

LAPORAN PRAKTIKUM

MODUL I

TIPE DATA PRIMITIF, TIPE DATA ABSTRAK DAN TIPE DATA KOLEKSI



Disusun Oleh:

Muhammad Raafi Al Hafiidh

2311102070

Dosen:

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

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

INSTITUT TEKNOLOGI TELKOM PURWOKERTO

2023

DASAR TEORI

Tipe Data Primitif

Tipe data primitif adalah jenis data dasar yang disediakan oleh bahasa pemrograman. Tipe data ini tidak dapat diubah atau diurai lebih lanjut. Berikut adalah beberapa contoh tipe data primitif:

- Bilangan bulat: byte, short, int, long
- Bilangan pecahan: float, double
- Karakter: char
- Nilai boolean: true, false

Tipe data primitif memiliki karakteristik sebagai berikut:

- Memiliki ukuran memori yang tetap
- Memiliki operasi yang sudah ditentukan
- Tidak dapat diubah atau diurai lebih lanjut

Tipe Data Abstrak

Tipe data abstrak adalah jenis data yang mendefinisikan sekumpulan operasi dan properti, tetapi tidak menentukan bagaimana operasi tersebut diimplementasikan. Tipe data abstrak memungkinkan programmer untuk fokus pada apa yang ingin dilakukan dengan data, tanpa harus khawatir tentang bagaimana data tersebut disimpan atau diproses.

Berikut adalah beberapa contoh tipe data abstrak:

- Stack
- Queue
- List
- Tree

Tipe data abstrak memiliki karakteristik sebagai berikut:

- Mendefinisikan sekumpulan operasi dan properti
- Tidak menentukan bagaimana operasi tersebut diimplementasikan
- Memungkinkan programmer untuk fokus pada apa yang ingin dilakukan dengan data

Tipe Data Koleksi

Tipe data koleksi adalah jenis data yang digunakan untuk menyimpan kumpulan data yang terstruktur. Tipe data koleksi menyediakan cara untuk menambahkan, menghapus, dan mencari data dalam kumpulan data.

Berikut adalah beberapa contoh tipe data koleksi:

- Array
- List
- Set
- Map

Tipe data koleksi memiliki karakteristik sebagai berikut:

- Digunakan untuk menyimpan kumpulan data yang terstruktur
- Menyediakan cara untuk menambahkan, menghapus, dan mencari data
- Memudahkan pengelolaan kumpulan data

LATIHAN DAN TUGAS

A. GUIDED

1. Guided 1

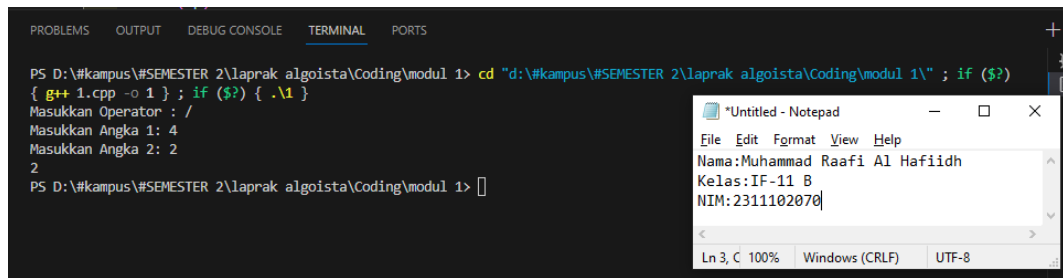
Source Code

```
#include <iostream>

using namespace std;

int main()
{
    char op;
    float num1, num2;
    // it allows user to enter operator i.e. +, -, *, /
    cout << "Masukkan Operator : ";
    cin >> op;
    // it allows user to enter the operands
    cout << "Masukkan Angka 1: ";
    cin >> num1;
    cout << "Masukkan Angka 2: ";
    cin >> num2;
    // switch statement begins
    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";
    } // switch statements ends
    return 0;
}
```

Screenshoot Program

The image shows a Windows environment with two windows. The background window is a terminal with a dark theme, showing the execution of a C++ program. The prompt is 'PS D:\#kampus\#SEMESTER 2\laprak algoista\Coding\modul 1>'. The user has run 'cd "d:\#kampus\#SEMESTER 2\laprak algoista\Coding\modul 1\" ; if (\$?) { g++ 1.cpp -o 1 } ; if (\$?) { .\1 }'. The program prompts 'Masukkan Operator : /', 'Masukkan Angka 1: 4', and 'Masukkan Angka 2: 2'. The prompt returns to 'PS D:\#kampus\#SEMESTER 2\laprak algoista\Coding\modul 1>'. The foreground window is a Notepad titled 'Untitled - Notepad', showing a menu bar (File, Edit, Format, View, Help) and text: 'Nama:Muhammad Raafi Al Hafidh', 'Kelas:IF-11 B', and 'NIM:2311102070'.

