

LAPORAN PRAKTIKUM

MODUL I TIPE DATA



Disusun oleh:
Reli Gita Nurhidayati
NIM: 2311102025

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

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

BAB I

TUJUAN PRAKTIKUM

A. 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 adalah sebuah pengklasifikasian data berdasarkan jenis data tersebut. Tipe data dibutuhkan agar kompiller 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 perbedaan antara Praktikum Struktur Data dan Algoritma

2

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

Source code

```
#include <iostream>
using namespace std;
// Main program
int main ()
{
    char op;
    float num1, num2;

    cin >> op;
    cin >> num1 >> num2;

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

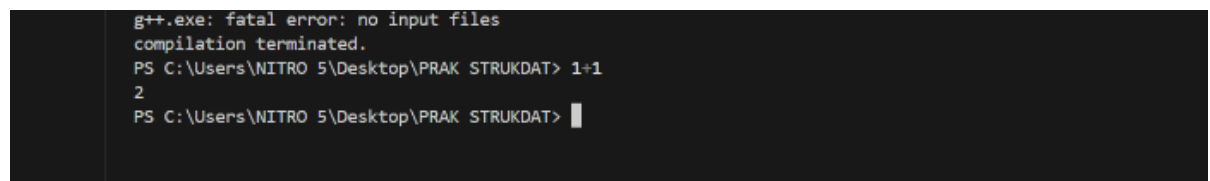
        case '-':
            cout << num1 - num2;
            break;

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

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

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

Screenshot program



```
g++.exe: fatal error: no input files
compilation terminated.
PS C:\Users\NITRO 5\Desktop\PRAK STRUKDAT> 1+1
2
PS C:\Users\NITRO 5\Desktop\PRAK STRUKDAT> █
```

Deskripsi program

Program ini merupakan kalkulator sederhana yang memungkinkan pengguna untuk melanjutkan operasi penjumlahan, pengurangan, perkalian, dan pembagian antara dua bilangan. Pengguna akan memasukkan operator (+, -, *, /) dan dua bilangan yang akan

dioperasikan. Setelah menerima input, program akan menggunakan struktur switch case untuk mengecek operator yang dimasukkan lalu melakukan operasi yang sesuai. Hasil operasi tersebut akan ditampilkan sebagai output. Jika operator yang dimasukkan tidak termasuk dalam operator yang valid (+, -, *, /), program akan menampilkan pesan kesalahan.

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 = "Dian";
    mhs1.address = "Mataram";
    mhs1.age = 22;
    mhs2.name = "Bambang";
    mhs2.address = "Surabaya";
    mhs2.age = 23;
    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\NITRO 5\Desktop\PRAK STRUKDAT> cd "c:\Users\NITRO 5\Desktop\PRAK STRUKDAT\" ; if ($?) { g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .
## Mahasiswa 1 ##
Nama: Dian
Alamat: Mataram
Umur: 22
## Mahasiswa 2 ##
Nama: Bambang
Alamat: Surabaya
Umur: 23
PS C:\Users\NITRO 5\Desktop\PRAK STRUKDAT>
```

Deskripsi Program

Program ini menggunakan struktur (struct) untuk merepresentasikan data mahasiswa. Dua objek Mahasiswa, yaitu mhs1 dan mhs2, dibuat dengan atribut *name*, *address*, dan *age*. Setiap objek diinisialisasi dengan nilai tertentu, Program kemudian mencetak informasi masing-masing mahasiswa seperti *name*, *address*, dan *age* ke layar menggunakan fungsi printf. Hal ini memungkinkan program untuk

mengelola dan menampilkan data mahasiswa dengan menggunakan struktur sebagai wadah informasi yang terkait.

