untuk menyelesaikan masalah dari berbagai kalangan pemakai, mulai dari para mahasiswa, pendidik, dan ilmuwan.

Salah satu kompiler pascal yang terkenal dan tercepat adalah Turbo PASCAL yang dibuat oleh perusahaan Borland (<http://www.borland.com>).

Adakah versi Turbo Pascal untuk pemrograman Visual ?

Pada trend pemrograman Visual, perusahaan Borland mengeluarkan Borland Delphi yang menggunakan dasar dialek Pascal pada lingkungan pemrograman Visual.

Apa artinya Pascal adalah bahasa pemrograman terstruktur ?

Pascal adalah suatu bahasa pemrograman terstruktur. Hal tersebut berarti semua program yang anda buat harus terstruktur dan teratur, dalam hal ini ada pemakaian goto dan jump.

Bagaimana Struktur program pascal yang paling sederhana ?

Struktur program pascal yang paling sederhana adalah :

```
uses ...;

var ...;

begin
    .... {Your program is here}
end.
```

Dapatkah anda memberi contoh sebuah program pascal yang sederhana ?

```
uses crt;

begin
    Clrscr;
    Writeln('Saya sedang belajar Pascal !');
    Writeln('Di PentaCOM');
end.
```

Bagaimana Struktur program pascal yang kompleks ?

Adapun struktur dasar suatu program pascal adalah sebagai berikut :

```
program    ... ;    { Program heading }
uses      ... ;    { Uses clause }
label     ... ;    { Labels }
const     ... ;    { Constants }
type      ... ;    { Types }
var       ... ;    { Variables }
procedure ... ;    { Procedures }
function  ... ;    { Functions }
begin
  statement;      { Statements }
  ...
end.
```

Program heading

Judul program dalam Turbo Pascal bersifat optional dan tidak ada pengaruhnya dalam program. Jika ditulis akan memberikan nama program dan suatu daftar parameter optional dimana program itu berkomunikasi. Daftar itu terdiri dari sederetan indentifier yang diakhiri dengan tanda kurung dan dipisahkan dengan tanda koma.

Contoh :

```
Program Perhitungan(Input,Output);
```

Uses Clause

Bagian uses clause digunakan untuk menentukan library yang dibutuhkan saat proses program. LIBRARY merupakan file penyimpan subroutine yang secara berulang digunakan saat proses program. Library ini diistilahkan sebagai unit dalam Turbo PASCAL yang terdiri dari SYSTEM, OVERLAY, GRAPH, DOS, CRT yang ditempatkan dalam file TURBO.TPL.

Contoh :

```
Uses Crt; (*menggunakan Unit Crt *)
```

Perhatian :

Unit system merupakan unit yang secara otomatis akan disertakan dalam setiap program, unit inilah mengatur semua perintah dasar input dan output pada Pascal . Sedangkan unit-unit lain hanya akan disertakan bila kita pilih pada bagian uses.

Declaration

Bagian ini meliputi deklarasi untuk label, const, type, var, procedure dan function.

Syarat terpenting dalam pembentukan suatu program adalah, bahwa setiap variabel, type non-standard, label, procedure non-standard, serta function non-standard yang dipakai didalamnya harus **nyatakan (deklarasi)** terlebih dahulu pada bagian deklarasi.

Setiap deklarasi tersebut harus bersifat unik (tidak boleh serupa satu sama yang lain).

Penulisan nama dalam deklarasi tersebut harus memenuhi syarat-syarat sebagai berikut :

- panjang nama maximum 63 character, diawali dengan alphabet
- tidak ada special character kecuali tanda garis bawah ("_")
- tidak boleh serupa dengan reserved word (kata tercadang)
- tidak boleh serupa dengan indentifier lainnya.

Variabel Declaration

Setiap variabel yang digunakan dalam program harus didefinisikan terlebih dahulu sebelum digunakan. Cara penulisan ini adalah :

```
Var  
  <nama var> : <type variable>;
```

Statement

Bagian ini merupakan inti dari program. Seluruh perintah dan urutannya serta proses Input/Output dalam program perlu disusun secara teratur oleh penyusun program. Segala yang ada pada bagian uses maupun deklarasi merupakan pendukung terhadap isi program.

Bagaimana membuat keterangan ?

Untuk membuat program anda menjadi mudah dibaca dan dimengerti, maka perlu diberi keterangan yang akan diabaikan oleh kompiler. Untuk membuat keterangan gunakan tanda kurawal { ini adalah keterangan }, atau (* ini adalah keterangan *).

Type data apa saja yang disediakan Turbo Pascal ?

Adapun type variable yang disediakan pada TURBO PASCAL :

1.Ordinal types

Type data yang mempunyai urutan pasti, dan masih terbagi menjadi

- Integer

Type variable yang berguna untuk pengolahan data yang bulat, type ini masih terbagi atas beberapa menurut jangkauan data dan ukurannya :

Type	Jangkauan	Ukuran
Shortint	-128..127	8 bit
Integer	-32768..32767	16 bit

Longint	-2147483648..2147483647	32 bit
Byte	0..255	8 bit
Word	0..65535	16 bit

- Char

Type variabel yang berguna untuk pengolahan character ASCII, type character ini penulisannya ditandai dengan dua buah petik tunggal seperti : 'A', '3','*',#7 untuk menyatakan ' harus ditulis ""

- Boolean

Type varibale yang berguna untuk pengolahan hal yang hanya mempunyai dua ketentuan yaitu benar(TRUE) dan salah(FALSE) saja.

2.Real types

Type variable yang beguna untuk pengolahan data yang tidak bulat, untuk type real ini juga terbagi atas beberapa :

Type	Jangkauan	Ketelitian	Ukuran
Real	2.9e-39..1.7e38	11-12 digit	6 bit
Single	1.5e-45..3.4e38 ³ 7-8	7-8 digit	4 bit
Double	5.0e-324..1.7e308	15-16 digit	8 bit
Extended	3.4e-4932..1.1e4932	19-20 digit	10 bit
Comp	-9.2e18..9.2e18	19-20 digit	8 bit

Untuk pengolahan type variabel diatas di sediakan berbagai jenis operator antara lain :

Operator	Integer Type	Real Type
+	Penjumlahan	Penjumlahan
-	Pengurangan	Pengurangan
*	Perkalian	Perkalian
/	Pembagian	Pembagian
DIV	Hasil bagi	
MOD	Sisa Bagi	

Operator	Integer Type	Boolean Type
NOT	Bitwise Negation	Logical Negation
AND	Bitwise AND	Logical AND
OR	Bitwise inclusive OR	Logical inclusive OR
XOR	Bitwise exclusive OR	Logical exclusive OR
SHL	Bitwise shift-left	
SHR	Bitwise shift-right	

Operator relasi yang mengembalikan hasil Boolean

Operator	Fungsi
:=	Menyatakan nilai
=	Sama dengan
<>	Tidak sama dengan
<	Lebih kecil
>	Lebih besar
<=	Lebih kecil atau sama dengan
>=	Lebih besar atau sama dengan

Bagaimana menampilkan tulisan ke layar ?

Pascal menyediakan perintah Write dan Writeln yang dapat anda gunakan untuk menampilkan tulisan ke layar.

Dapatkah anda memberi contoh program yang menggunakan variabel ?

```
{Deklarasi variabel}
var
  UmurKu      : Byte;
  Keterangan  : String;
(*Program anda*)
begin
  UmurKu:=19;
  Keterangan:='Hi, saya sedang belajar Pascal di PentaCOM';
  Writeln('Saya berumur ',UmurKu,' tahun');
  Writeln(Keterangan);
end.
```

Bagaimana kalau saya ingin membaca masukan dari pemakai ?

Perintah Readln dapat digunakan untuk membaca masukan dari pemakai dan menyimpannya ke suatu variabel.

```
var
  UmurKu      : Byte;
  Keterangan  : String;
begin
  Write('Berapa umur anda ? '); Readln(UmurKu);
  Write('Komentar anda: '); Readln(Keterangan);
  Writeln;
  Writeln('Saya berumur ',UmurKu,' tahun');
  Writeln(Keterangan);
end.
```

Bagaimana membatasi pencetakan tempat desimal untuk data Real type ?

Untuk menentukan jumlah tempat sebelum desimal dan sesudah desimal anda dapat menggunakan :x:y setelah variabel yang akan dibatasi pencetakan nilainya, dimana x adalah jumlah tempat sebelum desimal dan y adalah jumlah tempat setelah desimal. Silahkan coba koding berikut :

```
var
  Pi      : Real;
Begin
  Pi := 22/7;
  Writeln('Bilangan Pi adalah ', Pi);
  Writeln('Bilangan Pi adalah ', Pi:2:3);
end.
```

Modul 2

Unit apa yang perlu saya cantumkan pada bagian uses, kalau saya ingin melakukan pengaturan layar ?

Turbo Pascal menyediakan suatu unit CRT, yang berisi perintah-perintah pengaturan layar seperti membersihkan layar, mengatur warna tulisan, warna background, memindahkan posisi cursor, mode layar, dan membuat window

Ok, saya sudah tahu perintah Clrscr, apa perintah untuk mengatur warna tulisan ?

