

LAPORAN PRAKTIKUM

MODUL I TIPE DATA



**Disusun oleh:
Reza Alvonzo
NIM: 2311102026**

Dosen Pengampu:
Muhammad Afrizal Amrustian, S. Kom., M. Kom

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
PURWOKERTO
2024**

BAB I

TUJUAN PRAKTIKUM

1. Mahasiswa dapat mempelajari tipe data primitif, abstrak, dan kolektif.
2. Mahasiswa dapat memahami pengaplikasian pada tools yang digunakan.
3. Mahasiswa mengaplikasikan berbagai tipe data pada bahasa pemograman yang telah ditentukan.

BAB II

DASAR TEORI

Tipe data adalah suatu konsep pemograman yang mengacu pada jenis nilai yang dapat disimpan dan di operasikan dalam suatu program, secara garis besar tipe data dapat di bagi menjadi beberapa bentuk yaitu:

Tipe data primitif merujuk pada tipe data dasar atau bawaan yang didefinisikan oleh bahasa pemrograman itu sendiri. Tipe data primitif biasanya memiliki ukuran yang tetap dan dioperasikan langsung oleh CPU. Mereka merupakan fondasi dari struktur data yang lebih kompleks.

Tipe data abstrak 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.

Tipe data kolektif adalah tipe data yang menyimpan nilai objek secara bersamaan, tipe data ini memungkinkan pengguna mengakses, menyimpan dan mengelola data secara terstruktur.

BAB III

GUIDED

1. Guided 1

Source code

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

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

    // it allows user to enter operation i.e. +, -, /
    cin >> op;
    // it allow user to enter the operator
    cin >> num1 >> num2;

    // switch statement begins
    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 operathor is other than +, -, *, or /,
        // error message will display
        default:
            cout <<"error! operator is not correct";
    } // switch statement ends
    return 0;
}

```

Screenshoot program

```

PS C:\Users\fahri\OneDrive\Documents\Array> 3+3
6
PS C:\Users\fahri\OneDrive\Documents\Array> 

```

Deskripsi program

program di atas adalah program kalkulator sederhana di mana char op di gunakan untuk menyimpan bilangan yang akan di hitung dan operator op menggunakan cin, serta switch untuk memeriksa pertanyaan yang di berirak oleh user.

2. Guided 2

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("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

```
## mahasiswa 1 ##  
Nama: dian  
Umur: 22  
## Mahasiswa 2 ##  
Nama: bambang  
Alamat: surabaya  
Umur: 23
```

Deskripsi program

Program di atas berfungsi untuk menampilkan informasi tentang dua mahasiswa. Struct mahasiswa memiliki tiga anggota yaitu name, address, dan age, yang mewakili nama, alamat, dan usia mahasiswa. Dua variabel mhs1 dan mhs2 dideklarasikan dengan tipe struct mahasiswa, kemudian diisi dengan nilai masing-masing menggunakan operator penugasan. Selanjutnya, nilai dari masing-masing anggota struct diprint menggunakan fungsi printf()

3. Guided 3

Source code

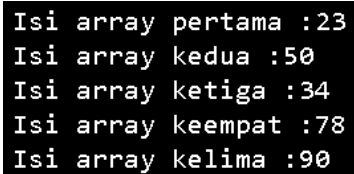
```
#include <iostream>  
  
using namespace std;  
  
int main() {  
    // deklarasi dan inisialisasi array  
    int nilai[5];  
    nilai[0] = 23;  
    nilai[1] = 50;  
    nilai[2] = 34;
```

```
        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



```
Isi array pertama :23
Isi array kedua :50
Isi array ketiga :34
Isi array keempat :78
Isi array kelima :90
```

Deskripsi program

Program di atas akan menampilkan array Array dengan cara, nilai dengan panjang 5 diinisialisasi dengan beberapa nilai integer. Setiap elemen array diakses dan dicetak satu per satu menggunakan operator indeks [] dan fungsi cout

UNGUIDED

1. Unguided 1

Source code

```
#include <iostream>

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

double kalikan(double x, double y) {
    return x * y;
}

int main() {
    int angka1 = 5;
    int angka2 = 3;
    double desimal1 = 2.0;
    double desimal2 = 5.0;

    std::cout << "Hasil penjumlahan: " << tambah(angka1, angka2)
    << std::endl;
    std::cout << "Hasil perkalian: " << kalikan(desimal1,
    desimal2) << std::endl;

    return 0;
}
```

Screenshoot program

```
Hasil penjumlahan: 8
Hasil perkalian: 10

Process returned 0 (0x0)   execution time : 0.076 s
Press any key to continue.
|
```

Deskripsi program

Program ini memiliki dua fungsi, yaitu tambah untuk menjumlahkan dua bilangan bulat dan kalikan untuk mengalikan dua bilangan desimal. Di dalam fungsi main, terdapat inisialisasi variabel *angka1*, *angka2*, *desimal1*, dan *desimal2*. Program mencetak hasil penjumlahan dari *angka1* dan *angka2* menggunakan fungsi *tambah*, serta hasil perkalian dari *desimal1* dan *desimal2* menggunakan fungsi *kalikan*.