Deskripsi Program

Program di atas merupakan sebuah program kalkulator sederhana yang dapat menghitung operasi matematika dasar seperti penjumlahan, pengurangan, perkalian, dan pembagian. Program tersebut menggunakan pemilihan *switch-case* untuk menentukan operasi matematika mana yang akan dilakukan berdasarkan operator yang dimasukkan oleh pengguna.

Untuk jalannya program tersebut pertama-tama, program akan meminta pengguna untuk memasukkan operator dengan menampilkan pesan "Masukkan Operator : " dan menggunakan `cin` untuk membaca input dari pengguna dan menyimpannya pada variabel `op` yang bertipe `char`. Selanjutnya, program akan meminta pengguna memasukkan dua angka yang akan dioperasikan dengan menampilkan pesan "Masukkan Angka 1: " dan "Masukkan Angka 2: ", dan menggunakan `cin` untuk membaca input dari pengguna dan menyimpannya pada variabel `num1` dan `num2` yang bertipe `float`.

Setelah itu, program menggunakan struktur pemilihan *switch-case* untuk mengecek operator yang dimasukkan oleh pengguna dan melakukan operasi matematika yang sesuai. Jika operator adalah '+', program akan menjumlahkan `num1` dan `num2` menggunakan operator '+' dan menampilkan hasilnya dengan menggunakan `cout`. Jika operator adalah '-', program akan mengurangi `num2` dari `num1` menggunakan operator '-' dan menampilkan hasilnya. Jika operator adalah '*', program akan mengalikan `num1` dan `num2` menggunakan operator '*' dan menampilkan hasilnya. Jika operator adalah '/', program akan membagi `num1` dengan `num2` menggunakan operator '/' dan menampilkan hasilnya. Jika operator yang dimasukkan tidak valid, program akan menampilkan pesan "Error! operator is not correct".

2. Guided 2

Source Code

```
#include <stdio.h>

// membuat 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;

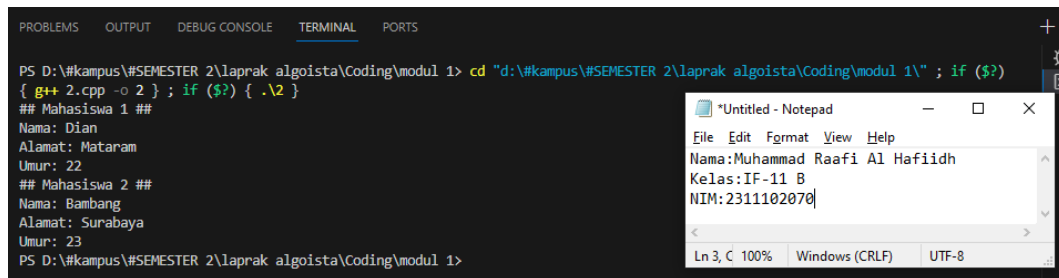
    mhs2.name = "Bambang";
    mhs2.address = "Surabaya";
    mhs2.age = 23;

    // 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;
}
```

Screenshoot Program



The image shows a screenshot of a C++ program being executed in a terminal window. The terminal output displays the program's execution, including the compilation command and the resulting output for two students. A Notepad window is also open, showing the output of the program.

```
PS D:\#kampus\#SEMESTER 2\laprak algoista\Coding\modul 1> cd "d:\#kampus\#SEMESTER 2\laprak algoista\Coding\modul 1\" ; if ($?) { g++ 2.cpp -o 2 } ; if ($?) { .\2 }  
## Mahasiswa 1 ##  
Nama: Dian  
Alamat: Mataram  
Umur: 22  
## Mahasiswa 2 ##  
Nama: Bambang  
Alamat: Surabaya  
Umur: 23  
PS D:\#kampus\#SEMESTER 2\laprak algoista\Coding\modul 1>
```

The Notepad window shows the following output:

```
File Edit Format View Help  
Nama:Muhammad Raafi Al Hafiidh  
Kelas:IF-11 B  
NIM:2311102070
```

Deskripsi Program

Program tersebut merupakan program yang menggunakan struct di bahasa pemrograman C++. *Struct* adalah tipe data yang memungkinkan pengguna untuk menggabungkan tipe data yang berbeda dalam satu variabel. Pada program tersebut, terdapat definisi *struct* Mahasiswa yang memiliki tiga anggota yaitu *name*, *address*, dan *age*.

