

**LAPORAN PRAKTIKUM STRUKTUR
DATA DAN ALGORITMA**

**MODUL I
TIPE DATA**



Disusun Oleh :

NAMA : RIFKY DWI MAHARDIKA
NIM : 2311102043

Dosen

Wahyu Andi Saputra, S.Pd., M.Eng

**PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2024**

A. Dasar Teori

Tipe Data

Tipe data memiliki peran penting dalam menentukan bentuk dan operasi yang dapat dilakukan oleh variabel. Tipe data dibutuhkan supaya compiler dapat mengetahui bagaimana sebuah data akan digunakan. Tipe data dibagi menjadi beberapa bagian, antara lain :

1) Tipe data Primitif

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 primitive antara lain :

- 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.

2) Tipe data Abstrak

Tipe data abstrak (ADT) adalah tipe data yang dibentuk oleh programmer sendiri ADT dapat berisi banyak tipe data, sehingga nilainya bisa lebih dari satu dan beragam. ADT umumnya dibuat menggunakan fitur Class dalam bahasa pemrograman berorientasi objek (OOP). Fitur Class mirip dengan fitur data structures Struct pada bahasa C. Keduanya berfungsi untuk membungkus tipe data di dalamnya sebagai anggota.

3) 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, antara lain:

- 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". Map digunakan Self-Balancing Tree khususnya Red-Black Tree.

B. Guided

Guided 1

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

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

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

    // Meminta pengguna untuk memasukkan operand
    cout << "Masukkan dua angka: ";
    cin >> num1 >> num2;

    // Switch statement
    switch (op)
    {
        case '+':
            cout << "Hasil penjumlahan: " << num1 + num2;
            break;
        case '-':
            cout << "Hasil pengurangan: " << num1 - num2;
            break;
        case '*':
            cout << "Hasil perkalian: " << num1 * num2;
            break;
        case '/':
            if (num2 != 0)
                cout << "Hasil pembagian: " << num1 / num2;
            else
                cout << "Error! Tidak dapat melakukan pembagian dengan nol.";
            break;
        default:
            cout << "Error! Operator tidak benar.";
    }

    return 0;
}
```

Screenshots Output

```
PS C:\Users\ASUS\Documents\semester 2\struktur data algoritma\laprak 1\tugas> cd "c:\Users\ASUS\Documents\semester 2\struktur data algoritma\laprak 1\type data primitif\" ; if ($?) { g++ Guided_1.cpp -o Guided_1 } ; if ($?) { .\Guided_1 }
Masukkan operator (+, -, *, /): +
Masukkan dua angka: 2 15
Hasil penjumlahan: 17
PS C:\Users\ASUS\Documents\semester 2\struktur data algoritma\laprak 1\tugas>
```

+

NAMA : RIFKY DWI MAHARDIKA
NIM : 2311102043

Deskripsi:

Sebuah program kalkulator sederhana yang meminta user untuk memasukkan operator matematika (+, -, *, /) dan dua operasi numerik. Setelah menerima input, program menggunakan struktur kontrol switch-case untuk melakukan operasi yang sesuai berdasarkan operator yang dimasukkan. Program ini memberikan solusi sederhana untuk melakukan operasi aritmatika dasar menggunakan operator yang dimasukkan oleh user.

Guided 2

```
#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
    //mhs 1
    mhs1.name = "Dian";
    mhs1.address = "Mataram";
    mhs1.age = 22;
    //mhs 2
    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;
}

```

Screenshots Output

```

PS C:\Users\ASUS\Documents\semester 2\struktur data algoritma\laprak 1\tipe d
ata algoritma\laprak 1\tipe data abstrak\> if ($?) { g++ Guided_2.cpp -o Gu
## Mahasiswa 1 ##
Nama: Dian
Alamat: Mataram
Umur: 22
## Mahasiswa 2 ##
Nama: Bambang
Alamat: Surabaya
Umur: 23

```

NAMA : RIFKY DWI MAHARDIKA
NIM : 2311102043

Deskripsi:

Sebuah contoh program penggunaan struct (struktur data) dalam bahasa pemrograman C. Struct digunakan untuk menggabungkan beberapa jenis data yang berbeda ke dalam satu kesatuan yang disebut sebagai tipe data yang didefinisikan oleh users. Program ini memanfaatkan struct untuk mengorganisir informasi tentang mahasiswa ke dalam satu kesatuan yang terstruktur.

Guided 3

```

#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;
}

```

Screenshots Output

