

Algoritma Pemrograman

TIPE- TIPE DATA

DALAM TURBO PASCAL

Secara umum, tipe data dalam pemrograman Pascal adalah:

- **Tipe data sederhana**
 - Tipe data standard (*standard data type*)
--> Integer, Real, Boolean, Char, dan String
 - Tipe data didefinisikan pemakai (*user defined data type*)
--> subrange dan enumerasi
- **Tipe data terstruktur**
--> Array, record, set, file
- **Tipe data penunjuk**
--> Pointer

Tipe data berfungsi membatasi jangkauan data yang akan dilaksanakan.

TIPE DATA SEDERHANA

Tipe Data Standard

- Integer

Tipe integer adalah tipe data yang nilainya bulat. Ada 5 macam tipe integer, yaitu:

Tipe	Jangkauan Nilai (Range)	Format
Shortint	-128 .. 127	Signed 8 bit
Integer	-32768 .. 32767	Signed 16 bit
Longint	-2147483648 .. 2147483647	Signed 32 bit
Byte	0 .. 255	Unsigned 8 bit
Word	0 .. 65535	Unsigned 16 bit

Contoh deklarasi:

```
Var      A, B, C : Integer;  
         D, E, F : Longint;
```

Operator-operator yang berlaku pada tipe data integer, yaitu :

Operator aritmatik : +, -, *, /, div, mod

Operator logic : <, =, >, <=, >=, <>

- **Real**

Real merupakan tipe bilangan pecahan. Bilangan real bisa dinyatakan dalam bentuk eksponensial. Ada 5 macam tipe real, yaitu:

<i>Tipe</i>	<i>Jangkauan Nilai</i>	<i>Digit Signifikan</i>	<i>Ukuran</i>
Real	$2,9 \times 10^{-38} .. 1,7 \times 10^{38}$	11 - 12	6 byte
Single	$1,5 \times 10^{-45} .. 3,4 \times 10^{38}$	7 - 8	4 byte
Double	$5.0 \times 10^{-324} .. 1,7 \times 10^{308}$	15 - 16	8 byte
Extended	$3,4 \times 10^{-4932} .. 1,11 \times 10^{4932}$	19 - 20	10 byte
Comp	$-2^{63} + 1 .. 2^{63} - 1$	19 - 20	8 byte

Contoh deklarasi:

```
Var      A, B, C : Real;  
         D, E, F : Double;
```

Operasi – operasi yang bisa dikerjakan oleh tipe data real adalah:

- Operasi biner, seperti penjumlahan (+), pengurangan (-), perkalian (*), dan pembagian (/)
- Operasi uner. Contoh: -4.25, +1.25

- **Boolean**

Boolean adalah tipe data yang hanya mempunyai dua kemungkinan, yaitu true dan false. Biasanya tipe data ini digunakan untuk perbandingan.

```
Var      A, B, C : Boolean;
```

Operator NOT, OR, AND, dan XOR dapat dibentuk secara bersamaan menjadi sebuah ungkapan Boolean yang rumit.

Misal:

```
Var A,B,C: Boolean;
```

```
Begin
```

```
    C := false; B := true;
```

```
    A := (Not(C) AND Not (B)) XOR (C);
```

```
    Write(A); readln;
```

```
End.
```

Menurut anda, berapakah nilai A?

- **Char**

Char adalah semua tombol yang terdapat pada keyboard atau lebih lengkapnya semua karakter yang terdapat pada kode ASCII.

Catatan : Apabila char ingin dijadikan sebagai konstanta maka karakter yang dimasukkan harus diapit dengan tanda kutip satu. Dan apabila karakter itu berupa tanda kutip satu maka harus diapit dengan dua tanda kutip satu

Beberapa fungsi untuk memanipulasi tipe data char:

- `Ord(x)` dengan `x` adalah data bertipe char. Fungsi ini digunakan untuk memperoleh nilai urutan dalam kode ASCII yang digunakan untuk melambangkan karakter tersebut. **Contoh: `Ord('C')` adalah `67`.**
- `Char(x)` dengan `x` adalah data bertipe byte. Fungsi ini adalah kebalikan dari fungsi `Ord`. Nilai yang diperoleh merupakan karakter ASCII yang dinyatakan dengan urutan ke `x`. **Contoh: `Char(67)` adalah `'C'`.**
- `Pred(x)` dengan `x` adalah data bertipe char. Fungsi ini digunakan untuk mengetahui karakter yang mendahului `x`. **Contoh: `Pred('h')` adalah `'g'`.**
- `Succ(x)` dengan `x` adalah data bertipe char. Fungsi ini digunakan untuk mengetahui karakter sesudah `x`. Contoh: `Succ('h')` adalah `'i'`. Jika `x` tidak mempunyai penerus, maka nilai fungsi ini tidak terdefiniskan.
- `Uppcase(x)` dengan `x` adalah data bertipe char. Jika `x` merupakan huruf kecil maka akan dikembalikan huruf kapitalnya. Contoh: `uppercase('a')` adalah `'A'`.

- **String**

String adalah data yang berisi sederet karakter yang terletak diantara tanda kutip satu. Jika karakter kutip merupakan bagian dari konstanta string, maka ditulis dengan menggunakan dua buah tanda kutip satu berurutan. Nilai data string akan menempati memori sebesar maksimal jumlah karakter yang dapat ditampung ditambah dengan 1 byte (index ke-0) untuk menyimpan panjang string yang sebenarnya. Jika panjang string tidak ditulis, maka panjang string dianggap 255 karakter. Panjang string yang diijinkan antara 1 sampai 255.

Bentuk umum dari deklarasi tipe string adalah:

```
Var pengenal : string[panjang];
```

dimana

pengenal : nama variabel

panjang : bilangan bulat yang menunjukkan banyaknya karakter (1 – 255).

Untuk tipe data string, operator yang berlaku adalah operator penggabungan (+).

Tipe data didefinisikan pemakai (*user-defined data type*)

– Subrange

Tipe data subrange adalah suatu range yang menunjukkan nilai terkecil dan nilai terbesar yang dapat dipergunakan. Deklarasi tipe data subrange mempunyai bentuk:

```
Type pengenal = konstanta1 .. konstanta2;
```

dimana:

pengenal : nama tipe data yang dideklarasikan

konstanta1 : batas bawah nilai data

konstanta2 : batas atas nilai data

Kedua konstanta di dalam subrange harus bertipe ordinal yang sama, di mana nilai konstanta pertama lebih kecil atau sama dengan nilai konstanta kedua. Tipe data real tidak dapat digunakan sebagai nilai subrange, karena bukan tipe ordinal. Contoh:

```
Type tanggal = 1 .. 31;
```

```
bulan = 1 .. 12;
```

– Enumerated

Tipe data enumerated (skalar) menunjukkan kumpulan dari nilai yang urutannya sudah pasti. Nilai dari tipe yang dideklarasikan ini akan diwakili dengan pengenal – pengenal (identifiers) yang akan menjadi nilai suatu konstanta. Contoh:

```
Type bahasa = (delphi, java, c, pascal, basic);
```

```
bulan = (maret, april, mei, juni, juli);
```

Tipe data bulan mempunyai 5 elemen dari maret sampai juli. Dari urutannya, maret adalah identifier berupa konstanta bernilai 0 dan juli bernilai 4. Yang perlu diperhatikan dengan tipe data skalar ini adalah tipe data ini sudah berbeda dengan tipe standar yang ada dan pascal tidak mengijinkan operasi dengan tipe data yang berbeda.

Derajat Operator

Anda sudah mengenal beberapa tipe data di dalam Pascal dan operatornya. Di dalam Pascal, operator – operator tersebut memiliki derajat atau tingkatan. Adapun kegunaan dari derajat ini adalah Pascal dapat menentukan operator mana yang akan di jalankan terlebih dahulu.

Misal: $x := 4 + 5 * 3$

Menurut anda, berapakah nilai dari x? Apakah 27 (hasil dari $9 * 3$) atau 19 (hasil dari $4 + 15$)

Berikut ini urutan operator berdasarkan derajatnya:

@, not, *, /, div, mod, as, and, shl, shr, +, -, or, xor, :=, <>, <, >, <=, >=, in

Berdasarkan dari urutan operator di atas, maka nilai x pada $x := 4 + 5 * 3$ adalah 19, karena operator * lebih dulu dikerjakan daripada operator +.

