

**LAPORAN PRAKTIKUM STRUKTUR  
DATA DAN ALGORITMA**

**MODUL I  
TIPE DATA**



**Disusun Oleh :**

**NAMA : FAISAL KHOIRUDDIN**

**NIM : 2311102046**

**Dosen**

**WAHYU ANDI SAPUTRA, S.Pd., M.Eng.**

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

## A. Dasar Teori

Tipe data merupakan sebuah pengklasifikasian data berdasarkan jenis data tersebut. Tipe data dibutuhkan agar kompiler dapat mengetahui bagaimana sebuah data akan digunakan. Tipe data yang akan dipelajari yaitu sebagai berikut :

1. Tipe data Primitif
2. Tipe data Abstrak
3. Tipe data Koleksi

### 1. Tipe data Primitif

Tipe data primitif merupakan tipe data yang sudah ditentukan oleh bahasa pemrograman, 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 yaitu :

#### 1) Int

Int merupakan tipe data yang digunakan untuk menyimpan bilangan bulat seperti 11, 2, 3, 4, dan lain-lain.

#### 2) Float

Float merupakan tipe data yang digunakan untuk menyimpan bilangan desimal seperti 1.2, 3.1, 3.14, dan lain-lain.

#### 3) Char

Char berfungsi untuk menyimpan data berupa sebuah huruf. Biasanya digunakan untuk simbol seperti A, B, C, D, E, F, G, dan seterusnya.

#### 4) Boolean.

Tipe data ini digunakan untuk menyimpan nilai boolean yang hanya memiliki dua nilai yaitu true(benar) dan false(salah).

### 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(OPP) pada bahasa C++ yang mirip dengan fitur data structures Struct pada bahasa C. Keduanya berfungsi untuk membungkus tipe data di dalamnya sebagai anggota. Perbedaan antara Struct dan Class yaitu pada akses default Struct bersifat publik dan akses default Class bersifat privat.

### 3. Tipe Data Koleksi

Tipe data koleksi (Collection Data Type) merupakan tipe data yang digunakan untuk mengelompokkan dan menyimpan beberapa nilai atau objek secara bersamaan. Tipe data koleksi memungkinkan 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:

#### 1) Array

Array merupakan 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.

#### 2) Vector

Vector merupakan 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.

#### 3) 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.

### B. Guided

#### Guided 1 : tipe data primitif

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

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

```

// Meminta pengguna untuk memasukkan operator: +, -, *, /
cout << "Masukkan operator (+, -, *, /): ";
cin >> op;

// Meminta pengguna untuk memasukkan operator
cout << "Masukkan dua angka: ";
cin >> num1 >> num2;

// Switch statement
switch (op) {
    case '+':
        cout << "Hasil penjumlahan: " << num1 + num2;
        break;
    case '-':
        cout << "Hasil pengurangan: " << num1 - num2;
        break;
    case '*':
        cout << "Hasil perkalian: " << num1 * num2;
        break;
    case '/':
        if (num2 != 0)
            cout << "Hasil pembagian: " << num1 / num2;
        else
            cout << "Error! Tidak dapat melakukan pembagian dengan nol.";
        break;
    default:
        cout << "Error! Operator tidak benar.";
}

return 0;
}

```

## Screenshots Output

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

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

PS C:\Users\ASUS\OneDrive\Dokumen\semester 2> & 'c:\Users\ASUS\.vscode\extensions\ms-vscode.cpptools-1.19.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-galiwcv4.fco' '--stdout=Microsoft-MIEngine-Out-mpyobd4p.wqn' '--stderr=Microsoft-MIEngine-Error-rxueurgz.d11' '--pid=Microsoft-MIEngine-Pid-ay3oxwhu.b0n' '--dbgExe=c:\mingw32\bin\gdb.exe' '--interpreter=mi'
Masukkan operator (+, -, *, /): +
Masukkan dua angka: 2 3
Hasil penjumlahan: 5
PS C:\Users\ASUS\OneDrive\Dokumen\semester 2>

```

Overlaid on the terminal is a Notepad window titled "\*Tidak berjudul - Notepad". It contains the following text:

