

LAPORAN PRAKTIKUM

MODUL I TIPE DATA



**Disusun oleh:
Bayu Kuncoro Adi
NIM: 2311102031**

**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

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

A. 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 :

- Tipe data Primitif
- Tipe data Abstrak
- Tipe data Koleksi

1. 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 :

- Int : adalah tipe data yang digunakan untuk menyimpan bilangan bulat seperti 12, 1, 4, dan sebagainya.
- Float : tipe data yang digunakan untuk menyimpan bilangan desimal seperti 1.5, 2.1, 3.14, dan sebagainya.
- Char : berfungsi untuk menyimpan data berupa sebuah huruf. Biasanya digunakan untuk simbol seperti A, B, C dan seterusnya
- 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 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.

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, dan di antaranya adalah:

- 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.
- 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
- 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>
#include
<iomanip> using
namespace std;

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

    // It allows user to enter operator i.e. +, -,
    // *, /cout << "Enter operator (+, -, *, /):
    ";
    cin >> op;

    // It allows user to enter the
    // operandscout << "Enter two
    // operands: ";
    cin >> num1 >> num2;

    // Switch statement
    switch (op) {
        // If user
        // enters +case
        // '+':
        //     cout << "Result: " << num1 +
        //     num2;break;
        // If user
        // enters -case '-'
        // '-':
        //     cout << "Result: " << num1 -
        //     num2;break;
        // If user
        // enters *case
        // '*':
        //     cout << "Result: " << num1 *
        //     num2;break;
```

```

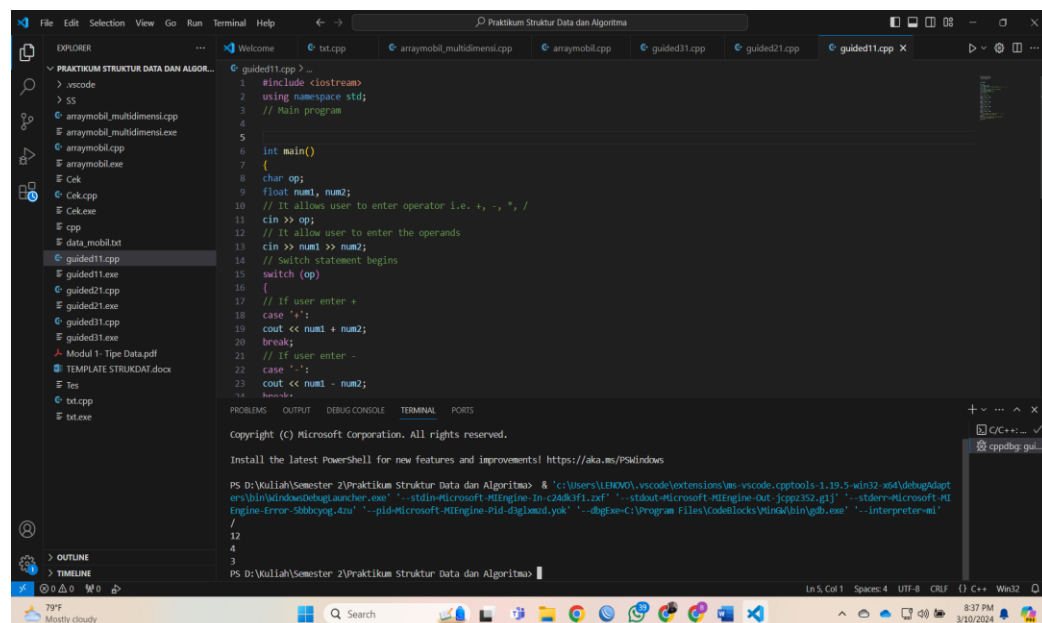
// If user
enters /case
'/':

    if (num2 != 0) {
        cout << "Result: " << fixed << setprecision(2)
        <<
num1 / num2;
    } else {
        cout << "Error! Division by zero is not
        allowed.";
    }
    break;
// If the operator is other than +, -, * or /,
// error message will display
default:
    cout << "Error! Operator is not correct";
} // switch statement ends

return 0;
}

```

Screenshoot program



Deskripsi program

Program C++ diatas terdapat switch, float, break. Program diatas berjalan seperti sebuah kalkulator, dengan pengguna menginputkan +,-,*,/ lalu menginputkan bilangan yang akan di operasikan. Dan menghasilkan sebuah ouput hasil dari perhitungan tersebut.

2. Guided 2

