

LAPORAN PRAKTIKUM

MODUL I TIPE DATA



**Disusun oleh:
Imelda Fajar Awalina Crisyanti
NIM: 2311102004**

**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

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 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(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](https://learn.microsoft.com/en-us/cpp/faq/struct-vs-class) 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

BAB III

GUIDED

1. Guided 1

Source code

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

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;
}
```

```
}
```

Screenshoot program

```
5 + 3  
Output: 8
```

Deskripsi program

Program ini merupakan sebuah kalkulator sederhana yang memungkinkan pengguna untuk melakukan operasi aritmatika dasar (penjumlahan, pengurangan, perkalian, dan pembagian) antara dua bilangan. Pengguna diminta untuk memasukkan operator aritmatika yang diinginkan dan dua bilangan yang akan dioperasikan, kemudian program akan menghitung hasilnya sesuai dengan operator yang dimasukkan. Jika operator yang dimasukkan tidak valid, program akan menampilkan pesan kesalahan.

2. Guided 2

```
#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;

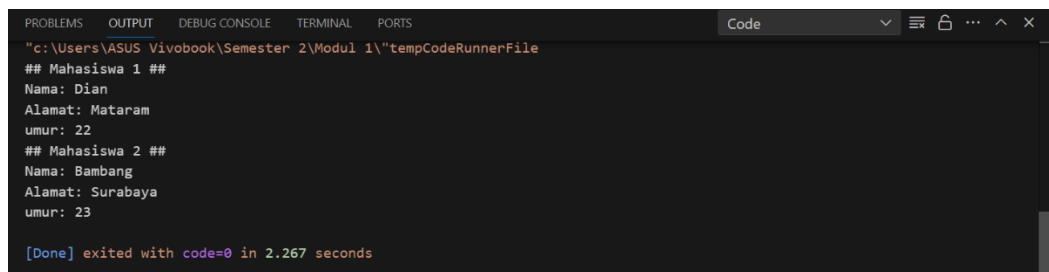
    printf ("## Mahasiswa 1 ##\n");
```

```

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



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code
"c:\Users\ASUS Vivobook\Semester 2\Modul 1\tempCodeRunnerFile"
## Mahasiswa 1 ##
Nama: Dian
Alamat: Mataram
umur: 22
## Mahasiswa 2 ##
Nama: Bambang
Alamat: Surabaya
umur: 23
[Done] exited with code=0 in 2.267 seconds

```

Deskripsi program

Program ini merupakan contoh penggunaan struktur dalam bahasa pemrograman C. Struktur Mahasiswa didefinisikan dengan tiga anggota yaitu name, address, dan age untuk merepresentasikan data seorang mahasiswa.

Dalam fungsi main(), dua variabel mhs1 dan mhs2 dideklarasikan sebagai objek dari struktur Mahasiswa. Kemudian, nilai-nilai atribut untuk masing-masing mahasiswa diatur.

Selanjutnya, program mencetak detail masing-masing mahasiswa ke layar menggunakan fungsi printf. Setiap detail ditampilkan dengan menggunakan format yang sesuai, seperti nama dengan %s untuk string dan umur dengan %d untuk integer.

Output dari program akan menampilkan detail dua mahasiswa yang telah diatur sebelumnya, termasuk nama, alamat, dan umur masing-masing mahasiswa

3. Guided 3

