

# **LAPORAN PRAKTIKUM**

## **MODUL I TIPE DATA**



**Disusun oleh:  
Brian Farrel Evandhika  
NIM: 2311102037**

**Dosen Pengampu:**  
Wahyu Andi Saputra, S.Pd., M.Eng.

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS INFORMATIKA  
INSTITUT TEKNOLOGI TELKOM PURWOKERTO  
PURWOKERTO  
2023**

# **BAB I**

## **TUJUAN PRAKTIKUM**

### **1. Memahami dan Menganalisis Tipe Data**

- **Tipe Data Primitif:** Memahami konsep, jenis, dan operasi pada tipe data primitif seperti integer, floating-point, karakter, dan boolean.
- **Tipe Data Abstrak:** Memahami konsep dan implementasi tipe data abstrak seperti string, date, dan time.
- **Tipe Data Kolektif:** Memahami konsep, jenis, operasi, dan kompleksitas algoritma pada struktur data kolektif seperti array, linked list, stack, queue, tree, dan hash table.

### **2. Mampu Memilih dan Mengaplikasikan Tools yang Tepat**

- Mampu memilih tools yang sesuai dengan kebutuhan untuk memanipulasi struktur data.
- Memahami cara kerja dan fitur-fitur utama dari tools yang dipilih.
- Mampu menggunakan tools dengan efektif dan efisien untuk menyelesaikan tugas terkait struktur data.

### **3. Mampu Mengimplementasikan Tipe Data dan Struktur Data dalam Bahasa Pemrograman**

- Mampu memilih dan mengimplementasikan struktur data yang tepat untuk menyelesaikan masalah pemrograman.
- Mampu menulis kode yang efisien dan mudah dibaca untuk memanipulasi struktur data.
- Mampu menganalisis kompleksitas waktu dan memori dari kode yang diimplementasikan.

### **4. Mengembangkan Kemampuan Berpikir Komputasi dan Pemecahan Masalah**

- Mampu menganalisis dan memahami masalah pemrograman yang melibatkan struktur data.
- Mampu merancang solusi yang efektif dan efisien dengan menggunakan struktur data yang tepat.
- Mampu menguji dan mengevaluasi solusi yang telah dirancang.
- Mampu mengembangkan kemampuan berpikir kritis dan logis dalam menyelesaikan masalah pemrograman.

## **BAB II**

### **DASAR TEORI**

Tipe data adalah adalah sebuah pengklasifikasian data berdasarkan jenis data tersebut. Tipe data dibutuhkan agar kompiler dapat mengetahui bagaimana sebuah data akan digunakan. Adapun tipe data yang akan dipelajari, sebagai berikut :

1. Tipe data Primitif
2. Tipe data Abstrak
3. Tipe data Koleksi

#### **Tipe data Primitif**

Tipe data primitif adalah tipe data yang sudah ditentukan oleh sistem, tipe data primitif ini disediakan oleh banyak bahasa pemrograman, perbedaannya terletak pada jumlah bit yang dialokasikan untuk setiap bit pada tipe data primitif tergantung pada bahasa pemrograman, compiler dan sistem operasinya. Contoh tipe data primitif adalah :

- a. Int: adalah tipe data yang digunakan untuk menyimpan bilangan bulat seperti 12, 1, 4, dan sebagainya.
- b. Float : tipe data yang digunakan untuk menyimpan bilangan desimal seperti 1.5, 2.1, 3.14, dan sebagainya.
- c. Char: berfungsi untuk menyimpan data berupa sebuah huruf. Biasanya digunakan untuk simbol seperti A, B, C dan seterusnya
- d. Boolean: tipe data ini digunakan untuk menyimpan nilai boolean yang hanya memiliki dua nilai yaitu true dan false.

#### **Tipe Data Abstrak**

Tipe data abstrak atau yang biasa disebut Abstrak Data Tipe (ADT) merupakan tipe data yang dibentuk oleh programmer itu sendiri. Pada tipe data abstrak bisa berisi banyak tipe data, jadi nilainya bisa lebih dari satu dan beragam tipe data. Fitur Class adalah fitur Object Oriented Program (OPP) pada bahasa C++ yang mirip dengan fitur data structures Struct pada bahasa C. Keduanya berfungsi untuk membungkus tipe data di dalamnya sebagai anggota. menurut [learn.microsoft.com](https://learn.microsoft.com) perbedaan antara Struct dan Class adalah pada akses defaultnya dimana Struct bersifat public dan Class bersifat private.

