

# JOB SHEET 03

## Variabel, Konstanta & Tipe Data

*"Everybody should learn to program a computer,  
because it teaches you how to think."*

### TUJUAN PEMBELAJARAN

1. Mampu menjelaskan dan mengimplementasikan *Variables, Constanta, Data Types, Type Casting, Keywords* dalam pemrograman menggunakan IDE.

### POKOK MATERI

1. Variabel & Konstanta
2. Tipe Data Fundamental
3. Type Casting
4. Keywords

### URAIAN MATERI

---

#### A. Variabel



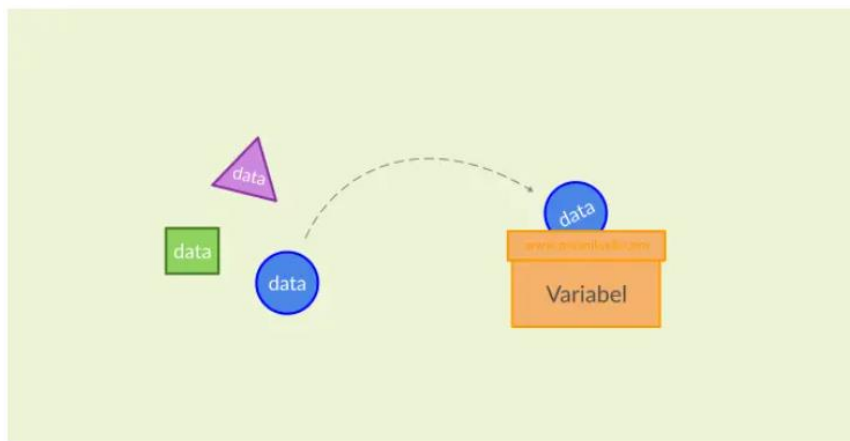
Inti dari sebuah program komputer adalah menerima input, melakukan pemrosesan, dan menghasilkan output. Nilai input bisa kita dapatkan dari keyboard, file, kamera, mikrofon, dan sebagainya. Sementara output dapat kita tampilkan ke monitor, cetak ke dokumen, atau ke dalam sebuah file. Pada tahap

pemrosesan, program membutuhkan bantuan variabel untuk menyimpan nilai sementara.

Variabel adalah sebuah tempat menyimpan sebuah nilai. Sementara tipe data adalah jenis nilai yang akan tersimpan dalam variabel. Pada pelajaran matematika, kita sering menemukan x dan y.

x dan y ini disebut variabel, karena tugasnya menyimpan nilai.

```
x = 3;  
y = 4;
```



Anggap saja variabel itu sebuah kotak, lalu tipe data itu jenis-jenis benda yang akan disimpan dalam kotak tersebut. Berikut ini cara membuat variabel pada C:

```
int nama_variabel;  
int nama_variabel = 0;
```

tipe data

jangan lupa  
akhiri dengan  
titik koma

variabel bisa juga  
langsung diisi

petanikode.com

Jadi, pertama kita harus menuliskan tipe data lalu diikuti dengan nama variabelnya.

```
int tinggi;
```

Artinya:

Kita membuat variabel dengan nama tinggi dan tipe datanya adalah int (integer). Saat membuat variabel, kita juga bisa mengisi langsung dengan nilai yang ingin kita simpan.

```
int berat = 49;
```

Artinya:

Kita akan membuat variabel dengan nama berat dan tipe data integer, lalu langsung diisi dengan nilai 49.

### B. Aturan Penulisan Variabel pada C

Ada beberapa aturan penulisan variabel yang harus kamu ketahui:

1. Nama variabel tidak boleh didahului dengan simbol dan angka.
2. Nama variabel tidak boleh menggunakan kata kunci yang sudah ada pada bahasa C, contoh: **if**, **int**, **void**, dll.
3. Nama variabel bersifat case sensitive, artinya huruf besar dan kecil dibedakan, contoh: **nama** dan **Nama** adalah dua variabel yang berbeda.
4. Disarankan menggunakan underscore untuk nama variabel yang terdiri dari dua suku kata, contoh: **nama\_lengkap**.

```

1  #include <stdio.h>
2
3  int main(){
4      // membuat variabel dengan tipe data integer
5      int berat;
6      // membuat variabel dan langsung mengisinya
7      int tinggi = 178;
8
9      // mengisi nilai ke variabel
10     // berat = 54;
11
12     // mencetak isi variabel ke layar
13     printf("Berat badan saya adalah %ikg dan tinggi %icm", berat, tinggi);
14
15     return 0;
16 }
17

```

Annotations in the image:

- Line 3: *membuat variabel berat* (points to `int berat;`)
- Line 5: *membuat variabel dengan tipe data integer* (points to `int`)
- Line 7: *membuat variabel dan langsung mengisinya* (points to `int tinggi = 178;`)
- Line 10: *pengisian variabel berat dihapus* (points to `// berat = 54;`)

Jadi kita cukup menuliskan nama variabel pada fungsi `printf()` untuk menampilkan isinya. Isi dari variabel berat akan sama dengan nol 0, karena kita tidak mengisinya. Ini adalah nilai default yang diberikan.

