

**Pertemuan 1-3 (4 maret 2014)**

1. Konsep Dasar Bahasa Pascal
2. Sejarah Singkat Bahasa Pascal
3. Struktur Dan Komponen Dasar Bahasa Pascal
  - (i). Judul Program (Header )
  - (ii). Blok Program
    - Blok Deklarasi
    - Blok statemen
4. Unit-unit Pada Pascal
5. Standart Type Data
  - (i). Standard type Data Integer
    - Boolean
    - Real
    - Char
  - (ii). User Define Type Data
    - Enumerated
    - Subrange
6. Konstanta, Variabel dan Ekspresi dan Operator
7. Statemen Read dan Readln
8. Statemen Write dan Writeln
9. Format Output
10. Definisi String
11. Deklarasi String
12. Procedure Standard pada Operasi String
13. Fungsi Standar pada Operasi String

# PASCAL

Pascal adalah bahasa tingkat tinggi yang orientasinya pada segala tujuan, dirancang oleh Prof. Niklaus Wirth dari Technical University di Zurich, Switzerland. Nama Pascal diambil sebagai penghargaan terhadap Blaise pascal, ahli matematik dan filosofi.

Kelebihan dari bahasa pemrograman Pascal adalah:

- Tipe Data Standar, tipe-tipe data standar yang telah tersedia pada kebanyakan bahasa pemrograman. Pascal memiliki tipe data standar: *boolean*, *integer*, *real*, *char*, *string*,
- User defined Data Types, programmer dapat membuat tipe data lain yang diturunkan dari tipe data standar.
- Strongly-typed, programmer harus menentukan tipe data dari suatu variabel, dan variabel tersebut tidak dapat dipergunakan untuk menyimpan tipe data selain dari format yang ditentukan.
- Terstruktur, memiliki sintaks yang memungkinkan penulisan program dipecah menjadi fungsi-fungsi kecil (*procedure* dan *function*) yang dapat dipergunakan berulang-ulang.
- Sederhana dan Ekspresif, memiliki struktur yang sederhana dan sangat mendekati bahasa manusia (bahasa Inggris) sehingga mudah dipelajari dan dipahami.

---

Contoh program [Hello World](#) menggunakan bahasa pascal adalah sebagai berikut:

```
Program HelloWorld;  
begin  
  writeln('Hello world');  
end.
```

## Struktur program :

Judul Program	PROGRAM nama-program;
Blok Program	
Bagian deklarasi	
- deklarasi label	LABEL nama-label;
- deklarasi konstanta	CONST.....;
- deklarasi tipe	TYPE .....
- deklarasi variabel	VAR .....
- deklarasi prosedur	PROCEDURE nama-prosedur;

```

.....;
- deklarasi fungsi          FUNCTION nama-fungsi;
.....;

```

Bagian Pernyataan

Begin

```
(statement)
```

```
.....;
```

```
.....;
```

end.

**Contoh :** Menghitung luas bidang berbentuk empat persegi panjang dengan panjang P dan lebar L.

```

PROGRAM Luas;                {Judul}
VAR P,L,Luas: real;          {Deklarasi variabel}
BEGIN
    Read (P,L);              {Statemant}
    Luas := P*L;             {Statement}
    Write (P,L,Luas);        {Statement}
END.

```

---

### **Beberapa aturan dalam program Pascal :**

- Akhir sebuah program Pascal ditandai dengan tanda baca titik (.) setelah END yang paling akhir.
- Tanda titik koma (;) merupakan pemisah antar instruksi satu dengan lainnya.
- Beberapa statement boleh ditulis menjadi satu baris dipisahkan dengan tanda baca titik koma (;)  
Contoh : `simpan := akhir; akhir := simpan + awal;`
- Baris komentar diletakkan diantara tanda (\* dan \*) atau diantara tanda { dan }  
Contoh : `Var rerata : real; (*nilai rata-rata*)`  
`Nilai : real; {nilai ujian}`

## **Statement (pernyataan)**

Adalah instruksi atau gabungan instruksi, yang menyebabkan komputer melakukan aksi.