Pada prinsipnya ada dua cara untuk mengatur warna tulisan yang dicetak dengan perintah Write dan Writeln, yang pertama adalah perintah :

```
TextColor(nomor warna);
```

Dan yang kedua adalah variabel internal dari unit CRT:

```
Textattr := nomor warna;
```

Hallo, anda belum memberitahukan kepada saya nomor warna yang dapat saya gunakan !

Baiklah, berikut ini adalah nomor warna yang mewakili masing-masing warna: 0 (black), 1 (blue), 2 (green), 3 (cyan), 4 (red), 5 (magenta), 6 (brown), 7 (lightgray), 8 (darkgray), (9) lightblue, 10 (lightgreen), 11 (lightcyan), 12 (lightred), 13 (lightmagenta), 14 (yellow), 15 (White).

Saya pernah melihat tulisan warna berkedip-kedip, dapatkah hal tersebut dilakukan ?

Anda dapat saja melakukan hal tersebut dengan menambahkan angka 128 pada nomor warna yang anda inginkan, misalnya :

```
TextColor(4 + 128);  
Writeln('Merah dan berkedip-kedip');
```

Atau :

```
TextColor(Red + Blink);  
Writeln('Merah dan berkedip-kedip');
```

Pada prinsipnya anda dapat menggunakan konstanta dari masing-masing nomor warna seperti black, blue, green, dll, untuk menggantikan pemakaian nomor.

Ok, saya sudah mengerti, bagaimana pula dengan warna latar belakang ?

Pengaturan warna latarbelakang dapat menggunakan perintah :

TextBackground(nomor warna);

Dalam hal ini nomor warna yang berlaku hanya 0 s/d 7.

Tadi anda ada menyinggung tentang memindahkan posisi cursor, apa gunanya dan apa perintahnya ?

Pada lingkungan dos, posisi tulisan dilayar dicetak berdasarkan posisi cursor yang sedang aktif, misalnya sekarang posisi cursor berada di kolom 10, baris 5, maka perintah :

```
Writeln('Sedang belajar Pascal di PentaCOM');
```

Akan menampilkan tulisan tersebut mulai kolom 10 di baris 5.

Jadi misalnya anda ingin mencetak tulisan tersebut pada kolom 20, baris 13, maka anda harus memindahkan cursor keposisi kolom 20, baris 13, diikuti dengan perintah pencetakan tulisan tersebut, contoh :

```
GotoXY(20,13);  
Writeln('Tulisan ini dicetak mulai kolom 20, di baris 13');
```

Bisakah saya mendapatkan nilai posisi cursor yang sedang aktif ?

Bisa, anda bisa menggunakan variabel internal unit CRT, yaitu WhereX, dan WhereY.

```
GotoXY(5,5);  
Write('Turbo');  
GotoXY(WhereX+10,3);  
Write('Pascal');
```

Ngomong-ngomong, berapa sih jumlah kolom dan baris pada layar normal ?

Normalnya 80 kolom, 25 baris.

Saya pernah melihat tulisan layar komputer di bandara Polonia yang lebih besar dari tulisan normal, bagaimana hal tersebut dapat dilakukan ?

Oh, ya. Hal tersebut dapat dilakukan dengan mengatur Mode layar dengan perintah :

```
TextMode(CO40);
```

Dan

```
TextMode(CO80);
```

Untuk menormalkannya.

Contoh :

```
Uses CRT;  
begin  
  TextMode(CO40);  
  Writeln('A Real BIG Characters on screen !');
```



```

Readln;
TextMode(CO80);
Writeln('Back to normal');
Readln;
end.

```

He-he-he, saya sudah mengerti trik yang mereka gunakan.

Tunggu dulu, ada satu hal lagi yang belum diberitahu, yaitu membatasi dari pencetakan dilayar berdasarkan koordinat kiri atas dan koordinat kanan bawah dengan perintah :

```
Window(x1, y1, x2, y2);
```

Dimana x1, y1 adalah koordinat kiri atas dan x2, y2 adalah koordinat kanan bawah.

Contoh :

```

uses Crt;
begin
  ClrScr;
  Writeln('Creating a window from 30,10 to 50,20');
  Window(30,10,50,20);
  Writeln('We are now writing in this small window we just created, we'+
    'can't get outside it when writing long lines like this one');
  Write('Press any key to clear the window');
  ReadKey;
  ClrScr;
  Write('The window is cleared, press any key to restore to fullscreen');
  ReadKey;
  {Full Screen is 80x25}
  Window(1,1,80,25);
  Clrscr;
  Writeln('Back in Full Screen');
end.

```

Ok, apakah anda sudah selesai ?

Belum, sebenarnya masih banyak perintah yang terdapat pada unit CRT, seperti :

```

Sound(frekuensi);
Delay(milidetik);
NoSound;

```

Apa pula fungsinya itu ?

Fungsi perintah Sound untuk membunyikan speaker komputer dengan frekuensi tertentu.

Contoh :

```
Sound(500);
```

Tolong, bunyinya tidak berhenti, walaupun programnya sudah selesai !

He-he-he, anda tidak menggunakan perintah NoSound untuk menghentikan efek dari perintah Sound.

Contoh :

```
Sound(500);
```

```
NoSound;
```

Kok sekarang speaker komputer tidak berbunyi sama sekali ?

Oh, ya, saya lupa menyisipkan delay diantara Sound dan NoSound.

Contoh :

```
Sound(500);  
Delay(1000);  
NoSound;
```

Sebagai programer pemula pascal, saya sering mendapatkan peringatan compiler akan kesalahan Type Mismatch, dapatkah anda menjelaskan hal tersebut ?

Ok, masalah ini sering saya dengar dari orang yang baru belajar pascal. Memang pascal adalah bahasa yang sangat ketat dalam variabel dan type data. Berikut ini saya akan menerangkan konvensi perhitungan dan type data yang dihasilkan :

Konvensi Penjumlahan (+), Pengurangan (-), dan Perkalian (*) :

```
integer dengan integer = integer  
integer dengan real   = real  
real   dengan real   = real
```

Konvensi Pembagian (/) :
Selalu menghasilkan real.

Pembagian bulat dengan Div :
Menghasilkan Integer.

Latihan di Lab :

(*Hangman.pas*)

```
uses crt;  
  
const  
havetry=10;  
  
s: array[0..18] of string=  
( 'hello', 'mouse', 'hacher', 'programmer', 'killer', 'teacher', 'splotchier',  
  'butcher', 'computer', 'pascal', 'house', 'poor', 'children', 'museum',  
  'security', 'spillway', 'stupidity', 'corrigenda', 'freebooter');  
  
var ch: char;  
    gue,b,now wrd,trynow: byte;  
    sl,wrs,alr: string;  
    r: boolean;  
    win: integer;  
  
begin  
win:=0;  
clrscr;  
Writeln('A.Larkin      Hangman      ESC for exit');  
writeln;
```

```

randomize;
repeat
  writeln;
  nowwr:=random(18);
  s1:=s[nowwr];
  gue:=0;
  wrs:='';
  alr:='';
  trynow:=havetry;
  for b:=1 to length(s[nowwr]) do write('_');

  repeat
    gotoxy(20,wherey);
    write('You have ',trynow,' tries ');
    ch:=readkey;
    r:=false;
    if ch<>#0 then
      repeat
        b:=pos(ch,s1);
        if (b<>0) and (s1[b]<>#0) then
          begin
            gotoxy(b,wherey);
            write(s1[b]);
            alr:=alr+ch;
            s1[b]:=#0;
            gue:=gue+1;
            gotoxy(40,wherey);
            write(' ');
            r:=true;
          end else
            if (pos(ch,alr)<>0) and (r=false) then
              begin
                gotoxy(40,wherey);
                write('Already tried!');
              end else
                if (b=0) and (r=false) then
                  begin
                    trynow:=trynow-1;
                    alr:=alr+ch;
                    wrs:=wrs+ch;
                    gotoxy(57,wherey);
                    write('Wrong: ',wrs);
                    gotoxy(40,wherey);
                    write(' ');
                  end;
                until (b=0) or (trynow=0);
            until (ch=#27) or (gue=length(s1)) or (trynow=0);
          if (trynow=0) then
            begin
              win:=win-1;
              delline;
              gotoxy(1,wherey);
              write(s[nowwr]);
              gotoxy(20,wherey);
              write('You lost! Your scores is ',win);
            end else
              if (gue=length(s1)) then
                begin
                  win:=win+1;
                  delline;
                  gotoxy(1,wherey);

```

```

    write(s[nowwrd]);
    gotoxy(20,wherey);
    write('You win!      Your scores is ',win);
end;
until ch=#27;

```

end.

Daftar Frekuensi

Nada	Frekuensi
C	262
D	294
E	330
F	350
G	392
A	440
B	494

Modul 3

Setelah mencoba program Hangman, sesuatu pertanyaan yang timbul adalah mengapa komputer dapat mengetahui huruf yang kita masukan ada dalam tulisan atau tidak ?

Ok, hal tersebut dapat terjadi karena adanya keputusan yang terprogram, artinya komputer dapat mengambil keputusan untuk berdasarkan aliran logika yang telah ditentukan sebelumnya yang dikenal sebagai algoritma.