### C. Tipe Data

Tipe data adalah jenis data.

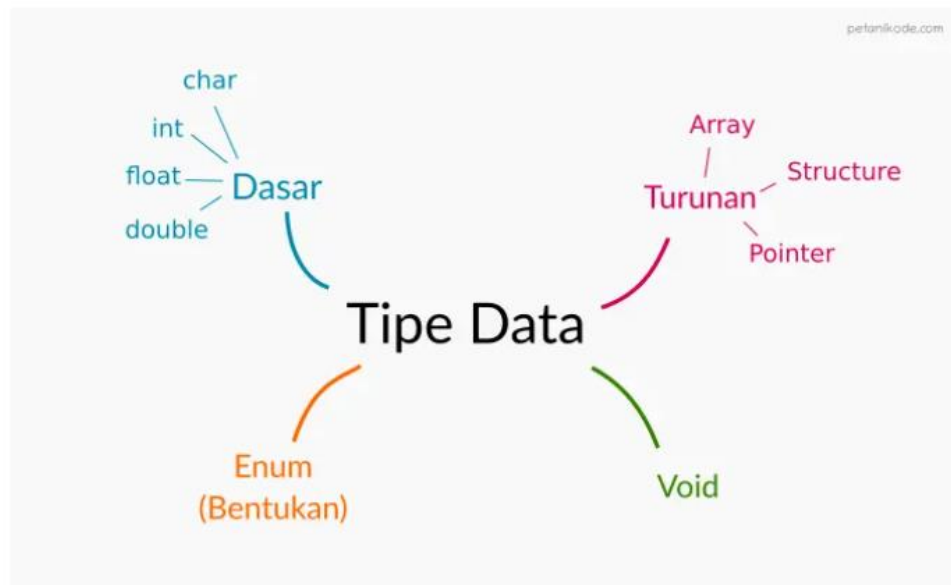
Menurut Wikipedia:

Tipe data atau kadang disingkat dengan 'tipe' saja adalah sebuah pengelompokan data untuk memberitahu compiler atau interpreter bagaimana programmer ingin mengolah data tersebut.

Ada delapan jenis tipe data pada C:

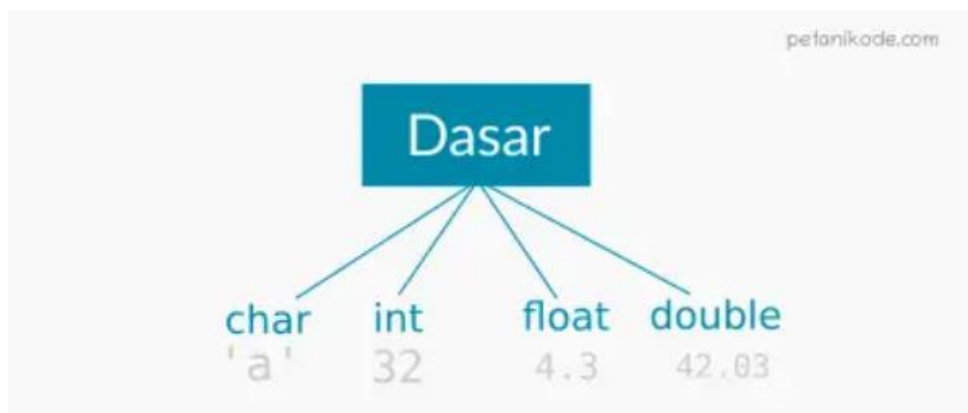
1. Char
2. Integer
3. Float
4. Array
5. Structure
6. Pointer
7. Enum
8. Void

Apabila kita kelompokkan, maka akan terdapat empat kelompok tipe data.



### Tipe Data Dasar

Sesuai namanya, tipe data dasar adalah tipe data yang paling dasar dalam bahasa pemrograman C.



Char : adalah tipe data yang berisi 1 huruf atau 1 karakter;

Integer : adalah tipe data yang berupa angka;

Float : adalah tipe data yang berupa bilangan pecahan

Double : adalah tipe data sama seperti float, namun double memiliki ukuran penyimpanan yang lebih besar dibandingkan float.

Setiap tipe data memiliki panjang dan ukuran, perhatikan tabel berikut.