Kemudian pada fungsi main, terdapat dua variabel *struct* Mahasiswa yaitu *mhs1* dan *mhs2*. Variabel-variabel tersebut kemudian diisi dengan nilai sesuai dengan definisi struct, dimana *mhs1* memiliki nilai *name* = "Dian", *address* = "Mataram", dan *age* = 22, sedangkan *mhs2* memiliki nilai *name* = "Bambang", *address* = "Surabaya", dan *age* = 23.

Setelah itu, nilai dari variabel-variabel tersebut dicetak menggunakan `printf()`. Masing-masing anggota *struct* dicetak menggunakan format string yang sesuai. Program tersebut berguna untuk menyimpan data mahasiswa dan memudahkan pengelolaan data tersebut.

3. Guided 3

Source Code

```
#include <iostream>

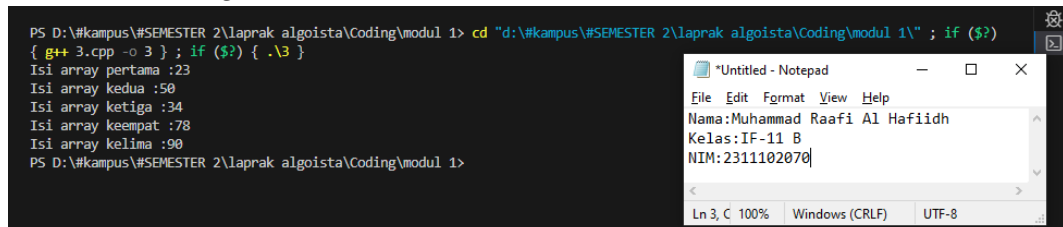
using namespace std;

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



Deskripsi Program

Program di atas merupakan contoh sederhana penggunaan array pada bahasa pemrograman C++. Array adalah tipe data yang dapat menampung sejumlah data dengan jenis yang sama dalam satu variabel.

Pada program di atas, terdapat array bernama 'nilai' dengan 5 elemen bertipe integer. Elemen array tersebut diisi dengan nilai-nilai integer secara manual dengan menggunakan operator indeks.

Kemudian, program mencetak nilai tiap elemen array menggunakan `cout` dan operator indeks juga. Output dari program ini akan menampilkan nilai masing-masing elemen array.

Secara keseluruhan, program ini menunjukkan bagaimana cara menggunakan array pada bahasa pemrograman C++ untuk menampung data dengan jenis yang

sama dalam satu variabel dan mengakses nilai tiap elemennya menggunakan operator indeks.

B. UNGUIDED

1. 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!

```
#include <iostream>

using namespace std;

int tambah(int a, int b)
{
    return a + b;
}
double bagi(double a, double b)
{
    return a / b;
}
bool isGenap(int a)
{
    return a % 2 == 0;
}

int main()
{
    int angka1, angka2;
    double pecahan1, pecahan2;

    cout << "Masukkan bilangan bulat 1: ";
    cin >> angka1;
    cout << "Masukkan bilangan bulat 2: ";
    cin >> angka2;

    cout << "Masukkan bilangan pecahan 1: ";
    cin >> pecahan1;
    cout << "Masukkan bilangan pecahan 2: ";
    cin >> pecahan2;

    int hasilTambah = tambah(angka1, angka2);
    double hasilBagi = bagi(pecahan1, pecahan2);
    bool apakahGenap = isGenap(angka1);

    cout << "\nHasil Tambah: " << hasilTambah << endl;
    cout << "Hasil Bagi: " << hasilBagi << endl;
```

```

        cout << "Apakah " << angka1 << " genap? " << apakahGenap
    << endl;
    return 0;
}

```

Screenshoot Program

```

Isi array kelima :90
PS D:\#kampus\#SEMESTER 2\laprak algoista\Coding\modul 1> cd "d:\#kampus\#SEMESTER 2\laprak algoista\Coding\modul 1\" ; if ($?) {
{ g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Masukkan bilangan bulat 1: 8
Masukkan bilangan bulat 2: 2
Masukkan bilangan pecahan 1: 8
Masukkan bilangan pecahan 2: 2