```

NIM : 2311102046
Nama : Faisal KHoiruddin
Kelas : IF-11-B

```

```
-ay3oxwhu.b0n' '--dbgExe=c:\mingw32\bin\gdb.exe' '--interpreter=mi'
Masukkan operator (+, -, *, /): +
Masukkan dua angka: 2 3
Hasil penjumlahan: 5
PS C:\Users\ASUS\OneDrive\Dokumen\semester 2> & 'c:\Users\ASUS\.vscode\extensions\ms-vscode.cpptools-1.19.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-5ugqwnf1.2qr' '--stdout=Microsoft-MIEngine-Out-gttobqvz.1kh' '--stderr=Microsoft-MIEngine-Error-u1qna4z2.aqj' '--pid=Microsoft-MIEngine-Pid-cjfls3p5.blm' '--dbgExe=c:\mingw32\bin\gdb.exe' '--interpreter=mi'
Masukkan operator (+, -, *, /): -
Masukkan dua angka: 3 5
Hasil pengurangan: -2
PS C:\Users\ASUS\OneDrive\Dokumen\semester 2> 
```

\*Tidak berjudul - Notepad

File Edit Lihat

NIM : 2311102046  
Nama : Faisal Khoiruddin  
Kelas : IF-11-B

```
PS C:\Users\ASUS\OneDrive\Dokumen\semester 2> & 'c:\Users\ASUS\.vscode\extensions\ms-vscode.cpptools-1.19.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-jfcmians.3hw' '--stdout=Microsoft-MIEngine-Out-xo30uwmi.crb' '--stderr=Microsoft-MIEngine-Error-m1flirmj.kl1' '--pid=Microsoft-MIEngine-Pid-hy3kogx3.bxs' '--dbgExe=c:\mingw32\bin\gdb.exe' '--interpreter=mi'
Masukkan operator (+, -, *, /): *
Masukkan dua angka: 2 3
Hasil perkalian: 6
PS C:\Users\ASUS\OneDrive\Dokumen\semester 2> 
```

\*Tidak berjudul - Notepad

File Edit Lihat

NIM : 2311102046  
Nama : Faisal Khoiruddin  
Kelas : IF-11-B