```

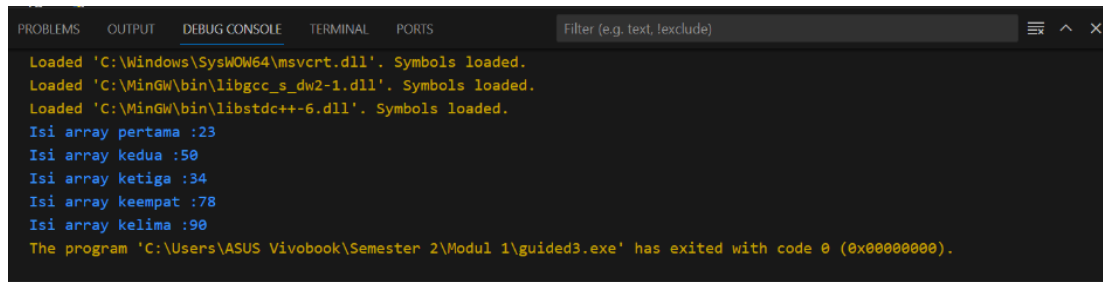
#include <iostream>
using namespace std;
int main()
{

```

```
int nilai[5];
nilai[0] = 23;
nilai[1] = 50;
nilai[2] = 34;
nilai[3] = 78;
nilai[4] = 90;

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

A screenshot of a terminal window showing the output of a C++ program. The window has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The OUTPUT tab is active, showing the following text: Loaded 'C:\Windows\SysWOW64\msvcrt.dll'. Symbols loaded. Loaded 'C:\MinGW\bin\libgcc_s_dw2-1.dll'. Symbols loaded. Loaded 'C:\MinGW\bin\libstdc++-6.dll'. Symbols loaded. Isi array pertama :23 Isi array kedua :50 Isi array ketiga :34 Isi array keempat :78 Isi array kelima :90 The program 'C:\Users\ASUS Vivobook\Semester 2\Modul 1\guided3.exe' has exited with code 0 (0x00000000).

```
Loaded 'C:\Windows\SysWOW64\msvcrt.dll'. Symbols loaded.
Loaded 'C:\MinGW\bin\libgcc_s_dw2-1.dll'. Symbols loaded.
Loaded 'C:\MinGW\bin\libstdc++-6.dll'. Symbols loaded.
Isi array pertama :23
Isi array kedua :50
Isi array ketiga :34
Isi array keempat :78
Isi array kelima :90
The program 'C:\Users\ASUS Vivobook\Semester 2\Modul 1\guided3.exe' has exited with code 0 (0x00000000).
```

Deskripsi program

Program ini adalah contoh sederhana dari penggunaan array dalam bahasa C++. Array nilai dengan panjang 5 elemen digunakan untuk menyimpan beberapa nilai integer. Setiap elemen array diisi dengan nilai yang berbeda. Kemudian, program mencetak isi dari setiap elemen array ke layar menggunakan cout, dengan menampilkan pesan yang bersesuaian dengan indeks elemen array.

Output dari program ini akan menampilkan nilai dari setiap elemen array nilai,

LATIHAN KELAS - UNGUIDED

1. Unguided 1

Source code

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

float tambah(float num1, float num2) {
    return num1 + num2;
}

float kurang(float num1, float num2) {
    return num1 - num2;
}

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

    cout << "Masukkan operator (+, -, ): ";
    cin >> op;

    cout << "Masukkan dua angka: ";
    cin >> num1 >> num2;

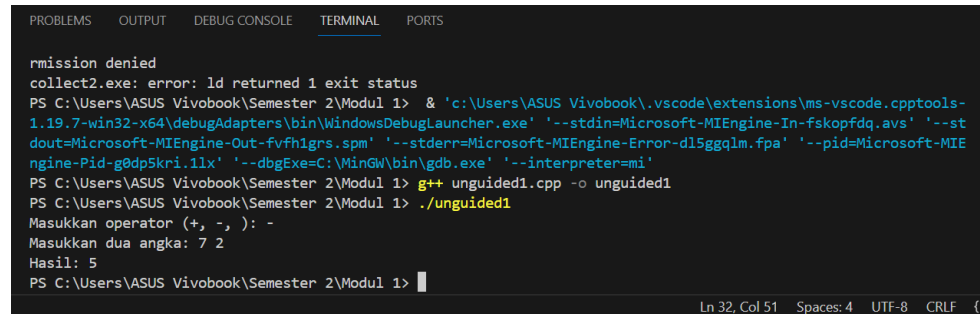
    switch (op) {
        case '+':
            cout << "Hasil: " << tambah(num1, num2);
            break;
        case '-':
            cout << "Hasil: " << kurang(num1, num2);
            break;