Hasil Tambah: 10
Hasil Bagi: 4
Apakah 8 genap? 1
PS D:\#kampus\#SEMESTER 2\laprak algoista\Coding\modul 1>

```

Deskripsi Program

Program di atas memiliki tiga fungsi yaitu `tambah`, `bagi`, dan `isGenap`, yang masing-masing melakukan operasi penjumlahan, pembagian, dan pengecekan bilangan genap menggunakan tipe data primitif `int` dan `double`. Fungsi `tambah` dan `bagi` sama seperti pada contoh sebelumnya, sedangkan fungsi `isGenap` menerima satu parameter bertipe `int` dan mengembalikan `true` jika bilangan tersebut genap dan `false` jika tidak.

Di dalam fungsi `main`, program meminta pengguna memasukkan dua bilangan bulat dan dua bilangan pecahan menggunakan perintah `cin`. Kemudian, program memanggil ketiga fungsi tersebut dengan parameter yang sesuai berdasarkan input dari pengguna. Hasilnya disimpan pada variabel-variabel `hasilTambah`, `hasilBagi`, dan `'apakah'Genap` masing-masing. Terakhir, program menampilkan hasilnya pada layar menggunakan perintah `cout`.

Dari program di atas, dapat disimpulkan bahwa tipe data primitif digunakan untuk memproses nilai tunggal seperti bilangan bulat (`int`) dan bilangan pecahan (`double`), serta dapat digunakan untuk menentukan sifat-sifat tertentu dari nilai tersebut seperti kegenapan. Program ini menunjukkan bahwa dengan menggabungkan fungsi-fungsi yang menggunakan tipe data primitif, kita dapat membuat program yang lebih kompleks dan dapat memproses input dari pengguna dengan bantuan perintah `cin`.

2. Unguided 2

Jelaskan fungsi dari class dan struktur secara detail dan berikan contoh programnya

Jawab :

Class dan **struct** adalah dua konsep penting dalam pemrograman berorientasi objek. Kedua konsep tersebut fungsinya mirip seperti kerangka atau *blueprint* untuk membuat objek atau *instance*.

Class adalah tipe data abstrak yang digunakan untuk mendefinisikan sebuah objek dengan properti dan metode yang terkait dengannya. Properti pada class disebut dengan variabel anggota (member variable), sedangkan metode disebut dengan fungsi anggota (member function). **Class** sifatnya dapat menyembunyikan atau mengatur data dan perilaku dari objek dengan tingkat aksesnya melalui modifier (public, private, dan protected).

Contoh program menggunakan *class*:

Source Code

```
#include <iostream>

using namespace std;

class Lingkaran
{
private:
    double jari_jari;

public:
    void setJariJari(double r)
    {
        jari_jari = r;
    }
    double hitungLuas()
    {
        return 3.14 * jari_jari * jari_jari;
    }
};

int main()
{
    Lingkaran lingkaran1;
    lingkaran1.setJariJari(21);
    cout << "Luas lingkaran: " <<
    lingkaran1.hitungLuas();
    return 0;
}
```

Screenshoot Program

```
Masukkan bilangan pecahan 2: 2

Hasil Tambah: 10
Hasil Bagi: 4
Apakah 8 genap? 1
PS D:\#kampus\SEMESTER 2\laprak algoista\Coding\modul 1> cd "d:\#kampus\SEMESTER 2\laprak algoista\Coding\modul 1" & g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile & if ($?) { .\tempCodeRunnerFile }
Luas lingkaran: 153.86
PS D:\#kampus\SEMESTER 2\laprak algoista\Coding\modul 1> cd "d:\#kampus\SEMESTER 2\laprak algoista\Coding\modul 1" & g++ u2.cpp -o u2 & if ($?) { .\u2 }
Luas lingkaran: 1384.74
PS D:\#kampus\SEMESTER 2\laprak algoista\Coding\modul 1> 
```

Untitled - Notepad
File Edit Format View Help
Nama:Muhammad Raafi Al Hafidh
Kelas:IF-11 B
NIM:2311102070
Ln 3, C 100% Windows (CRLF) UTF-8

Deskripsi Program

Pada contoh program di atas, kita mendefinisikan class Lingkaran yang memiliki variabel anggota `jari_jari` dan fungsi anggota `setJariJari` dan `hitungLuas`. Fungsi `setJariJari` digunakan untuk mengisi nilai jari-jari lingkaran, sedangkan fungsi `hitungLuas` digunakan untuk menghitung luas lingkaran.

Contoh program menggunakan *struct*:

Source Code