Tipe data primitif seperti *int* dan *double* digunakan untuk menyimpan nilai sederhana seperti bilangan bulat dan desimal. Dalam program ini, tipe data primitif digunakan untuk melakukan operasi aritmatika sederhana seperti penjumlahan dan perkalian. Penggunaan tipe data primitif memudahkan dalam melakukan operasi matematika dasar dan pengolahan data numerik dalam pemrograman.

2. Unguided 2

Class adalah dasar konstruksi pemograman yang berbasis objek, yang memungkinkan pengunnanya penggunanya membuat data baru(variable) dan fungsi yang beroperasi pada data tersebut

```
#include <iostream>

class Mahasiswa {
public:
    std::string nama;
    int umur;

    void displayInfo() {
        std::cout << "Nama: " << nama << std::endl;
        std::cout << "Umur: " << umur << std::endl;
    }
};

struct Point {
    int x;
    int y;

    void displayCoordinates() {
        std::cout << "X: " << x << std::endl;
        std::cout << "Y: " << y << std::endl;
    }
};
```

```

int main() {
    Mahasiswa mhs;
    mhs.nama = "Jane Doe";
    mhs.umur = 22;
    mhs.displayInfo();

    Point p;
    p.x = 3;
    p.y = 7;
    p.displayCoordinates();

    return 0;
}

```

Screenshoot Program

```

Nama: Jane Doe
Umur: 22
X: 3
Y: 7

Process returned 0 (0x0)   execution time : 0.073 s
Press any key to continue.
|

```

Deskripsi Program

Dalam program di atas, terdapat dua entitas data yang berbeda, yaitu Mahasiswa (class) dan Point (struct). Kedua entitas tersebut memiliki variabel data dan fungsi untuk menampilkan informasi terkait data yang dimiliki. Class Mahasiswa memiliki variabel nama dan umur, serta fungsi displayInfo untuk menampilkan informasi mahasiswa. Sementara itu, struct Point memiliki variabel x dan y, serta fungsi displayCoordinates untuk menampilkan koordinat titik. Program ini menciptakan objek dari kedua entitas tersebut dan memanggil fungsi yang sesuai untuk menampilkan informasi yang relevan.

3. Unguided 3

```

#include <iostream>
#include <map>

int main() {

    std::map<int, std::string> dataMahasiswa;

    dataMahasiswa[101] = "John Doe";
    dataMahasiswa[102] = "Jane Smith";
    dataMahasiswa[103] = "Alice Johnson";

    std::cout << "Data Mahasiswa:" << std::endl;
    for (const auto& pair : dataMahasiswa) {
        std::cout << "NIM: " << pair.first << ", Nama: " <<
pair.second << std::endl;
    }

    return 0;
}

```

Screenshoot Program

```
Data Mahasiswa:  
NIM: 101, Nama: John Doe  
NIM: 102, Nama: Jane Smith  
NIM: 103, Nama: Alice Johnson  
  
Process returned 0 (0x0)   execution time : 0.018 s  
Press any key to continue.  
|
```

Deskripsi Program

Program di atas menggunakan `std::map` untuk menyimpan data mahasiswa dengan key (NIM) sebagai integer dan value (nama) sebagai string. Data mahasiswa ditambahkan ke dalam map menggunakan operator `[]`. Untuk menampilkan data mahasiswa, digunakan iterator untuk mengiterasi melalui setiap pasangan key-value dalam map.

Perbedaan utama antara array dan map adalah dalam cara elemen-elemen disimpan dan diakses. Array menggunakan indeks numerik untuk akses, sementara map menggunakan key yang dapat berupa tipe data apa pun. Map juga memiliki fleksibilitas lebih besar dalam menambah dan menghapus elemen tanpa perlu khawatir tentang ukuran map secara eksplisit.

BAB IV

KESIMPULAN

Dari hasil laporan di atas kesimpulan yang dapat di ambil adalah Program- program tersebut menunjukkan penggunaan tipe data primitif (seperti char, int, float), tipe data abstrak (seperti struct), dan tipe data kolektif (seperti array).Struktur program yang baik menggunakan konsep tipe data yang sesuai dengan kebutuhan, seperti penggunaan struct untuk mengelola data yang terkait dan array untuk menyimpan kumpulan nilai yang serupa.Program- program tersebut memberikan pemahaman dasar tentang penggunaan tipe data dan struktur dasar dalam pemrograman yang baik .