#### **Tipe Data Koleksi**

Tipe data koleksi (Collection Data Type) adalah tipe data yang digunakan untuk mengelompokkan dan menyimpan beberapa nilai atau objek secara bersamaan.

Tipe data koleksi memungkinkan Anda menyimpan, mengelola, dan mengakses sejumlah besar data dengan cara yang terstruktur. Ada beberapa tipe data koleksi yang umum digunakan dalam pemrograman, dan di antaranya adalah:

a. Array: Array adalah struktur data statis yang menyimpan elemen-elemen dengan tipe data yang sama. Elemen-elemen tersebut dapat diakses dengan menggunakan indeks. Array memiliki ukuran tetap yang ditentukan saat deklarasi.

b. Vector: Vector adalah Standard Template Library (STL) jika di dalam C/C++ memiliki bentuk `std::vector`. Umumnya, vector mirip seperti array yang memiliki kemampuan untuk menyimpan data dalam bentuk elemen-elemen yang alokasi memorinya dilakukan otomatis dan bersebelahan. Kemampuan vector bukan hanya pada jumlah elemen yang dinamis, vector pada C/C++ juga dilengkapi dengan fitur-fitur pelengkap seperti element access, iterators, capacity, modifiers

c. Map: Map terasa mirip dengan array namun dengan index yang memungkinkan untuk berupa tipe data selain integer. Pada map, indeks tersebut diberi nama "key". Pada `std::map` digunakan Self-Balancing Tree khususnya Red-Black Tree.

## BAB III

### GUIDED

#### 1. Guided 1 : Tipe Data Primitif

##### Source Code

```
#include <iostream>
using namespace std;
// Main program
int main()
{
    char op;
    float num1, num2;
    // It allows user to enter operator i.e. +, -, *, /
    cin >> op;
    // It allow user to enter the operands
    cin >> num1 >> num2;
    // Switch statement begins
    Praktikum Struktur Data dan Algoritma 3
    switch (op)
    {
        // If user enter +
        case '+':
            cout << num1 + num2;
            break;
        // If user enter -
        case '-':
            cout << num1 - num2;
            break;
        // If user enter *
        case '*':
            cout << num1 * num2;
            break;
        // If user enter /
        case '/':
            cout << num1 / num2;
            break;
        // If the operator is other than +, -, * or /,
        // error message will display
        default:
```

```
cout << "Error! operator is not correct";  
} // switch statement ends  
return 0;  
}
```

### Screenshoot program

```
Warning: PowerShell detected that you might be using a screen reader and has disabled PSReadLine for compatibility purposes. If you want to re-enable it, run 'Import-Module PSReadLine'.
```

```
PS C:\Users\MSI GAMING> cd "c:\Users\MSI GAMING\Downloads\New folder\" ; if ($?) { g++ Untitled-2.cpp -o Untitled-2 } ; if ($?) { .\Untitled-2 }
```

```
+  
1  
2  
3
```

```
PS C:\Users\MSI GAMING\Downloads\New folder>
```

### Deskripsi program

Program tersebut adalah sebuah kalkulator sederhana yang memungkinkan pengguna untuk melakukan operasi aritmetika dasar (penjumlahan, pengurangan, perkalian, dan pembagian) antara dua bilangan floating-point.

## 2. Guided 2 : Tipe Data Abstrak

### Source Code

```
#include <stdio.h>
//Struct
struct Mahasiswa
{
    const char *name;
    const char *address;
    int age;
};
int main()
{
    // menggunakan struct
    struct Mahasiswa mhs1, mhs2;
    // mengisi nilai ke struct
    mhs1.name = "Dian";
    mhs1.address = "Mataram";
    mhs1.age = 22;
    mhs2.name = "Bambang";
    mhs2.address = "Surabaya";
    mhs2.age = 23;
    // mencetak isi struct
    printf("## Mahasiswa 1 ##\n");
    printf("Nama: %s\n", mhs1.name);
    printf("Alamat: %s\n", mhs1.address);
    printf("Umur: %d\n", mhs1.age);
    printf("## Mahasiswa 2 ##\n");
    printf("Nama: %s\n", mhs2.name);
    printf("Alamat: %s\n", mhs2.address);
    printf("Umur: %d\n", mhs2.age);
    return 0;
}
```