Bagaimana penulisan perintah pengambilan keputusan pada Pascal ?

Salah satu perintah struktur pengambilan keputusan adalah :

```
If condition Then statement;
```

Dimana condition adalah sesuatu yang bernilai True atau False, dan statement adalah perintah yang akan dijalankan.

Dapatkan anda memberikan sebuah contoh pemakaian perintah If ?

Baiklah, misalnya kita akan membuat program menentukan pembayaran berdasarkan berdasarkan jumlah belanja, jika diatas 100000 (seratus ribu) mendapatkan potongan 3%, maka programnya adalah sebagai berikut :

```

Uses Crt;
Var
Belanja : Real;

Begin
Clrscr;
Write('Jumlah belanja ? '); Readln(Belanja);

```

```

If Belanja > 100000 Then Belanja := Belanja * 0.97;

Writeln('Jumlah yang harus anda bayar ',Belanja:10:2);
Readln;
End.

```

Ok, saya mengerti bahwa pada prinsipnya Statement setelah Then akan dijalankan kalau condition setelah If menghasilkan nilai True. Bagaimana kalau condition True Statement1 dijalankan, dan sebaliknya Statement2 dijalankan ?

Untuk keputusan seperti ini, pascal menyediakan struktur pengambilan keputusan berikut:

```

If condition Then Statement1 Else Statement2;

```

Atau lebih baik ditulis sebagai

```

If conditon Then
    Statement1
Else
    Statement2;

```

Langsung saya buat contoh :

```

Uses Crt;
Var
    Bilangan : Integer;

Begin
    Clrscr;
    Write('Masukan Bilangan ? '); Readln(Bilangan);

    If (Bilangan Mod 2) = 0 Then
        Writeln ('Genap')
    Else
        Writeln ('Ganjil');

    Readln;
End.

```

He-he-he, bagaimana kalau keputusannya lebih dari 2, misalnya 4 Statement yang harus dijalankan berdasarkan masing-masing condition ?

Oh, hal itu bisa dilakukan dengan merangkai beberapa struktur If, misalnya :

```

If condition1 Then
    Statement1
Else
    If condition2 Then
        Statement2
    Else
        If condition3 Then
            Statement3
        Else
            Statement4;

```

Jadi pada prinsipnya adalah terdiri dari tiga struktur If.

Misalnya kita akan membuat nilai huruf dari angka dengan kriteria 80 keatas mendapat A, 70 s/d 79 mendapat B, 60 s/d 69 mendapat C, 50 s/d 59 mendapat D, dan dibawah 49 mendapat E, maka dapat ditulis menjadi :

```
if mark>=80 then
  grade:='A'
else { 79 or below goes here }
  if mark>=70 then
    grade:='B'
  else { 69 or below goes here }
    if mark>=60 then
      grade:='C'
    else { 59 or below goes here }
      if mark>=50 then
        grade:='D'
      else { 49 or below goes here }
        grade:='E';
```

Wah panjang banget, adakah cara lain untuk melakukan hal tersebut ?

Selain struktur kendali If, pascal juga menyediakan suatu struktur Case, yang akan menjalankan statement berdasarkan range tertentu, adapun syntaxnya adalah sebagai berikut :

```
Case variabel Of
  Range1 : Statement1;
  Range2 : Statement2;
  Range3 : Statement3;
  ...
  Else StatementN;
End;
```

Saya akan menggulangi contoh diatas dengan struktur Case :

```
Case mark of
  80..100: grade:='A';
  70..79 : grade:='B';
  60..69 : grade:='C';
  50..59 : grade:='D';
  Else   grade:='E';
End;
```

Lebih sederhana bukan.

Opss, hampir lupa, bagaimana kalau statement yang harus dijalankan pada masing-masing condition lebih dari 1 ?

Ya, benar, sering kita perlu menjalankan beberapa Statement pada masing-masing condition, untuk keperluan tersebut kita dapat memblok perintah-perintah tersebut dengan Begin ... End;

Contoh :

```

If Belanja > 100000 Then
  Begin
    Belanja := Belanja * 0.97;
    Writeln ('Anda berhak mendapat potongan 3%');
  End;

```

Pertanyaan yang terakhir, bagaimana penulisan condition yang terdiri dari beberapa logika ?

Pertanyaan yang tepat sekali, untuk condition yang terdiri dari beberapa logika dapat anda gabungkan dengan operasi AND, OR. Misalnya kita akan mencari tahun kabisat.

In the Gregorian calendar, which is the calendar used by most modern countries, the following rules decides which years are leap years:

1. Every year divisible by 4 is a leap year.
2. But every year divisible by 100 is NOT a leap year
3. Unless the the year is also divisible by 400, then it is still a leap year.

Maka penulisan programnya menjadi :

```

If ((tahun Mod 4) = 0) And Not (tahun Mod 100 = 0) Or (tahun Mod 400) = 0 Then
  Writeln ('Tahun Kabisat !')
Else
  Writeln ('Bukan Tahun Kabisat !');

```

Latihan 3

1. What does Pascal provide in order to perform conditional branching ?
2. Explain briefly each syntax !
3. How could we do nested branching ? Is it legal ?
4. Tell us how to trap arrow keys !
5. How could we trap function keys (F1 through F10) ? How is it similar to trapping arrow keys?
6. Determine the output if i=5 and j=3 for the following excerpt :

```

if i<4 then writeln('Need more experience ...')
else
begin
  writeln('Yes, you have it !');
  if j<=3 then writeln('But, I would prefer better ones.')
  else writeln('Superb !');
  writeln('Congratulations !');
end;

```

```
writeln('Thank you for using this program !');
```

7. Make a program to ask ages. Classify the ages into these categories :

```
* age < 2          ==> "You are a baby !"
* 2 to age < 12   ==> "You are a kid !"
* 12 to age < 18  ==> "You are a teenager !"
* 18 to age < 24  ==> "You are a young adult !"
* 24 to age < 40  ==> "You are an adult !"
* 40 to age < 55  ==> "You are middle aged !"
* 55 to age < 65  ==> (Give comments yourself)
* age to 65       ==> (Give comments yourself)
```

Silahkan mencoba :

Readkey is a function, returning a char value. OK, let's see how it works !

```
program test_readkey;
uses crt;
var
  c : char;
begin
  c:=readkey;
  case c of
    #8 : writeln('You presses backspace');
    #9 : writeln('You presses tab');
    #13: writeln('You presses enter');
    #27: writeln('You presses escape');
    #32: writeln('You presses space');
    else writeln('You presses other key');
  end;
end.
```

Let us detect the arrow keys. Arrow keys, just like function keys, are extended. It means it generates a special codes. Now, here is the code to trap them :

```
program trap_arrow;
uses crt;
var
  c : char;
begin
  c:=readkey;
  if c=#0 then { If extended codes, }
  begin
    c:=readkey; { read the code once more }
    case c of
      #72: writeln('Up arrow');
      #75: writeln('Left arrow');
      #77: writeln('Right arrow');
      #80: writeln('Down arrow');
    end;
  end;
end.
```


Modul 4

Hallo, ketemu lagi, pada pertemuan ini kita akan membahas tentang Looping.

Apa itu Looping

Secara sederhana looping diartikan sebagai proses berulang terhadap statement maupun serangkaian statement lebih dari satu kali.

Aduh, kurang jelas, dapatkah anda memberikan contoh pemakaian looping dari program yang pernah kita buat sebelumnya ?

He-he-he, mungkin anda sudah lupa dengan program Hangman yang pernah kita buat pada pertemuan kedua, disana komputer secara berulang menanyakan huruf tebakan kita sebanyak 10 kali kesempatan, dalam hal ini kita telah menggunakan looping.

Ok, sekarang saya sudah punya gambaran. Apakah hal tersebut tidak dapat dilakukan tanpa looping ?

Tentu saja bisa, tetapi akibatnya program kita menjadi panjang dan tidak efisien, coba bayangkan misalnya anda ingin mencetak tulisan PentaCOM sebanyak 10 baris di layar, bisa saja anda menggunakan perintah `Writeln('PentaCOM');` sebanyak 10 kali. Bukankah lebih baik kita memerintahkan komputer untuk secara berulang menjalankan perintah `Writeln('PentaCOM');` sebanyak 10 kali.

Benar juga ya, tetapi bagaimana struktur perintah perulangan pada pascal ?

Salah satu struktur perintah perulangan pada pascal adalah yang menggunakan counter.

```
For variabel := nilai awal To nilai akhir Do Statement;
```

Atau

```
For variabel := nilai awal DownTo nilai akhir Do Statement;
```

Dimana variabel harus ordinal type

Contoh :

```
Var I : Integer;  
Begin  
For I := 1 To 10 Do Writeln('PentaCOM');  
End.
```

Tapi tadi saya lihat yang satu pakai To dan yang satu lagi pakai DownTo, apa sih bedanya ?

Kita menggunakan To kalau nilai awal < nilai akhir, dan sebaliknya pakai DownTo

Contoh :

```
Var I : Integer;  
Begin  
For I := 10 Downto 1 Do Writeln('PentaCOM');  
End.
```

Apa maksudnya variabel counter harus ordinal type ?

