

LAPORAN PRAKTIKUM

MODUL I TIPE DATA



Disusun oleh:
Arjun Werdho Kumoro
NIM: 2311102009

Dosen Pengampu:
Muhammad Afrizal Amrustian, S. Kom., M. Kom

**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 pemrograman 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

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 :

- a. **Int** : adalah tipe data yang digunakan untuk menyimpan bilangan bulat
- b. seperti 12, 1, 4, dan sebagainya.
- c. **Float** : tipe data yang digunakan untuk menyimpan bilangan desimal
- d. seperti 1.5, 2.1, 3.14, dan sebagainya.
- e. **Char** : berfungsi untuk menyimpan data berupa sebuah huruf. Biasanya
- f. digunakan untuk simbol seperti A, B, C dan seterusnya
- g. **Boolean** : tipe data ini digunakan untuk menyimpan nilai boolean yang
- h. 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:

- 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 Tipe data primitif

Source code

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

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

    // Meminta pengguna untuk memasukkan operator dan dua angka
    cout << "Enter an operator (+, -, *, /): ";
    cin >> op;
    cout << "Enter two numbers: ";
    cin >> num1 >> num2;

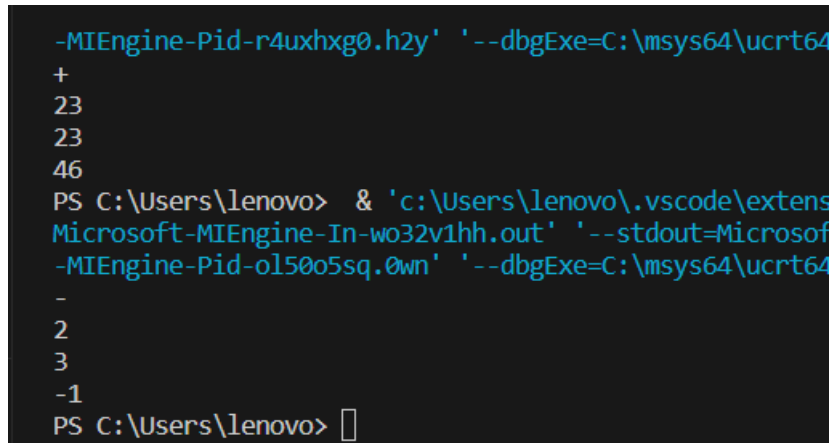
    // Memulai pemilihan kasus berdasarkan operator yang dimasukkan
    switch (op) {
        // Jika pengguna memasukkan operator +
        case '+':
            cout << "Result: " << num1 + num2;
            break;
        // Jika pengguna memasukkan operator -
        case '-':
            cout << "Result: " << num1 - num2;
            break;
        // Jika pengguna memasukkan operator *
        case '*':
            cout << "Result: " << num1 * num2;
            break;
        // Jika pengguna memasukkan operator /
        case '/':
            // Mengecek apakah num2 tidak sama dengan 0
            if (num2 != 0)
                cout << "Result: " << num1 / num2;
            // Jika num2 sama dengan 0, tampilkan pesan kesalahan
            else
                cout << "Error! Cannot divide by zero!";
            break;
        // Jika operator yang dimasukkan tidak valid
        default:
            cout << "Error! Operator is not correct!";
    }
```

```
}

return 0;