### Screenshoot program

```
PS C:\Users\MSI GAMING> cd "c:\Users\MSI GAMING\Downloads\New folder\" ; if ($?) { g++ Untitled-2.cpp -o Untitled-2 } ; if ($?) { .\Untitled-2 }  
## Mahasiswa 1 ##  
Nama: Dian  
Alamat: Mataram  
Umur: 22  
## Mahasiswa 2 ##  
Nama: Bambang  
Alamat: Surabaya  
Umur: 23  
PS C:\Users\MSI GAMING\Downloads\New folder>
```

### Deskripsi program

Program ini adalah program sederhana yang menggunakan struktur (struct) dalam bahasa pemrograman C untuk menyimpan informasi tentang dua mahasiswa. Program ini menciptakan dua variabel struktur **mhs1** dan **mhs2**, masing-masing merepresentasikan data mahasiswa.



### 3. Guided 3 : Tipe Data Koleksi

#### Source Code

```
#include <iostream>
using namespace std;
int main()
{
    //deklarasi dan inisialisasi array
    int nilai[5];
    nilai[0] = 23;
    nilai[1] = 50;
    nilai[2] = 34;
    nilai[3] = 78;
    nilai[4] = 90;
    //mencetak array
    cout << "Isi array pertama :" << nilai[0] << endl;
    cout << "Isi array kedua :" << nilai[1] << endl;
    cout << "Isi array ketiga :" << nilai[2] << endl;
    cout << "Isi array keempat :" << nilai[3] << endl;
    cout << "Isi array kelima :" << nilai[4] << endl;
    return 0;
}
```

#### Screenshoot program

```
PS C:\Users\MSI GAMING> cd "c:\Users\MSI GAMING\Downloads\New folder\" ; if ($?) { g++ Untitled-2.cpp -o Untitled-2 } ; if ($?) { .\Untitled-2 }
Isi array pertama :23
Isi array kedua :50
Isi array ketiga :34
Isi array keempat :78
Isi array kelima :90
PS C:\Users\MSI GAMING\Downloads\New folder>
```

#### Deskripsi program

Program ini adalah program sederhana yang menggunakan struktur (struct) dalam bahasa pemrograman C untuk menyimpan informasi tentang dua mahasiswa. Program ini menciptakan dua variabel struktur **mhs1** dan **mhs2**, masing-masing merepresentasikan data mahasiswa.

## LATIHAN KELAS - UNGUIDED

### 1. Unguided 1

#### Source code

```
#include <iostream>

using namespace std;

int hitungLuas(int panjang, int lebar) {
    int luas = panjang * lebar;
    return luas;
}

int main() {
    int panjang, lebar;

    cout << "Masukkan panjang persegi panjang: ";
    cin >> panjang;
    cout << "Masukkan lebar persegi panjang: ";
    cin >> lebar;

    int luas = hitungLuas(panjang, lebar);

    cout << "Luas persegi panjang adalah: " << luas << endl;

    return 0;
}
```

## Screenshoot program

```
PS C:\Users\MSI GAMING> cd "c:\Users\MSI GAMING\Downloads\New folder\" ; if ($?) { g++ Untitled-2.cpp -o Untitled-2 } ; if ($?) { .\Untitled-2 }
Masukkan panjang persegi panjang: 2
Masukkan lebar persegi panjang: 2
Luas persegi panjang adalah: 4
PS C:\Users\MSI GAMING\Downloads\New folder>
```

## Deskripsi program

Kode tersebut adalah program sederhana yang menggunakan tipe data primitive untuk menghitung luas persegi panjang berdasarkan panjang dan lebar yang dimasukkan oleh pengguna.

## 2. Unguided 2

```
#include <iostream>

using namespace std;

class Persegi {
public:
    double sisi;

    Persegi(double s) : sisi(s) {}

    double luas() {
        return sisi * sisi;
    }
};

class Kotak {
private:
    double panjang;
    double lebar;
    double tinggi;
```

```

public:
    Kotak(double p, double l, double t) : panjang(p), lebar(l),
    tinggi(t) {}

    double volume() {
        return panjang * lebar * tinggi;
    }
};

int main() {

    Persegi persegi(2);
    cout << "Luas persegi: " << persegi.luas() << endl;

    Kotak kotak(1, 2, 3);
    cout << "Volume kotak berbentuk balok: " << kotak.volume() <<
endl;

    return 0;
}

