

**LAPORAN PRAKTIKUM STRUKTUR
DATA DAN ALGORITMA**

**MODUL I
PENGENALAN CODE BLOCKS**



Disusun Oleh :

NAMA : Pandia Arya Brata

NIM : 2311102076

Dosen

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

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

MODUL 1

TIPE DATA

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 pemrograman yang telah ditentukan.

B. 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 (OOP) 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

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.

C. GUIDED

Guided 1 : tipe data primitif

```
#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
```

```

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;
}

```

Guided 2 : tipe data abstrak

```

#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;
}

```

Guided 3 : tipe data koleksi

```

#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;
}

```

D. UNGUIDED

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 main()
{
    int nilai0;
    float nilai1, nilai2;
    float tambahan, kurangan, kalian, bagian;

    cout << "KALKULATOR ANAK DIAMOND \n";
    cout << "===== "<< endl;
    cout << "Tambahan (1)" << endl;
    cout << "Kurangan (2)" << endl;
    cout << "Perkalian (3)" << endl;
    cout << "Pembagian (4)" << endl;
    cout << "===== "<< endl;
    cout << "Masukan pilihan : ";
    cin >> nilai0;
    cout << endl;
    cout << "===== "<< endl;

    cout << "Nilai 1 : ";
    cin >> nilai1;
    cout << "Nilai 2 : ";
    cin >> nilai2;

    tambahan = nilai1 + nilai2;
    kurangan = nilai1 - nilai2;
    kalian = nilai1 * nilai2;
    bagian = nilai1 / nilai2;

    if (nilai0==1){
        cout << "Hasil : " << tambahan << endl;
    }
    else if (nilai0==2){
        cout << "Hasil : " << kurangan << endl;
    }
    else if (nilai0==3){
        cout << "Hasil : " << kalian << endl;
    }
    else if (nilai0==4){
```

```

        cout << "Hasil : " << bagian << endl;
    }

    cout << "===== "<< endl;

    return 0;
}

```

Screenshot program :

```

PS C:\Users\Administrator> cd 'c:\Users\Administrator\Desktop\STRUKDAT (PRAKTIK)\Pertemuan 1\output'
PS C:\Users\Administrator\Desktop\STRUKDAT (PRAKTIK)\Pertemuan 1\output> & .\'soal1.exe'
KALKULATOR ANAK DIAMOND
=====
Tambahkan (1)
Kurangan (2)
Perkalian (3)
Pembagian (4)
=====
Masukan pilihan : 1
=====
Nilai 1 : 10
Nilai 2 : 10
Hasil : 20
=====
PS C:\Users\Administrator\Desktop\STRUKDAT (PRAKTIK)\Pertemuan 1\output> 

```

Deskripsi Program :

Program diatas adalah program kalkulator dengan menggunakan tipe data primitif yang menggunakan tipe data int tipe data yang digunakan untuk menyimpan bilangan bulat seperti 12, 1, 4, dan sebagainya, dan float, tipe data yang digunakan untuk menyimpan bilangan desimal seperti 1.5, 2.1, 3.14, dan sebagainya.

2. Jelaskan fungsi dari class dan struct secara detail dan berikan contoh programnya

Jawab :

Class dan structure adalah dua cara yang digunakan dalam pemrograman berorientasi objek untuk mengenali dan menggambarkan objek dan perilakunya. Namun, kedua konstruk ini memiliki perbedaan-perbedaan dalam cara mereka bekerja dan mengorganisir data. Berikut adalah penjelasan lebih detail tentang fungsi dari class dan struktur.

Fungsi Class

Class adalah sebuah konstruk dalam pemrograman berorientasi objek yang memungkinkan programmer untuk mendefinisikan kerangka atau blueprint dari suatu objek. Dalam class, terdapat atribut, metode, dan properti yang mendefinisikan karakteristik dan perilaku suatu objek. Beberapa fungsi dari class adalah:

a. Encapsulation

Dalam class, encapsulation digunakan untuk menyembunyikan sebagian atau seluruh data dan metode dari luar class. Hal ini memungkinkan programmer untuk mengendalikan akses ke data dan menghindari perubahan yang tidak terduga.

b. Inheritance

Dalam class, inheritance digunakan untuk membuat class baru yang dapat mengambil karakteristik dan perilaku dari class yang sudah ada atau parent class. Inheritance memungkinkan programmer untuk membuat class yang lebih spesifik atau lebih umum.

c. Polymorphism

Dalam class, polymorphism digunakan untuk membuat metode atau fungsi yang dapat mengambil berbagai jenis argumen atau tipe data. Hal ini memungkinkan programmer untuk membuat kode yang lebih fleksibel dan dapat digunakan kembali.

Fungsi Structure

Structure adalah sebuah konstruk dalam pemrograman berorientasi objek yang digunakan untuk mengenali dan mendefinisikan objek dengan tipe data kompleks. Dalam struktur, terdapat atribut atau variabel yang menyimpan nilai data. Beberapa fungsi dari structure adalah:

a. Mendefinisikan tipe data kompleks

Structure memungkinkan programmer untuk mendefinisikan tipe data kompleks yang terdiri dari beberapa nilai data. Hal ini memudahkan programmer dalam menyimpan dan mengakses nilai data dalam satu objek.

b. Memproses beberapa data sekaligus

Structure memungkinkan programmer untuk memproses beberapa data sekaligus dengan menggunakan loop atau array. Hal ini memungkinkan programmer untuk mengelola data dalam satu objek dengan lebih efisien.

c. Penggunaan yang lebih ringan

Structure dapat digunakan untuk menyimpan data tanpa harus membuat objek dengan metod atau fungsi. Oleh karena itu, penggunaan structure lebih ringan dibandingkan dengan class.

Contoh Program :