}
```

Screenshoot program



```
-MIEngine-Pid-r4uxhxo.h2y' '--dbgExe=C:\msys64\ucrt64...
+
23
23
46
PS C:\Users\lenovo> & 'c:\Users\lenovo\.vscode\extensions\
Microsoft-MIEngine-In-wo32v1hh.out' '--stdout=Microsoft
-MIEngine-Pid-ol50o5sq.0wn' '--dbgExe=C:\msys64\ucrt64...
-
2
3
-1
PS C:\Users\lenovo> 
```

Deskripsi program

1. Program dimulai dengan mendeklarasikan variabel `op` untuk menyimpan operator dan variabel `num1` dan `num2` untuk menyimpan dua angka yang akan dioperasikan.
2. Pengguna diminta untuk memasukkan operator dan dua angka menggunakan `cin`.
3. Kemudian, dilakukan pengecekan menggunakan struktur `switch` untuk mengecek nilai `op`:
4. Jika operator yang dimasukkan adalah `+`, program akan mencetak hasil penjumlahan `num1` dan `num2`.
5. Jika operator yang dimasukkan adalah `-`, program akan mencetak hasil pengurangan `num1` dan `num2`.
6. Jika operator yang dimasukkan adalah `*`, program akan mencetak hasil perkalian `num1` dan `num2`.
7. Jika operator yang dimasukkan adalah `/`, program akan mencetak hasil pembagian `num1` dan `num2`.
8. Jika operator yang dimasukkan tidak sesuai dengan `+`, `-`, `*`, atau `/`, program akan mencetak pesan kesalahan.
9. Program kemudian mengembalikan nilai `0`, menandakan bahwa program telah berakhir dengan sukses.

2. Guided 2 Tipe data abstrak

Source Code

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

// Struct
struct Mahasiswa {
    const char *name;
    const char *address;
    int age;
};

int main() {
    // Menggunakan struct
    Mahasiswa mhs1, mhs2;

    // Mengisi nilai ke struct
    mhs1.name = "arjun";
    mhs1.address = "kalimanah";
    mhs1.age = 19;

    mhs2.name = "ulima";
    mhs2.address = "bukateja";
    mhs2.age = 20;

    // Mencetak isi struct
    cout << "## Mahasiswa 1 ##" << endl;
    cout << "Nama: " << mhs1.name << endl;
    cout << "Alamat: " << mhs1.address << endl;
    cout << "Umur: " << mhs1.age << endl;

    cout << "## Mahasiswa 2 ##" << endl;
    cout << "Nama: " << mhs2.name << endl;
    cout << "Alamat: " << mhs2.address << endl;
    cout << "Umur: " << mhs2.age << endl;

    return 0;
}}
```

Screenshot program

```
PS C:\Users\lenovo> & 'c:\Users\lenovo\.vscode\external\Microsoft-MIEngine-In-l04ihn12.amd' '--stdout=Microsoft-MIEngine-Pid-u2hrqfqv.0kd' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe'
## Mahasiswa 1 ##
Nama: arjun
Alamat: kalimanah
Umur: 19
## Mahasiswa 2 ##
Nama: ulima
Alamat: bukateja
Umur: 20
PS C:\Users\lenovo> 
```

Deskripsi Program

1. Deklarasi Struktur: Script dimulai dengan mendeklarasikan sebuah struktur bernama Mahasiswa, yang memiliki tiga anggota: name (nama mahasiswa), address (alamat mahasiswa), dan age (umur mahasiswa).
2. Inisialisasi Struktur: Dua variabel struktur mhs1 dan mhs2 dideklarasikan untuk menampung informasi dua mahasiswa.
3. Pengisian Nilai: Informasi untuk setiap mahasiswa diisi ke dalam variabel struktur mhs1 dan mhs2 menggunakan operator penugasan.
4. Cetak Informasi: Informasi dari setiap mahasiswa kemudian dicetak ke layar menggunakan fungsi printf(). Setiap atribut dari struktur diakses menggunakan operator titik (.).
5. Pengembalian Nilai: Program mengembalikan nilai 0, menandakan bahwa program telah berakhir dengan sukses.

3. Guided 2 Tipe data koleksi

Source code

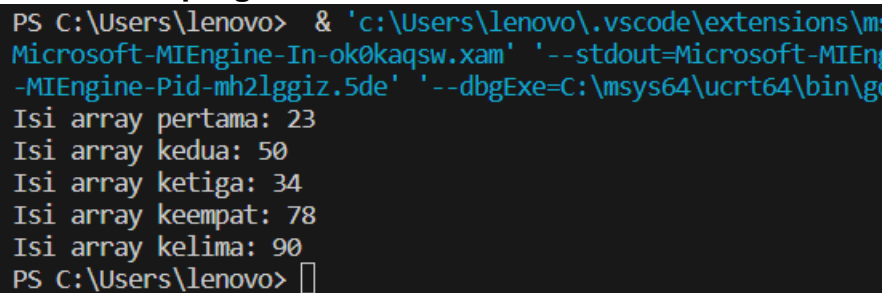
```
#include <iostream>
using namespace std;

int main() {
    // Deklarasi dan inisialisasi array
    int nilai[5] = {23, 50, 34, 78, 90};

    // Mencetak isi 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;
}
```

