

# **LAPORAN PRAKTIKUM**

## **MODUL I TIPE DATA**



**Disusun oleh:**  
Ahmadan Syaridin  
2311102038

**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. Dapat Mempelajari tipe data primitif, abstrak , dan kolektif.
2. Dapat memahami pengaplikasian pada tools yang digunakan.
3. Dapat mengaplikasikan berbagai tipe data pada bahasa pemograman.

## **BAB II**

### **DASAR TEORI**

Tipe data adalah sebuah pengklasifikasian data berdasarkan jenis data tersebut. Tipe data dibutuhkan agar kompiuter dapat mengetahui bagaimana sebuah data akan digunakan.

#### **Tipe data Primitif**

Tipe data primitif adalah tipe data yang sudah ditentukan oleh sistem, tipe data primitif ini disediakan oleh banyak pemrograman, perbedaannya terletak pada bahasa pemrograman compiler dan sistem operasinya.

- a. Int : adalah tipe data yang digunakan untuk menyimpan bilangan bulat seperti 12, 1, 4 dan lainnya.
- b. Float : tipe data yang digunakan untuk menyimpan bilangan desimal seperti 1.5, 2.1, 3.14, dan lainnya.
- c. Char : berfungsi untuk menyimpan data berupa sebuah huruf. Biasanya digunakan untuk symbol seperti A, B, C dan seterusnya.
- d. Boolean : tipe data 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 ini 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](http://learn.microsoft.com) perbedaan struct dan class pada akses defaultnya Dimana struct bersifat public dan class bersifat private.

## BAB III

### GUIDED

#### 1. Guided 1

##### Source code

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

int main()
{
    char op;
    float num1, num2;

    cin >> op;

    cin >> num1 >> num2;

    switch (op)
    {
        case '+':
            cout << num1 + num2;
            break;

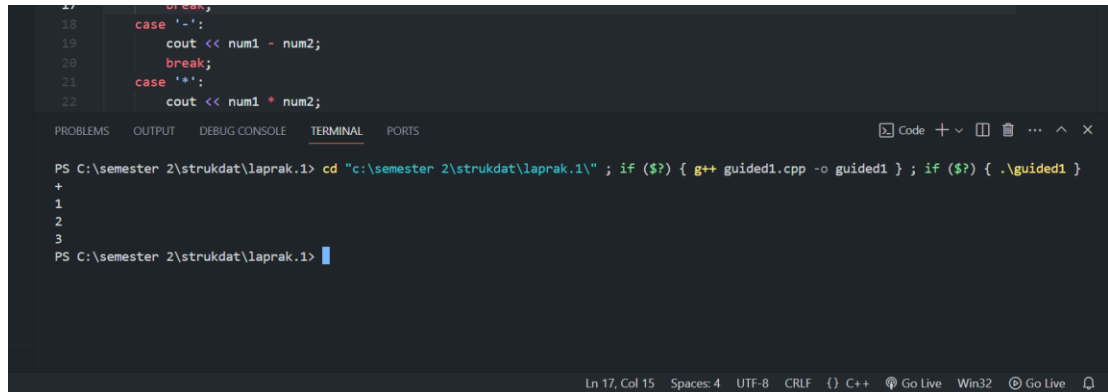
        case '-':
            cout << num1 - num2;
            break;

        case '*':
            cout << num1 * num2;
            break;

        case '/':
            cout << num1 / num2;
            break;

        default:
            cout << " Error! operator is not correct";
    }
    return 0;
}
```

## Screenshoot



```
27         break;
28     case '-':
29         cout << num1 - num2;
30         break;
31     case '+':
32         cout << num1 + num2;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\semester 2\strukdat\laprak.1> cd "c:\semester 2\strukdat\laprak.1" ; if ($?) { g++ guided1.cpp -o guided1 } ; if ($?) { .\guided1 }
+
1
2
3
PS C:\semester 2\strukdat\laprak.1>
```

Ln 17, Col 15 Spaces: 4 UTF-8 CRLF {} C++ Go Live Win32 Go Live

## Deskripsi program

Code program di atas merupakan kalkulator sederhana yang meminta pengguna untuk memasukan pilihan matematika(+, -, \*, /) dan dua angka untuk di oprasikan. Kemudian switch mengecek operator yang di masukan pengguna dan Melakukan operasi matematika yang sesuai.

## 2. Guided 2

### Source code