Maksudnya variabel counter harus berupa salah satu type seperti Byte, Shortint, Integer, Longint, Char, Boolean.

Misalnya Ch adalah Char, maka dapat dilakukan

```
For Ch := 'a' to 'z' Do Write(Ch, ' ');
```

Dan bahkan misalnya Bo adalah boolean, dapat juga

```
For Bo :=false to true do writeln(Bo);
```

Adakah struktur perulangan lainnya pada bahasa pascal ?

Ya, ada lagi, yaitu yang melakukan perulangan yang menggunakan condition.

While condition Do Statement;

Contoh :

```
I := 1;  
While I <= 10 Do  
  Begin  
    Writeln('PentaCOM');  
    I := I + 1;  
  End;
```

Apakah maksudnya I := I + 1; saya belum pernah melihat persamaan seperti ini secara matematika !

Ya, memang hal ini tidak mungkin ada secara matematika, tetapi pada dunia pemrograman artinya menaikkan nilai variabel I sebesar 1. Supaya tidak membingungkan pascal menyediakan suatu function Inc(variabel); sebagai pengganti I:= I + 1;

Ok, sekarang saya sudah jelas dengan struktur looping pada bahasa pascal.

Tunggu dulu, masih ada satu struktur perulangan dengan condition, yaitu :

```
Repeat  
  Statement;  
  ...  
Until condition;
```

Contoh :

```
I := 1;
Repeat
  Writeln('PentaCOM');
  Inc(I);
Until I <= 10;
```

Kalau begitu ya sama saja dengan yang While, bukankah begitu ?

Tentu saja ada perbedaannya, coba perhatikan contoh berikut

```
I := 11;
While I <= 10 Do
  Begin
    Writeln('PentaCOM');
    Inc(I);
  End;
```

Dan

```
I := 11;
Repeat
  Writeln('PentaCOM');
  Inc(I);
Until I <= 10;
```

Pada contoh 1, tidak akan mencetak apa-apa, sedangkan pada contoh 2 akan mencetak tulisan PentaCOM minimal 1 kali.

Pada prinsipnya struktur While melakukan pemeriksaan awal, dan struktur Repeat melakukan pemeriksaan di akhir.

Bagaimana kalau kita ingin keluar dari pertengahan looping walaupun conditionnya masih terpenuhi ?

Mudah saja, gunakan perintah Break, dan anda juga dapat menggunakan perintah Continue untuk kembali ke While ataupun Repeat.

Contoh :

```
I := 1;
While I <= 10 Do
  Begin
    Writeln('PentaCOM');
    I := I + 1;
    Break;
  End;
```

Akan mencetak PentaCOM sekali saja, karena begitu ketemu perintah Break, maka proses keluar dari looping. Bagaimana dengan yang berikut ini :

```
I := 1;
While I <= 10 Do
  Begin
```

```
Writeln('PentaCOM');  
I := I + 1;  
Continue;  
End;
```

Pelajaran tambahan :

Random dan randomize

Kedua berfungsi menghasilkan bilangan random, bilangan random sering digunakan untuk mensimulasikan dunia nyata yang penuh dengan ketidakpastian. Kita tidak akan mendiskusikan hal ini, tetapi pemakaian kedua perintah tersebut.

Randomize digunakan untuk menginisialisasi suatu bibit random secara acak yang cukup dipanggil sekali saja.

Random digunakan untuk mendapatkan bilangan random(bilangan), yang akan menghasilkan bilangan 0 s/d bilangan-1. Misalnya num:=random(50); berarti variabel num mungkin berisi salah satu bilangan 0 s/d 49.

Cobalah contoh berikut :

```
var  
  i : byte;  
begin  
  randomize;  
  for i:=1 to 10 do  
    write(random(30),' ');  
end.
```

Coba hilangkan "randomize", dan jalankan beberapa kali, bagaimana dengan bilangan yang dihasilkan apakah sama pada setiap percobaan ?

Jadi, selalu gunakan "randomize" untuk menjamin kerandoman bilangan yang dihasilkan pada setiap run.

Latihan IV

1. Explain the characteristics of all three loop syntaxes in Pascal !
2. Explain the differences between them !
3. Suppose the output is :
1
2
3
4
5
6

Write an excerpt using all three syntaxes !

4. Explain how to nest one syntax to another !

Latihan di laboratorium

1. Input : 5
Output : 1 2 3 4 5 4 3 2 1
Input : 1
Output : 1

2. Input : 5
Output : 1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5

3. Input : 5
Output : 1 2 3 4 5
2 3 4 5 1
3 4 5 1 2
4 5 1 2 3
5 1 2 3 4

4. Input : 5
Output : 1
1 2
1 2 3
1 2 3 4
1 2 3 4 5

5. Input : 5
Output : 1
2 2
3 3 3
4 4 4 4
5 5 5 5 5

6. Input : 5
Output : 1 1 1 1 1
2 2 2 2 2
3 3 3 3 3
4 4 4 4 4
5 5 5 5 5

7. Input : 5
Output : 1
1 2 1
1 2 3 2 1
1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1
1 2 3 4 4 4 4 4 3 2 1
1 2 3 3 3 3 3 3 3 3 2 1
1 2 2 2 2 2 2 2 2 2 2 2 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

8. Input : 5
Output : 1 1 1 1 1
1 2 2 2 1
1 2 3 2 1
1 2 2 2 1
1 1 1 1 1

9. Make fibonacci series. If input is n, write all the series up to n numbers :

Input : 10

Output : 1 1 2 3 5 8 13 21 34 55

The first and the second numbers of fibonacci series are 1. The third is the sum of the first and the second. The fourth is the sum of the second and the third. The fifth is the sum of the third and the fourth, so on.

10. Make a factor tree.

Input : 100

Output :

```
      100
     /  \
    2    50
       /  \
      2  25
         /  \
        5  5
```

Input : 1001

Output :

```
      1001
     /  \
    7   143
       /  \
      11  13
```

Input : 5

Output : 5 is a prime !

Tugas Rumah :

Create a game just like a space invader (you've got to know that), but in normal text mode (80x25). Use colors as pretty as possible. Normally space invader have a matrix of enemies, but I make allowances to you. You just make only one enemy. If the enemy is destroyed, it just reappears at a different place and the player gains a point. Each players have 3 ships to play. If all ships are used up (shot down) you show "Game Over" message and ask if the player want to play again. Each enemy ship scores 10. Player got an extra ship when reaches 500 and its multipliers (1000, 1500, 2000,...). Players can only have one shot. Before the fire is out, players can not trigger another shot. The same rule applies for the enemy. Player could only move his/her ship horizontally (left or right), not vertically or diagonally. The enemy may come closer to the ship after it reaches the edge (of left or right). I mean that the enemy ship go left or right, after it touches the edge, it advances a bit -- just like a normal space invader game. Make it as fancy as possible.

Modul 5

Hallo, ketemu lagi, adakah pertanyaan sejauh ini ?

Ya, setelah saya pelajari program Hangman, terutama pada bagian var dimana saya temui deklarasi variabel dengan awalan Array of, apa sih Array itu ?

Baiklah, array adalah variabel yang dapat menampung sejumlah data yang ditandai dengan suatu index pada masing-masing elemennya.

Contoh :

```
Var  
Nilai : Array [1..10] of Integer;
```

Dalam hal ini variabel Nilai adalah suatu array yang memiliki 10 element, yaitu Nilai[1], Nilai[2], Nilai[3], ..., Nilai[10]. Pada masing-masing elemennya dapat menampung satu nilai Integer.

Contoh :

```
Nilai[1] := 60;  
Nilai[2] := 75;  
...  
Nilai[10] := 90;  
Writeln(Nilai[1]);  
Writeln(Nilai[2]);  
...  
Writeln(Nilai[10]);
```

Kalau begitu kita juga dapat melakukan hal yang sama dengan memesan 10 variabel, misalnya Nilai1, Nilai2, Nilai3, dst, dapatkah kira-kira anda memberikan contoh keunggulan dari pemakaian Array ?

Ya, memang hal tersebut dapat dilakukan, tetapi sesuatu yang perlu kita perhatikan adalah kemudahan pengolahan data Variabel tersebut, misalnya kita akan menjumlahkan total nilai dari variabel tersebut tanpa array maka dapat ditulis :

```
Total := Nilai1+Nilai2+Nilai3+Nilai4+Nilai5+Nilai6+Nilai7+Nilai8+Nilai9+Nilai10;
```

Dan kalau pakai array :

```
Total := 0;
For I:= 1 To 10 Do Total := Total + Nilai[I];
```

Dan bagaimana kalau ada elemennya ada 100. Bisa anda bayangkan.

Adakah bentuk array yang lain, selain yang anda sebutkan ?

Ada, array yang telah kita sebutkan tersebut adalah array 1 dimensi, kita dapat juga membuat multi dimensi sesuai dengan kebutuhan, contoh :

```
Var
Matrix : Array [1..3,1..3] Of Integer;
```

Adalah contoh array 2 dimensi, yang terdiri dari elemen Matrix[1,1], Matrix [1,2], Matrix[1,3], Matrix[2,1], Matrix[2,2], Matrix[2,3], Matrix[3,1], Matrix[3,2], dan Matrix[3,3].