Screenshot program



```
PS C:\Users\lenovo> & 'c:\Users\lenovo\.vscode\extensions\ms-vscode-  
Microsoft-MIEngine-In-0k0kaqsw.xam' '--stdout=Microsoft-MIEng  
-MIEngine-Pid-mh2lggiz.5de' '--dbgExe=C:\msys64\ucrt64\bin\g  
Isi array pertama: 23  
Isi array kedua: 50  
Isi array ketiga: 34  
Isi array keempat: 78  
Isi array kelima: 90  
PS C:\Users\lenovo> █
```

Deskripsi Program

1. Deklarasi Array: Script dimulai dengan mendeklarasikan sebuah array bernama nilai yang memiliki kapasitas untuk menyimpan 5 bilangan bulat.
2. Inisialisasi Array: Nilai-nilai array diinisialisasi dengan menggunakan operator indeks, yaitu nilai[0], nilai[1], nilai[2], nilai[3], dan nilai[4].
3. Pengisian Nilai: Setiap elemen array diisi dengan nilai tertentu, seperti 23, 50, 34, 78, dan 90.
4. Cetak Nilai Array: Nilai-nilai array kemudian dicetak ke layar menggunakan cout. Setiap elemen array diakses menggunakan indeksnya, misalnya nilai[0], nilai[1], dan seterusnya.
5. Pengembalian Nilai: Program mengembalikan nilai 0, menandakan bahwa program telah berakhir dengan sukses.

LATIHAN KELAS - 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!

Source code

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

// Fungsi untuk menghitung volume balok
float hitungVolumeBalok(float panjang, float lebar, float tinggi) {
    return panjang * lebar * tinggi;
}

// Fungsi untuk menghitung luas permukaan balok
float hitungLuasPermukaanBalok(float panjang, float lebar, float tinggi) {
    return 2 * ((panjang * lebar) + (panjang * tinggi) + (lebar * tinggi));
}

int main() {
    float panjang, lebar, tinggi;

    cout << "Masukkan panjang balok: ";
    cin >> panjang;

    cout << "Masukkan lebar balok: ";
    cin >> lebar;

    cout << "Masukkan tinggi balok: ";
    cin >> tinggi;

    // Hitung volume balok
    float volume = hitungVolumeBalok(panjang, lebar, tinggi);
    cout << "Volume balok: " << volume << endl;

    // Hitung luas permukaan balok
```

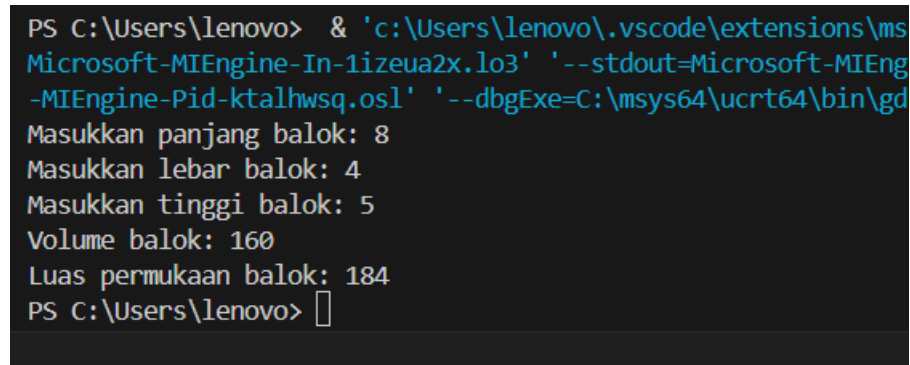
```
float luasPermukaan = hitungLuasPermukaanBalok(panjang, lebar,
tinggi);

cout << "Luas permukaan balok: " << luasPermukaan << endl;

return 0;