```
#include
<stdio.h>
#include
<string.h>

//Struct

struct Mahasiswa

{

    char
    name[50];
    char
    address[100]
    ;int age;

};

int main()

{

// menggunakan struct
struct Mahasiswa mhs1, mhs2;

// mengisi nilai ke struct strcpy(mhs1.name, "Dian");
strcpy(mhs1.address, "Mataram");mhs1.age = 22;
strcpy(mhs2.name, "Bambang");

strcpy(mhs2.address, "Surabaya");mhs2.age = 23;

// mencetak isi struct printf("## Mahasiswa 1 ##\n");
printf("Nama: %s\n", mhs1.name);

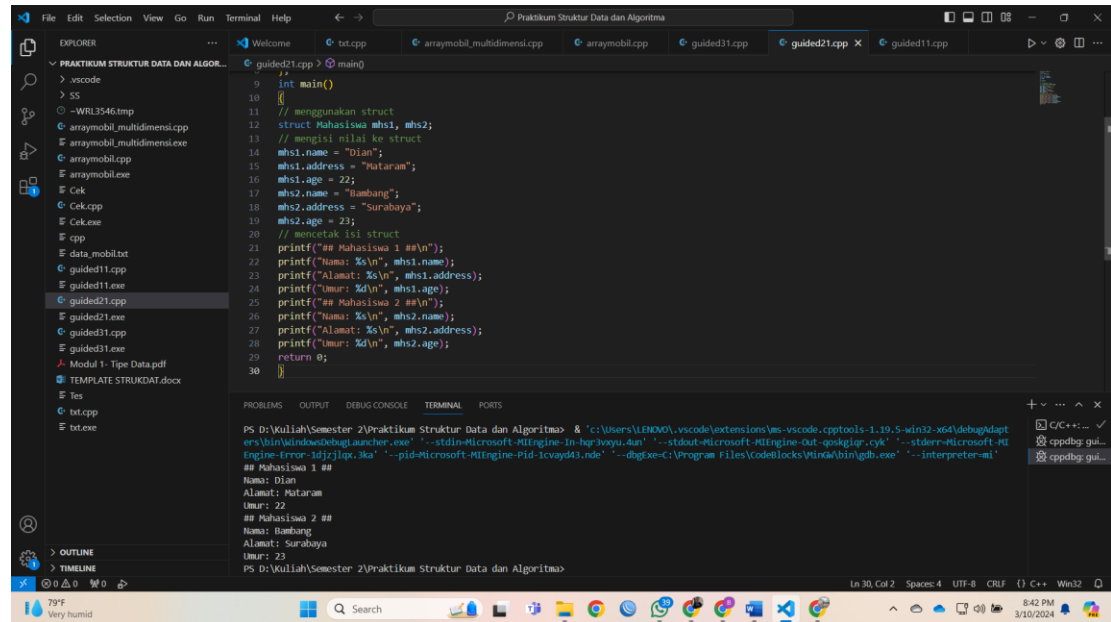
printf("Alamat: %s\n", mhs1.address);printf("Umur: %d\n",
mhs1.age); printf ("\n");

