



Variabel dan Tipe Data

Tim Olimpiade Komputer Indonesia

Pendahuluan

Melalui dokumen ini, kalian akan:

- Mengenal konsep variabel.
- Mempelajari berbagai tipe data.
- Mempelajari cara deklarasi variabel.
- Mengenal operasi assignment.



Kilas Balik

- Mari kita lihat kembali program halo.pas.

```
begin  
    writeln('Halo Dunia!');  
end.
```

- Pada program tersebut, terdapat kata kunci **begin** dan **end** yang diakhiri dengan titik. Kedua kata kunci tersebut membentuk sesuatu yang disebut dengan blok program utama.
- Ketika halo.pas dieksekusi, seluruh perintah yang dituliskan dalam blok program utama akan dieksekusi secara sekuensial (berurutan).



Baris Perintah Program

- Pada `halo.pas`, satu-satunya perintah yang ada adalah **`writeln('Halo Dunia!');`**
- Pada Pascal, **`writeln(x)`** merupakan fungsi untuk mencetak `x` ke layar. Untuk **`halo.pas`**, `x = 'Halo Dunia!'`.



Bagian 1

Konsep Variabel



Perkenalan Variabel

Variabel

Merupakan istilah yang diadopsi dari dunia matematika, yang memetakan sebuah nama ke suatu nilai.

- Setiap kali suatu variabel digunakan dalam ekspresi matematika, yang diacu sebenarnya adalah nilai yang dipetakan oleh nama variabel tersebut.
- Contoh: jika kita menyatakan $x = 5$, maka hasil dari $3x^2 + x$ adalah 80.
- Dalam pemrograman, kita bisa membuat variabel, mengisikan nilai pada variabel, dan mengacu nilai yang dipetakan variabel tersebut.



Aturan Penamaan Variabel

- Variabel bebas diberi nama apapun, tetapi terbatas pada beberapa aturan berikut:
 - Terdiri dari kombinasi karakter huruf, angka, dan underscore (-).
 - Tidak boleh dimulai dengan angka. Artinya "2kar" bukan merupakan nama variabel yang sah.
 - Huruf kapital dan huruf kecil dianggap sama. Artinya "a1" dan "A1" dianggap merupakan dua variabel yang sama.
 - Tidak boleh merupakan *reserved word*. Contoh *reserved word* pada Pascal: **begin**, **end**, **var**, **if**, atau **then**.
- Lebih jauh lagi, aturan ini berlaku pada seluruh penamaan *identifier*, yaitu nama variabel, fungsi, dan prosedur (akan dipelajari selanjutnya).



Assignment

Assignment

Pengisian nilai yang diacu oleh variabel dengan suatu nilai disebut *assignment*.

- Operator untuk *assignment* adalah $:=$
- Isikan ruas kiri dengan nama suatu variabel, dan ruas kanan dengan nilai yang ingin diisikan ke variabel tersebut.
- Tipe data dari variabel dan nilai yang diacu **harus sesuai**.



Contoh Program: assign.pas

- Perhatikan contoh program assign.pas berikut. Tuliskan, lalu jalankan program ini.
-

```
var
  x: integer;
begin
  x := 12;
  writeln('Nilai = ', x);
end.
```



Penjelasan Program: assign.pas

- Keluaran yang dihasilkan dari program itu adalah sebuah baris berisikan:

Nilai = 12

- Pada program tersebut, **x** merupakan suatu variabel.
- Variabel **x** didaftarkan terlebih dahulu dengan menuliskan **x: integer** pada bagian antara **var** dengan **begin**. Semua variabel yang akan digunakan pada program utama wajib didaftarkan pada bagian ini.
- Pada blok program utama, **x** diisi dengan nilai 12, lalu perintah **writeln** dieksekusi.



Contoh Program: assign2.pas

- Berikut adalah contoh program yang melibatkan beberapa variabel.

```
var
  x: integer;
  y: longint;
begin
  x := 12;
  y := 123456;
  writeln('Nilai x = ', x);
  writeln('Nilai y = ', y);

  x := 15;
  writeln('Sekarang nilai x = ', x);
end.
```



Penjelasan Program: assign2.pas

- Keluaran yang dihasilkan dari program itu adalah:

Nilai x = 12

Nilai y = 123456

Sekarang nilai x = 15

- Apa maksud dari kata kunci **integer** dan **longint**? Dijelaskan pada bagian selanjutnya.



Bagian 2

Tipe Data Variabel



Tipe Data Variabel

- Setiap variabel pada Pascal wajib memiliki sebuah **tipe data**.
- Jenis tipe data dasar dari suatu variabel pada:
 - Bilangan bulat: hanya bisa berisikan bilangan bulat.
 - Bilangan riil: bisa berisikan bilangan riil (termasuk pecahan).
 - Karakter: merepresentasikan karakter, seperti 'a', 'b', '3', atau '?'.
 - String: merepresentasikan untaian karakter, seperti 'aku' atau 'halo dunia'.
 - Nilai kebenaran: merepresentasikan benar (**TRUE**) atau salah (**FALSE**).