}
```

Screenshoot program



```
PS C:\Users\lenovo> & 'c:\Users\lenovo\.vscode\extensions\ms
Microsoft-MIEngine-In-1izeua2x.lo3' '--stdout=Microsoft-MIEng
-MIEngine-Pid-ktalhwsq.osl' '--dbgExe=C:\msys64\ucrt64\bin\gd
Masukkan panjang balok: 8
Masukkan lebar balok: 4
Masukkan tinggi balok: 5
Volume balok: 160
Luas permukaan balok: 184
PS C:\Users\lenovo> 
```

Deskripsi program

Penjelasan Program:

1. Program ini meminta pengguna memasukkan panjang, lebar, dan tinggi balok.
2. Setelah itu, program akan menghitung volume dan luas permukaan balok berdasarkan nilai yang dimasukkan.
3. Program kemudian menampilkan hasil volume dan luas permukaan balok.

Kesimpulan dari Materi Tipe Data Primitif:

1. Program ini menggunakan tipe data primitif float untuk menyimpan nilai panjang, lebar, tinggi, volume, dan luas permukaan balok. Tipe data primitif float dipilih karena memiliki presisi yang cukup untuk representasi bilangan real.
2. Dengan menggunakan tipe data primitif, program menjadi lebih efisien dalam penggunaan memori dan eksekusi, karena tipe data float hanya membutuhkan 4 byte dalam penyimpanan.
3. Pemilihan tipe data primitif yang sesuai dengan kebutuhan program dapat membantu mengoptimalkan kinerja dan penggunaan memori program secara keseluruhan.

2. Unguided 2

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

Class adalah struktur data yang dapat digunakan untuk menggabungkan data dan fungsi (metode) ke dalam sebuah entitas tunggal. Ini memungkinkan kita untuk memodelkan objek dalam program dan bekerja dengan konsep pemrograman berorientasi objek (OOP). Di dalam class, kita dapat mendefinisikan variabel anggota (atau disebut juga data member) dan fungsi anggota (metode).

Dalam class:

- Data member adalah variabel yang menyimpan data terkait dengan objek dari class tersebut.
- Member functions (atau metode) adalah fungsi yang bekerja pada objek dari class tersebut, dan biasanya digunakan untuk memanipulasi data member.

Contoh Source code Class

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

// Mendefinisikan class Mahasiswa
class Mahasiswa {
public:
    // Data member
    string nama;
    int usia;

    // Member function
    void displayInfo() {
        cout << "Nama: " << nama << endl;
        cout << "Usia: " << usia << endl;
    }
};

int main() {
    // Membuat objek dari class Mahasiswa
    Mahasiswa mhs1;
```

```

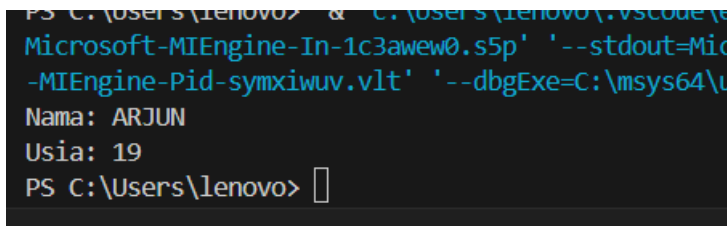
// Mengatur data member objek mhs1
mhs1.nama = "ARJUN";
mhs1.usia = 19;

// Memanggil member function untuk menampilkan informasi mahasiswa
mhs1.displayInfo();

return 0;
}

```

Screenshoot program



```

PS C:\Users\lenovo> & C:\Users\lenovo\.vscode\extensions\ms-vscode.cmake-tools\bin\Microsoft-MIEngine-In-1c3awew0.s5p' '--stdout=Microsoft-MIEngine-Pid-symxiwuv.vlt' '--dbgExe=C:\msys64\ut
Nama: ARJUN
Usia: 19
PS C:\Users\lenovo>

```

Deskripsi program

1. Penggunaan Class: Program menggunakan class Mahasiswa untuk menggambarkan entitas mahasiswa. Class ini memiliki dua data member (nama dan usia) yang merepresentasikan atribut-atribut yang dimiliki oleh seorang mahasiswa.
2. Objek Mahasiswa: Dalam fungsi main(), objek mhs1 dari class Mahasiswa dibuat. Ini menunjukkan konsep dari instansiasi, di mana kita membuat objek dari sebuah class.
3. Inisialisasi Data Member: Setelah objek mhs1 dibuat, data member objek tersebut diatur dengan nilai. Dalam hal ini, nama mahasiswa diatur menjadi "John" dan usianya diatur menjadi 20.
4. Member Function: Program juga menggunakan member function displayInfo() untuk menampilkan informasi tentang objek mahasiswa ke layar. Ini adalah contoh bagaimana fungsi anggota dapat digunakan untuk memanipulasi data anggota dalam class.
5. Output: Program mencetak informasi mahasiswa (nama dan usia) ke layar menggunakan fungsi displayInfo().