Jadi jumlah elemennya adalah $3 \times 3 = 9$ elemen.

Dapatkah kita buat 3 Dimensi, 4 Dimensi, dst ?

Oh, bisa, hal ini tergantung bagaimana kita mendeklarasikan variabel array tersebut.

Adakah hal-hal yang khusus dalam mendeklarasikan array pada Pascal ?

Ada, anda bisa mendeklarasikan array seperti :

```
x : Array[3..10] of Integer; {elemennya x[3], x[4], x[5], ..., x[10]}
idx : Array['A'..'Z'] of Integer;
a : Array['a'..'z'] of Byte;
n : Array[byte] of Integer; { The same as array[0..255] of integer; }
```

Saya bingung dengan yang Array ['A'..'Z'] of Integer, bagaimana kira-kira cara pemakaiannya ?

Cara pemakaiannya seperti array bisanya, Cuma indexnya ditulis sebagai character, contoh:

```
Idx['A'] = 100;
```

Dan kalau pakai looping :

```
For c:='A' to 'Z' do
  idx[c]:= 0;
```

Latihan V

1. What is an array ?
2. What is it for ?
3. How is the declaration ?
4. Explain how to use it !
5. How can we declare two or more dimension in array and explain the usage.
6. Suppose we have `n : array[char]` of byte; Is it valid ? Explain.

Latihan di Laboratorium

1. Make a Pascal triangle, example :

```
Input   : 7
Output  :
          1
         1 1
        1 2 1
       1 3 3 1
      1 4 6 4 1
     1 5 10 10 5 1
    1 6 15 20 15 6 1
```

2. Suppose you have an array 100 elements of random numbers. Find its maximum and minimum value of the elements.
3. Make matrix operations : addition, subtraction, multiplication and division. You may limit the maximum dimension of the matrix.

Latihan di rumah :

Make a jackpot game. It has 4 digits. Each digits is represented by numbers. The players has \$500 at start and must place their bet of \$20 every game. Winning points :

- If any two numbers are the same : bet returned plus 10% bonus
- If any three are the same : bet returned plus 25% bonus
- If all four numbers are the same : bet returned doubled
- If all four are in sequence e.g. 1 2 3 4 or 5 4 3 2 : bet returned plus 50% bonus
- If all four are even or all odd e.g. 1 7 5 3 or 2 6 4 8 : bet returned plus 5% bonus
- If all four are prime numbers : bet returned plus 10% bonus
- If two numbers are the same and the other two are the same too :
bet returned plus 75% bonus

If none of above : bet lost.

Remember, in one digit, possible primes are : 2, 3, 5, and 7 only.

Example :	Bonus :
1 1 2 2	75%
1 2 2 1	75%
2 1 2 1	75%
2 1 2 2	25%
9 5 3 7	5%
7 2 3 5	10%
4 3 2 1	50%
1 8 3 2	lost

Space invader revisited.

Last time you made space invader. Now you make a full version of Space Invader with matrices of enemies. You may extend it with bonus stages or UFO. All rules remain the same.

Modul 6

Pada contoh maupun tugas modul-modul sebelumnya kita telah banyak menggunakan string, apakah sebenarnya string pada Pascal ?

Sebenarnya string adalah suatu array dari char, maksudnya anda dapat mengambil masing-masing character dari suatu string dengan memperlakukannya sebagai elemen dari array.

Contoh :

```
Var
Nama : String;
Begin
Nama := 'PentaCOM';
Writeln(Nama[1]);
Writeln(Nama[8]);
Nama[6] := 'K';
Writeln(Nama);
End.
```

Maka Nama[1] adalah huruf 'P', dan Nama[8] adalah huruf 'M'.

Sedangkan Nama[6] := 'K'; akan menyebabkan string nama berubah menjadi 'PentaKOM'.

Oh, begitu saja, jadi kita sudah selesai hari ini ?

Tunggu dulu, sebenarnya kalau kita dapat juga mendeklarasikan string dengan :

```
Var
Nama : String;
Alamat : String[40];
```

Dalam hal ini variabel Nama dapat menampung maksimal 255 character, sedangkan untuk variabel Alamat dapat menampung maksimal 40 character.

Jadi kita dapat mendeklarasikan variabel string dengan panjang tertentu, berapa panjang maksimal yang diperbolehkan ?

Sebenarnya kalau kita tidak menentukan panjang string ketika deklarasi, maka otomatis pascal menyediakan 255 character untuk string kita, dalam hal ini merupakan panjang maksimal yang diperbolehkan untuk suatu type data string.

Bolehkan kita memesan variabel array string ?

Boleh, anda dapat melakukannya.

Contoh :

```
NamaSiwa : Array[1..10] Of String[30];
```

Pada program Hangman, saya melihat adanya pemakaian fungsi Length, Pos, apa fungsinya, dan adakah yang lain ?

Ya, itu adalah fungsi bantu yang disediakan oleh Pascal untuk melakukan pengolahan data string,

Adapun fungsi bantu untuk string adalah sebagai berikut :

Length, mendapatkan panjang string

Syntax : length(s)

Contoh : : n:=length(s);

Misalnya s:='Apa kabar?'; n akan berisi 11.

Copy, mendapatkan bagian dari suatu string.

Syntax : copy(s,from,howmuch)

Contoh : : st:=copy(s,5,5);

Menggambil 5 character mulai posisi ke 5

Misalnya s:='Apa kabar?'; st akan berisi 'kabar'.

Pos, mengambil posisi substring dari suatu string.

Syntax : Pos(substr,s)

Contoh : : n:=pos('kabar','Apa kabar?'); { n:=5; }

Jika substring tidak ditemukan, maka akan mengembalikan 0.

Val, mengkonversi string menjadi numerik.

Syntax : val(strvar,numvar,errorcode)

strvar adalah suatu variabel string yang akan di konversi
numvar adalah variabel integer atau real
errorcode adalah variabel integer yang akan berisi nilai kesalahan, jika berisi 0 sukses.

Contoh :

```
Var
  s : string;
  e : integer;
  r : real;

Begin
  Write('Enter a number : '); readln(s);
  val(s,r,e);
  if e<>0 then
    writeln('Error at position : ',e);
  else
    writeln('That was : ',r:4:3);
end.
```

Str, mengkonversi numerik menjadi string.

Syntax : (numvar,strvar)

Contoh :

```
var
  s : string;
  i : integer;
begin
  write('Input an integer : '); readln(i);
  str(i,s);
  writeln('That was : ',s);
end.
```

Jika anda bekerja dengan type real, anda perlu melakukan pemformatan sebelum konversi ke string.

Contoh :

```
var
  s : string;
  r : real;
begin
  write('Input a real : '); readln(r);
  str(r:4:3,s);
  writeln('That was : ',s);
end.
```

Concat, menggabung dua string atau lebih

Syntax : concat(s1,s2,...,sn)

Contoh : st:=concat(s1,s2);

Jika s1 adalah 'Penta', dan s2 adalah 'COM', maka st menjadi 'PentaCOM'

Sebenarnya kita juga dapat menggunakan operator + untuk menggabung string.

Contoh : `st:=s1 + s2;` {adalah sama dengan `st:=concat(s1,s2);`}

Insert, menyisip suatu string kedalam string lain dari posisi tertentu

Syntax : `insert(source,target,index)`

Contoh : :

```
var
  s1, s2 : string;
begin
  s1:='not ';
  s2:='I do love you';
  insert(s1,s2,6);
  writeln(s2);      { I do not love you }
end.
```

Delete, menghapus sejumlah character dari string mulai posisi ke i

Syntax : `delete(s,i,n);`

```
var
  s : string;
begin
  s:='I am not responsible for that !';
  delete(s,6,3);
  writeln(s);      { I am responsible for that }
end.
```

Latihan 6

1. What is a string ?
2. Explain what the relationship between string and character in pascal !
3. What is the maximum length of a string ?
4. How is the declaration ?
5. Explain how to merge two string, using function and operator !
6. Explain how to convert string to value and vice versa !
7. Explain how to know the length of a string !

Latihan di laboratorium

8. Make programs for several task like this:
 - a. Make all letters in a string upcase.
 - b. Eliminate trailing spaces.
 - c. Eliminate leading spaces.
 - d. Change leading spaces with leading zeroes.
 - e. Count the number of word in a sentence
9. Make a text animation. Display a sentence in the center of the screen. And scroll it Left To Right !
10. Make a text animation. Display a sentence in the center of the screen. And scroll it Right To Left !
11. Make a text animation. Display a sentence in the center of the screen. Let all characters fall one by one. Skip the spaces !

Latihan di rumah

Make a simple calculator that can perform addition, subtraction User input the equation then press enter. You should give the answer. Example :

```
Input   : 5+10
Output  : 15
```

Modul 7

Hallo, masihkah anda ingat bahwa pascal adalah bahasa pemrograman yang terstruktur, yang artinya program yang kita buat harus terstruktur dan teratur. Salah satu usaha untuk usaha kearah sana adalah memecah program kita menjadi sub-sub program dengan menggunakan procedure dan function.

Apa itu procedure dan function ?