Type statement dalam Pascal terdiri atas :

### 1. Sederhana :

- menandai sebuah item data ke sebuah variabel (assignment statement)

contoh :  $X := Y * 4.135$

- pemanggilan procedure dan goto statement

### 2. Terstruktur:

- Compound Statement

contoh : Begin

    read (x) ;

    y := x + 5;

    write (y)

End.

- Repetitive Statement

contoh :       For I := 1 to 100 do

                write (count);

- Conditional Statement

contoh :       If x > 100 then write (s)

                  else write (p) ;

## **Komponen Dasar Program Pascal**

Pola susun bahasa Pascal dibentuk dengan menggunakan komponen bahasa pemrograman yang umum, yaitu :

1. Simbol Dasar
2. Reserved Word (kata pasti)
3. Identifier (penyebut)

### **1. Simbol Dasar.**

Simbol dasar terdiri atas :

1. Simbol huruf, yaitu huruf A sampai dengan Z atau a sampai dengan z. (huruf besar dan kecil).
2. Simbol angka atau digit yaitu : 0,1,2,3,4,5,6,7,8,9.
3. Simbol khusus, yaitu  
+ - \* / ; := , ' = < > <= >= <> : { } ( ) [ ]

## 2. Reserved Word (kata pasti)

Reserved Word adalah suatu kata yang secara mutlak tidak boleh diartikan lain dan harus digunakan sebagaimana yang telah didefinisikan atau ditentukan kegunaanya oleh bahasa Pascal. Reserved word ini tidak boleh didefinisikan ulang oleh pemakai, sehingga tidak dapat dipergunakan sebagai pengenal (identifikasi)

Reserved Word ini jumlahnya berbeda untuk masing-masing bahasa Pascal.

Contoh beberapa reserved word yang telah didefinisikan oleh bahasa pascal antara lain

AND	ELSE	LABEL	SET
ARRAY	END	OF	TYPE
BEGIN	FUNCTION	OR	UNTIL
CASE	FOR	PROCEDUREVAR	
CONST	GOTO	PROGRAM	WHILE
DO	IF	RECORD	WITH
DOWNTON	IN	REPEAT	dsb

## 3. Identifier (sebutan/pengenal)

Identifier merupakan sebuah kata yang digunakan sebagai nama atau sebutan terhadap sesuatu didalam program. Pemakai dapat mendefinisikan sendiri suatu nama sebagai identifier.

Identifier ini terdiri atas :

1. Identifier Standar, yaitu identifier yang telah didefinisikan oleh bahasa pascal.  
Contoh dari Identifier standar ini antara lain:

ABS	LN
ARCTAN	ODB
BOOLEAN	PRED
CHAR	ROUND
CHR	READ

COS	READLN
EOF	SQR
EOLN	SQRT
EXP	SUCC

Dan masih banyak lagi.

2. Identifier Non Standar; yaitu identifier yang didefinisikan oleh pemakai bahasa pascal; misalnya;
3. nama suatu program
4. nama suatu konstanta
5. nama suatu variabel
6. nama suatu procedure  
Identifier ini bebas, tetapi dengan ketentuan-ketentuan sebagai berikut :
7. terdiri dari gabungan huruf dan angka dengan karakter pertama harus berupa huruf. Huruf besar dan huruf kecil dianggap sama.
8. Tidak boleh mengandung blank.
9. Tidak boleh mengandung simbol-simbol khusus, kecuali garis bawah.
10. Panjangnya bebas, tetapi hanya 63 karakter pertama yang dianggap signifikan.

Contoh :

Identifier	Keterangan
GajiKaryawan	Benar
No_Mhs	Benar
P3K	Benar
1X	Salah, karakter pertama harus huruf
A&B	Salah, tidak boleh mengandung simbol khusus
A B	Salah, tidak boleh mengandung blank

### Jenis – jenis data

Dalam bahasa Pascal terdapat beberapa jenis tipe data yang bisa digunakan untuk sebuah variabel atau konstanta pada program. Tipe Data tersebut antara lain adalah

<b>Tipe Data</b>	<b>Deskripsi (range variabel)</b>
<u>Byte</u>	angka dari 0 sampai 255
<u>Integer</u>	angka dari -32768 to 32767
<u>Real</u>	semua nilai pecahan dari 1E-38 to 1E+38
<u>Boolean</u>	nilai TRUE atau FALSE
<u>Char</u>	semua karakter dari tabel ASCII
<u>String</u>	semua huruf, spasi, frasa