```
#include <stdio.h>

struct mahasiswa
{
    const char *name;
    const char *address;
    int age;
};

int main()
{
    struct mahasiswa mhs1, mhs2;

    mhs1.name = "Adan";
    mhs1.address = "Cirebon";
    mhs1.age = 18;
    mhs2.name = "Ahmadan";
    mhs2.address = "Purwokerto";
    mhs2.age = 18;
```

```

printf("##mahasiswa 1 ##\n");
printf("nama : %s\n", mhs1.name);
printf("alamat : %s\n", mhs1.address);
printf("nama : %d\n", mhs1.age);
printf("##mahasiswa 2 ##\n");
printf("nama : %s\n", mhs2.name);
printf("alamat : %s\n", mhs2.address);
printf("nama : %d\n", mhs2.age);

return 0;
}

```

## Screenshot

```

18 mhs2.address = "Purwokerto";
19 mhs2.age = 18;
20
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\semester 2\strukdat\laprak.1> cd "c:\semester 2\strukdat\laprak.1"; if ($?) { g++ guided2.cpp -o guided2 }; if ($?) { .\guided2 }
##mahasiswa 1 ##
nama : Adan
alamat : Cirebon
nama : 18
##mahasiswa 2 ##
nama : Ahmadan
alamat : Purwokerto
nama : 18
PS C:\semester 2\strukdat\laprak.1>

```

## Deskripsi

Program penggunaan struct untuk menyimpan informasi tentang mahasiswa. Struct “mahasiswa” digunakan untuk menyimpan nama, Alamat, dan umur. Fungsi main() dua variable ‘mhs1, mhs2’ bertipe ‘mahasiswa’ dideklarasikan menyimpan informasi tentang dua mahasiswa berbeda.

## 3. Guided 3

### Source code

```

using namespace std;
int main()
{
    int nilai[5];

```

```

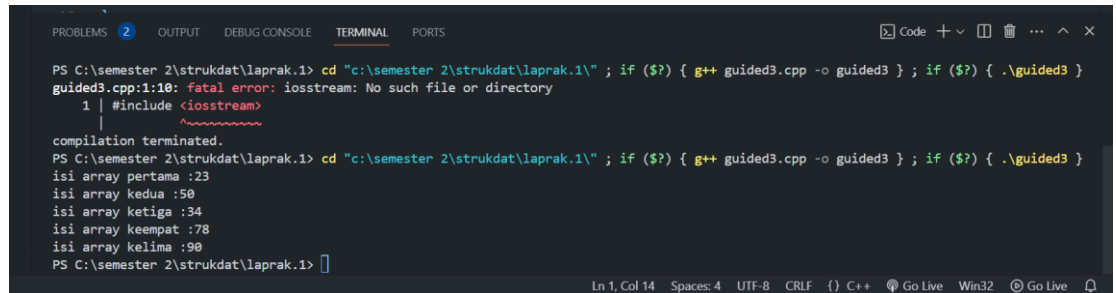
    nilai[0] = 23;
    nilai[1] = 50;
    nilai[2] = 34;
    nilai[3] = 78;
    nilai[4] = 90;