```

PS C:\Users\ASUS\Documents\semester 2\struktur data algoritma\laprak 1\tipe data abstrak> cd "c:\U
ata algoritma\laprak 1\tipe data koleksi\" ; if ($?) { g++ Guided_3.cpp -o Guided_3 } ; if ($?) {
Isi array pertama :23
Isi array kedua :50
Isi array ketiga :34
Isi array keempat :78
Isi array kelima :90
PS C:\Users\ASUS\Documents\semester

```

+

NAMA : RIFKY DWI MAHARDIKA
NIM : 2311102043

Deskripsi:

Program C++ sederhana yang mendemonstrasikan penggunaan array untuk menyimpan dan mencetak beberapa nilai int. Program ini memberikan contoh sederhana tentang bagaimana menggunakan array untuk menyimpan beberapa nilai dalam urutan tertentu dan kemudian mengakses dan mencetak nilai-nilai tersebut. Dalam contoh ini, array digunakan untuk menyimpan nilai-nilai int, tetapi konsepnya dapat diterapkan pada jenis data lainnya.

C. Unguided

Unguided 1

```
#include <iostream>
#include <cmath>

using namespace std;

// Fungsi menghitung luas dan volume persegi panjang
float hitungLuasPersegiPanjang(float panjang, float lebar)
{
    return panjang * lebar;
}

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

// Fungsi menghitung luas dan volume lingkaran
float hitungLuasLingkaran(float jari_jari)
{
    return M_PI * pow(jari_jari, 2);
}

float hitungVolumeBola(float jari_jari)
{
    return (4.0 / 3.0) * M_PI * pow(jari_jari, 3);
}

int main()
{
    int pilihan;
    cout << "\n";
    cout << "Pilihan Menu:\n";
    cout << "1. Persegi Panjang (Balok)\n";
    cout << "2. Lingkaran (Bola)\n";
    cout << "Masukkan pilihan Anda: ";
    cin >> pilihan;

    if (pilihan == 1)
    {
        float panjang, lebar, tinggi;
        cout << "\n";
        cout << " ## PERSEGI PANJANG (BALOK) ## \n";
        cout << "Masukkan panjang persegi panjang: ";
    }
}
```



```

        cin >> panjang;
        cout << "Masukkan lebar persegi panjang: ";
        cin >> lebar;
        cout << "Masukkan tinggi balok (untuk hitung volume): ";
        cin >> tinggi;

        float luas = hitungLuasPersegiPanjang(panjang, lebar);
        float volume = hitungVolumeBalok(panjang, lebar, tinggi);

        cout << "\n";
        cout << " ##  HASIL LUAS & VOLUME  ## \n";
        cout << "Luas persegi panjang = " << luas << endl;
        cout << "Volume balok dengan tinggi " << tinggi << " = " <<
volume << endl;
    }
    else if (pilihan == 2)
    {
        float jari_jari;
        cout << "\n";
        cout << " ##  LINGKARAN (BOLA)  ## \n";
        cout << "Masukkan jari-jari lingkaran: ";
        cin >> jari_jari;

        float luas = hitungLuasLingkaran(jari_jari);
        float volume = hitungVolumeBola(jari_jari);
        cout << "\n";
        cout << " ##  HASIL LUAS & VOLUME  ## \n";
        cout << "Luas lingkaran = " << luas << endl;
        cout << "Volume bola = " << volume << endl;
    }
    else
    {
        cout << "Pilihan tidak valid!\n";
    }

    return 0;
}

```

Screenshots Output

Menu 1

```
PS C:\Users\ASUS\Documents\semester 2\struktur data algoritma\laprak 1\tugas> cd "c:\Users\ASUS\Documents\semester 2\struktur data algoritma\laprak 1\tugas\" ; if ($?) { g++ unguided_1.cpp -o unguided_1 } ; if ($?) { .\unguided_1 }

Pilihan Menu:
1. Persegi Panjang (Balok)
2. Lingkaran (Bola)
Masukkan pilihan Anda: 1

## PERSEGI PANJANG (BALOK) ##
Masukkan panjang persegi panjang: 5
Masukkan lebar persegi panjang: 2
Masukkan tinggi balok (untuk hitung volume): 5

## HASIL LUAS & VOLUME ##
Luas persegi panjang = 10
Volume balok dengan tinggi 5 = 50
PS C:\Users\ASUS\Documents\semester 2\struktur data algoritma\laprak 1\tugas>
```

+

NAMA : RIFKY DWI MAHARDIKA
NIM : 2311102043

Menu 2

```
PS C:\Users\ASUS\Documents\semester 2\struktur data algoritma\laprak 1\tugas> cd "c:\Users\ASUS\Documents\semester 2\struktur data algoritma\laprak 1\tugas\" ; if ($?) { g++ unguided_1.cpp -o unguided_1 } ; if ($?) { .\unguided_1 }