Struct (singkatan dari structure) adalah tipe data yang digunakan untuk menyimpan sekumpulan variabel dengan tipe data yang berbeda dalam satu unit. Struct memungkinkan kita untuk membuat kumpulan data yang terorganisir, mirip dengan class, tetapi dengan perbedaan utama yaitu default akses ke anggota yang berbeda (public untuk struct dan private untuk class).

Dalam struct:

- Anggota struct (data member) dapat diakses secara langsung tanpa harus menggunakan fungsi anggota.
- Struct biasanya digunakan untuk menyimpan data terkait, tanpa menyediakan banyak fungsi untuk memanipulasi data.

Contoh Source code struct

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

// Mendefinisikan struct Mahasiswa
struct Mahasiswa {
    // Data member
    string nama;
    int usia;
};

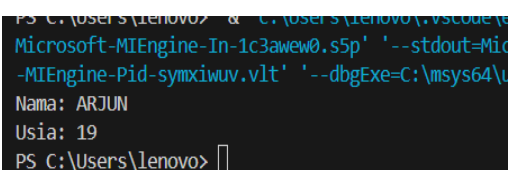
int main() {
    // Membuat objek dari struct Mahasiswa
    Mahasiswa mhs1;

    // Mengatur data member objek mhs1
    mhs1.nama = "ARJUN";
    mhs1.usia = 19;

    // Menampilkan informasi mahasiswa
    cout << "Nama: " << mhs1.nama << endl;
    cout << "Usia: " << mhs1.usia << endl;

    return 0;
}
```

Screenshoot program



```
PS C:\Users\lenovo> g++ C:\Users\lenovo\vscode\Microsoft-MIEngine-In-1c3awew0.s5p' '--stdout=Microsoft-MIEngine-Pid-symxiwuv.vlt' '--dbgExe=C:\msys64\usr\bin\g++.exe'
Nama: ARJUN
Usia: 19
PS C:\Users\lenovo> █
```

Deskripsi program

1. Penggunaan Struct: Program menggunakan struct Mahasiswa untuk menggambarkan entitas mahasiswa. Struct ini memiliki dua data member (nama dan usia) yang merepresentasikan atribut-atribut yang dimiliki oleh seorang mahasiswa.
2. Objek Mahasiswa: Dalam fungsi main(), objek mhs1 dari struct Mahasiswa dibuat. Ini menunjukkan konsep dari instansiasi, di mana kita membuat objek dari sebuah struct.
3. Inisialisasi Data Member: Setelah objek mhs1 dibuat, data member objek tersebut diatur dengan nilai. Dalam hal ini, nama mahasiswa diatur menjadi "ARJUN" dan usianya diatur menjadi 19.
4. Akses Data Member: Program menggunakan operator dot (.) untuk mengakses data member dari objek struct mhs1. Ini memungkinkan program untuk mengakses dan memanipulasi data anggota dari objek struct.
5. Output: Program mencetak informasi mahasiswa (nama dan usia) ke layar menggunakan cout.

3. Unguided 3

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

Source code

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

int main() {
    // Mendefinisikan map untuk menyimpan jadwal puasa Idul Fitri
    map<int, std::string> jadwal_puasa_idul_fitri;

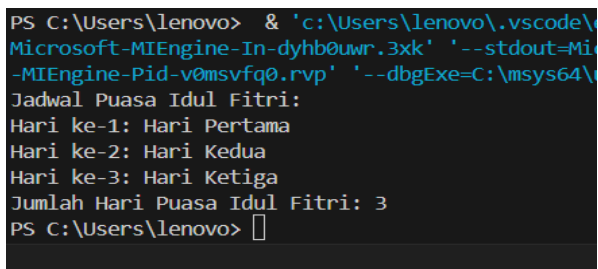
    // Menambahkan data jadwal puasa Idul Fitri ke dalam map
    jadwal_puasa_idul_fitri[1] = "Hari Pertama";
    jadwal_puasa_idul_fitri[2] = "Hari Kedua";
    jadwal_puasa_idul_fitri[3] = "Hari Ketiga";

    // Menampilkan jadwal puasa Idul Fitri
    cout << "Jadwal Puasa Idul Fitri:\n";
    for (const auto& pair : jadwal_puasa_idul_fitri) {
        cout << "Hari ke-" << pair.first << ": " << pair.second
        << endl;
    }

    // Menampilkan jumlah hari puasa Idul Fitri
    cout << "Jumlah Hari Puasa Idul Fitri: " <<
    jadwal_puasa_idul_fitri.size() << endl;

    return 0;
}
```