    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;
}

```

## Screenshot



```

PS C:\semester 2\strukdat\laprak.1> cd "c:\semester 2\strukdat\laprak.1\" ; if ($?) { g++ guided3.cpp -o guided3 } ; if ($?) { .\guided3 }
guided3.cpp:1:10: fatal error: iosstream: No such file or directory
   1 | #include <iosstream>
     |          ^~~~~~
compilation terminated.
PS C:\semester 2\strukdat\laprak.1> cd "c:\semester 2\strukdat\laprak.1\" ; if ($?) { g++ guided3.cpp -o guided3 } ; if ($?) { .\guided3 }
isi array pertama :23
isi array kedua :50
isi array ketiga :34
isi array keempat :78
isi array kelima :90
PS C:\semester 2\strukdat\laprak.1>

```

## Deskripsi

Program menggunakan array untuk menyimpan dan mencetak nilai – nilai interger. Array ‘nilai’ dengan Panjang 5 dinisialisasi oleh beberapa nilai interger. Setelah itu, mencetak isi array ke layer menggunakan ‘cout’

## LATIHAN KELAS - UNGUIDED

### 1. Unguided 1

#### Source code

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

float tambah(float a, float b) {
    return a + b;
}

float kurang(float a, float b) {
    return a - b;
}

float kali(float a, float b) {
    return a * b;
}

float bagi(float a, float b) {
    return a / b;
}

int main() {
    char op;
    float num1, num2;

    cout << "Masukkan operator (+, -, *, /): ";
    cin >> op;

    cout << "Masukkan dua bilangan: ";
    cin >> num1 >> num2;

    switch (op) {
        case '+':
            cout << "Hasil penjumlahan: " << tambah(num1,
num2);
            break;
        case '-':
```



```

        cout << "Hasil pengurangan: " << kurang(num1,
num2) ;
        break;
    case '*':
        cout << "Hasil perkalian: " << kali(num1, num2);
        break;
    case '/':
        if (num2 != 0) {
            cout << "Hasil pembagian: " << bagi(num1,
num2) ;
        } else {
            cout << "Error! Pembagian dengan nol tidak bisa
dilakukan.";
        }
        break;
    default:
        cout << "Error! Operator tidak valid.";
    }

    return 0;
}

```

### Screenshoot program

```

27
28      cout << "Masukkan operator (+, -, *, /): ";
...
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\semester 2\strukdat\laprak.1> cd "c:\semester 2\strukdat\laprak.1\" ; if ($?) { g++ unguided1.cpp -o unguided1 } ; if ($?) { .\unguid
ed1 }
Masukkan operator (+, -, *, /): -
Masukkan dua bilangan: 1000 78,9
Hasil pengurangan: 922
PS C:\semester 2\strukdat\laprak.1>

```

### Deskripsi program

Program meminta pengguna untuk memasukkan operator matematika dan dua bilangan. Switch untuk operasi yang di hasilkan operator,hasil operasi kemudian

di tampilkan ke layar. Kesimpulannya tipe data primitive dapat digunakan untuk menyimpan data sederhana seperti bilangan bulat, bilangan riil, dan karakter. Tipe data primitif 'char' untuk menyimpan operator matematika, dan 'float' untuk menyimpan bilangan. Dengan menggunakan tipe data primitive program dapat melakukan operasi matematika secara efisien dan dengan menggunakan sedikit memori.

## 2. Unguided 2

Source code "Srutct"

```
3. #include <stdio.h>
4.
5. struct mahasiswa
6. {
7.     const char *name;
8.     const char *address;
9.     int age;
10.};
11.
12.int main()
13.{
14.    struct mahasiswa mhs1, mhs2;
15.
16.    mhs1.name = "Adan";
17.    mhs1.address = "Cirebon";
18.    mhs1.age = 18;
19.    mhs2.name = "Ahmadan";
20.    mhs2.address = "Purwokerto";
21.    mhs2.age = 18;
22.
23.    printf("##mahasiswa 1 ##\n");
24.    printf("nama : %s\n", mhs1.name);
25.    printf("alamat : %s\n", mhs1.address);
26.    printf("nama : %d\n", mhs1.age);
27.    printf("##mahasiswa 2 ##\n");
28.    printf("nama : %s\n", mhs2.name);
29.    printf("alamat : %s\n", mhs2.address);
30.    printf("nama : %d\n", mhs2.age);
31.
```

```

32.     return 0;
33.
34.}

```

## Screenshot



```

18     mhs2.address = "Purwokerto";
19     mhs2.age = 18;
20
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\semester 2\strukdat\laprak.1> cd "c:\semester 2\strukdat\laprak.1\"; if ($?) { g++ guided2.cpp -o guided2 }; if ($?) { .\guided2 }
##mahasiswa 1 ##
nama : Adan
alamat : Cirebon
nama : 18
##mahasiswa 2 ##
nama : Ahmadan
alamat : Purwokerto
nama : 18
PS C:\semester 2\strukdat\laprak.1>