```
PS C:\Users\ASUS\OneDrive\Dokumen\semester 2> & 'c:\Users\ASUS\.vscode\extensions\ms-vscode.cpptools-1.19.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-0rou2aiy.cb1' '--stderr=Microsoft-MIEngine-Error-g5a-c3wa32bx.2nq' '--dbgExe=c:\mingw32\bin\gdb.exe' '--interpreter=mi'
Masukkan operator (+, -, *, /): /
Masukkan dua angka: 4 2
Hasil pembagian: 2
PS C:\Users\ASUS\OneDrive\Dokumen\semester 2> 
```

\*Tidak berjudul - Notepad

File Edit Lihat

NIM : 2311102046  
Nama : Faisal Khoiruddin  
Kelas : IF-11-B

## Deskripsi:

Program tersebut merupakan contoh dari program data primitif yaitu program kalkulator sederhana untuk melakukan operasi penjumlahan, pengurangan, perkalian, dan pembagian. Pengguna memasukkan dua angka sesuai keinginan kemudian muncul output yang merupakan operasi dari kedua bilangan tersebut.

- `#include <iostream>` ➔ merupakan input output stream header yang digunakan sebagai standar input output operasi yang digunakan di c++
- `using namespace std;` ➔ digunakan untuk mendeklarasikan/ memberitahukan kepada compiler bahwa kita akan menggunakan semua fungsi/class/file yang terdapat dalam namespace std
- `int main() {` ➔ merupakan fungsi utama yang akan dieksekusi saat program dijalankan.
- `char op;` ➔ deklarasi variable op sebagai char
- `float num1, num2;` ➔ mendeklarasikan variabel num1 dan num2 sebagai float untuk menyimpan dua angka.
- `cout << "Masukkan operator (+, -, *, /): ";` ➔ cout menampilkan dan meminta

pengguna untuk memasukkan operator

- `cin >> op;` ➔ membaca input operator dari pengguna
- `cout << "Masukkan dua angka: ";` ➔ `cout` menampilkan statement Memasukkan dua angka untuk meminta pengguna memasukkan dua angka
- `cin >> num1 >> num2;` ➔ membaca dua angka yang diinputkan oleh pengguna
- `switch (op) {` ➔ switch statement
- `case '+':` ➔ jika operator adalah '+', maka lakukan penjumlahan
- `cout << "Hasil penjumlahan: " << num1 + num2;` ➔ `cout` menampilkan statement hasil penjumlahan dilanjutkan dengan memanggil dan menjumlahkan kedua variabel
- `break;` ➔ keluar dari switch setelah menjalankan kode di case '+'
- `case '-':` ➔ jika operator adalah '-', maka lakukan pengurangan
- `cout << "Hasil pengurangan: " << num1 - num2;` ➔ `cout` menampilkan statement hasil pengurangan dilanjutkan dengan memanggil dan mengurangi kedua variabel
- `break;` ➔ Keluar dari switch setelah menjalankan kode di case '-'
- `case '*':` ➔ jika operator adalah '\*', maka lakukan perkalian
- `cout << "Hasil perkalian: " << num1 * num2;` ➔ `cout` menampilkan statement hasil perkalian dilanjutkan dengan memanggil dan mengalikan kedua variabel
- `break;` ➔ Keluar dari switch setelah menjalankan kode di case '\*'
- `case '/':` ➔ jika operator adalah '/', maka lakukan pembagian
- `if (num2 != 0)` ➔ percabangan if dengan kondisi jika variable num2 tidak sama dengan nol, maka lakukan pembagian
- `cout << "Hasil pembagian: " << num1 / num2;` ➔ `cout` menampilkan statement "Hasil pembagian: " dilanjutkan dengan memanggil dan membagi kedua variabel
- `else` ➔ jika num2 sama dengan nol maka tampilkan pesan error
- `cout << "Error! Tidak dapat melakukan pembagian dengan nol.";` ➔ `cout` menampilkan statement "Error! Tidak dapat melakukan pembagian dengan nol.";
- `break;` ➔ Keluar dari switch setelah menjalankan kode di case '/'
- `default:` ➔ jika operator yang dimasukkan selain dari operator diatas maka tampilkan pesan error
- `cout << "Error! Operator tidak benar.";` ➔ `cout` menampilkan "Error! Operator tidak benar."

- return 0; ➔ program akan mengembalikan (return) nilai 0 ke operating sistem yang menjalankan program tersebut

## Guided 2 : tipe data abstrak

```
#include <stdio.h>

//Struct

struct Mahasiswa {

    const char *name;

    const char *address;

    int age;

};

int main() {

    // Deklarasi dua variabel menggunakan struct Mahasiswa

    struct Mahasiswa mhs1, mhs2;

    // mengisi nilai ke dalam struct mhs1

    mhs1.name = "Dian";

    mhs1.address = "Mataram";

    mhs1.age = 22;

    // Mengisi nilai ke dalam struct mhs2

    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;

}

```

### Screenshots Output

The screenshot shows a Windows command prompt window with the following text:

```

PS C:\Users\ASUS\OneDrive\Dokumen\semester 2> & 'c:\Users\ASUS\.vscode\extensions\ms-vscode.cpptools-1.19.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-t52rj4d1.uqr' '--stdout=Microsoft-MIEngine-Out-gjrnysp2.2hb' '--stderr=Microsoft-MIEngine-Error-pnjvtbx4.ce0' '--pid=Microsoft-MIEngine-Pid-wiai3myq.jti' '--dbgExe=c:\mingw32\bin\gdb.exe' '--interpreter=mi'
## Mahasiswa 1 ##
Nama: Dian
Alamat: Mataram
Umur: 22
## Mahasiswa 2 ##
Nama: Bambang
Alamat: Surabaya
Umur: 23
PS C:\Users\ASUS\OneDrive\Dokumen\semester 2>

```

Overlaid on the command prompt is a Notepad window titled "\*Tidak berjudul - Notepad". It contains the following text:

```

NIM : 2311102046
Nama : Faisal KHoairuddin
Kelas : IF-11-B

```

### Deskripsi:

Program tersebut merupakan contoh dari program data abstrak yaitu program sederhana menggunakan struct untuk menyimpan dan menampilkan informasi tentang dua mahasiswa. Program struct dengan nama mahasiswa yang memiliki tiga elemen yaitu name, address, dan age. Program mendeklarasikan variabel mhs1 dan mhs2 dengan tipe mahasiswa, mengisi nilai ke dalam kedua struct dan mencetak isi dari kedua struct tersebut.

- `#include <stdio.h>` ➔ merupakan *header* yang berisi fungsi-fungsi, makro dan tipe yang digunakan untuk melakukan operasi *input output*. file `stdio.h` berisi kode program agar nantinya kita bisa mengakses perintah `printf`.
- `struct Mahasiswa {` ➔ Mendeklarasikan struct dengan nama mahasiswa dengan tiga elemen yaitu name, address, dan age
- `const char *name;` ➔ deklarasi variabel pointer name
- `const char *address;` ➔ deklarasi variabel pointer address
- `int age;` ➔ deklarasi variabel age dengan nama tipe data integer
- `int main() {` ➔ merupakan fungsi utama yang akan dieksekusi saat program dijalankan.
- `struct Mahasiswa mhs1, mhs2;` ➔ deklarasi dua variabel menggunakan struct Mahasiswa
- `mhs1.name = "Dian";` ➔ mengisi nilai ke dalam struct mhs1 dengan variabel



name = "Dian";

- mhs1.address = "Mataram"; ➔ mengisi nilai ke dalam struct mhs1 dengan variabel address = "Mataram";
- mhs1.age = 22; ➔ mengisi nilai ke dalam struct mhs1 yaitu mhs1.age = 22
- mhs2.name = "Bambang"; ➔ mengisi nilai ke dalam struct mhs2 yaitu mhs2.name = "Bambang"
- mhs2.address = "Surabaya"; ➔ mengisi nilai ke dalam struct mhs2 yaitu mhs2.address = "Surabaya"
- mhs2.age = 23; ➔ mengisi nilai ke dalam struct mhs2 yaitu mhs2.age = 23
- printf("## Mahasiswa 1 ##\n"); ➔ mencetak judul mahasiswa 1
- printf("Nama: %s\n", mhs1.name); ➔ mencetak kata nama diikuti oleh nilai dari mhs1.name
- printf("Alamat: %s\n", mhs1.address); ➔ mencetak kata alamat diikuti oleh nilai dari mhs1.address
- printf("Umur: %d\n", mhs1.age); ➔ mencetak kata umur diikuti oleh nilai dari mhs1.age
- printf("## Mahasiswa 2 ##\n"); ➔ mencetak judul mahasiswa 1
- printf("Nama: %s\n", mhs2.name); ➔ mencetak kata nama diikuti oleh nilai dari mhs2.name
- printf("Alamat: %s\n", mhs2.address); ➔ mencetak kata alamat diikuti oleh nilai dari mhs2.address
- printf("Umur: %d\n", mhs2.age); ➔ mencetak kata alamat diikuti oleh nilai dari mhs2.address
- return 0; ➔ program akan mengembalikan (return) nilai 0 ke operating sistem yang menjalankan program tersebut

### 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;
}
```

### Screenshots Output

```
PS C:\Users\ASUS\OneDrive\Dokumen\semester 2> & 'c:\Users\ASUS\.vscode\extensions\ms-vscode.cpptools-1.19.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-vxvnmql4.kku' '--stdout=Microsoft-MIEngine-Out-ppk1myf4.13x' '--stderr=Microsoft-MIEngine-Error-pita05tc.lbi' '--pid=Microsoft-MIEngine-Pid-jvwel1cd.jpg' '--dbgExe=c:\mingw32\bin\gdb.exe' '--interpreter=mi'
Isi array pertama :23
Isi array kedua :50
Isi array ketiga :34
Isi array keempat :78
Isi array kelima :90
PS C:\Users\ASUS\OneDrive\Dokumen\semester 2>
```

### Deskripsi:

Program tersebut merupakan contoh dari program data koleksi yaitu program Array yang menampilkan nilai dari array pertama hingga array kelima. Program tersebut mendeklarasikan dan menginisialisasi array kemudian mencetak isi array tersebut. Output dari program tersebut yaitu menampilkan isi dari array pertama hingga array kelima.

