

PEMROGRAMAN BERORIENTASI OBJEK

Tipe Data

TIPE DATA

- ***Tipe Data*** berfungsi untuk merepresentasikan jenis dari nilai yang terdapat dalam program.
- Contoh apabila mempunyai suatu data dengan nilai 2, maka 2 termasuk kedalam *tipe bilangan bulat*.

TIPE DATA

- Kesalahan dalam menyebutkan ***tipe data*** akan menyebabkan program yang dibuat tidak dapat dijalankan, walaupun dipaksa untuk dijalankan maka program akan memberikan nilai yang *tidak akurat* sesuai yang diharapkan.

TIPE DATA

Tipe data terbagi menjadi 2 yaitu :

- Tipe Data ***Dasar*** dan;
- Tipe Data ***Bentukan***.

TIPE DATA DASAR

- Dalam C++ terdapat beberapa ***tipe data dasar*** yang telah didefinisikan , yaitu digolongkan kedalam tipe data ***bilangan bulat*** (Integer), ***bilangan riil*** (Floating Point), tipe data ***logika*** (Boolean), dan tipe data ***Karakter***.
- ***Tipe-tipe data*** tersebut adalah tipe yang siap digunakan tanpa harus didefinisikan terlebih dahulu.

TIPE DATA BILANGAN BULAT

- ***Tipe Data Bilangan Bulat ini*** digunakan untuk merepresentasikan data numerik yang berupa bilangan bulat, yaitu bilangan yang tidak mengandung angka dibelakang koma.
- Contoh data bilangan bulat adalah 30 atau 6 atau 92 dan seterusnya.

TIPE DATA BILANGAN BULAT

TIPE DATA	UKURAN (bit)	RENTANG
Int	16 atau 32	-32.768 s/d 32.767 atau -2.147.483.648 s/d 2.147.483.647
Unsigned Int	16 atau 32	0 s/d 65.535 atau 0 s/d 4.294.967.295
Signed Int	16 atau 32	Sama seperti Int
Short Int	16	-32.768 s/d 32.767
Unsigned Short Int	16	0 s/d 65.535
Signed Short Int	16	Sama seperti Short Int
Long Int	32	-2.147.483.678 s/d 2.147.483.647
Signed Long Int	32	Sama dengan Long Int
Unsigned Long Int	32	0 s/d 4.294.967.295

TIPE DATA BILANGAN BULAT

```
#include <iostream>

using namespace std;

int main() {
    // Mendeklarasikan variabel Usia dengan tipe data int
    int Usia;

    // Melakukan assignment terhadap variabel Usia
    Usia = 15;
    cout<<"Usia : "<<Usia; cout<<" "; cout<<"Tahun";

    return 0;
}
```

TIPE DATA BILANGAN RIIL

- ***Tipe Data Bilangan Riil*** ini adalah tipe data yang merepresentasikan data – data bilangan yang mengandung angka dibelakang koma misalnya : 4 . 87 atau 15 . 369 dan sebagainya.

TIPE DATA BILANGAN RIIL

Tipe Data	Ukuran (dalam bit)	Rentang	Presisi (dalam digit)
Float	32	1.2E-38 s/d 3.4E+38	6
Double	64	2.3E-308 s/d 1.7E+308	15
Long Double	80	3.4E-4932 s/d 1.1E+4932	19

TIPE DATA BILANGAN RIIL

```
#include <iostream>

using namespace std;

int main() {
    // Mendeklarasikan variabel nilai
    // dengan tipe data double
    double nilai;

    // Melakukan assignment terhadap variabel nilai
    nilai = 486.295;
    cout<<"Nilai Akhir : "<<nilai;

    return 0;
}
```

TIPE DATA BILANGAN LOGIKA

- ***Tipe Data Bilangan Logika*** ini merepresentasikan data – data yang hanya mengandung dua buah nilai, yaitu ***nilai logika (boolean)***.
- ***Nilai logika*** itu sendiri hanya terdiri dari nilai benar (direpresentasikan dengan selain ***nilai 0***, biasanya ***nilai 1***) dan salah (direpresentasikan dengan ***nilai 0***).

TIPE DATA BILANGAN LOGIKA

- Dalam bahasa C++ standar, tipe ini telah dinyatakan dalam tipe `bool` yang nilainya berupa `true` (benar) atau `false` (salah).