```

### Screenshoot program

```

PS C:\Users\MSI GAMING> cd "c:\Users\MSI GAMING\Downloads\New folder\" ; if ($?) { g++ Untitled-2.cpp -o Untitled-2 } ; if ($?) { .\Untitled-2 }
Luas persegi: 4
Volume kotak berbentuk balok: 6
PS C:\Users\MSI GAMING\Downloads\New folder>

```

### Deskripsi Program

Pada dasarnya dalam bahasa C++ struct dan class mempunyai pengertian yang sama. Deklarasi class dengan struct mempunyai anggota dengan akses public, Namun mempunyai beberapa perbedaan mari kita lihat defenisinya dulu

#### Defenisi Struct

Struct atau structure merupakan tipe data yang menyimpan beberapa data dengan default public, Struktur bermanfaat untuk mengelompokkan sejumlah data dengan tipe yang berlainan.

#### Defenisi Class

Class merupakan tipe data defaultnya itu private, jadi jika objek anggota dari class itu sendiri ingin dipublikasikan, maka harus menggunakan public

### Perbedaan Class dan Struct

Class adalah sesuatu yang mempunyai data (sifat) dan fungsi (kelakuan) untuk mendeskripsikan suatu objek.

Struct mendefinisikan tipe data dan memberitahukan kepada Visual C++ untuk memperlakukan koleksi tipe data dasar sebagai satu kesatuan tipe data, yang darinya dapat mendefinisikan variabel dan array variabel.

### 3. Unguided 3

```
#include <iostream>
#include <map>

using namespace std;

int main() {
    // Deklarasi array
    int array[] = {1, 2, 3, 4, 5};

    // Deklarasi map
    map<string, int> ages = {{"Alice", 30}, {"Bob", 25}, {"Charlie", 35}};

    // Menggunakan array
    for (int i = 0; i < sizeof(array) / sizeof(array[0]); i++)
    {
        cout << "Array element " << i << ": " << array[i] << endl;
    }

    // Menggunakan map
    for (auto const& age : ages) {
        cout << "Name: " << age.first << ", Age: " << age.second << endl;
    }

    return 0;
}
```

**Screenshoot program**

```
PS C:\Users\MSI GAMING> cd "c:\Users\MSI GAMING\Downloads\New folder\" ; if ($?) { g++ Untitled-2.cpp -o Untitled-2 } ; if ($?) { .\Untitled-2 }
Array element 0: 1
Array element 1: 2
Array element 2: 3
Array element 3: 4
Array element 4: 5
Name: Alice, Age: 30
Name: Bob, Age: 25
Name: Charlie, Age: 35
PS C:\Users\MSI GAMING\Downloads\New folder>
```

### Deskripsi program

Array: Merupakan struktur data yang menyimpan elemen-elemen bertipe sama dalam urutan tertentu. Elemen-elemen dapat diakses menggunakan indeks numerik yang dimulai dari 0.

Map: Merupakan struktur data asosiatif yang menyimpan pasangan kunci-nilai. Kunci dan nilai dapat memiliki tipe data yang berbeda, dan kunci digunakan untuk mengakses nilai terkait.

## **BAB IV**

### **KESIMPULAN**

Tipe data C++ adalah elemen penting dalam pemrograman yang menentukan:

Jenis nilai yang dapat disimpan variabel.

Operasi yang dapat dilakukan pada nilai tersebut.

Memahami tipe data dengan baik membantu programmer dalam:

Menulis program yang lebih efisien: Memilih tipe data yang tepat dapat mengoptimalkan penggunaan memori dan waktu proses.

Meningkatkan akurasi: Tipe data yang tepat membantu menghindari kesalahan dan menghasilkan hasil yang diharapkan.

Mempermudah pembacaan kode: Penggunaan tipe data yang jelas membuat kode lebih mudah dipahami dan dipelihara.

Beberapa poin penting tentang tipe data C++:

Beragam tipe data: C++ menyediakan berbagai tipe data bawaan untuk menyimpan data numerik, karakter, boolean, dan lainnya.

Tipe data primitif dan kompleks: Tipe data primitif adalah tipe data dasar, sedangkan tipe data kompleks tersusun dari beberapa tipe data primitif.

Konversi tipe data: Konversi antara tipe data dapat dilakukan dengan menggunakan operator atau fungsi konversi.

Memahami dan menggunakan tipe data dengan tepat adalah kunci untuk menulis program C++ yang efektif dan efisien.

## **DAFTAR PUSTAKA**

Asisten Praktikum, "Modul 1 Tipe Data", Learning Management System, 2024