Tipe Data	Ukuran	Panjang	Contoh
char	1 byte	-128 – 127 atau 0 – 255	'A', 'a', '3'
int	2 byte	-32,768 – 32,767 atau -2,147,483,648 – 2,147,483,647	32, 1, 4
float	4 byte	1.2E-38 – 3.4E+38	4.3, 2.2, 6.0
double	8 byte	2.3E-308 – 1.7E+308	4.2, 4.22, 3.2

Format penulisan koma untuk tipe data float dan double menggunakan tanda titik, mengikuti format internasional. Jika kamu menulis seperti ini 2,5 maka akan salah. Penulisan yang benar adalah 2.5.

Lalu untuk penulisan tipe data char harus diapit dengan tanda petik. Mengapa? Karena kalau misalnya karakter itu adalah spasi— tanpa tanda petik—kita tidak akan bisa mengisinya dengan karakter spasi.

```
#include <stdio.h>

int main(){
    int usia = 19;
    float berat = 65.3;
    double tinggi = 178.43;
    char jenis_kelamin = 'L';

    printf("Usia: %i tahun\n", usia);
    printf("Tinggi: %.2f cm\n", tinggi);
    printf("Berat: %.2f Kg\n", berat);
    printf("Jenis kelamin: %c\n", jenis_kelamin);

    return 0;
}
```

Mengapa kita menggunakan `%.2f` pada saat mencetak tipe data float dan double? Kita bisa saja mencetaknya dengan format `%f` saja, akan tetapi akan dicetak dengan angka yang panjang.

```
Tinggi: 178.430000 cm  
Berat: 65.300003 Kg
```

Kita menggunakan `%.2f` untuk menentukan batas jumlah angka setelah koma. Angka 2 sebelum f artinya dua angka di belakang koma.

```
Tinggi: 178.43 cm  
Berat: 65.30 Kg
```

### Tipe Data Void

Tipe data void adalah tipe data khusus untuk menyatakan tidak ada data.

Apakah nol 0 sama dengan void?

Tidak, nol memiliki nilai yaitu 0 sementara void tidak ada.



**0**



**null**



**undefined**

Nol 0 artinya tisunya habis, null atau void artinya tisunya tidak ada. Lalu *undefined* artinya variabelnya belum dibuat dalam memori. Biasanya, *undefined* akan kamu temukan saat belajar Javascript.

```
void main(){  
    //...  
}
```

Artinya fungsi main() tidak akan mengembalikan nilai apa-apa.

#### D. Konversi Tipe Data

Konversi tipe data atau type casting adalah proses mengubah tipe data menjadi tipe yang lain. Kenapa kita perlu konversi tipe data?

Untuk menjawabnya, perhatikan ilustrasi berikut ini:



Air yang bentuknya cair tidak bisa disimpan dalam kardus. Karena itu, air harus konversi dulu menjadi bentuk padat (es) agar bisa disimpan dalam kardus. Begitu juga dengan variabel



Cara mengubah tipe data pada C, cukup dengan menuliskan nama tipe datanya di depan nilai atau variabel yang ingin dibuat.

```
float a = (float) 10;
```

Artinya nilai 10 akan diubah menjadi tipe float.

```
#include <stdio.h>

void main(){
    int a = 5;
    int b = 2;

    float c = a / b;

    printf("Hasil a / b = %.2f\n", c);
}
```

Program di atas akan menghasilkan keluaran sebagai berikut:

```
Hasil a / b = 2.00
```

Mengapa 2.00? bukannya 2.50?

Ini disebabkan karena variabel a dan b bertipe integer. Karena hal tersebut hasil pembagian dari tipe data integer akan selalu dibulatkan.

```

1  #include <stdio.h>
2
3  void main(){
4      int a = 5;
5      int b = 2;
6
7      float c = (float) a / (float) b;
8
9      printf("Hasil a / b = %.2f\n", c);
10 }

```

type casting menjadi float

### E. Konstanta

Konstanta adalah sebuah nilai tetapan.

Bisa juga dibilang sebagai variabel yang tidak bisa diubah nilainya.

Ada dua cara pembuatan konstanta pada C:

1. Menggunakan #define

```

#include <stdio.h>
#define SEPULUH 10
#define VERSI 4.5
#define JENIS_KELAMIN 'L'

void main(){
    printf("isi konstanta SEPULUH adalah %i\n", SEPULUH);
    printf("isi konstanta VERSI adalah %f\n", VERSI);
    printf("isi konstanta JENIS_KELAMIN adalah %c\n", JENIS_KELAMIN);
}