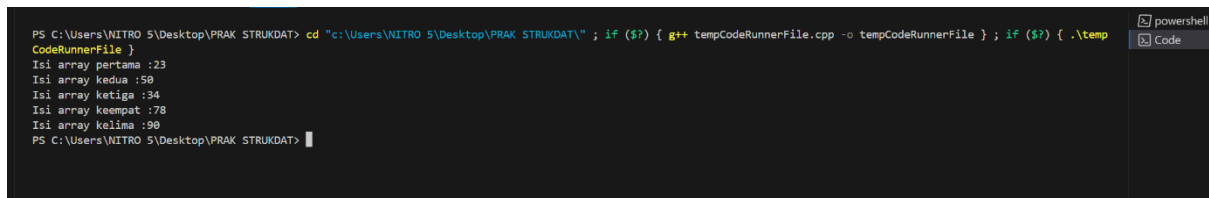
3. Guided 3 Source Code

```
#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



```
PS C:\Users\NITRO 5\Desktop\PRAK STRUKDAT> cd "c:\Users\NITRO 5\Desktop\PRAK STRUKDAT\" ; if ($?) { g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Isi array pertama :23
Isi array kedua :50
Isi array ketiga :34
Isi array keempat :78
Isi array kelima :90
PS C:\Users\NITRO 5\Desktop\PRAK STRUKDAT>
```

Deskripsi Program

Program ini merupakan array dalam c++, berguna untuk menyimpan dan mencetak sejumlah nilai. Array 'nilai' dengan Panjang 5 elemen dideklarasikan dan diinisialisasi dengan nilai tertentu, setiap elemen array akan dicetak ke layar menggunakan pernyataan cout. Dengan demikian, program ini mengilustrasikan penggunaan array untuk menyimpan dan mengakses sejumlah nilai secara terurut.

UNGUIDED

1. Tugas 1

Buatlah program menggunakan tipe data primitif minimal dua fungsi dan bebas. Menampilkan program, jelaskan program tersebut dan ambil kesimpulan dari materi tipe data primitif!

Source code

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

int hitungLuasPersegiPanjang(int panjang, int lebar) {
    return panjang * lebar;
}

float hitungVolumeBalok(float panjang, float lebar, float tinggi) {
    return panjang * lebar * tinggi;
}

int main() {
    int panjang_persegi, lebar_persegi;
    float panjang_balok, lebar_balok, tinggi_balok;

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

    cout << "Luas persegi panjang: " <<
    hitungLuasPersegiPanjang(panjang_persegi, lebar_persegi) << endl;

    cout << "Masukkan panjang balok: ";
    cin >> panjang_balok;
    cout << "Masukkan lebar balok: ";
    cin >> lebar_balok;
    cout << "Masukkan tinggi balok: ";
    cin >> tinggi_balok;

    cout << "Volume balok: " << hitungVolumeBalok(panjang_balok,
    lebar_balok, tinggi_balok) << endl;

    return 0;
}
```


Screenshoot program

```
PS C:\Users\NITRO 5\Desktop\PRAK STRUKDAT> cd "c:\Users\NITRO 5\Desktop\PRAK STRUKDAT\" ; if ($?) {  
Masukkan panjang persegi panjang: 12  
Masukkan lebar persegi panjang: 4  
Luas persegi panjang: 48  
Masukkan panjang balok: 15  
Masukkan lebar balok: 5  
Masukkan tinggi balok: 10  
Volume balok: 750  
PS C:\Users\NITRO 5\Desktop\PRAK STRUKDAT> }
```

Deskripsi program

Penjelasan Program

Program ini memiliki fungsi yaitu **hitungLuasPersegiPanjang** untuk menghitung luas persegi Panjang dan **hitungVolumeBalok** untuk menghitung volume balok. Di dalam fungsi **main**, program akan meminta pengguna untuk memasukkan Panjang dan lebar persegi Panjang serta Panjang, lebar, dan tinggi balok.

Setelah mendapatkan input, maka program akan memanggil **hitungLuasPersegiPanjang** untuk menghitung luas persegi Panjang dan menampilkan hasilnya. Kemudian, program akan memanggil fungsi **hitungVolumeBalok** untuk menghitung volume balok dan menampilkan hasilnya.

Kesimpulan:

Tipe data primitif seperti int dan float sangat berguna dalam pemrograman c++ untuk merepresentasikan bilangan bulat dan bilangan riil. Menggunakan tipe data primitif membuat kita dapat melakukan operasi aritmatika dengan mudah. Penggunaan tipe data Primitif juga memungkinkan kita untuk membuat program yang lebih efisien dan sederhana, serta melakukan perhitungan matematika yang dibutuhkan dalam berbagai konteks.

2. Tugas 2

Jelaskan fungsi dari class dan struct secara detail dan berikan contoh programnya!