Akan tetapi, jika kita ingin agar operator + dikerjakan terlebih dahulu, kita bisa meletakkan operasi penjumlahan tersebut di dalam kurung, misalnya: $x := (4 + 5) * 3$. Di sini nilai x itu adalah 27, karena di dalam Pascal operasi yang di dalam kurung akan dilakukan terlebih dahulu, tidak dipengaruhi oleh derajat operator.

INPUT OUTPUT DI DALAM PASCAL

Write dan Writeln

Perintah ini digunakan untuk mencetak kata-kata atau variable ke layar komputer. Sebagai contoh untuk mencetak kata 'Sedang Belajar Pascal', perintah yang diberikan adalah:

```
Write('Sedang Belajar Pascal');
```

atau

```
Writeln('Sedang Belajar Pascal');
```

Jika menggunakan write, setelah mencetak kata – kata atau variable, kursor akan diletakkan di samping hasil cetakan. Jika menggunakan perintah writeln, setelah mencetak kata – kata atau variable, kursor akan dipindahkan ke bawah satu baris dengan posisi horizontal pada awal baris. Untuk mencetak suatu variable, tidak perlu tanda petik satu. Contoh:

```
Var A : integer;  
Begin  
    A:= 9;  
    Writeln(A);  
End.
```

Dengan menggunakan write atau writeln, kita bisa menampilkan kata – kata dan variable dengan tanda koma (,) sebagai pemisah antara variable dan kata - kata, Contoh:

```
Var A : integer;  
Begin  
    A:= 9;  
    Writeln('Nilai A adalah ', A);  
End.
```

Terkadang kita menemukan prosedur write atau writeln yang ditulis seperti ini, writeln('Asyik': 10) → Jika ditulis seperti ini, maka akan menghasilkan space yang disediakan untuk menuliskan kata asyik di layar monitor adalah 10 karakter, mulai dari kiri ke kanan. Coba tuliskan program berikut ini:

```
begin
```

```
    write('Sehat': 10); write('Senang': 10);  
    readln;  
end.
```

Perhatikan apa yang terjadi pada penggalan program di atas!

Selain itu, dengan menggunakan write atau writeln, kita dapat mengatur berapa banyak angka di belakang koma yang akan ditampilkan di layar monitor jika kita ingin menampilkan bilangan real, caranya adalah dengan menuliskan seperti program di bawah ini

```
Begin  
Write(1/3: 5 : 2); write(2/3: 5: 2);  
Readln;  
End.
```

Read dan Readln

Read atau readln digunakan untuk menerima masukan dari user untuk disimpan ke dalam suatu variable. Contoh:

```
Var A: Integer;  
Begin  
    Write('Masukkan nilai A: ');readln(A);  
    Writeln('Nilai yang anda masukkan adalah: ', A);  
End.
```

Catatan:

Terkadang kita menggunakan fungsi readln pada satu baris sebelum kata end.. Hal ini dimaksudkan agar suatu program tidak akan langsung *terminate* (berhenti) tanpa menunggu suatu masukan dari user. Adapun masukan tersebut berupa penekanan tombol enter pada keyboard.

Komentar

Komentar adalah bagian dari program yang tidak akan diproses oleh compiler (Free Pascal). Komentar hanya digunakan untuk dokumentasi saja. Cara pemberian komentar adalah dengan menambahkan // untuk 1 baris atau {...} untuk lebih dari 1 baris. Contoh pemberian komentar:

```
Program Komentar;  
Begin  
    //Ini Adalah Komentar  
    (Pernyataan - pernyataan ini  
    Tidak kan Diproses}  
End.
```

Unit

Di dalam Pascal, ada bagian yang disebut sebagai unit. Unit ini menyimpan fungsi dan prosedur standard yang sudah didefinisikan oleh pembuat compiler. Beberapa unit standard yang ada dalam Pascal adalah Crt, System, Graph, Dos, Printer, dan Overlay. Cara penggunaan unit adalah dengan menambahkan statemen:

```
Uses nama_unit;
```

pada **bagian Deklarasi**.

Misal:

```
Uses Crt;
```

Beberapa fungsi dan prosedur yang ada dalam unit Crt:

Clrscr

Clrscr adalah suatu prosedur untuk membersihkan layar. Contoh:

```
Uses crt;  
Begin  
    Writeln('Layar Sudah Bersih');  
    Readln;  
End.
```

Readkey

Readkey adalah fungsi masukan yang hanya akan mengambil satu karakter masukan. Jenis variable yang dapat diinput oleh readkey pun hanya jenis variable karakter.