Procedure dan function merupakan sub-program yang memiliki bagian deklarasi dan begin-end; tersendiri layaknya suatu program pascal. Tujuan pemakaian procedure maupun function adalah untuk memecah program kita menjadi kesatuan logika yang lebih kecil (sub-sub program) untuk menghindari perulangan pengetikan dan memudahkan pembacaan.

Contoh pemakaian procedure :

```
uses crt;

begin
  clrscr;
  writeln('ABC');
  writeln('Press any key when ready ...');
```

```

    readkey;
    writeln('Hey, let''s pause !');
    writeln('Press any key when ready ...');
    readkey;
    writeln('Pause again !');
    writeln('Press any key when ready ...');
    readkey;
end.

```

Dengan memecah program kita menjadi kesatuan yang lebih kecil :

```

uses crt;

procedure pause;
begin
    writeln('Press any key when ready ...');
    readkey;
end;

begin
    clrscr;
    writeln('ABC');
    pause;
    writeln('Hey, let''s pause !');
    pause;
    writeln('Pause again !');
    pause;
end.

```

Contoh pemakaian function :

```

Var
I : Byte;
Akar,Bagi : Word;
Begin
    Writeln(Upper('Bilangan Prima dari 1 s/d 100'));
    Writeln(Replicate('-',30));
    For I := 1 To 100 do
        If (Bil > 1 And Bil < 4) Then
            Write(I : 4)
        Else
            Begin
                Akar := Round(Sqrt(Bil));
                Bagi := 3;
                While (Bagi <= Akar) And (Bil Mod Bagi <> 0) Do
                    Bagi := Bagi + 1;
                If Bagi > Akar Then
                    Write(I : 4);
            End;
    End.

```

Dan contoh berikut yang menggunakan function :

```

Function IsPrime (Num : Longint) : Boolean;
Var
    Sq ,Dv : Word;          (*local variable *)
Begin
    If Num < 2 Then
        IsPrime := False

```

```

Else
  If Num < 4 Then
    IsPrime := Bil > 1
  Else
    Begin
      Sq := Round(Sqrt(Num));
      Dv := 3;
      While (Dv <= Sq) And (Num Mod Dv <> 0) Do
        Dv := Dv + 1;
      IsPrime := Dv > Sq;
    End;
End;

(* Main program*)
Var
I : Byte;          (*local variable*)
Begin
  Writeln(Upper('Prime number from 1 to 100'));
  Writeln(Replicate('-',30));
  For I := 1 To 100 do
    If IsPrima(I) Then
      Write(I : 4);
End.

```

Tadi saya melihat ada keterangan variabel local, apa maksudnya ?

Pada program utama, procedure maupun function dapat memiliki variabel masing-masing, dimana ruang lingkup pemakaiannya hanya berlaku pada program utama, procedure maupun function dimana variabel tersebut dideklarasikan.

Contoh :

```

procedure foo;
var a : byte;
begin
  a:=10;
  writeln(a);
end;

begin      { main begin...end block }
  foo;
  a:=15;   { This will be an error, since a is only known inside foo }
end.

```

Kalau begitu berarti ada variabel global dong ?

Ya, benar, global variabel adalah variabel yang ruang lingkup pemakaiannya meliputi seluruh program, baik program utama, procedure maupun function.

Contoh:

```

var
  b : byte;      { global variabel }

procedure foo;
var a : byte;
begin
  a:=10;
  b:=15;        { This is legal }
end.

```

```

end;

begin          { main begin...end block }
  foo;
  a:=15;      { This is illegal }
  b:=5;      { This is perfectly legal }
end.

```

Jadi variabel global harus dideklarasikan di atas semua procedure maupun function.

Bagaimana kalau ada variabel local dan variabel global yang sama namanya ?
 Global variabel tersebut akan di override. Pada suatu program yang terstruktur dengan baik, usahakan untuk menggunakan variabel Global seminimal mungkin.

Contoh :

```

uses crt;
var
  a : byte;          { This is the global variable }

procedure foo;
var
  a : byte;          { This is local variable }

begin
  a:=15;
  writeln('foo is ',a); { Will print 15}
end;

begin
  a:=10;
  writeln(a);          { Will print 10}
  foo;
  writeln(a);          { Still print 10}
end.

```

Dari tadi anda menyebutkan procedure dan function, apa sih bedanya ?
 Pada prinsipnya function mengembalikan nilai.

Syntax untuk procedure :

```

Procedure namaprocedure (deklarasi parameter);
Deklarasi variabel local
begin
  :
  { commands }
  :
  :
end;

```

Syntax untuk function :

```

function namafunction (deklarasi parameter) : returntype;
Deklarasi variabel local
begin
  :
  { commands }
end;

```



```

:
namafunction := return value;
end;

```

Coba perhatikan kembali contoh Isprima untuk lebih jelas.

Sebenarnya apa fungsi deklarasi parameter pada procedure maupun function ?

Pada deklarasi bagian deklarasi parameter kita menentukan parameter dan type data dari masing-masing parameter yang akan dikirim kedalam procedure maupun function.

```

procedure makewin(x1,y1,x2,y2 : byte);
var
  i,j : byte;
begin
  gotoxy(x1,y1); write(#201);   {top}
  for i:=x1+1 to x2-1 do write(#205);
  write(#187);
  for i:=y1+1 to y2-1 do      {middle}
  begin
    gotoxy(x1,i); write(#186);
    for j:=x1+1 to x2-1 do write(' ');
    write(#186);
  end;

  gotoxy(x1,y2); write(#200);   {bottom}
  for i:=x1+1 to x2-1 do write(#205);
  write(#188);
end;

begin
  makewin(1,1,30,8);
  makewin(10,10,60,18);
end.

```

Dan contoh untuk function :

```

function factorial(n : byte):longint;
var
  i      : byte;
  result : longint;
begin
  result:=1;
  for i:=1 to n do
    result:=result*i;
  factorial:=result;
end;

var
  x : byte;
begin
  writeln('Enter a value : '); readln(x);
  writeln(x,'! is ',factorial(x));
end.

```

Kalau kita bisa mengirim nilai kedalam procedure maupun function, apakah bisa juga sebaliknya ?

Pada contoh diatas nilai parameter di kirim secara PASS BY VALUE. Kita dapat juga mengirim parameter secara PASS BY REFERENCE dengan menambah Var pada saat deklarasi masing-masing parameter, dalam hal ini yang dikirim kedalam procedure maupun function adalah alamat dari variabel tersebut.

Contoh :

```

(*Pass by value*)
procedure foo(a : byte);
begin
  writeln(a); {15}
  a:=10;
  writeln(a); {10}
end;

var
  x : byte;
begin
  x:=15;
  writeln(x); {15}
  foo(x);
  writeln(x); {15}
end.

(*Pass by reference*)
procedure foo(var a : byte);
begin
  writeln(a); {15}
  a:=10;
  writeln(a); {10}
end;

var
  x : byte;
begin
  x:=15;
  writeln(x); {15}
  foo(x);
  writeln(x); {10}
end.

```

Bolehkan suatu function memanggil dirinya sendiri ?

Boleh, dan hal tersebut banyak ditemukan pada persamaan matematika, misalnya dalam menghitung faktorial.

```

5! = 4! * 5
4! = 3! * 4
2! = 1! * 2
1! = 1

```

Jadi kita secara programming dapat dilakukan dengan :

```

function factorial (n:byte):factorial;
begin
  if n<2 then      { This is the terminating condition }
    factorial:=1;
  else
    factorial:=n*factorial(n-1); { This is the recursive part }
end;

var
  x : byte;
begin
  writeln('Enter a value : '); readln(x);
  writeln(x,'! is ',factorial(x));
end.

```

Sesuatu hal yang perlu diperhatikan dalam pembuatan fungsi rekursif adalah harus adanya suatu titik akhir, sehingga rekursif dapat selesai.

Latihan 7

1. Explain the syntax of procedures and functions.
2. Explain the differences between them.
3. Explain the differences between pass by value and pass by reference. Give an example.
4. Can procedures nest each other ? Explain and give example !
5. Explain the differences between global variable and local variable.
6. Explain the rule of calling procedures / functions.
7. Explain how recursive.

Latihan di laboratorium

8. Make a recursive algorithm of fibonacci series !
9. Make a guessing number game using functions and procedures. You hold a random number between 1 to 100, and user have to guess. If the guess is smaller than it suppose to be, you should say "lower". If the guess is higher, say "higher". If it is the same, then user wins. User has 8 chances to guess. If chances are used up, game over. Score is number of chances left times 10. If user wins, get another number and go on. Don't forget to restore the chances back to 8. If game over, ask him/her to play again.

Here are some hint :

```
function compare(number, guess : byte):shortint;  
function advice(comparison : shortint):string;  
function asknumber: byte;  
function yesno : boolean;  
function score(guessleft : byte):longint;  
procedure writeat(x,y : byte; sentence:string);  
procedure setupscreen;
```

Latihan di rumah

Make a program to translate a number into word using function.

Input : 1537000

Output : Satu juta lima ratus ribu tiga puluh tujuh ribu rupiah.