```

## Deskripsi

Program merupakan program sederhana yang menggunakan struktur struct untuk merepresntasikan informasik mengenai mahasiswa,struktur struct tersebut Bernama ‘mahasiswa’ dan memiliki dua anggota data : name untuk menyimpan nama mahasiswa, adres untuk menyimpan Alamat mahasiswa, dan age untuk menyimpan usia mahasiswa.dalam fungsi utama (main),dua variable dari tipe data struct mahasiswa di deklarasikan, yaitu mhs1 dan mhs2, yang masing -masing diinisialisasi dengan data nama,Alamat,dan usia mahasiswa.selanjutnya program menampilkan informasi masing-masing mahasiswa menggunakan perintah cout. Fungsi utama dari struct adalah untul mengorganisir dan mengelompokam data secara terstrujtur,sehingga dapat membentuk satu entitas yang leboh kompleks,jadi fungsi penggunaan struct pada program tersebut untuk menciptakan struktur data yang lebih memudahkan repesentasi informasi mahasiswa.

## Source “Class”

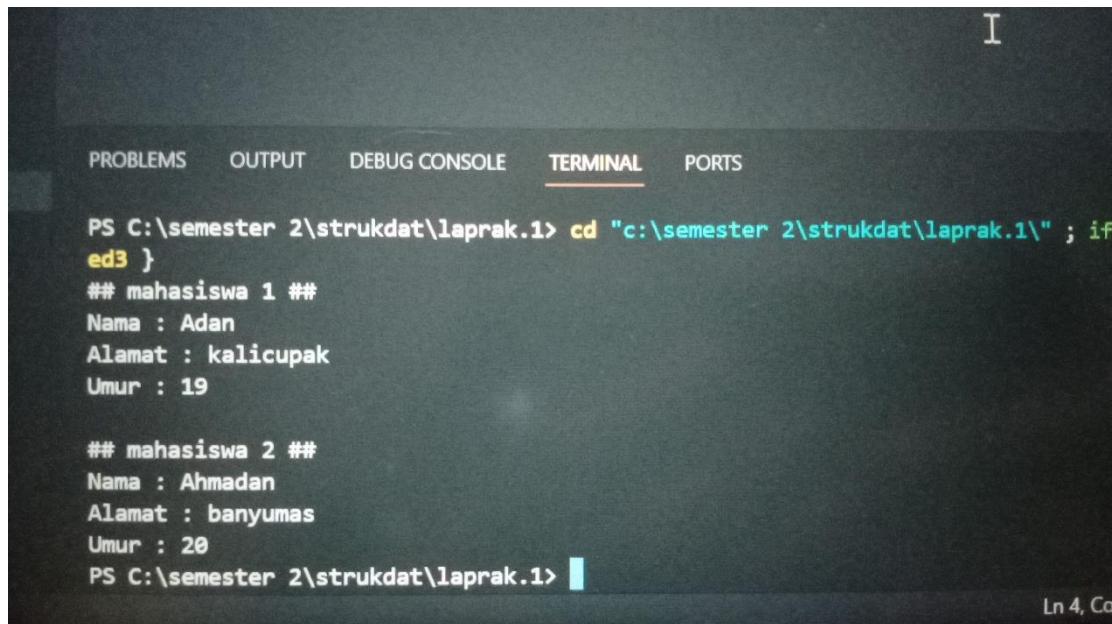
```

#include <iostream>
#include <string>
using namespace std;
class mahasiswa {
public:

```

```
string name;
string address;
int age;
// Constructor untuk inisialisasi objek mahasiswa
mahasiswa (string n, string addr, int a) : name(n), address(addr),
age(a)
{}
// Metode untuk menampilkan informasi mahasiswa
void displayInfo() {
cout << "Nama : " << name << endl;
cout << "Alamat : " << address << endl;
cout << "Umur : " << age << endl;
}
};
int main() {
// Membuat objek mahasiswa menggunakan constructor
mahasiswa mhs1("Adan", "kalicupak", 19);
mahasiswa mhs2("Ahmadan", "banyumas", 20);
cout << "## mahasiswa 1 ##" << endl;
mhs1.displayInfo();
cout << "\n## mahasiswa 2 ##" << endl;
mhs2.displayInfo();
return 0;
}
```