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



```
9  int main()
10  {
11      // menggunakan struct
12      struct Mahasiswa mhs1, mhs2;
13      // mengisi nilai ke struct
14      mhs1.name = "Dian";
15      mhs1.address = "Mataram";
16      mhs1.age = 22;
17      mhs2.name = "Bangang";
18      mhs2.address = "Surabaya";
19      mhs2.age = 23;
20      // mencetak isi struct
21      printf("## Mahasiswa 1 ##\n");
22      printf("Nama: %s\n", mhs1.name);
23      printf("Alamat: %s\n", mhs1.address);
24      printf("Umur: %d\n", mhs1.age);
25      printf("## Mahasiswa 2 ##\n");
26      printf("Nama: %s\n", mhs2.name);
27      printf("Alamat: %s\n", mhs2.address);
28      printf("Umur: %d\n", mhs2.age);
29      return 0;
30  }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\Kuliah\Semester 2\Praktikum Struktur Data dan Algoritma> & 'c:\Users\LEKNO\vscode\extensions\ms-vscode.cpptools-1.19.5-win32-x64\debuggadp
ers\bin\WindowsDebugLauncher.exe' '--stdin-microsoft-qt-engine-in-hq\vsyu4un' '--stdout-microsoft-qt-engine-out-qskqlp,cy' '--stderr-microsoft-qt
engine-errord-jlqpx_ika' '--pid-microsoft-qt-engine-Pid-1cvayd41.nde' '--dbgexe=c:\Program Files\Codeblocks\Win64\bin\gdb.exe' '--interpreter=mi'

## Mahasiswa 1 ##
Nama: Dian
Alamat: Mataram
Umur: 22

## Mahasiswa 2 ##
Nama: Bangang
Alamat: Surabaya
Umur: 23

PS D:\Kuliah\Semester 2\Praktikum Struktur Data dan Algoritma>
```

Deskripsi Program

Program C++ diatas terdapat intger, printf, dan char. Program diatas menghasilkan output Nama mahasiswa, Alamat, dan Umur. Terdapat sebuah array yang berisi data mahasiswa.

3. Guided 3

```
#include
<iostream>
#include
<array> 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;

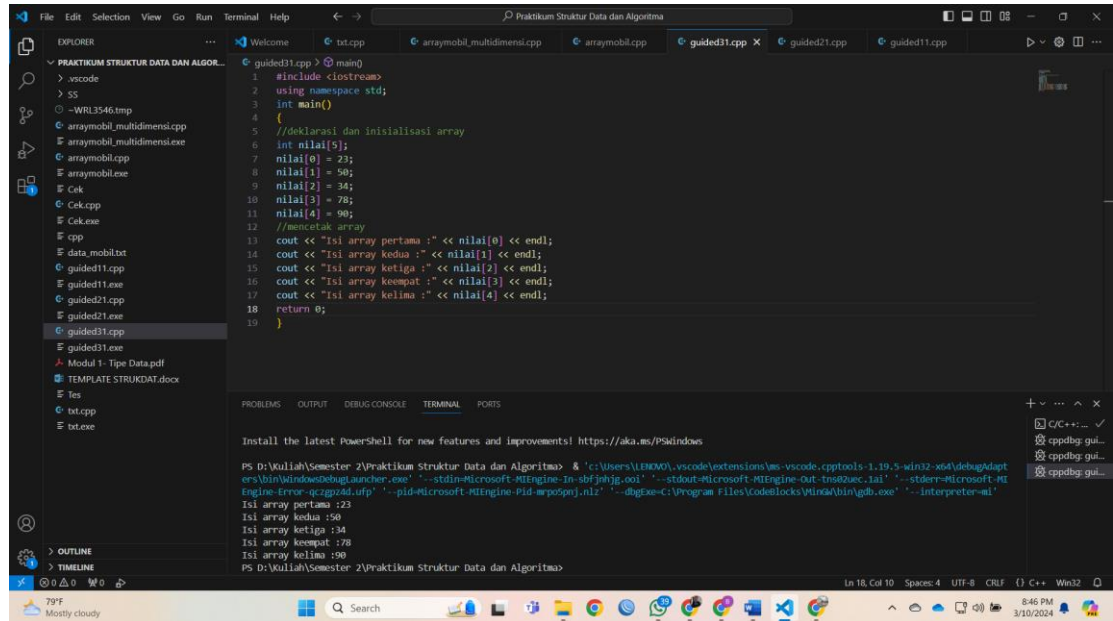
    // Mencetak array dengan tab

    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



The screenshot shows a Visual Studio Code editor window with a C++ program. The Explorer panel on the left shows a project named 'PRAKTIKUM STRUKTUR DATA DAN ALGORITMA'. The main editor displays the file 'guided31.cpp' with the following code:

```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     //deklarasi dan inisialisasi array
6     int nilai[5];
7     nilai[0] = 23;
8     nilai[1] = 50;
9     nilai[2] = 34;
10    nilai[3] = 78;
11    nilai[4] = 90;
12    //mencetak array
13    cout << "Isi array pertama : " << nilai[0] << endl;
14    cout << "Isi array kedua : " << nilai[1] << endl;
15    cout << "Isi array ketiga : " << nilai[2] << endl;
16    cout << "Isi array keempat : " << nilai[3] << endl;
17    cout << "Isi array kelima : " << nilai[4] << endl;
18    return 0;
19 }
```

The TERMINAL panel at the bottom shows the output of the program:

```
PS D:\Vulliah\Semester 2\Praktikum Struktur Data dan Algoritma>
Isi array pertama :23
Isi array kedua :50
Isi array ketiga :34
Isi array keempat :78
Isi array kelima :90
PS D:\Vulliah\Semester 2\Praktikum Struktur Data dan Algoritma>
```

Deskripsi Program

Program C++ diatas terdapat integer, array, dan cout. Dalam program diatas menghasilkan output dari array yang dimasukkan pengguna secara manual atau langsung ada pada program tersebut. Program tersebut menghasilkan isi array pertama sampai kelima.

LATIHAN KELAS - UNGUIDED

1. Unguided 1

Source code

```
#include <iostream>

// Fungsi untuk menjumlahkan dua bilangan bulat
int tambah(int a, int b) {
    return a + b;
}

// Fungsi untuk mengalikan dua bilangan pecahan
float kali(float a, float b) {
    return a * b;
}

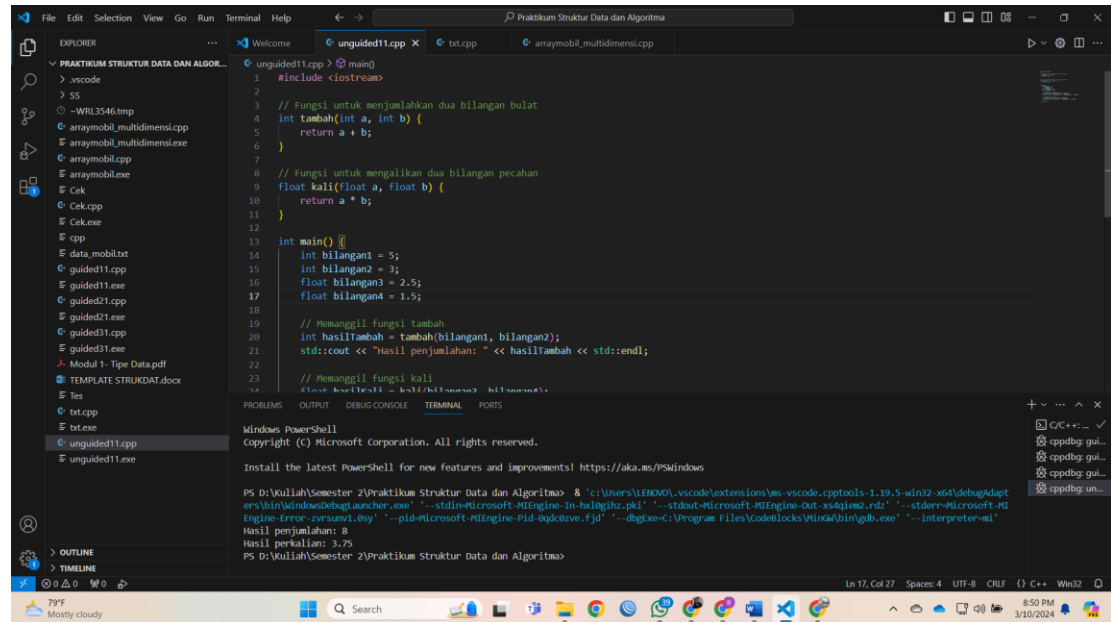
int main() {
    int bilangan1 = 5;
    int bilangan2 = 3;
    float bilangan3 = 2.5;
    float bilangan4 = 1.5;