1. Jenis data sederhana
  - a. Jenis data yang standar; yaitu :
    - Integer
    - Real
    - Karakter
    - Boolean
  - b. Jenis data yang non standar (user defined), yaitu;
    - Enumerated
    - Sub-range
2. Jenis data berstruktur, yaitu:
  - a. Array
  - b. Record
  - c. Set
  - d. File
3. Jenis Data Pointer

### **INTEGER**

Jenis data ini terdiri atas integer positif, integer negatif dan nol. Merupakan nilai bilangan bulat.

Pada TURBO PASCAL jenis data ini di bagi atas beberapa bagian, yaitu :

Type	Ukuran memori (dalam byte)	Jangkauan nilai
BYTE	1	0..255
SHORTINT	1	-128..127
INTEGER	2	-32768..32767
WORD	2	0..65535
LONGINT	4	-2147483648..2147483647

Operator Integer terdiri atas : + , - , \* , / , DIV dan MOD

Contoh :

Var

Jumlah : byte;

Begin

Jumlah := 200;

WriteLn('Nilai JUMLAH = ',Jumlah);

End.

Hasilnya bila dijalankan :

Nilai JUMLAH = 200

### **REAL**

Penulisan untuk jenis data ini selalu menggunakan titik desimal. Nilai konstanta numerik real berkisar dari 1E-38 sampai dengan 1E+38 dengan mantissa yang signifikan sampai dengan 11 digit. E menunjukkan nilai 10 pangkat. Nilai konstanta numerik real menempati memori sebesar 6 byte.

Contoh :

123.45



12345. → salah, titik desimal tidak boleh dibelakang

12E5

12E+5

-12.34

.1234 → salah, titik desimal tidak boleh dimuka

Pada TURBO PASCAL, jenis data ini dibedakan atas :

Tipe	Ukuran memori (dalam byte)	Jangkauan nilai	Digit signifikan
SINGLE	4	$1.5 \times 10^{-45} \dots 3.4 \times 10^{E38}$	7-8
DOUBLE	8	$5.0 \times 10^{-324} \dots 1.7 \times 10^{E308}$	15-16
EXTENDED	10	$1.9 \times 10^{-4951} \dots 1.1 \times 10^{E4932}$	19-20
COMP	8	$-2E+63+1 \dots 2E+63-1$	19-20

Operator untuk jenis data ini terdiri atas : + , - , \* dan /

### KARAKTER

Yang dimaksud dengan jenis data ini adalah karakter tunggal atau sebuah karakter yang ditulis diantara tanda petik tunggal, seperti misalnya 'A', 'a', '!', '5' dsb.

Dasarnya adalah ASCII CHARACTER SET.

Misalnya : 032 pada tabel ASCII CHARACTER SET menunjukkan karakter.

Blank.

033 ————— !

048 ————— 0

076 ————— L

dst

## STRING

Nilai data string merupakan urutan dari karakter yang terletak di antara tanda petik tunggal. Nilai data string akan menenpati memori sebesar banyaknya karakter stringnya ditambah dengan 1 byte. Bila panjang dari suatu string di dalam deklarasi variabel tidak disebutkan, maka dianggap panjangnya adalah 255 karakter.

Contoh :

Var

```
Kampus : string[10];
```

Begin

```
Kampus := 'Gunadarma';
```

```
Write(Kampus);
```

End.

## BOOLEAN

Jenis data ini mempunyai nilai TRUE atau FALSE.

Operator untuk jenis data ini adalah :

1. Logical Operator, yaitu : NOT, AND dan OR
2. Relational Operator, yaitu : >, <, >=, <=, <> dan =\

## JENIS DATA NON-STANDARD (USER DEFINED)

### 1. ENUMERATED.

Jenis data ini terdiri atas barisan identifier yang terurut dimana setiap identifier tersebut dianggap sebagai suatu individual data item (elemen data yang berdiri sendiri). Pada saat mendeklarasikan jenis data ini kita harus menuliskan semua elemen-elemennya.