Tipe Data: Bilangan Bulat

| Nama | Jangkauan | Ukuran |
|----------|-----------------------|--------|
| shortint | $-2^7..2^7 - 1$ | 1 byte |
| byte | $0..2^8 - 1$ | 1 byte |
| integer | $-2^{15}..2^{15} - 1$ | 2 byte |
| word | $0..2^{16} - 1$ | 2 byte |
| longint | $-2^{31}..2^{31} - 1$ | 4 byte |
| longword | $0..2^{32} - 1$ | 4 byte |
| int64 | $-2^{63}..2^{63} - 1$ | 8 byte |
| qword | $0..2^{64} - 1$ | 8 byte |

- Biasa disebut dengan *integer* saja.
- Pascal menawarkan beberapa tipe data bilangan bulat yang variasinya terletak pada jangkauan nilai yang bisa direpresentasikan dan ukurannya pada memori.
- Dalam memprogram, yang umum digunakan adalah **longint** dan **int64**.



Tipe Data: Bilangan Riil

| Nama | Jangkauan (magnitudo) | Akurasi | Ukuran |
|--------|---|-------------|--------|
| single | 1.5×10^{-45} .. 3.4×10^{38} | 7-8 digit | 4 byte |
| double | 5.0×10^{-324} .. 1.7×10^{308} | 15-16 digit | 8 byte |

- Biasa disebut dengan *floating point*.
- Tipe data *floating point* bisa merepresentasikan negatif atau positif dari magnitudonya.
- Pada pemrograman, umumnya tipe data *floating point* dihindari karena kurang akurat. Representasi 3 pada *floating point* bisa jadi 2.999999999999999 atau 3.0000000000000001 karena keterbatasan pada struktur penyimpanan bilangan pecahan pada komputer.
- Tipe yang umum digunakan adalah **double**.



Tipe Data: Karakter

- Merupakan tipe data untuk merepresentasikan karakter menurut ASCII (*American Standard Code for Information Interchange*).
- Dalam ASCII, terdapat 128 karakter yang direpresentasikan dengan angka dari 0 sampai 127.
- Misalnya, kode ASCII untuk karakter spasi (' ') adalah 32, huruf 'A' adalah 65, 'B' adalah 66, huruf 'a' adalah 97, dan huruf 'b' adalah 98.



Tipe Data: Karakter (lanj.)

- Pada Pascal, tipe data ini dinyatakan sebagai **char**, dengan ukuran 1 byte.
- Dalam memprogram, kita bisa menggunakan fungsi **ord(x)** untuk mendapatkan nilai ASCII dari **char x**, dan **chr(x)** untuk mendapatkan karakter dari kode ASCII **x**.



Contoh Program: karakter.pas

- Tulis dan coba eksekusi program berikut.
-

```
begin
  writeln(chr(65));
  writeln(chr(66));
  writeln(chr(49));
  writeln(ord('A'));
  writeln(ord('a'));
  writeln(ord('1'));
  writeln(chr(ord('a')));
end.
```



Penjelasan Program: karakter.pas

- Berikut adalah keluaran dari contoh program tersebut.
-

A

B

1

65

97

49

a

- Dari contoh tersebut, kalian dapat memahami penggunaan **chr** dan **ord**.
- Perhatikan pula penulisan suatu karakter perlu menggunakan tanda petik tunggal (').
- Bedakan antara 1 dengan '1'. Menuliskan **ord(1)** bisa mengakibatkan *error* dan program tidak bisa dikompilasi.



Tipe Data: String

- Tipe data yang merepresentasikan untaian dari **char**, contohnya 'kucing', 'dan', '' (string kosong). Tipe data ini dinyatakan sebagai **string**.
- Pada Pascal, **string** bisa menampung antara 0 sampai 255 karakter. Untuk menampung lebih dari 255 karakter, gunakan tipe data **ansistring**.



Tipe Data: Boolean

- Merupakan tipe data yang menyimpan nilai kebenaran, yaitu hanya **TRUE** atau **FALSE**.
- Tipe data ini akan lebih terasa kebermanfaatannya ketika kita sudah mempelajari struktur percabangan dan **array**.
- Pada Pascal, kalian dapat menggunakan tipe data **boolean**.



Deklarasi Variabel

- Mendaftarkan nama-nama dan tipe variabel yang akan digunakan disebut dengan deklarasi variabel.
- Pada saat dideklarasikan, setiap variabel perlu disertakan tipe datanya.



Deklarasi Variabel (lanj.)

- Pada Pascal, variabel dideklarasikan di antara **var** dengan **begin**.
- Tipe data dituliskan sesudah tanda titik dua (:), setelah nama variabel dituliskan.
Contoh: **nilai: longint** atau **rerata: double**.
- Beberapa variabel juga bisa dideklarasikan secara bersamaan jika memiliki tipe data yang sama. Contoh: **x, y: double**.