    // Memanggil fungsi tambah
    int hasilTambah = tambah(bilangan1, bilangan2);
    std::cout << "Hasil penjumlahan: " << hasilTambah <<
std::endl;

    // Memanggil fungsi kali
    float hasilKali = kali(bilangan3, bilangan4);
    std::cout << "Hasil perkalian: " << hasilKali << std::endl;

    return 0;
}
```

Screenshoot program



The screenshot shows the Visual Studio Code interface with a C++ file named 'unguided11.cpp' open. The code defines two functions: 'tambah' for adding two integers and 'kali' for multiplying two floats. The 'main' function calls these functions with specific values and prints the results. The terminal at the bottom shows the execution of the program, displaying the output: 'Hasil penjumlahan: 8' and 'Hasil perkalian: 3.75'.

```
1 // unguided11.cpp > main()
2 #include <iostream>
3
4 // Fungsi untuk menjumlahkan dua bilangan bulat
5 int tambah(int a, int b) {
6     return a + b;
7 }
8
9 // Fungsi untuk mengalikan dua bilangan pecahan
10 float kali(float a, float b) {
11     return a * b;
12 }
13
14 int main() {
15     int bilangan1 = 5;
16     int bilangan2 = 3;
17     float bilangan3 = 2.5;
18     float bilangan4 = 1.5;
19
20     // Memanggil fungsi tambah
21     int hasilTambah = tambah(bilangan1, bilangan2);
22     std::cout << "Hasil penjumlahan: " << hasilTambah << std::endl;
23
24     // Memanggil fungsi kali
25     float hasilKali = kali(bilangan3, bilangan4);
26     std::cout << "Hasil perkalian: " << hasilKali << std::endl;
27 }
```

Terminal Output:

```
PS D:\Kuliah\Semester 2\Praktikum Struktur Data dan Algoritma>
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\Kuliah\Semester 2\Praktikum Struktur Data dan Algoritma> & 'C:\Users\LEKNO\.vscode\extensions\ms-vscode.cpptools-1.19.5-win32-x64\debugAdapt
ers\bin\windowsdebuglauncher.exe' '--stdin=Microsoft-MiEngine-In-hu0gihz.pki' '--stdout=Microsoft-MiEngine-Out-xsqilem2.rdz' '--stderr=Microsoft-Mi
Engine-Error-zvrsunw1.bsy' '--pid=Microsoft-MiEngine-Pid-0gd0rve.fjd' '--dbgexe=C:\Program Files\CodeBlocks\Win64\bin\gdb.exe' '--interpreter=mi'
Hasil penjumlahan: 8
Hasil perkalian: 3.75
PS D:\Kuliah\Semester 2\Praktikum Struktur Data dan Algoritma>
```

Penjelasan Program

DefinisiProgram:

- 'Tambah' dan 'kali' adalah dua tugas program; masing-masing memiliki dua argumen dan mengembalikan hasil penjumlahan dan perkalian.
- Kita mendeklarasikan beberapa variabel bertipe data primitif di dalam fungsi 'main': dua integer ('bilangan1' dan 'bilangan2') dan dua float ('bilangan3' dan 'bilangan4').
- Setelah itu, variabel-variabel tersebut digunakan sebagai argumen untuk memanggil fungsi 'tambah' dan 'kali'

Kesimpulan Program

Kesimpulan dari materi yang termasuk dalam kategori data primitif:

- Dalam program C++, tipe data primitif seperti int dan float digunakan untuk menyimpan data dasar. Tipe data int dapat menyimpan bilangan bulat, sedangkan tipe data float dapat menyimpan bilangan pecahan.
 - Kedua jenis data primitif ini memiliki batasan untuk menunjukkan nilai. Misalnya, tipe data int hanya dapat menunjukkan nilai dalam batas waktu tertentu.
 - Pemilihan tipe data yang tepat sangat penting dalam pemrograman untuk memastikan penggunaan memori yang efisien dan untuk menghindari kesalahan dalam perhitungan atau manipulasi data.
 - Kita melihat bagaimana program di atas menggunakan tipe data primitif untuk melakukan operasi sederhana seperti penjumlahan dan perkalian, yang merupakan operasi pemrograman dasar.
-

2. Unguided 2

Class :

- Class dalam C++ digunakan untuk mendefinisikan sebuah tipe data baru yang memiliki data anggota (variabel) dan fungsi anggota (metode) yang terkait. Ini memungkinkan konsep pemrograman berorientasi objek di mana Anda dapat menggabungkan data dan fungsi yang beroperasi pada data tersebut dalam satu unit yang disebut sebagai objek.
- Class juga memungkinkan untuk menerapkan konsep enkapsulasi, di mana Anda dapat menyembunyikan detail implementasi di dalam class dan hanya mengekspos antarmuka publik untuk berinteraksi dengan objek.
- Selain itu, class juga mendukung konsep inheritance, di mana class baru dapat dibangun berdasarkan class yang sudah ada dengan mewarisi sifat-sifat dan perilaku yang ada.

Contoh Program :