Pilihan Menu:
1. Persegi Panjang (Balok)
2. Lingkaran (Bola)
Masukkan pilihan Anda: 2

## LINGKARAN (BOLA) ##
Masukkan jari-jari lingkaran: 5

## HASIL LUAS & VOLUME ##
Luas lingkaran = 78.5398
Volume bola = 523.599
PS C:\Users\ASUS\Documents\semester 2\struktur data algoritma\laprak 1\tugas>
```

+

NAMA : RIFKY DWI MAHARDIKA
NIM : 2311102043

Deskripsi:

Program di atas adalah program sederhana yang mendemonstrasikan penggunaan array untuk menyimpan dan mencetak beberapa nilai int. Program ini memberikan contoh sederhana tentang bagaimana menggunakan array untuk menyimpan beberapa nilai dalam urutan tertentu dan kemudian mengakses dan mencetak nilai-nilai tersebut. Dalam contoh ini, array digunakan untuk menyimpan nilai-nilai integer, tetapi konsepnya dapat diterapkan pada jenis data lainnya.

Kesimpulan dari materi tipe data primitif adalah tipe data dasar yang terdapat di dalam suatu bahasa pemrograman, contohnya int, float, char, dan lainnya. Tipe data ini bisa menyimpan nilai sederhana kemudian dapat melakukan operasi matematika dasar, pemrosesan data, dan logika pada program.

Unguided 2

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

// struct
struct MahasiswaStruct
{
    string nama;
    int umur;
};

// class
class MahasiswaClass
{
private:
    string nama;
    int umur;

public:
    MahasiswaClass(string n, int u)
    {
        nama = n;
        umur = u;
    }

    void tampilkanData()
    {
        cout << "## MENGGUNAKAN CLASS ##\n";
        cout << "Nama : " << nama << endl;
        cout << "Umur : " << umur << endl;
    }
};

int main()
{
    // struct
    MahasiswaStruct mhsStruct;
    mhsStruct.nama = "Jono";
    mhsStruct.umur = 20;

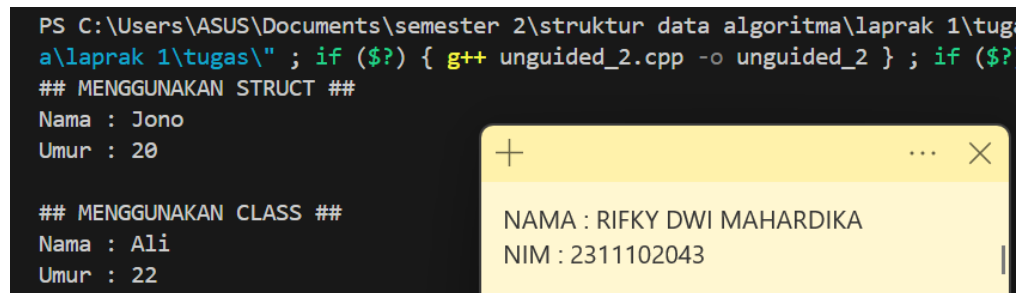
    cout << "## MENGGUNAKAN STRUCT ##\n";
    cout << "Nama : " << mhsStruct.nama << endl;
    cout << "Umur : " << mhsStruct.umur << endl;

    cout << endl;
```

```
// class
MahasiswaClass mhsClass("Ali", 22);
mhsClass.tampilkanData();

return 0;
}
```

Screenshots Output



```
PS C:\Users\ASUS\Documents\semester 2\struktur data algoritma\laprak 1\tugas\
a\laprak 1\tugas\> if ($?) { g++ unguided_2.cpp -o unguided_2 } ; if ($?)
## MENGGUNAKAN STRUCT ##
Nama : Jono
Umur : 20

## MENGGUNAKAN CLASS ##
Nama : Ali
Umur : 22
```

Deskripsi:

Program di atas mendemonstrasikan penggunaan struct dan class dalam C++. Program ini memberikan contoh penggunaan struct dan class dalam C++, dan menunjukkan perbedaan dalam penggunaan keduanya. Struct digunakan untuk membuat tipe data baru yang terdiri dari beberapa anggota, sedangkan class digunakan untuk membuat objek yang memiliki anggota data dan fungsi-fungsi (metode) yang beroperasi pada anggota data tersebut.