Modul 8

Hallo, ketemu lagi, apakah anda telah siap ? Ok saya rasa siap, pada hari ini kita akan membahas beberapa fasilitas yang tersedia pada pascal dan akan membantu membuat program anda menjadi lebih terstruktur dan mudah di maintenance.

Kalau begitu artinya, walaupun tidak menggunakan fasilitas kita tetap dapat membuat program ?

Benar, tetapi sebagai programmer yang baik, tentu saja anda harus mampu menghasilkan program yang terstruktur dan mudah dimaintenance.

Apa saja fasilitas yang anda maksud tersebut ?

Pertama kita akan membahas tentang Constants, Set, dan Type.

Oh, konstanta, di sekolah kami ada belajar $\pi = 3.1415926513...$, $e = 2.7182818284529...$, apakah maksud anda seperti itu ?

Ya, benar, pengertian konstanta di pascal hampir sama dengan konstanta di dunia matematika maupun fisika, yaitu besaran-besaran yang memiliki ketetapan-ketetapan tertentu. Tetapi pemakaian konstanta di pascal programmer bebas membuat konstanta sesuai dengan kebutuhan programnya.

Dapatkah anda memberi sebuah contoh program yang menggunakan konstanta ?

Contoh :

```
Uses crt;
Const
    PPN = 0.1;           {deklarasi konstanta}
                        {PPN 10%}
Var
    Penjualan, SetelahPPN : Real;
Begin
    Write('Nilai penjualan :') Readln(Penjualan);
    SetelahPPN = NilaiPenjualan * (1+PPN);
    Writeln('Nilai penjualan setelah PPN adalah ', SetelahPPN:10:2);
End.
```

Kalau begitu bukankan sama saja dengan menulis langsung `SetelahPPN = NilaiPenjualan * (1+0.1);` dan apa hubungannya dengan membuat program kita lebih mudah di maintenance ?

Pada prinsipnya sama, tetapi coba anda bayangkan, misalnya terjadi perubahan nilai PPN menjadi 12.5%, berdasarkan contoh diatas kita tinggal mengubah

```
Const
    PPN = 0.125;        {deklarasi konstanta}
                        {PPN 12.5%, tidak pakai :}
```

Dalam hal ini anda tidak perlu mengubah isi program, sehingga maintenance program anda menjadi mudah.

Apakah ada kegunaan lain dari konstanta pada pascal ?

Ada, yaitu memberikan nilai awal kepada variabel, contoh yang diambil dari Hangman Modul 2 :

```
const
havetry=10;

s: array[0..18] of string=
('hello', 'mouse', 'hacher', 'programmer', 'killer', 'teacher', 'splotchier',
 'butcher', 'computer', 'pascal', 'house', 'poor', 'children', 'museum',
 'security', 'spillway', 'stupidity', 'corrigenda', 'freebooter');
```

dan contoh lain :

```
Uses crt;
Const                               {deklarasi konstanta}
    Gaji   : Real = 500000;
Var
    Omzet  : Real;
Begin
    Write('Omzet penjualan :') Readln(Omzet);
    If Omzet >= 10000000 Then
        Gaji := Gaji + 10000;
    Writeln('Gaji yang diterima ', Gaji:10:2);
End.
```

Pada contoh diatas konstanta Gaji yang dideklarasikan beserta type dapat berperilaku sebagai variabel, sehingga nilainya boleh diubah didalam program.

Ok, sekarang akan kita lanjutkan ke pembahasan tentang Set.

Set adalah sesuatu fasilitas khusus pada pascal yang tidak dimiliki oleh bahasa pemrograman lainnya. Set pada pascal adalah merupakan himpunan seperti pada matematika. Pemakaian Set dapat menyederhanakan algoritma program, dan memudahkan pembacaan, contoh :

```
uses crt;

var
    c       : char;
    vowel   : set of 'A','E','I','O','U';   {deklarasi variabel Set}
    letter  : set of 'A'..'Z';             {range A s/d Z}

begin
    write('Enter a vowel');
    repeat
        c:=upcase(readkey);
    until c in vowel;
    write('Enter a letter');
    repeat
        c:=upcase(readkey);
    until c in letter;
end.
```

Contoh lain pemakaian set

```
uses crt;
```

```

var
  c : char;

begin
  write('Yes or no (Y/N) ?');
  repeat
    c:=upcase(readkey);
  until c in ['Y','N'];
end.

```

Apakah fungsi operator in pada operasi Set ?

Oh, ya, fungsi operator in pada Set adalah menyatakan bagian dari himpunan tertentu.

Apakah masih ada fasilitas lain yang khusus pada pascal ?

Ya, masih adalah beberapa yaitu user defined type, dimana pemakai dapat membuat type ciptaan sesuai dengan kebutuhan program.

Dapatkah anda menjelaskan lebih lanjut apa itu type ciptaan dari sisi penerapan pada program, dan kalau bisa kasih contoh dong !

Sebagaimana kita ketahui, pascal menyediakan beberapa ordinal type seperti shortint, integer, longint, byte, char dan boolean. Didalam pemakaian type ini kadang-kadang kurang sesuai dengan kebutuhan, misalnya kita ingin mengolah data nilai siswa yang kemungkinan nilainya berkisar 0 s/d 100. Pada prinsipnya kita dapat menggunakan type byte untuk keperluan tersebut.

Contoh :

```

Var
  Fisika : Byte;

```

Tetapi akan lebih baik kalau dengan contoh berikut ;

```

Type
  Nilai = 0..100;
Var
  Fisika : Nilai;

```

Dalam hal ini type ciptaan seperti ini disebut sebagai Subrange.

Dapatkah anda menjelaskan lebih lanjut kira-kira bagaimana pengertian dari subrange tersebut!

Ok, sebenarnya subrange tersebut merupakan bagian dari range nilai type lain, misalnya type Byte dimana nilai yang dapat ditampung berkisar 0 s/d 255, padahal deklarasi type diatas :

```

Type
  Nilai = 0 .. 100;

```

Dalam hal ini sebenarnya type nilai merupakan subrange dari type ordinal yang telah ada.

Adakah type ciptaan lain yang dapat membuat program kita menjadi lebih terstruktur ?

Ya, masih ada, salah satunya adalah type substitusi, sesuai dengan namanya, type ini sebenarnya adalah pemberian nama terhadap type berstruktur seperti array dan string, sehingga selanjutnya kita dapat menggunakan nama tersebut dalam deklarasi variabel.

Contoh :

```
Type
  STRING30 = String[30];
Var
  Nama : STRING30;
```

Sebenarnya sama saja kalau kita tulis sebagai berikut :

```
Var
  Nama : STRING30;
```

Adakah type ciptaan yang lain ?

Masih ada dua yaitu Enumerated type dan Record Type, pada modul ini kita belum akan membahas tentang Record type.

Apa itu Enumerated type ?

Pada pascal, user diberi kesempatan untuk membentuk ordinal type yang sesuai dengan kebutuhan program, misalnya :

Contoh :

```
Var
  Today : Byte;
Begin
  Today := 7;
  If Today = 4 then Writeln('Jangan lupa sembayang DATUK');
End.
```

Bandingkan dengan :

```
Type
  DOW = (senin, selasa, rabu, kamis, jumat, sabtu, minggu);
Var
  Today : DOW;
Begin
  Today := minggu;
  If Today = kamis then Writeln('Jangan lupa sembayang DATUK');
End.
```

Dapatkan Enumerated type mengalami input-output dan operasi matematika ?

Enumerated type tidak dapat mengalami input output maupun operasi matematika, tetapi ada beberapa fungsi bantu ordinal yang dapat digunakan untuk mengolah enumerated type, yaitu :

Pred(X), function mendapatkan nilai predecessor dari suatu nilai ordinal type.

Succ(X), function mendapatkan nilai predecessor dari suatu nilai ordinal type.

Dec(X), procedure mengurangi nilai 1 ke variabel X.

Inc(X), procedure menambah nilai 1 ke variabel X.

Contoh :

```

Type
  DOW = (senin, selasa, rabu, kamis, jumat, sabtu, minggu);
Var
  Today : DOW;
Begin
  Today := minggu;
  Today := Pred(Today);      {sabtu}
  Inc(Today);                {minggu}
End.

```

Latihan 8

1. Explain what is constants in pascal ?
2. Explain how constants make your program more easy maintenance ?
3. Can we change the value of constant in program ?
4. Explain how to declare set variable !
5. Explain the function of in operator in set !
6. Explain what is Sub Ranges, Subtitution and Enumerated in type declaration ?
7. Can we do input output for Enumerated type ?

Latihan di Laboratorium

8. Change your space invader game into enumerated types, i.e. :

```

  enemy = (robot, alien, ufo); etc...

```

Latihan di Rumah

Make a pac-man game, using your own random number generator. You have 4 ghosts in the game, player has 5 lives. It has a "super bullet" that makes ghosts eatable. Just like a normal pac-man. You must record the highest score, but not in file (since I haven't taught you yet). Hint :

- * Use C as the Pac-man
- * Use . as the small bullet
- * Use * as the big bullet
- * For ghosts you may use U

Fungsi bantu konversi ini terdiri atas

ROUND(Real)

FUNCTION untuk menghasilkan data type INTEGER dari suatu REAL dengan memperhatikan aturan pembulatan.