```
#include <iostream>

// Definisi class
class Mobil {
private:
    std::string merek;
    std::string model;
    int tahun;

public:
    // Constructor
    Mobil(std::string mrk, std::string mdl, int thn) {
        merek = mrk;
        model = mdl;
        tahun = thn;
    }

    // Method untuk menampilkan informasi mobil
```

```
    void tampilkanInfo() {  
        std::cout << "Mobil: " << merek << " " << model << " (" << tahun << ")" << std::endl;  
    }  
};  
  
int main() {  
    // Membuat objek dari class Mobil  
    Mobil mobil1("Toyota", "Camry", 2020);  
  
    // Memanggil method tampilkanInfo dari objek  
    mobil1.tampilkanInfo();  
  
    return 0;  
}
```


Struct :

- Struktur dalam C++ mengelompokkan data dalam satu unit, seperti class. Namun, perbedaan utamanya adalah bahwa anggota struct secara default bersifat publik, yang berarti mereka dapat diakses dari luar struct tanpa perlu menggunakan fungsi anggota.
- Struktur sering digunakan untuk menyimpan sekumpulan data terkait yang lebih sederhana tanpa metode atau operasi yang kompleks,
- Dapat juga mendefinisikan fungsi di dalam struktur jika diperlukan.

Contoh Program :

```
#include <iostream>

// Definisi struct
struct Mahasiswa {
    std::string nama;
    int umur;
    float ipk;
};

int main() {
    // Mendefinisikan objek dari struct Mahasiswa
    Mahasiswa mhs1;
    mhs1.nama = "John Doe";
    mhs1.umur = 20;
    mhs1.ipk = 3.75;

    // Menampilkan informasi mahasiswa
    std::cout << "Nama: " << mhs1.nama << std::endl;
    std::cout << "Umur: " << mhs1.umur << std::endl;
    std::cout << "IPK: " << mhs1.ipk << std::endl;

    return 0;
}
```

3. Unguided 3

Fungsi map dalam C++ adalah cara yang bagus untuk memetakan kunci ke nilai. Fungsi map adalah bagian dari Library Template Standar, atau STL, yang memungkinkan implementasi struktur data penyimpanan kunci-nilai yang disusun secara terurut berdasarkan kunci.

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

int main() {
    // Mendeklarasikan map dengan kunci bertipe string dan nilai
    // bertipe int
    std::map<std::string, int> umur;

    // Menambahkan pasangan kunci-nilai ke dalam map
    umur["Alice"] = 25;
    umur["Bob"] = 30;
    umur["Charlie"] = 20;

    // Mengakses nilai dengan menggunakan kunci
    std::cout << "Umur Alice: " << umur["Alice"] << std::endl;
    std::cout << "Umur Bob: " << umur["Bob"] << std::endl;
    std::cout << "Umur Charlie: " << umur["Charlie"] << std::endl;

    return 0;
}
```

Penjelasan Program

Untuk menyimpan umur, program mendeklarasikan sebuah map dengan kunci bertipe string dan nilai bertipe int. Selanjutnya, ia menggunakan operator [] untuk menambahkan beberapa pasangan kunci-nilai ke dalam map. Setelah itu, kunci yang tepat digunakan untuk mengakses nilai-nilai tersebut. Program menghitung umur dari tiga orang yang terdaftar di map.