Penjelasan pada contoh program diatas, antara lain:

- Struct digunakan untuk mengelompokkan beberapa variabel dengan tipe data yang berbeda ke dalam satu unit. Pada program ini, “MahasiswaStruct” adalah sebuah struct yang memiliki dua anggota: “nama” dengan tipe string dan “umur” dengan tipe int. Struct ini tidak memiliki metode atau fungsi terkait, hanya digunakan untuk menyimpan data.
- Class adalah struktur data yang dapat menggabungkan data variabel dan fungsi

metode ke dalam satu unit. Pada program ini, “MahasiswaClass” adalah sebuah class yang memiliki dua anggota: nama dengan tipe string dan “umur” dengan tipe int. Class ini memiliki constructor “MahasiswaClass” ‘string n, int u’ yang digunakan untuk menginisialisasi objek dari class tersebut dengan nilai yang diberikan. Class juga memiliki metode tampilkanData() yang digunakan untuk menampilkan data yang disimpan dalam objek class.

Ketika program dijalankan:

- Bagian pertama menunjukkan penggunaan struct. Sebuah objek dari struct “MahasiswaStruct” dibuat dan diisi dengan nilai “nama” dan “umur”. Kemudian nilai-nilai tersebut ditampilkan menggunakan “cout”.
- Bagian kedua menunjukkan penggunaan class. Sebuah objek dari class “MahasiswaClass” dibuat menggunakan constructor yang disediakan. Data kemudian ditampilkan menggunakan metode tampilkanData().

Unguided 3

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

int main()
{
    map<int, string> NIK;

    NIK[3335340001] = "Didik Sutikno";
    NIK[3932530009] = "Jatmiko Adi";
    NIK[3253300002] = "Joko Widodo";

    cout << "Data Warga Penerima Bantuan 2024: " << endl;
    cout << "1. " << NIK[3335340001] << endl;
    cout << "2. " << NIK[3932530009] << endl;
    cout << "3. " << NIK[3253300002] << endl;
    cout << endl;
```

```
    return 0;
}
```

Screenshots Output

```
a\laprak 1\tugas\" ; if ($?) { g++ unguided_3.cpp -o unguided_3 }
Data Warga Penerima Bantuan 2024:
1. Didik Sutikno
2. Jatmiko Adi
3. Joko Widodo

PS C:\Users\ASUS\Documents\semeste
```

+

NAMA : RIFKY DWI MAHARDIKA
NIM : 2311102043

Deskripsi:

Program tersebut adalah contoh penggunaan struktur data map dalam bahasa pemrograman C++. Program ini menggunakan map untuk menyimpan dan mengakses data warga berdasarkan NIK. Map memungkinkan kita untuk dengan mudah mencari nilai (nama warga) berdasarkan kunci (NIK) tanpa harus memperhatikan indeks atau urutan penyimpanan data.

perbedaan dari array dengan map, antara lain :

- Array adalah struktur data linear yang menyimpan elemen-elemennya secara berurutan dalam memori. Setiap elemen diakses menggunakan indeks numerik yang berurutan, yang dimulai dari 0. Ukuran array harus ditentukan pada saat kompilasi, dan tidak bisa berubah selama program berjalan. Array cocok digunakan ketika kita memiliki kumpulan data dengan ukuran tetap dan kita hanya perlu mengakses elemen secara berurutan.
- Map adalah struktur data yang memetakan kunci ke nilai. Kunci dalam map bersifat unik dan dapat berupa tipe data apa pun, sedangkan nilai dapat berupa tipe data apa pun juga. Map menyediakan cara untuk mengakses nilai berdasarkan kunci yang terkait, bukan dengan indeks numerik seperti pada array.

D. Kesimpulan

Tipe data pemrograman merupakan atribut yang berkaitan dengan data yang akan memberi tahu sistem komputer. Sehingga nantinya bisa menafsirkan nilai dari data tersebut. Secara khusus, tipe data adalah format penyimpanan data. Data bisa dalam bentuk variabel untuk tipe data tertentu. Tipe data bisa mencakup angka, huruf dan karakter lainnya. Pada dasarnya ada banyak karakter berbeda, namun dianggap sebagai teks. Pemahaman tipe data pemrograman tidak hanya dibutuhkan oleh programmer saja. Orang awam pun bisa mempelajarinya.

E. Referensi

MEILINAEKA. (2022). Mengenal Tipe Data Pemrograman. diakses dari <https://it.telkomuniversity.ac.id/6-tipe-data-pemrograman-yang-harus-anda-pahami-apa-saja/>

TylerMSFT. (n.d.). Collections (C++/CX). diakses dari <https://learn.microsoft.com/en-us/cpp/cppcx/collections-c-cx?view=msvc-170>

Martin Mulyo Syahidin. (2019). Mengenal Jenis-jenis Tipe Data dalam Bahasa C++