Bentuk umum deklarasinya adalah :

```
TYPE nama = (data_item_1, data_item_2, ....., data_item_n);
```

Contoh :

```
TYPE hari = (sen, sel, rab, kam, jum, sab, ming);
```

```
TYPE warna = (red, blue, green, yellow, black, white);
```

Setelah jenis data ini dideklarasikan, maka selanjutnya kita dapat mendeklarasikan suatu variabel yang berjenis data sama dengan jenis data ini.

Misalnya :

```
TYPE nama_hari = (sen,sel,rab,kam,jum,sab,ming);  
VAR libur : nama_hari;
```

Fungsi standar yang dapat digunakan pada jenis data ini adalah :

PRED, SUCC dan ORD

Misalnya :

```
PRED (sel) = sen  
SUCC (sen) = sel  
ORD (sen) = 0  
ORD (sel) = 1
```

## 2. SUB-RANGE.

Jenis data ini berupa range dari suatu kumpulan data yang mempunyai urutan..

Bentuk umum deklarasinya adalah :

```
TYPE nama = data_item_pertama .. data_item_terakhir;
```

Contoh :

1. TYPE jam\_kuliah = 1 .. 10;  
tanggal = 1 .. 31;  
abjad = 'A' .. 'Z';
2. TYPE bulan = (jan,feb,mar,apr,mei,jun,jul,agt,sep,okt,nov,des);  
hari = (sen,sel,rab,kam,jum,sab,ming);  
ata = agt .. jan;  
pta = feb .. jun;  
hari\_kerja = sen .. jum;

## TANDA OPERASI

Tanda operasi (operator) di dalam bahasa Pascal dikelompokkan ke dalam 9 kategori, yaitu

1. Assignment operator.
2. Binary operator.
3. Unary operator.
4. Bitwise operator.
5. Relational operator.
6. Logical operator.
7. Address operator.
8. Set operator.
9. String operator.

### Assignment operator

Assignment operator (operator pengerjaan) menggunakan simbol titik dua diikuti oleh tanda sama dengan (:=).

Contoh :

A:=B;

### Binary operator

Digunakan untuk mengoperasikan dua buah operand. Operand dapat berbentuk konstanta ataupun variabel. Operator ini digunakan untuk operasi aritmatika yang berhubungan dengan nilai tipe data integer dan real.

Operat or	Operasi	Tipe operand	Tipe hasil
*	Perkalian	real,real integer,integer real,integer	real integer real
DIV	Pembagian bulat	integer,integer	integer
/	Pembagian real	real,real integer,integer real,integer	real real real
MOD	Sisa pembagian	integer,integer	integer
+	Pertambahan	real,real	real

		integer,integer real,integer	integer real
-	pengurangan	real,real integer,integer real,real	real integer real

Contoh :

15*5	hasilnya	75
20/3	hasilnya	6.6666666667E+00
20 div 3	hasilnya	6
20 mod 3	hasilnya	2

### Unary operator

Operator ini hanya menggunakan sebuah operand saja. Dapat berupa unary minus dan unary plus. Unary minus digunakan untuk menunjukkan nilai negatif, baik pada operang numerik real maupun integer. Unaru plus adalah operator untuk memberai tanda plus.

Contoh :

-5	+7
-2.5	+2.5

### Bitwise operator

Digunakan untuk operasi bit per bit pada nilai integer. Terdiri dari operator NOT, AND, OR, XOR, Shl, Shr.

### Relational operator

Relational operator digunakan untuk membandingkan hubungan antara dua buah operand dan akan didapatkan hasil tipe boolean, yaitu True atau False. Terdiri dari operator : =, <, >, <=, >=, <>

### **Logical operator**

Terdapat 4 buah logical operator yaitu : NOT, AND, OR dan XOR. Operator ini bekerja dengan nilai-nilai logika, yaitu True dan False.

### **Set operator**

Digunakan untuk operasi himpunan.

### **String operator**

Digunakan untuk operasi string. Hanya ada sebuah operator string saja, yaitu operator + yang digunakan untuk menggabungkan dua buah nilai string.

Contoh :

```
Nama1 := 'Arief';  
Nama2 := 'Kurniawan';  
Nama3 := Nama1 + Nama2;
```

### **Konstanta, Variabel dan Ekspresi.**

#### **Konstanta.**

Konstanta adalah suatu identifier non-standar yang nilainya telah ditetapkan dalam suatu program dan dideklarasikan pada bagian deklarasi.