Contoh:

```
Uses crt;
```

```
Var A : Char;
```

```
Begin
```

```
    Write('Tekan karakter apa saja di dalam keyboard');
```

```
    A := Readkey;
```

```
    Writeln('Karakter yang barusan anda tekan adalah: ', A);
```

```
    Readln;
```

```
End.
```

TextBackground

TextBackground merupakan prosedur untuk mengganti warna background tulisan yang ditampilkan di layer dengan warna yang kita inginkan. Perintah yang digunakan adalah:

```
Textbackground(warna);
```

Warna merupakan variable yang bias diganti dengan nama warna atau nilai warna tersebut:

Warna	Konstanta	Nilai
Hitam	Black	0
Biru	Blue	1
Hijau	Green	2
Cyan (Biru Laut)	Cyan	3
Merah	Red	4
Magenta	Magenta	5
Coklat	Brown	6
Coklat Muda	Light Gray	7

Warna	Konstanta	Nilai
Coklat Tua	Dark Brown	8
Biru Muda	Light Blue	9
Hijau Muda	Light Green	10
Cyan Muda	Light Cyan	11
Merah Muda	Light Red	12
Magenta Muda	Light Magenta	13
Kuning	Yellow	14
Putih	White	15

Contoh:

```
Uses Crt;
```

```
Begin
```

```
    Textbackground(Red);
```

```
    Writeln('Laboratorium Komputasi FMIPA UGM');
```

```
    Readln;
```

```
End.
```

Textcolor

Textcolor merupakan prosedur yang digunakan untuk mewarnai tulisan. Perintahnya adalah:

```
    Textcolor(warna);
```

Cara pengisian warna sama dengan Textbackground.

Contoh:

```
Uses crt;
```

```
Begin
```

```
    Textbackground(Red);
```

```
    Textcolor(Blue);
```

```
    Writeln('Laboratorium Komputasi FMIPA UGM');
```

```
    Readln;
```

```
End.
```

Beberapa fungsi dan prosedur untuk operasi Aritmatika:

Sqr(Value) → fungsi ini digunakan untuk menghasilkan nilai perpangkatan

Sqrt(Value) → fungsi ini digunakan untuk menghasilkan nilai akar pangkat dua

Abs(Value) → untuk menghasilkan nilai mutlak

Ln(Value) → untuk mencari nilai logaritma natural

Exp(Value) → untuk mencari nilai eksponennya

Trunc(Value) → untuk memotong nilai real ke suatu nilai integer

Round(Value) → untuk membulatkan nilai real ke nilai integer yang terdekat

STRUKTUR PERCABANGAN

Struktur percabangan atau sering disebut dengan struktur kontrol ini memungkinkan programmer untuk membuat program yang dapat memilih satu langkah di antara sejumlah langkah untuk dikerjakan.

Dalam Pascal disediakan 2 buah struktur kontrol seleksi, yaitu:

a. Struktur **IF.....THEN.....**

Merupakan struktur kontrol di mana suatu aksi dilaksanakan berdasarkan kondisi logikanya (benar atau salah). Struktur if .. then ... sendiri memiliki 4 jenis, yaitu:

- Bentuk 1
if (kondisi) **then** aksi;
{Jika kondisi benar maka aksi akan dilaksanakan dan sebaliknya }
- Bentuk 2
if (kondisi) **then** aksi1 **else** aksi2 ;
{Jika kondisi benar maka aksi1 dilaksanakan, jika kondisi salah maka aksi2 akan dilaksanakan}
- Bentuk 3
if (kondisi1) **then** aksi1 **else**
 if (kondisi2) **then** aksi2 **else**

 if (kondisi-n) **then** aksi-n ;
{Jika kondisi1 benar maka aksi1 dilaksanakan tanpa membaca kondisi2 dan seterusnya. Jika kondisi1 salah maka aksi2 dilaksanakan tanpa membaca aksi3 dan selanjutnya. Demikian seterusnya}
- Bentuk 4
if (kondisi1) **then** aksi1 ;
if (kondisi2) **then** aksi2 ;
.....
if (kondisi-n) **then** aksi-n ;
{Masing-masing kondisi akan dibaca dan aksi akan tetap dilaksanakan. Jadi masing-masing struktur terpisah satu dengan yang lain}