```
#include <iostream>

using namespace std;

// Struct
struct PUBG_M_Player
{
    string nama;
    string tim;
    int umur;
};

// Class
class MLBB_Player
{
public:
    string nama;
    string tim;
    int umur;

    void printPlayer()
    {
        cout << "Nama : " << nama << endl;
        cout << "Tim : " << tim << endl;
        cout << "Umur : " << umur << endl;
    }
};
```



```

    }
};

int main()
{
    // Struct
    cout << "PlayerUnknown's Battlegrounds\n";
    cout << "=====\n";
    PUBGM_Player player1 = {"Mas Faiz", "Ngawi Esport", 17};
    cout << "Nama: " << player1.nama << endl;
    cout << "Tim: " << player1.tim << endl;
    cout << "Umur: " << player1.umur << endl;

    // Class
    cout << "\tMobile Legends\n";
    cout << "=====\n";
    MLBB_Player player2;
    player2.nama = "Mr.Ironi";
    player2.tim = "Ngawi Glory";
    player2.umur = 15;
    player2.printPlayer();

    return 0;
}

```

Screenshot program :

The screenshot shows a terminal window with the following output:

```

PS C:\Users\Administrator\Desktop\STRUKDAT (PRAKTIK)\Pertemuan 1> cd 'c:\Users\Administrator\Desktop\STRUKDAT (PRAKTIK)\Pertemuan 1\output' & .\502.exe
PlayerUnknown's Battlegrounds
=====
Nama: Mas Faiz
Tim: Ngawi Esport
Umur: 17

    Mobile Legends
=====
Nama : Mr.Ironi
Tim : Ngawi Glory
Umur : 15
PS C:\Users\Administrator\Desktop\STRUKDAT (PRAKTIK)\Pertemuan 1\output>

```

Overlaid on the terminal is a small text editor window titled 'Nama Pa...'. It contains the following text:

```

Nama : Pandia Arya Brata
Nim : 2311102076

```

The terminal window has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active. The text editor window has tabs for File, Edit, and View. The status bar at the bottom of the text editor shows 'Ln 2, Col 18 | 42 characters | 100% | Window UTF-8'.

Deskripsi program :

1. Program di atas menggunakan struct PUBGM_Player untuk menyimpan informasi tentang seorang pemain game PUBG , yaitu nama, asal tim, dan umur.
2. Program juga menggunakan class game MLBB_Player untuk menyimpan informasi tentang seorang pemain game MLBB , yaitu nama, tim, dan umur.
3. Di dalam class MLBB_Player , terdapat fungsi printPlayer() yang digunakan untuk mencetak informasi tentang seorang pemain game MLBB.
4. Di dalam fungsi main(), program membuat objek player1 dari struct PUBGM_Player, serta objek player2 dari class MLBB_Player.

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

Source Code :

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

using namespace std;

int main()
{
    // membuat map dengan tipe data <string, int>
    map<string, int> m;

    // memasukkan data ke dalam map
    m["apel"] = 10;
    m["jeruk"] = 20;
    m["pisang"] = 30;

    // mengakses data dalam map
    cout << "Jumlah apel: " << m["apel"] << endl;
    cout << "Jumlah jeruk: " << m["jeruk"] << endl;

    // mengubah data dalam map
    m["jeruk"] = 25;

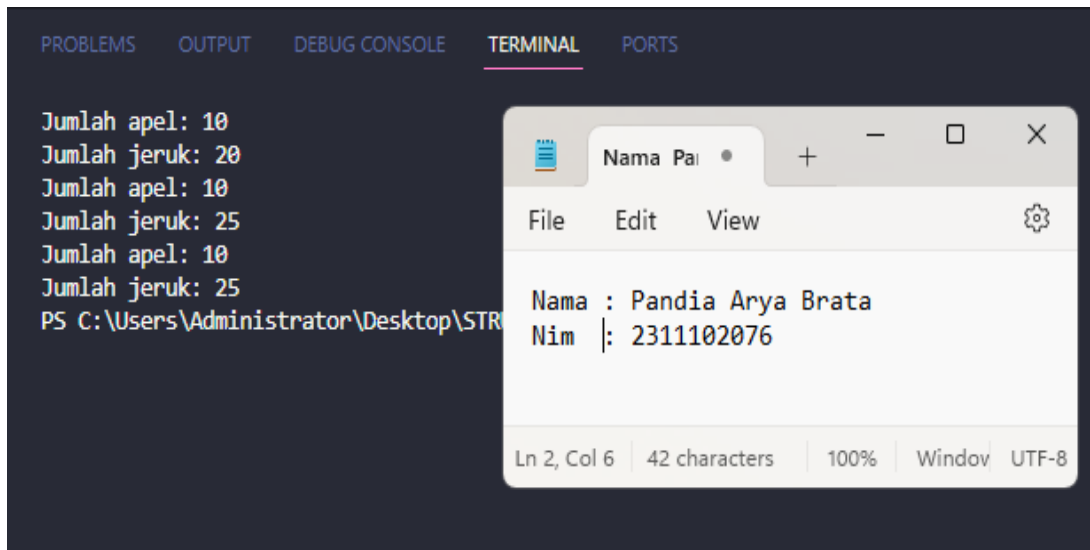
    // menampilkan data dalam map setelah diubah
    cout << "Jumlah apel: " << m["apel"] << endl;
    cout << "Jumlah jeruk: " << m["jeruk"] << endl;

    // menghapus data dalam map
    m.erase("pisang");

    // menampilkan data dalam map setelah dihapus
    cout << "Jumlah apel: " << m["apel"] << endl;
    cout << "Jumlah jeruk: " << m["jeruk"] << endl;

    return 0;
}
```