Class berfungsi sebagai konsep dasar dalam pemrograman berorientasi objek yang digunakan untuk membuat cetak biru untuk membuat objek. Ini akan memungkinkan pengguna untuk menggabungkan data dan fungsi terkait dalam satu kesatuan.

Struct merupakan singkatan dari structure, merupakan tipe data pengguna yang digunakan sebagai penggabung beberapa jenis data yang berbeda-beda dalam satu unit.

Class dan struct dalam C++ memiliki fungsi yang mirip, yaitu untuk mendefinisikan tipe data baru yang dapat menyimpan data dan fungsi-fungsi yang dapat dioperasikan pada data tersebut. Perbedaan utama antar keduanya terletak pada default aksesibilitas anggotanya: class secara default bersifat *private*, sedangkan struct secara default bersifat *public* atau umum.

Contoh Program

Source Code

```
#include <iostream>
#include <string>

class PersonClass {
private:
    std::string name;
    int age;

public:
    void setName(std::string inputName) {
        name = inputName;
    }

    void setAge(int inputAge) {
        age = inputAge;
    }

    void displayInfo() {
        std::cout << "Nama: " << name << ", Umur: " << age << " tahun" <<
std::endl;
    }
};

struct PersonStruct {
    std::string name;
    int age;

    void setName(std::string inputName) {
        name = inputName;
    }

    void setAge(int inputAge) {
        age = inputAge;
    }

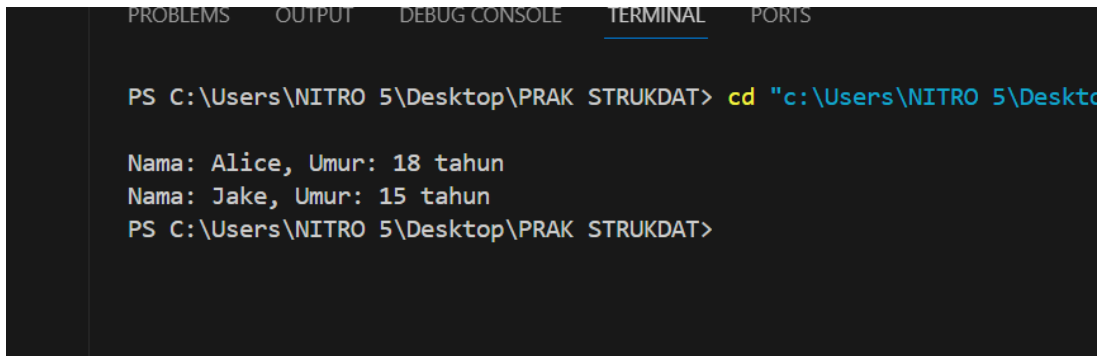
    void displayInfo() {
        std::cout << "Nama: " << name << ", Umur: " << age << " tahun" <<
std::endl;
    }
};

int main() {
    PersonClass personClass;
    personClass.setName("Alice");
    personClass.setAge(18);
    personClass.displayInfo();

    PersonStruct personStruct;
    personStruct.setName("Jake");
    personStruct.setAge(15);
    personStruct.displayInfo();

    return 0;
}
```