Contoh program 1:

Program menghitung akar kwadrat

```

uses crt;
var A,B,C:integer;
    x1,x2,D:real;
begin
  clrscr;
  write('Baca koefisien:');readln(A,B,C);writeln;
  writeln(A,'x*x + (' ,B,') x +',C);
  if A=0 then writeln('Bukan persamaan kwadrat') else
  begin
D:=(B*B) - (4*A*C);
writeln('Determinannya :',D:2:2);readln;
if D>0 then
  begin
    writeln('Persamaan kwadrat mempunyai 2 akar yang berbeda');
    x1:= (-B + sqrt(D))/(2*A);
    x2:= (-B - sqrt(D))/(2*A);
    writeln('Akar-akarnya adalah:',x1:2:2,'dan',x2:2:2);
  end else
if D=0 then
  begin
    writeln('Persamaan kwadrat mempunyai akar yang sama');
    x1:= -B/(2*A);
    x2:= -B/(2*A);
    writeln('Akar-akanya adalah:',x1:2:2);
  end else
    writeln('Tidak memiliki akar riil');
end;
readln;
end.

```

Contoh Program 2:

Program Konversi nilai

```

Uses Crt;
var Nilai : Byte;
begin
  clrscr;
  write('Baca nilai :');readln(Nilai);
  if nilai>=80 then write('Nilai = A') else
    if nilai>=65 then write('Nilai = B') else
      if nilai>=41 then write('Nilai = C') else
        if nilai>=26 then write('Nilai = D') else
          write('Nilai = E');
  readln;
end.

```

Catatan:

Jika anda menggunakan else, perlu diinga bahwa satu baris sebelum else tidak diperkenankan mengandung tanda ;

b. Struktur CASE...OF...

Merupakan peluasan dari struktur IF. Karena kalau dalam struktur IF hanya disediakan dua pilihan (berdasarkan kondisi logikanya) maka dalam struktur Case ..of dimungkinkan untuk memilih satu pilihan di antara banyak pilihan yang ada. Bentuk umumnya :

```
Case var_pilih of  
    Pilih1 : aksi1 ;  
    Pilih2 : aksi2 ;  
    ..... ;  
    pilih-n : aksi-n ;  
end;
```

atau

```
Case var_pilih of  
    pilih1 : aksi1 ;  
    pilih2 : aksi2 ;  
    ..... ;  
    pilih-n : aksi n;  
else aksi-n+1  
end;
```

Catatan :

*Ekspresi yang digunakan dalam statemen Case adalah yang **mempunyai tipe ordinal** yaitu dengan batas antara (-32768 s/d 32767). Sehingga tipe lain seperti integer yang bertipe longint, tipe string atau real tidak boleh digunakan.*

Contoh program 3:

Program Konversi nilai 2

```
Uses Crt;  
Var Nilai : integer;  
begin  
    Clrscr;  
    write('Baca nilai =');readln(Nilai);  
    Case Nilai of  
        0..25 : writeln('Nilainya = E');  
        26..39 : writeln('Nilainya = D');  
        40..64 : writeln('Nilainya = C');  
        65..79 : writeln('Nilainya = B');  
        80..100: writeln('Nilainya = A');  
    else  
        writeln('Tidak ada nilai yang dimaksud');
```

```
    end;readln;  
end.
```

Catatan : *Program ini akan memberikan nilai yang sama persis dengan yang menggunakan struktur IF.*

Latihan

1. Buatlah program untuk menentukan suatu bilangan yang dimasukkan oleh user, merupakan bilangan genap atau ganjil
2. Buatlah sebuah program untuk menentukan bilangan tahun yang dimasukkan oleh user merupakan bilangan tahun kabisat atau bukan
3. Buatlah sebuah program yang akan menerima masukan bilangan bulat dari user. Jika bilangan yang dimasukkan 0, maka program akan menampilkan string 'Minggu'; jika user memasukkan bilangan 1, maka program akan menampilkan string 'Senin'; dan seterusnya sampai Sabtu; jika user memasukkan bilangan lebih dari 6, maka akan keluar string 'Hari tidak valid'.