Screenshoot program



```
PS C:\Users\lenovo> & 'c:\Users\lenovo\.vscode\
Microsoft-MIEngine-In-dyhb0uwr.3xk' '--stdout=Mi
-MIEngine-Pid-v0msvfq0.rvp' '--dbgExe=C:\msys64\
Jadwal Puasa Idul Fitri:
Hari ke-1: Hari Pertama
Hari ke-2: Hari Kedua
Hari ke-3: Hari Ketiga
Jumlah Hari Puasa Idul Fitri: 3
PS C:\Users\lenovo> 
```


Deskripsi program

1. Data jadwal puasa Idul Fitri ditambahkan ke dalam map menggunakan `jadwal_puasa_idul_fitri[...] =`
2. Program kemudian menampilkan jadwal puasa Idul Fitri dengan melakukan iterasi melalui setiap pasangan nomor hari dan deskripsi puasa dalam map.
3. Program juga menampilkan jumlah hari puasa Idul Fitri menggunakan `jadwal_puasa_idul_fitri.size()`.

Perbedaan antara Array dan Map:

1. Array: Array adalah struktur data yang menyimpan kumpulan elemen dengan tipe data yang sama dalam urutan tertentu. Akses ke elemen array dilakukan menggunakan indeks numerik. Ukuran array harus ditentukan pada saat deklarasi dan tidak bisa diubah selama runtime.
2. Map: Map adalah struktur data yang menyimpan kumpulan pasangan kunci-nilai di mana setiap kunci unik terkait dengan satu nilai tertentu. Map menyediakan pencarian cepat berdasarkan kunci, dan ukuran map dapat berubah dinamis selama runtime. Map lebih fleksibel daripada array karena tidak memerlukan penentuan ukuran di awal, dan dapat digunakan untuk memetakan kunci yang tidak berurutan ke nilai-nilai yang terkait. Dalam konteks jadwal puasa Idul Fitri, map memungkinkan kita untuk memetakan nomor hari ke deskripsi puasa dengan mudah.

BAB IV

KESIMPULAN

- **Tipe Data Primitif:** Tipe data primitif adalah tipe data yang sudah ditentukan oleh sistem dan disediakan oleh bahasa pemrograman. Contoh tipe data primitif antara lain int, float, char, dan boolean. Program yang menggunakan tipe data primitif memiliki kelebihan dalam efisiensi penggunaan memori dan eksekusi karena sederhana dan langsung diakses oleh kompiler.

- **Tipe Data Abstrak:** Tipe data abstrak adalah tipe data yang dibentuk oleh programmer sendiri, seperti class dalam pemrograman berorientasi objek. Class memungkinkan kita untuk menggabungkan data dan fungsi ke dalam sebuah entitas tunggal. Sedangkan struct juga merupakan tipe data abstrak, namun lebih sederhana karena memiliki default akses ke anggota yang berbeda (public untuk struct dan private untuk class).

- **Tipe Data Koleksi:** Tipe data koleksi adalah tipe data yang digunakan untuk mengelompokkan dan menyimpan beberapa nilai atau objek secara bersamaan. Contoh tipe data koleksi antara lain array, vector, dan map. Array adalah struktur data statis yang menyimpan elemen-elemen dengan tipe data yang sama, sedangkan map adalah struktur data yang menyimpan kumpulan pasangan kunci-nilai.