Perbedaan Map dan Array

Array :

- Merupakan struktur data linear yang menyimpan elemen dalam urutan tertentu. Ini memiliki indeks berurutan yang dimulai dengan nol.
- Array ukuran harus ditetapkan pada saat deklarasi dan tidak boleh diubah selama runtime. Indeks digunakan untuk mengakses elemen-elemen array.
- Jika Anda memiliki kumpulan data dengan indeks berurutan dan ukuran tetap, ini akan sangat membantu.

Map :

- Merupakan struktur data asosiatif di mana pasangan kunci-nilai disimpan.
 - Tidak ada indeks yang disusun secara berurutan; sebaliknya, elemen disimpan berdasarkan kunci-kuncinya.
 - Ukuran map dapat berubah selama runtime karena ia dapat berubah sesuai dengan jumlah pasangan nilai kunci yang ditambahkan atau dihapus.
 - Kunci digunakan untuk mengakses nilai-nilai map.
 - Jika Anda perlu menghubungkan kunci yang berbeda ke nilai tertentu, ini cocok untuk Anda.
-

BAB IV

KESIMPULAN

Tipe data adalah pengklasifikasian data berdasarkan jenisnya, dan terdiri dari tipe data primitif, tipe data abstrak, dan tipe data koleksi. Tipe data primitif seperti int, float, char, dan boolean telah ditentukan oleh sistem dan digunakan untuk menyimpan data dasar seperti bilangan bulat, desimal, karakter, dan nilai boolean. Tipe data abstrak, seperti yang diwakili oleh fitur class dalam pemrograman berorientasi objek, dibentuk oleh pemrogram dan dapat menggabungkan berbagai tipe data dan fungsi di dalamnya. Tipe data koleksi, seperti array, vector, dan map, digunakan untuk mengelompokkan dan menyimpan beberapa nilai atau objek secara bersamaan, memungkinkan pengelolaan data yang terstruktur dengan berbagai fitur tambahan seperti alokasi memori dinamis dan indeks non-integer.

Dalam program-program yang disajikan, kita dapat melihat penerapan berbagai konsep dasar dalam pemrograman C++. Pada program dengan fungsi primitif, penggunaan tipe data seperti int dan float untuk operasi matematika dasar seperti penjumlahan dan perkalian mengilustrasikan penggunaan variabel dengan tipe data primitif. Selanjutnya, implementasi program dengan class dan struct memperlihatkan bagaimana kita dapat mengorganisir data dan fungsi terkait dalam satu unit yang disebut class atau struct. Perbedaan aksesibilitas default antara class dan struct juga dijelaskan, dimana class memiliki aksesibilitas default private sedangkan struct memiliki aksesibilitas default public.

Selain itu, program dengan fungsi map mengeksplorasi fitur STL untuk membuat pemetaan kunci-nilai menggunakan map. Pemahaman tentang cara menambahkan pasangan kunci-nilai ke dalam map dan mengakses nilai-nilai tersebut menyoroti fleksibilitas dan kekuatan penggunaan struktur data yang lebih kompleks seperti map dalam pengembangan program. Dari ketiga program tersebut, kita dapat melihat bagaimana konsep-konsep yang berbeda dapat diterapkan untuk memenuhi kebutuhan pemrograman yang beragam, menunjukkan keluwesan bahasa pemrograman C++ dalam menangani berbagai jenis masalah pemrograman.

DAFTAR PUSTAKA

Buku "C++ Primer" oleh Stanley B. Lippman, Josée Lajoie, dan Barbara E. Moo

c-classes-and-objects diakses dari <https://www.geeksforgeeks.org/c-classes-and-objects/>

c-structures diakses dari <https://www.geeksforgeeks.org/c-structures/>

pp-programming/class-objects diakses dari <https://www.programiz.com/cpp-programming/class-objects>

cpp-programming/structure diakses dari <https://www.programiz.com/cpp-programming/structure>

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

std::map diakses dari <https://en.cppreference.com/w/cpp/container/map>

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