```

2. Menggunakan const

```

#include <stdio.h>

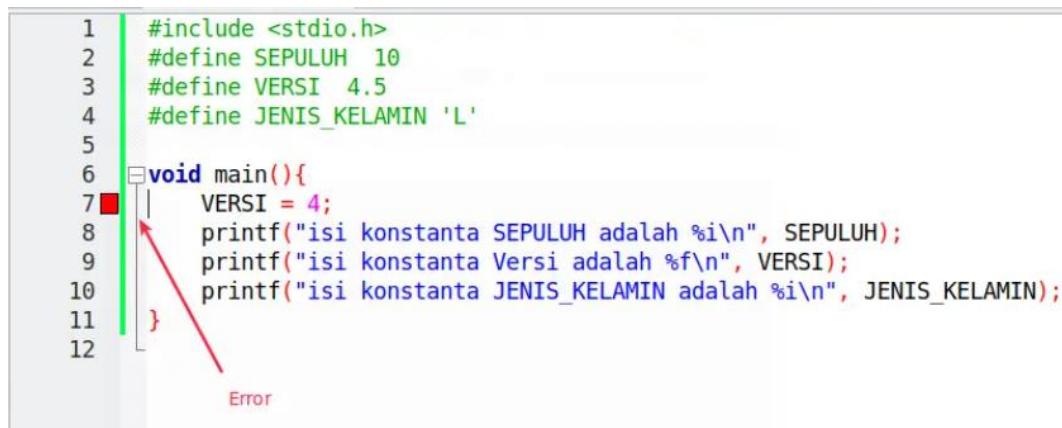
void main(){
    const double PI = 3.14;
    const char JENIS_KELAMIN = 'P';
    const int VERSI = 11;

    printf("isi konstanta PI adalah %f\n", PI);
    printf("isi konstanta JENIS_KELAMIN adalah %c\n", JENIS_KELAMIN);
    printf("isi konstanta VERSI adalah %i\n", VERSI);
}

```

Perbedaan pembuatan konstanta yang menggunakan `#define` dan `const` terletak pada format penulisannya. Pada `#define` kita tidak perlu menuliskan tipe data, sedangkan `const` harus. Pada `#define` kita tidak membutuhkan titik koma di akhir, sedangkan pada `const` kita harus menuliskan titik koma. Posisi penulisan untuk `#define` dan `const` bisa ditulis di dalam `main()` maupun di luar.

Untuk nama konstanta disarankan menggunakan huruf kapital untuk menandakan itu sebuah konstanta. Apa yang akan terjadi jika saya mencoba mengisi nilai ke dalam konstanta? Ya programnya akan error.



```
1  #include <stdio.h>
2  #define SEPULUH 10
3  #define VERSI 4.5
4  #define JENIS_KELAMIN 'L'
5
6  void main(){
7      VERSI = 4;
8      printf("isi konstanta SEPULUH adalah %i\n", SEPULUH);
9      printf("isi konstanta Versi adalah %f\n", VERSI);
10     printf("isi konstanta JENIS_KELAMIN adalah %i\n", JENIS_KELAMIN);
11 }
12
```

Error

## LATIHAN

1. Buatlah sebuah program konversi mata uang Rupiah (Rp) ke mata uang Dollar (\$)! jika, 1\$ = Rp. 14.250.

**Kasus:**

Anda mempunyai jasa money changer, kemudian ada yang meminta penukaran uang rupiah ke dolar, berapa dolar yang harus anda berikan kepada si penukar untuk Rp. 2.500.000?

Hitung menggunakan program yang anda buat!

2. Buatlah sebuah program konversi suhu dengan inputan data dari keyboard. Dengan ketentuan Rumus sebagai Berikut:

Kelvin  $k = c + 273.15$

Fahrenheit  $f = (c \times 1.8) + 32$

Reamur  $r = c \times 0.8$

**Hasil keluaran output adalah sebagai berikut:**

Input Suhu (Celcius) =

Suhu dalam Kelvin =

Suhu dalam Fahrenheit =

Suhu dalam Reamur =

3. Diketahui sebuah bola basket memiliki diameter 12 cm. Buatlah program dalam bahasa C untuk menghitung dan menampilkan volume dan luas permukaan dari bola tersebut. Gunakan konstanta untuk nilai Phi!

Volume Bola	Luas Permukaan Bola
$V = \frac{4}{3} \pi r^3$	$L = 4 \pi r^2$

## REFERENCES

---

1. The C Programming Language. 2nd Edition
2. [https://en.wikipedia.org/wiki/C\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/C_(programming_language))
3. [https://en.wikipedia.org/wiki/Imperative\\_programming](https://en.wikipedia.org/wiki/Imperative_programming)
4. <https://www.petanikode.com/tutorial/c/>
5. <https://www.learn-c.org/>
6. <https://www.tutorialspoint.com/cprogramming/index.htm>
7. <https://www.programiz.com/>
8. <https://www.dicoding.com/>
9. <https://data-flair.training/blogs/c-tutorials-home/>