Screenshot program :



The screenshot shows a C++ IDE interface. The 'TERMINAL' tab is active, displaying the following output:

```
Jumlah apel: 10
Jumlah jeruk: 20
Jumlah apel: 10
Jumlah jeruk: 25
Jumlah apel: 10
Jumlah jeruk: 25
PS C:\Users\Administrator\Desktop\STR
```

Overlaid on the terminal is a text editor window titled 'Nama Pai'. It contains the following text:

```
Nama : Pandia Arya Brata
Nim  |: 2311102076
```

The text editor window also shows a status bar at the bottom: 'Ln 2, Col 6 | 42 characters | 100% | Window | UTF-8'.

Deskripsi program :

Program di atas menggunakan fungsi map untuk membuat sebuah map dengan tipe data `<string, int>`. Kemudian program memasukkan beberapa data ke dalam map, mengakses data dalam map, mengubah data dalam map, dan menghapus data dalam map.

Perbedaan antara array dan map adalah bahwa array merupakan struktur data yang dapat menampung nilai-nilai dengan berbagai tipe data seperti string, number, boolean, objek, dan lain-lain. Sedangkan map merupakan fungsi yang biasanya digunakan untuk memproses array dan menghasilkan array baru dengan nilai yang sudah diproses. Dengan menggunakan map, kita dapat memproses setiap elemen dalam sebuah array tanpa melakukan perulangan dan melakukan operasi secara langsung ke dalam setiap elemen. Sedangkan array memerlukan perulangan untuk melakukan operasi pada setiap elemen.

KESIMPULAN

Tipe data primitif adalah jenis data dasar yang tersedia dalam bahasa pemrograman, seperti integer, float, boolean, dan character. Tipe data primitif ini digunakan untuk menyimpan data sederhana dan umumnya dikelompokkan berdasarkan jenis data yang mereka wakili. Tipe data primitif biasanya memiliki ukuran tetap dan nilai default yang ditentukan.

Tipe data abstrak adalah jenis data yang digunakan untuk mewakili suatu konsep atau objek dalam program, seperti stack, queue, dan list. Tipe data abstrak ini memiliki operasi dan perilaku khusus yang terkait dengan konsep atau objek yang mereka wakili, namun implementasinya tersembunyi dari pengguna. Pengguna hanya perlu menggunakan operasi dan perilaku yang didefinisikan untuk tipe data abstrak tanpa harus memahami bagaimana tipe data abstrak tersebut diimplementasikan.

Tipe data koleksi adalah jenis data yang digunakan untuk menyimpan kumpulan data, seperti array, list, set, dan map. Tipe data koleksi ini memungkinkan pengguna untuk menyimpan dan mengakses banyak data dalam satu waktu. Tipe data koleksi umumnya terdiri dari banyak elemen atau item yang dapat diakses secara individu melalui indeks atau kunci. Beberapa tipe data koleksi juga memiliki operasi khusus yang dapat digunakan untuk memanipulasi elemen dalam koleksi.

E. 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>