Contoh Program: `tipedasar.pas`

- Pahami program berikut ini dan coba jalankan!
-

```
var
  p1, p2: longint;
  x, y: double;
  teks: string;
begin
  p1 := 100;
  p2 := p1;
  writeln('p1: ', p1, ' p2: ', p2);

  x := 3.1418;
  y := 234.432;
  writeln('x: ', x);
  writeln('y: ', y);

  teks := 'ini adalah string';
  writeln('teks: ', teks);
end.
```



Penjelasan Program: `tipedasar.pas`

- Berikut adalah keluaran dari program `tipedasar.pas`:

```
p1: 100 p2: 100  
x: 3.141800000000000E+000  
y: 2.344320000000000E+002  
teks: ini adalah string
```

- Perhatikan bahwa perintah **`p2 := p1`** sama artinya dengan **`p2 := 100`**, karena **`p1`** sendiri mengacu pada nilai 100.
- Untuk tipe data *floating point*, bilangan tercetak dalam notasi *scientific*, yaitu 3.1418×10^0 dan 2.34432×10^2 .



Tipe Data Komposit: Record

- Kadang-kadang, kita membutuhkan suatu tipe data yang sifatnya komposit; terdiri dari beberapa data lainnya.
- Contoh kasusnya adalah ketika kita butuh suatu representasi dari titik. Setiap titik pada bidang memiliki dua komponen, yaitu **x** dan **y**.



Tipe Data Komposit: Record (lanj.)

- Memang bisa saja kita mendeklarasi dua variabel, yaitu **x** dan **y**. Namun bagaimana jika kita hendak membuat beberapa titik? Apakah kita harus membuat **x1, y1, x2, y2, ...**?
Sungguh melelahkan!
- Karena itulah Pascal memiliki suatu tipe data komposit, yaitu **record**.



Tipe Data Komposit: Record (lanj.)

- **Record** dapat dideklarasikan pada blok **type**, yang letaknya sebelum **var**.
-

type

```
<nama record> =  
  record  
    <variabel1>: <tipe1>;  
    <variabel2>: <tipe2>;  
    ...  
  end;
```

var

...

- Setelah dideklarasikan, sebuah tipe data **<nama record>** sudah bisa digunakan.
- Untuk mengakses nilai dari **<variabel 1>** dari suatu variabel bertipe **record**, gunakan tanda titik (.).



Tipe Data Komposit: Record (lanj.)

- Sebagai contoh, perhatikan contoh program titik.pas berikut:

```
type
    titik =
        record
            x, y: longint;
        end;
var
    t1, t2: titik;
begin
    t1.x := 5;
    t1.y := 3;

    t2.x := 1;
    t2.y := 2;

    writeln(t1.x, ', ', t1.y);
    writeln(t2.x, ', ', t2.y);
end.
```



Konsumsi Memori Record

- Memori yang dibutuhkan bagi sebuah tipe data **record** sama dengan jumlah memori tipe data yang menyusunnya.
- Artinya, **record** bernama **titik** pada contoh titik.pas mengkonsumsi memori yang sama dengan dua buah longint, yaitu 8 byte.



Ordinalitas

- Menurut keberurutannya, tipe data dapat dibedakan menjadi tipe data **ordinal** atau **non-ordinal**.
- Suatu tipe data memiliki sifat ordinal jika untuk suatu elemennya, kita bisa mengetahui secara pasti apa elemen sebelum atau selanjutnya. Contoh:
 - Diberikan bilangan bulat 6, kita tahu pasti sebelumnya adalah angka 5 dan sesudahnya adalah angka 7.
 - Diberikan karakter 'y', kita tahu pasti sebelumnya adalah karakter 'x' dan sesudahnya adalah karakter 'z'.
- Dengan demikian, seluruh tipe data bilangan bulat dan karakter adalah tipe data ordinal.



Ordinalitas (lanj.)

- Kebalikannya, suatu tipe data dinyatakan memiliki sifat non-ordinal jika kita tidak bisa menentukan elemen sebelum dan sesudahnya. Contohnya:
 - Diberikan bilangan riil 6, apakah elemen sesudahnya 7, atau 6.1, atau 6.01, atau 6.001, atau 6.000000000001?
 - Diberikan string 'telur', apakah elemen sesudahnya? Apakah 'ayam'? 'bebek'?
- **String** dan bilangan *floating point* termasuk dalam tipe data non-ordinal.



Yang Sudah Kita Pelajari...

- Mengenal konsep variabel.
- Mempelajari berbagai tipe data.
- Mempelajari cara deklarasi variabel.
- Mengenal operasi *assignment*.



Selanjutnya...

- Mengenal tentang ekspresi.
- Mengenal input dan output.