TRUNC(REAL)

FUNCTION untuk mengambil bagian INTEGER dari suatu REAL.

ORD(Char);
FUNCTION untuk menghasilkan data type Byte dari suatu Character.

CHR(Byte);
FUNCTION untuk menghasilkan data type Character dari suatu Byte.

Fungsi Bantu Matematika :

ABS(Numerik);
FUNCTION untuk mengambil harga mutlak suatu nilai

FRAC(Real);
FUNCTION untuk mengambil bagian pecahan dari suatu REAL

INT(Real);
FUNCTION untuk mengambil bilangan bulat dari suatu REAL

SQR(Real);
FUNCTION untuk menghasilkan nilai kuadrat suatu bilangan

SQRT(Real)
FUNCTION untuk menghasilkan akar kwadrat suatu bilangan

COS(Real)
FUNCTION untuk menghasilkan Cosinus suatu sudut

SIN(real);
FUNCTION untuk menghasilkan Sinus suatu sudut (dalam radian $\pi/180$)

ARCTAN(real);
FUNCTION untuk menghasilkan sudut yang diketahui nilai tangennya

PI;
FUNCTION untuk menghasilkan nilai pi (3.1415926...)

LN(Real);
FUNCTION untuk menghasilkan logaritma natural dari suatu nilai

EXP(real);
FUNCTION untuk menghasilkan eksponensial dari suatu nilai.

Modul 9

Hallo, kita akan melanjutkan pembahasan tentang type ciptaan yang akan membuat program anda lebih terstruktur didalam pengolahan data yang dikenal dengan istilah Record type.

Apa itu Record type?

Record type adalah suatu type data terstruktur yang merupakan pengelompokan beberapa type data lainnya.

Dapatkan anda memberi contoh suatu Record type ?

Baiklah, misalnya kita akan mengolah data nilai ujian pascal programming siswa sebagai berikut:

Nis	Nama	Tugas	Teori	Praktek
02001	Hendra Soewarno	80	90	90

Dari tabel diatas kita dapat menyusun variabel data sebagai berikut :

```
NIS : String[9];  
NAMA : String[30];  
Tugas, Teori, Praktek : Byte;
```

Misalnya jumlah siswa adalah 10 orang, maka kita perlu mendeklarasikan variabel tersebut menjadi array:

```
Var  
NIS : Array [1..10] Of String[9];  
NAMA : Array [1..10] Of String[30];  
Tugas, Teori, Praktek : Array [1..10] Of Byte;
```

Sedangkan kalau kita menggunakan fasilitas Record Type, maka :

```
Type  
TSiswa = Record  
    NIS : String[9];  
    NAMA : String[30];  
    Tugas, Teori, Praktek : Byte;  
End;
```

```
Var  
Siswa : Array [1..10] of TSiswa;
```

Bukankan menjadi lebih sederhana dan terstruktur.

Ok, I got It, tetapi bagaimana cara pemakaiannya dalam program ?

Misalnya kita anda mengisi data untuk Siswa yang pertama.

```
Siswa[1].Nis := "02001";  
Siswa[1].Nama := "Hendra Soewarno";  
Siswa[1].Tugas := 80;  
Siswa[1].Teori := 90;  
Siswa[1].Praktek := 90;
```

Bisakah anda bayangkan kalau yang tidak pakai Record Type !

Adakah cara untuk melakukan hal yang sama ?

Ya, kita dapat menggunakan block With ... End;

```
With Siswa[1] Do Begin
  Nis := "02001";
  Nama := "Hendra Soewarno";
  Tugas := 80;
  Teori := 90;
  Praktek := 90;
End;
```

Bagaimana untuk proses input-outputnya ?

Untuk proses input-output sama saja dengan variabel biasanya.

Contoh :

```
For I := 1 To 10 Do Begin
  Write('No. Induk siswa :'); Readln(Siswa[I].Nis);
  Write('Nama siswa :'); Readln(Siswa[I].Nama);
  Write('Tugas :'); Readln(Siswa[I].Tugas);
  Write('Teori :'); Readln(Siswa[I].Teori);
  Write('Praktek :'); Readln(Siswa[I].Praktek);
End;
```

```
For I := 1 To 10 Do Begin
  With Siswa[I] Do Begin
    Rata := (Tugas+Teori+Praktek)/3;
    Write(Nis:9, ' ');
    Write>Nama:30, ' ');
    Write(Tugas:6:2, ' ');
    Write(Teori:6:2, ' ');
    Write(Praktek:6:2, ' ');
    Write(Rata:6:2);
  End;
End;
```

Berbicara tentang pengolahan data, saya jadi teringat bagaimana melakukan pengurutan data pada komputer ?

Mengurut data pada komputer membutuhkan algoritma, beberapa diantaranya adalah :

1. Bubble Sort (Sangat sederhana tetapi lambat)
2. Quick Sort (Cepat tetapi agak rumit)
3. Shell Sort (Paling cepat tetapi sangat rumit)

Bagaimana konsep kerja dari Bubble Sort ?

Konsep dasar bubble sort adalah memindahkan data yang lebih kecil ke depan, dengan cara membandingkan seluruh data.

```
for i:=1 to NumberOfData-1 do
  for j:=i+1 to NumberOfData do
    if Data[i]>Data[j] then
      begin
        t:=Data[i];
        Data[i]:=Data[j];
        Data[j]:=t;
      end;
```

Pada prinsipnya adalah ambil data ke 1, bandingkan dengan data ke 2, 3, 4, s/d jumlah data, jika data ke 1 lebih besar, maka pindah tempat.
 Selanjutnya dilakukan terhadap data ke 2, dengan data 3,4,5, s/d jumlah data
 Selanjutnya dilakukan terhadap data ke 3, dengan data 4,5,6 s/d jumlah data
 Proses yang sama dilakukan sampai terhadap data ke jumlah data - 1

Bagaimana pula konsep kerja Shell Sort ?

Algoritma ini ditemukan oleh Donald Shell. Konsepnya adalah membandingkan data dengan jarak tertentu dalam array

```

for gap:=(NumberOfData div 2) downto 1 do
  for i:=1 to (NumberOfData-gap) do
    if Data[i]>Data[i+gap] then
      begin
        t:=Data[i];
        Data[i]:=Data[i+gap];
        Data[i+gap]:=t;
      end;

```

Misalnya kita ada 8 data, maka gap akan mulai dari 4 s/d 1.

Pada gap = 4, maka :

Bandingkan data 1 dengan 5, 2 dengan 6, 3 dengan 7, 4 dengan 8.

Pada gap =3, maka :

Bandingkan data 1 : 4, 2 : 5, 3 : 6, 4 : 7, 5 : 8

Pada gap = 2, maka :

Bandingkan data 1 : 3, 2 : 4, 3 : 5, 4 : 6, 5 : 7, 6 : 8

Pada gap = 1, maka :

Bandingkan data 1 : 2, 2 : 3, 3 : 4, 4 : 5, 5 : 6, 6 : 7, 7 : 8

Bagaimana dengan yang Quick Sort ?

Algoritma ini merupakan metode pengurutan data yang tercepat didunia, yang ditemukan oleh E. Hoare. Algoritma ini menggunakan metode recursive.

Prinsip kerjanya adalah membagi data menjadi dua bagian dengan data ditengah sebagai titik pivot. Pindahkan data sisi kiri yang lebih besar dari pada pivot dengan data sisi kanan yang lebih kecil dari pada pivot.

```

procedure qsort(lower, upper : byte)
var
  left, right, pivot : byte;
begin
  pivot:=Data[(lower+upper) div 2];
  left:=lower;
  right:=upper;

  while left<=right do
  begin
    while Data[left] < pivot do left:=left+1; { Parting for left }
    while Data[right] > pivot do right:=right-1;{ Parting for right}
    if left<=right then { Validate the change }
    begin
      swap Data[left] with Data[right];
      left:=left+1;

```

```

        right:=right-1;
    end;
end;
if right>lower then qsort(lower,right); { Sort the LEFT part }
if upper>left  then qsort(left ,upper); { Sort the RIGHT part }
end;

```

Pemakaian : qsort(1,NumberOfData);

Latihan 9

1. Explain what is Record Type and give an example !
2. Explain what With ... End block function !
3. Suppose we have an array contain 8 data (7,8.5,4,2,1,3,6), please write down step by step data changing when sorting using Bubble Sort, Shell Sort, and Quick Sort !

Latihan di Laboratorium

4. Make a database. The record tag, TStudent consists of these fields :

Field	Length
~~~~~	~~~~~
Name	20
Address	40
Phone	15
Quiz	number, from 1 to 100
Midtest	number, from 1 to 100
Finaltest	number, from 1 to 100

User could : add new student, edit existing student, view student in a list, delete the record, sort the record according to :

- * Quiz
- * Midtest
- * Finaltest
- * Overall value  
(Overall value = 0.1*Quiz + 0.4*Mid + 0.5*Final)

You must :

- * Make simple menu.
- * At least be able to handle 200 students.
- * Apply quick sort for at least one key-sort.
- * Apply search for student name first before editing the data.

Tip : Use array of records.

* * *