## Screenshot



```
PS C:\semester 2\strukdat\laprak.1> cd "c:\semester 2\strukdat\laprak.1\" ; if
ed3 }
## mahasiswa 1 ##
Nama : Adan
Alamat : kalicupak
Umur : 19

## mahasiswa 2 ##
Nama : Ahmadan
Alamat : banyumas
Umur : 20
PS C:\semester 2\strukdat\laprak.1>
```

## Deskripsi

Program ini mendefinisikan sebuah class Bernama 'mahasiswa' yang memiliki tiga atribut data 'name' (nama mahasiswa).address (Alamat mahasiswa) , dan age ( usia mahasiswa). Dalam case mahasiswa terdapat constructor untuk inisialisasi objek mahasiswa dengan nilai nilai awal yang di berikan saat pembuatan objek. Selain itu, terdapat metode "displayInfo()" yang digunakan untuk menampilkan informasi mahasiswa.Dalam fungsi utama (main), dua objek mahasiswa, yaitu "mhs1" dan "mhs2". dibuat dengan menggunakan constructor. Program kemudian menampilkan informasi masing-masing mahasiswa dengan memanggil metode "displayInfo()" pada objek tersebut. Keseluruhan program memberikan contoh penggunaan class dan objek untuk mengorganisir dan mengakses data terkait mahasiswa dengan cara yang lebih terstruktur. Penggunaan class pada program tersebut memberikan kerangka kerja yang jelas dan terstruktur untuk merepresentasikan dan mengelola informasi terkait mahasiswa, memanfaatkan konsep pemrograman berorientasi objek (OOP) untuk meningkatkan keterbacaan, modularitas, dan pemeliharaan program.

### 3. Unguided 3

#### Source code

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

using namespace std;

int main() {
    map<string, int> m;
    m["Adan"] = 20;
    m["Ahmadan"] = 25;
    m["Syaridin"] = 30;

    for (auto it = m.begin(); it != m.end(); ++it) {
        cout << it->first << " -> " << it->second << endl;
    }
    auto it = m.find("Syaridin");
    if (it != m.end()) {
        cout << "Nilai untuk key 'Syaridin': " << it->second << endl;
    } else {
        cout << "Key 'Syaridin' tidak ditemukan" << endl;
    }

    return 0;
}
```

#### Deskripsi program

Dalam program ini, struktur data digunakan untuk menghubungkan nama (string) dengan usia (int) dan beberapa individu. Tiga entri dimasukkan ke Dalam map, dengan “Adan” memiliki usia 20, Ahmadan 25 dan syaridin 30. tentu program Melakukan interaksi melalui map menggunakan loop for, dan untuk setiap entri, menampilkan nama dan usia di layar menggunakan fungsi `it->first` dan `it->second`. Jika kunci tersebut ditemukan, program memberikan pesan bahwa kunci tidak ditemukan.

## **Perbedaan**

Perbedaan dengan array cocok untuk situasi Dimana indeks berurutan dan jumlah elemen diketahui di awal, sementara map lebih fleksibel an berguna ketika kita perlu menghasilkan nilai dengan kata kunci yangd apat bervariasi

## **BAB IV**

### **KESIMPULAN**

Memahami berbagai macam jenis tipe data sangat penting untuk memulai Dalam menulis program C++ yang efektif. Tipe data primitive coco digunakan untuk menyimpan nilai sederhana contohnya ada bool, char, int ,float, double, void. Tipe data abstrak menyembunyikan detail implementasi dan menyediakan operasi yang dapat dilakukan pada data contohnya ada class, tipe data koleksi digunakan untuk menyimpan kumoulan – kumoulan data. Juga menyediakan cara untuk mengorganisir dan mengakses secara efisien. Contohnya ada vector,list,map,array.

### **DAFTAR PUSTAKA**

Karumanchi, N. (2016). *Data Structures and algorithms made easy: Concepts, problems, Interview Questions*. CareerMonk Publications.

TylerMSFT. (n.d.). Collections (C++/CX). diakses dari

<https://learn.microsoft.com/en-us/cpp/cppcx/collections-c-cx?view=msvc-170>