        default:
            cout << "Error! Operator tidak benar";
    }

    return 0;
}
```

```
}
```

Screenshoot program



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

rmission denied
collect2.exe: error: ld returned 1 exit status
PS C:\Users\ASUS Vivobook\Semester 2\Modul 1> & 'c:\Users\ASUS Vivobook\.vscode\extensions\ms-vscode.cpptools-1.19.7-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-fskopfdq.avs' '--stdout=Microsoft-MIEngine-Out-fvfh1grs.spm' '--stderr=Microsoft-MIEngine-Error-d15ggqlm.fpa' '--pid=Microsoft-MIEngine-Pid-g8dp5kri.1lx' '--dbgExe=C:\WinGW\bin\gdb.exe' '--interpreter=mi'
PS C:\Users\ASUS Vivobook\Semester 2\Modul 1> g++ unguided1.cpp -o unguided1
PS C:\Users\ASUS Vivobook\Semester 2\Modul 1> ./unguided1
Masukkan operator (+, -, *): -
Masukkan dua angka: 7 2
Hasil: 5
PS C:\Users\ASUS Vivobook\Semester 2\Modul 1> |
```

Deskripsi program

Program di atas adalah sebuah kalkulator sederhana yang memungkinkan pengguna untuk melakukan operasi penambahan dan pengurangan antara dua bilangan float. Berikut adalah deskripsi singkat dari program tersebut:

1. Fungsi tambah(float num1, float num2): Fungsi ini menerima dua parameter float num1 dan num2, dan mengembalikan hasil penambahan dari kedua parameter tersebut.
2. Fungsi kurang(float num1, float num2): Fungsi ini menerima dua parameter float num1 dan num2, dan mengembalikan hasil pengurangan num1 dikurangi num2.
3. Fungsi main(): Fungsi utama program dimulai di sini. Pertama, variabel op, num1, dan num2 dideklarasikan. Kemudian, pengguna diminta untuk memasukkan operator (op) dan dua angka (num1 dan num2) melalui input menggunakan cin.
4. Switch Statement: Program menggunakan switch untuk mengevaluasi nilai dari variabel op. Bergantung pada nilai op, program akan memanggil fungsi tambah() atau kurang() untuk melakukan operasi penambahan atau pengurangan.
5. Output: Hasil operasi aritmatika akan ditampilkan ke layar menggunakan cout.
6. Default Case: Jika operator yang dimasukkan tidak valid (bukan '+' atau '-'), program akan menampilkan pesan kesalahan.

2. Unguided 2

Struct

Struct adalah kumpulan beberapa tipe data yang berbeda yang menjadi satu kesatuan. Struct dalam C++ biasanya digunakan untuk mengelompokkan beberapa variabel yang berhubungan menjadi satu entitas yang lebih besar. Dalam struct, semua anggota (variabel) secara default bersifat publik. Ini berarti bahwa Anda dapat mengakses dan mengubah nilainya langsung dari luar struct. Berikut adalah contoh penggunaan struct:

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

struct Mahasiswa {
    string nama;
    int umur;
};

int main() {
    Mahasiswa mhs1;
    mhs1.nama = "Ali";
    mhs1.umur = 20;

    cout << "Nama: " << mhs1.nama << endl;
    cout << "Umur: " << mhs1.umur << endl;

    return 0;
}
```

Class

Class adalah struktur data yang lebih kompleks dibandingkan dengan struct. Class memungkinkan Anda untuk mendefinisikan tipe data sendiri dengan menggabungkan data bersama dengan fungsi-fungsi yang beroperasi pada data tersebut. Secara default, semua anggota class bersifat private, yang berarti mereka hanya dapat diakses dari dalam class itu sendiri. Anda dapat mendefinisikan fungsi-fungsi publik (atau bersifat privat) untuk memberikan akses ke anggota-anggota ini dari luar class. Berikut adalah contoh penggunaan class:

```

#include <iostream>
using namespace std;

class Mahasiswa {
private:
    string nama;
    int umur;

public:
    void setNama(string nm) {
        nama = nm;
    }

    void setUmur(int um) {
        umur = um;
    }

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

int main() {
    Mahasiswa mhs1;
    mhs1.setNama("Ali");
    mhs1.setUmur(20);
    mhs1.display();

    return 0;
}

```

3. Unguided 3

Program Menggunakan Fungsi map

```

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

int main() {

```

```

// Membuat map untuk menyimpan nama dan umur
map<string, int> umur;

// Menambahkan data ke dalam map
umur["Ali"] = 20;
umur["Budi"] = 22;
umur["Charlie"] = 25;

// Menampilkan umur
cout << "Umur Ali: " << umur["Ali"] << endl;
cout << "Umur Budi: " << umur["Budi"] << endl;
cout << "Umur Charlie: " << umur["Charlie"] << endl;

return 0;
}

```

Dalam program di atas, kita menggunakan **map** untuk menyimpan data pasangan kunci-nilai, di mana kunci adalah nama dan nilai adalah umur. Setiap elemen dalam map adalah pasangan yang terdiri dari sebuah kunci dan nilai yang sesuai. Kunci harus unik dalam map. Program di atas menambahkan beberapa entri ke dalam map, kemudian menampilkan umur dari beberapa orang yang telah ditentukan sebelumnya.

Perbedaan antara Array dan Map

Array:

1. Array adalah struktur data linear yang berisi kumpulan elemen yang tipe datanya sama dan terletak pada alamat memori yang berurutan.
2. Elemen-elemen dalam array dapat diakses menggunakan indeks.
3. Ukuran array biasanya tetap dan harus ditentukan pada saat deklarasi.

Map:

4. Map adalah struktur data asosiatif yang mengaitkan sebuah kunci dengan sebuah nilai.
5. Map tidak memiliki batasan pada jumlah elemen yang dapat disimpan, dan ukurannya dapat berubah selama program berjalan.
6. Elemen-elemen dalam map tidak memiliki urutan tertentu dan diatur berdasarkan kunci.
7. Kunci dalam map harus unik.

Perbedaan utama antara array dan map adalah bahwa array menyimpan elemen dalam urutan tertentu dan diakses dengan indeks numerik, sedangkan map menyimpan elemen dalam pasangan kunci-nilai, di mana setiap nilai memiliki kunci yang unik. Selain itu, ukuran array tetap saat dideklarasikan, sedangkan ukuran map bisa berubah sesuai dengan kebutuhan.

BAB IV

KESIMPULAN

Laporan praktikum ini membantu dalam memahami konsep dasar mengenai tipe data dalam pemrograman. Tipe data merupakan klasifikasi nilai yang dapat disimpan dalam variabel atau digunakan dalam operasi.

Terdapat beberapa jenis tipe data yang diajarkan dalam praktikum ini, seperti tipe data numerik (integer, float), tipe data string, dan tipe data boolean. Setiap tipe data memiliki karakteristik dan penggunaan yang berbeda-beda.

Dalam pemrograman, pemilihan tipe data yang tepat sangat penting untuk memastikan keakuratan dan efisiensi program. Laporan ini memberikan pemahaman yang baik tentang bagaimana memilih tipe data yang sesuai dengan kebutuhan program.

Modul ini merupakan langkah awal dalam mempelajari pemrograman dan merupakan fondasi penting untuk memahami konsep yang lebih kompleks di masa depan. Praktikum ini memberikan pemahaman yang solid tentang dasar-dasar tipe data, yang merupakan pondasi penting bagi pengembangan keterampilan pemrograman.