Bagian umum deklarasinya adalah :

```
CONST identifier = nilai;
```

Contoh :

```
CONST    phi = 3.14;  
        x = 2;  
        y = 'S';
```

#### **Variabel.**

Variabel adalah suatu identifier non-standar yang nilainya tidak tetap atau nilainya merupakan hasil dari suatu proses.

(variabel yang dimaksudkan disini sama seperti halnya arti variabel pada aljabar).

Bentuk umum deklarasinya adalah :

1. Jika hanya 1 (satu) variabel yang dideklarasikan :  
VAR identifier : jenis data;
2. Jika lebih dari 1 (satu) variabel dan masing-masing memiliki jenis data yang sama :  
VAR id-1, id-2, ..., id-3 : jenis data;
3. Jika beberapa variabel yang berbeda jenis datanya :  
VAR identifier-1 : jenis data;  
identifier-2 : jenis data;  
. . .  
identifier-n : jenis data;
4. Jika variabel yang dideklarasikan berjenis data non-standar :  
TYPE id = (data\_item\_1,data\_item\_2,...,data\_item\_n);  
VAR id-v : id;  
SUB-RANGE :  
TYPE id = data\_item\_pertama .. data\_item\_terakhir;  
VAR id-v : id;

Contoh :

1. VAR x : integer;
2. VAR p,q,r : real;
3. VAR a,b : char;  
m,n : boolean;
4. TYPE warna = (merah,hitam,biru,putih,kuning);  
VAR x1,x2 : warna;
5. TYPE tahun = 1900 .. 1993;  
VAR thn-1,thn-2,thn-3 : tahun;

## Ekspresi.

Sebuah ekspresi merupakan kumpulan dari operand-operand (seperti : bilangan, konstanta, variabel dll) yang bersama-sama dengan operator membentuk suatu bentuk aljabar dan menyatakan suatu nilai.

Ada 2 (dua) jenis ekspresi dalam bahasa Pascal, yaitu :

1. Ekspresi numerik / aritmatika, yaitu suatu ekspresi yang menghasilkan nilai numerik / aritmatika.
2. Ekspresi Boolean atau ekspresi logika, yaitu suatu ekspresi yang menghasilkan nilai boolean / logika (true/false).

Contoh :

1.  $(b * b - 4 * a * c) / (2 * a) / (2 * a) \Rightarrow$  ekspresi numerik, jika a,b dan c adalah bilangan (variabel bernilai numerik).
2.  $Upah < 1000.0 \Rightarrow$  ekspresi boolean (“upah” adalah suatu variabel bernilai real).

## Input dan output (Standar I/O).

Dalam bahasa Pascal untuk keperluan input (membaca input) digunakan identifier standar READ atau READLN.

Identifier standar ReadLn sedikit berbeda dengan Read. ReadLn digunakan untuk memasukkan data perbaris, artinya setelah tombol Enter ditekan, maka akan ganti baris, sedangkan Read tidak ganti baris, masih dalam baris yang sama.

Sedangkan untuk keperluan output (mencetak output) digunakan identifier standar WRITE atau WRITLN.

Perbedaan Write dengan WriteLn adalah bahwa Write menampilkan hasil tanpa ganti baris dan tampilan berikutnya akan disambung dalam baris yang sama. Sedang WriteLn digunakan untuk menampilkan tampilan perbaris, akan ganti baris untuk tampilan berikutnya.

Bentuk umum penulisannya :

1. READ (variabel input) ;  
Atau  
    READLN (variabel output);
2. WRITE (variabel output);  
    WRITELN (variabel output);

Contoh :

```
Var nama : string{15};
```

```
Begin nama := 'Dewi';
```



```
write('nama : ');  
write(nama);
```

End.

Hasilnya :

Nama : Dewi

Contoh :

```
Var nama : string{15};  
Begin nama := 'Dewi';  
WriteLn('Nama : ');  
WriteLn>Nama);
```

End.

Hasilnya :

Nama :  
Dewi

Resensi:

Wikipedia

Universitas Gunadarma staffsite