Screenshoot Program



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\NITRO 5\Desktop\PRAK STRUKDAT> cd "c:\Users\NITRO 5\Desktop\PRAK STRUKDAT"
Nama: Alice, Umur: 18 tahun
Nama: Jake, Umur: 15 tahun
PS C:\Users\NITRO 5\Desktop\PRAK STRUKDAT>
```

Deskripsi Program

Pada program ini mendefinisikan sebuah class **PersonClass** dan sebuah struct **personStruct** yang digunakan untuk merepresentasikan informasi suatu individu. Kedua struktur tersebut memiliki data anggota *name*(nama) dan *age* (umur), serta metode untuk mengatur data anggota tersebut dan menampilkan informasi individu tersebut.

PersonClass memiliki data anggota *name* dan *age* yang bersifat *private*, sehingga hanya dapat diakses dari dalam class tersebut. Metode **setName** dan **setAge** digunakan untuk **displayinfo** yang digunakan untuk menampilkan informasi dan umur.

PersonStruct memiliki data anggota *name* dan *age* yang bersifat *public* secara default, sehingga dapat diakses dari luar struct. Struct ini juga memiliki metode **setName**, **setAge**, dan **displayInfo** yang serupa dengan **PersonClass**.

Dalam fungsi **main**, objek dari class **PersonClass** dan struct **PersonStruct** dibuat, data anggota objek diisi berupa nama dan umur yang sesuai, lalu informasi individu akan ditampilkan dengan memanggil metode **displayInfo**.

3. Tugas 3

Buat dan jelaskan program menggunakan fungsi map dan jelaskan perbedaan dari array dengan map.

Contoh Program

Source Code

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

int main() {

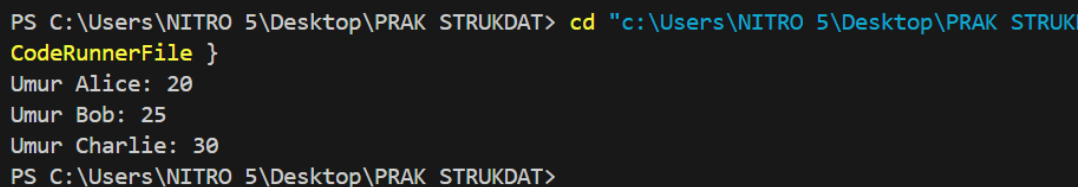
    map<string, int> umur;

    umur["Alice"] = 20;
    umur["Bob"] = 25;
    umur["Charlie"] = 30;

    cout << "Umur Alice: " << umur["Alice"] << endl;
    cout << "Umur Bob: " << umur["Bob"] << endl;
    cout << "Umur Charlie: " << umur["Charlie"] << endl;

    return 0;
}
```

Screenshoot Program



```
PS C:\Users\NITRO 5\Desktop\PRAK STRUKDAT> cd "c:\Users\NITRO 5\Desktop\PRAK STRUKDAT"
CodeRunnerFile }
Umur Alice: 20
Umur Bob: 25
Umur Charlie: 30
PS C:\Users\NITRO 5\Desktop\PRAK STRUKDAT>
```

Deskripsi Program

Program ini menggunakan fungsi **std::map** untuk menyimpan pasangan data nama dan umur. Pada library yang digunakan adalah **iostream** dan **using namespace std**. **Iostream** berguna untuk input-output dan **map** untuk menggunakan struktur data map. Sedangkan statement **using namespace std**; digunakan untuk menghilangkan kebutuhan menulis **std::** sebelum pemanggilan fungsi dari namespace **std**.

Membuat dan mengisi Map, map umur dibuat dengan key bertipe string dan value bertipe integer untuk menyimpan data nama dan umur. Untuk bagian akses dan tampilan data, data diakses dan ditampilkan menggunakan key sebagai referensi. Ini akan menghasilkan output umur dari setiap orang yang tersimpan dalam map.

Program di atas menggunakan map untuk mengaitkan nama dengan umur. Dengan menggunakan map, data dapat diakses menggunakan key yang bersesuaian dengan data yang disimpan, menjadikan struktur data ini sangat berguna untuk menyimpan informasi yang memerlukan hubungan antara dua nilai yang terkait.

Perbeddaan utama antara array dan map adalah array menggunakan indeks numerik untuk mengakses elemennya, sedangkan map menggunakan key untuk mengakses nilai-nilainya. Map lebih fleksibel karena key tidak terbatas pada tipe data numerik atau urutan tertentu seperti pada array, sehingga memungkinkan pemetaan yang lebih kompleks antara key dan value.

BAB IV

KESIMPULAN

Pemahaman mendalam tentang tipe data, class, struct, array, dan map dalam pemrograman c++ memberikan dasar yang kuat untuk pengembangan program yang efisien dan terstruktur. Tipe data memungkinkan pengguna untuk memilih representasi data yang tepat untuk setiap kebutuhan, sementara class dan struct memfasilitasi penggunaan konsep pemrograman berorientasi objek dalam mengorganisir data dan fungsionalitas terkait dalam satu kesatuan. Penggunaan array memungkinkan penyimpanan data dalam bentuk pasangan key-value yang memudahkan pencarian dan manipulasi data.

Dengan memahami tipe data, class, struct, array, dan map, pengembang dapat meningkatkan efisiensi program, mengoptimalkan penggunaan memori, dan mempercepat proses pengembangan. Selain itu, penguasaan konsep-konsep ini memungkinkan untuk menciptakan program yang lebih mudah dipelihara, dimodifikasi, dan diperluas.