```
#include <iostream>

using namespace std;

struct Mahasiswa
{
    string nama;
    string nim;
    float ipk;
};

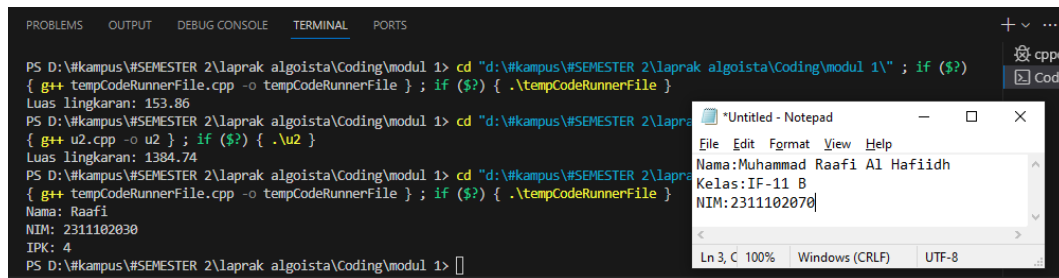
int main()
{
    Mahasiswa mhs1;

    mhs1.nama = "Raafi";
    mhs1.nim = "2311102030";
    mhs1.ipk = 4;

    cout << "Nama: " << mhs1.nama << endl;
    cout << "NIM: " << mhs1.nim << endl;
    cout << "IPK: " << mhs1.ipk << endl;

    return 0;
}
```

Screenshoot Program



```
PS D:\#kampus\#SEMESTER 2\laprak algoista\Coding\modul 1> cd "d:\#kampus\#SEMESTER 2\laprak algoista\Coding\modul 1\" ; if ($?) { g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Luas lingkaran: 153.86
PS D:\#kampus\#SEMESTER 2\laprak algoista\Coding\modul 1> cd "d:\#kampus\#SEMESTER 2\laprak algoista\Coding\modul 1\" ; if ($?) { g++ u2.cpp -o u2 } ; if ($?) { .\u2 }
Luas lingkaran: 1384.74
PS D:\#kampus\#SEMESTER 2\laprak algoista\Coding\modul 1> cd "d:\#kampus\#SEMESTER 2\laprak algoista\Coding\modul 1\" ; if ($?) { g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Nama: Raafi
NIM: 2311102030
IPK: 4
PS D:\#kampus\#SEMESTER 2\laprak algoista\Coding\modul 1> 
```

Deskripsi Program

Pada contoh program di atas, kita mendefinisikan struct Mahasiswa yang memiliki tiga properti yaitu nama, nim, dan ipk. Kemudian kita membuat objek `mhs1` dari struct Mahasiswa dan mengisi nilai propertinya. Terakhir, kita menampilkan nilai properti dari objek `mhs1` menggunakan fungsi `cout`.

Kesimpulan dari materi tipe data primitif adalah bahwa tipe data primitif hanya dapat menyimpan satu nilai pada satu waktu dan tidak dapat diubah menggunakan cara yang sama seperti tipe data non-primitif. Tipe data primitif akan dianggap sama jika nilainya sama. Oleh karena itu, dalam pemrograman yang lebih kompleks, seringkali kita memerlukan tipe data yang lebih kompleks seperti class.

3. Unguided 3

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

Source Code map:

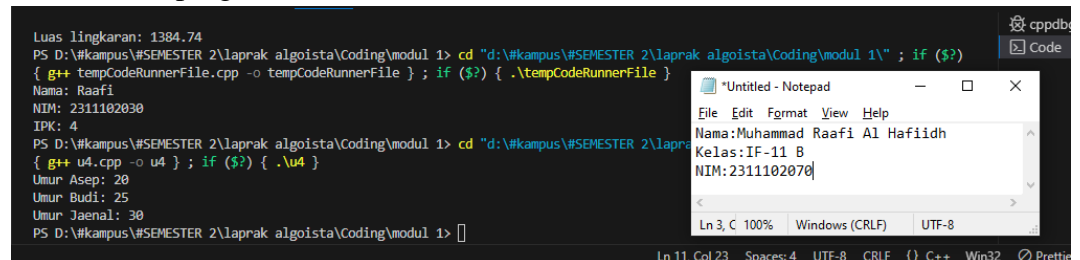
```
#include <iostream>
#include <map>
#include <string>
using namespace std;
int main()
{
    // membuat map untuk menyimpan umur berdasarkan nama
    map<string, int> umur;