TIPE DATA BILANGAN LOGIKA

```
#include <iostream>

using namespace std;

int main() {
    // Mendeklarasikan variabel a dan b
    // dengan tipe data bool
    bool benar;

    // Meminta input dari user
    int a, b;
    cout<<"Masukkan nilai a : "; cin>>a;
    cout<<"Masukkan nilai b : "; cin>>b;

    // Melakukan assignment terhadap variabel benar
    benar = a > b;

    if (benar == true) { // bisa ditulis: if (benar) {
        cout<<"a lebih besar dari b";
    } else {
        cout<<"a lebih kecil dari atau sama dengan b";
    }

    return 0;
}
```

TIPE DATA KARAKTER

- ***Tipe Data Karakter*** ini digunakan untuk merepresentasikan data bertipe ***karakter***, seperti : 'B', 'd', '8', '@', dan sebagainya
- Dalam bahasa C++, ***tipe data karakter*** dinyatakan dalam **`char`**
- Untuk ***karakter Unicode*** (Wide Character) dinyatakan dalam **`wchar_t`**

TIPE DATA KARAKTER

Tipe Data	Ukuran (Dalam Bit)	Rentang
Char	8	-128 s/d 127 atau 0 s/d 255
Unsigned Char	8	0 s/d 255
Signed Char	8	-128 s/d 127
Wchar_t	16 atau 32	0 s/d 65.535

TIPE DATA KARAKTER

- Dalam bahasa C++, *tipe karakter* selalu diapit oleh petik tunggal.
- Satu karakter yang diapit oleh tanda *petik ganda* akan dianggap sebagai **string**, sebagai contoh 'A' adalah karakter sedangkan "A" adalah **string** yang terdiri dari satu karakter
- String akan dibahas kemudian

TIPE DATA KARAKTER

```
#include <iostream>

using namespace std;

int main() {
    // Mendeklarasikan variabel Karakter
    // dan mengisinya dengan nilai 'Z'
    char Karakter = 'Z';

    // Mendeklarasikan variabel Teks
    // dan mengisinya dengan nilai "Sistem"
    char* Teks = (char*) "Sistem";
    char TEKS[10] = "Komputer";
    cout<<Karakter<<endl;
    cout<<Teks<<endl;
    cout<<TEKS<<endl;

    return 0;
}
```

TIPE DATA BENTUKAN

- ***Tipe Data Bentukan*** adalah **tipe data** yang dibuat sendiri sesuai kebutuhan dalam program yang akan dibuat.
- Tipe ini lebih dikenal dengan sebutan **user define type**.

TIPE DATA BENTUKAN

- Adapun yang termasuk kedalam *tipe data bentukan* adalah 1. **Struktur**, 2. **Array (larik)**, dan 3. **Enumerasi** termasuk juga 4. **String**.
- Sedangkan kalau dalam bahasa pascal maka struktur seperti ini sering dikenal dengan istilah **rekaman (record)**.

I. TIPE DATA STRUKTUR

- Sesuai dengan namanya, “**struktur**” adalah tipe data bentukan yang menyimpan lebih dari satu variabel bertipe sama maupun berbeda.
- Untuk membuat tipe data **struktur** dalam C++, maka harus dideklarasikan dengan menggunakan keyword **struct**.

I. TIPE DATA STRUKTUR

Bentuk umum pendeklarasian tipe data *struktur* didalam C++ adalah :

```
Struct nama_struktur {  
    tipe_data variabel 1;  
    tipe_data variabel 2;  
    ...  
};
```

I. TIPE DATA STRUKTUR

- Dalam pendeklarasian ***struktur*** harus mengakhirinya dengan tanda titik koma atau semicolon (;)
- Apabila telah membuat tipe data bentukan baru yang berjenis ***struktur*** tentunya akan digunakan didalam *pendeklarasian sebuah variabel.*

I. TIPE DATA STRUKTUR

- Pada saat itu maka, harus dilakukan pengaksesan terhadap isi dari **struktur** tersebut dengan menggunakan operator titik(.)
- Sebagai contoh sederhana untuk menerangkan tipe data **struktur** adalah dengan melihat suatu objek, misalnya mahasiswa.

I. TIPE DATA STRUKTUR

- Objek mahasiswa pasti mempunyai **NIM** (*nomor induk mahasiswa*), **Nama**, **Alamat** serta data pribadi lainnya.
- Kasus ini dapat diimplementasikan kedalam tipe data ***struktur***.

I. TIPE DATA STRUKTUR

```
#include <iostream>
#include <cstring>
using namespace std;

int main() {
    struct MAHASISWA {
        char NIM[10];
        char Nama[25];
        char Alamat[25];
        char Kota[25];
    };

    MAHASISWA A;          // Mendeklarasikan variabel A
                          // yang bertipe MAHASISWA

    strcpy(A.NIM, "123456789");
    strcpy(A.Nama, "Budhi Irawan");
    strcpy(A.Alat, "Buah Batu");
    strcpy(A.Kota, "Bandung");

    // Menampilkan nilai yang diisikan ke layar
    cout<<A.NIM<<endl;
    cout<<A.Nama<<endl;
    cout<<A.Alat<<endl;
    cout<<A.Kota<<endl;
    return 0;
}
```