    // mengisi map dengan data umur
    umur["Asep"] = 20;
    umur["Budi"] = 25;
    umur["Jaenal"] = 30;

    // menampilkan umur berdasarkan nama
    cout << "Umur Asep: " << umur["Asep"] << endl;
    cout << "Umur Budi: " << umur["Budi"] << endl;
    cout << "Umur Jaenal: " << umur["Jaenal"] << endl;
    return 0;
}
```

```
}
```

Screenshoot program



The screenshot shows a Windows terminal window with the following content:

```
Luas lingkaran: 1384.74
PS D:\#kampus\#SEMESTER 2\laprak algoista\Coding\modul 1> cd "d:\#kampus\#SEMESTER 2\laprak algoista\Coding\modul 1\" ; if ($?) { g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Nama: Raafi
NIM: 2311102030
IPK: 4
PS D:\#kampus\#SEMESTER 2\laprak algoista\Coding\modul 1> cd "d:\#kampus\#SEMESTER 2\laprak algoista\Coding\modul 1\" ; if ($?) { g++ u4.cpp -o u4 } ; if ($?) { .\u4 }
Umur Asep: 20
Umur Budi: 25
Umur Jaenal: 30
PS D:\#kampus\#SEMESTER 2\laprak algoista\Coding\modul 1>
```

Overlaid on the terminal is a Notepad window titled "Untitled - Notepad" containing the following text:

```
Nama:Muhammad Raafi Al Hafiidh
Kelas:IF-11 B
NIM:2311102070
```

Deskripsi Program

Pada program di atas, pertama-tama dibuat sebuah objek map bernama umur yang menyimpan data umur berdasarkan nama. Kemudian, data umur dimasukkan ke dalam map dengan menggunakan operator '['. Akhirnya, data umur dapat diakses dengan menggunakan operator '[' juga.

Perbedaan utama antara array dan map adalah bahwa array menyimpan data dalam bentuk urutan dengan indeks numerik, sedangkan map menyimpan data dalam bentuk pasangan kunci-nilai (*key-value*) dengan kunci yang bisa berupa tipe data apa saja (baik numerik maupun string). Dengan demikian, saat menggunakan array, kita hanya bisa mengakses data dengan menggunakan indeks numerik, sedangkan dengan map, kita bisa mengakses data dengan menggunakan kunci yang bisa berupa tipe data apa saja. Selain itu, ukuran array sudah ditentukan pada saat deklarasi, sedangkan ukuran map bisa bertambah atau berkurang sesuai dengan jumlah data yang disimpan di dalamnya.

KESIMPULAN

Kesimpulan yang dapat diambil dari laporan praktikum dengan “*Modul 1: Tipe Data Primitif, Tipe Data Abstrak, dan Tipe Data Koleksi*” adalah bahwa praktikum ini bertujuan untuk mempelajari berbagai jenis tipe data dalam pemrograman, yaitu tipe data primitif, abstrak, dan koleksi. Tipe data primitif hanya dapat menyimpan satu nilai pada satu waktu dan tidak dapat diubah, sedangkan tipe data abstrak dan koleksi lebih kompleks dan dapat dibuat sendiri oleh *programmer*.

Selain itu, praktikum ini juga membahas pengaplikasian tipe data yang digunakan dan penggunaan bahasa pemrograman yang telah ditentukan. Dalam praktikum ini, contoh implementasi data disajikan melalui sebuah program kalkulator sederhana menggunakan bahasa pemrograman C++. Program tersebut dapat menghitung operasi matematika dasar seperti penjumlahan, pengurangan, perkalian, dan pembagian.

Lalu juga di modul membahas tentang perbedaan penggunaan di tipe data abstrak, yaitu penggunaan *class* dan *struct* yang dimana *class* adalah salah satu dari konsep OOP (Objek Oriented Program) yang digunakan untuk membungkus sebuah prosedur, sedangkan *struct* merupakan sebuah variabel yang dapat menampung berbagai tipe data yang berbeda dalam satu buah variabel.

Referensi

Asisten Praktikum. “Modul 1 Tipe Data”. Learning Management System. 2024.

Modern C++ Programming: A Comprehensive Guide (2020) by Daniel J. Saks