2. TIPE DATA ARRAY

- *Tipe Data Array* ini digunakan untuk merepresentasikan kumpulan data sejenis atau data yang bertipe sama.
- Banyaknya data ditentukan dengan menyebutkan nilai didalam tanda **bracket** (**[]**)
- Contoh :

```
int nomorhari [7] ;  
char nama [25] ;
```

2. TIPE DATA ARRAY

- Pada contoh diatas tadi dideklarasikan variabel **nomorhari** bertipe **array** dengan jumlah elemen sebanyak **7** buah
- Ini berarti bahwa variabel tersebut mampu menampung **7** buah nilai yang masing-masing bertipe **integer**.

2. TIPE DATA ARRAY

- Begitu juga dengan **nama**, variabel tersebut dapat diisi oleh **25** karakter ***Array*** yang akan dibahas terpisah selanjutnya.

3. TIPE DATA ENUMERASI

- ***Tipe Data Enumerasi*** ini adalah tipe data yang nilainya terbatas pada nilai-nilai yang telah didefinisikan saja.
- ***Tipe enumerasi*** digunakan untuk membentuk tipe data yang nilainya bersifat pasti, misalnya untuk mendefinisikan tipe **jenis kelamin, nama hari, warna primer**, dan sebagainya

3. TIPE DATA ENUMERASI

- Contoh dari untuk jenis kelamin dapat kita bentuk kedalam tipe ***enumerasi***.
- Dalam bahasa C++, untuk mendefinisikan tipe ***enumerasi*** adalah dengan menggunakan keyword **`enum`**.

3. TIPE DATA ENUMERASI

- Bentuk umum untuk mendefinisikan tipe data *enumerasi* adalah :

```
enum nama_tipe {nilai_1, nilai_2, ...};
```

- **Contoh :**

```
enum JENIS_KELAMIN {Pria, Wanita};
```

```
enum HARI {Senin, Selasa, Rabu, Kamis,  
          Jumat, Sabtu, Minggu};
```

```
enum WARNA_PRIMER {Merah, Biru, Hijau};
```

3. TIPE DATA ENUMERASI

```
#include <iostream>
#include <cstring>
using namespace std;

// Mendefinisikan tipe JENIS_KELAMIN
enum JENIS_KELAMIN {Pria, Wanita};

int main() {

    struct MAHASISWA {
        char NIM[10];
        char Nama[25];
        JENIS_KELAMIN gender;
    } A;

    strcpy(A.NIM, "123456789");
    strcpy(A.Nama, "Budhi Irawan");
    A.gender = Pria;

    cout<<"NIM : "<<A.NIM<<endl;
    cout<<"Nama : "<<A.Nama<<endl;
    cout<<"Gender : "<<A.gender<<endl;

    return 0;
}
```

4. TIPE DATA STRING

- **Tipe Data String** ini digunakan untuk merepresentasikan data yang berupa teks (*kumpulan karakter*).
- Contoh data string adalah “**komputer**”, “**C++**” dan sebagainya.
- Bahasa C++ mendukung dua tipe string yaitu string gaya bahasa C dan string gaya bahasa C++.

4. TIPE DATA STRING

- Dalam bahasa C, **string** merupakan array dari tipe karakter yang diakhiri oleh karakter **null** (`'\0'`)
- Karena dalam bahasa C, **array** bisa diakses melalui **pointer**, maka dapat juga dideklarasikan **string** sebagai **pointer** ke tipe karakter

4. TIPE DATA STRING

Contoh

```
Char namadepan[25]; // string yang berupa array  
Char *namabelakang; // string yang berupa pointer
```

- Untuk mengisi data kedalam string yang berupa **array** maka dapat digunakan fungsi **strcpy()** seperti kode berikut :

```
strcpy (namadepan, "Budhi");
```

4. TIPE DATA STRING

- Untuk **string** yang berupa **pointer** dapat digunakan operator sama dengan (=) seperti contoh berikut :

```
namabelakang = (char *) "Budhi Irawan";
```

- **(char *)** merupakan kode yang digunakan untuk melakukan **typecasting** (pemeranan tipe data) terhadap teks yang akan diisikan
- Dalam contoh ini, **(char *)** berfungsi untuk menganggap bahwa teks "Budhi Irawan" sebagai tipe **char**

4. TIPE DATA STRING

```
#include <iostream>
#include <cstring>

using namespace std;

int main() {

    char namadepan[25];
    char *namabelakang;

    strcpy(namadepan, "Budhi");
    namabelakang = (char *) "Irawan Pratama";

    cout<<"Nama lengkap : "<<namadepan
        <<" "
        <<namabelakang;

    return 0;
}
```