- `#include <iostream>` ➔ merupakan input output stream header yang digunakan sebagai standar input output operasi yang digunakan di c++
- `using namespace std;` ➔ digunakan untuk mendeklarasikan/ memberitahukan kepada compiler bahwa kita akan menggunakan semua fungsi/class/file yang terdapat dalam namespace std
- `int main()` ➔ merupakan fungsi utama yang akan dieksekusi saat program dijalankan.
- `int nilai[5];` ➔ deklarasi array nilai dengan nama tipe data integer dengan ukuran 5
- `nilai[0] = 23;` ➔ mengisi array pertama dengan nilai 23
- `nilai[1] = 50;` ➔ mengisi array kedua dengan nilai 50
- `nilai[2] = 34;` ➔ mengisi array ketiga dengan nilai 34
- `nilai[3] = 78;` ➔ mengisi array keempat dengan nilai 78
- `nilai[4] = 90;` ➔ mengisi array kelima dengan nilai 90

- `cout << "Isi array pertama : " << nilai[0] << endl;` ➔ mencetak array pertama dilanjutkan dengan memanggil nilai[0]
- `cout << "Isi array kedua : " << nilai[1] << endl;` ➔ mencetak array kedua dilanjutkan dengan memanggil nilai[1]
- `cout << "Isi array ketiga : " << nilai[2] << endl;` ➔ mencetak array ketiga dilanjutkan dengan memanggil nilai[2]
- `cout << "Isi array keempat : " << nilai[3] << endl;` ➔ mencetak array keempat dilanjutkan dengan memanggil nilai[3]
- `cout << "Isi array kelima : " << nilai[4] << endl;` ➔ mencetak array kelima dilanjutkan dengan memanggil nilai[4]
- `return 0;` ➔ program akan mengembalikan (return) nilai 0 ke operating sistem yang menjalankan program tersebut

### C. Unguided/Tugas

#### 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>
#include <stdlib.h>
#include <iomanip>

using namespace std;

double penjumlahan (double a, double b)
{
    double jumlah = a + b;
    return jumlah;
}

double pengurangan (double a, double b)
{
    double kurang = a - b;
    return kurang;
}

double perkalian (double a, double b)
{
    double kali = a * b;
    return kali;
}
```

```

double pembagian (double a, double b)
{
    double bagi = a / b;
    return bagi;
}

int main()
{
    int pilihan;
    double a, b;
    cout << "\\Program Kalkulator Sederhana\\n" << endl;
    cout << "1. Penjumlahan" << endl;
    cout << "2. Pengurangan" << endl;
    cout << "3. Perkalian" << endl;
    cout << "4. Pembagian\\n" << endl;

    cout << "Masukkan Pilihan : ";
    cin >> pilihan;
    switch (pilihan)
    {
        case 1:
            cout << "\\n== Penjumlahan ==\\n" << endl;
            cout << "Masukkan bilangan pertama : ";
            cin >> a;
            cout << "Masukkan bilangan kedua : ";
            cin >> b;
            cout << "Hasil penjumlahan : " << a << " + " << b << " = " << penjumlahan
(a,b);
            break;
        case 2:
            cout << "\\n== Pengurangan ==\\n" << endl;
            cout << "Masukkan bilangan pertama : ";
            cin >> a;
            cout << "Masukkan bilangan kedua : ";
            cin >> b;
            cout << "Hasil pengurangan : " << a << " - " << b << " = " << pengurangan
(a,b);
            break;
        case 3:
            cout << "\\n== Perkalian ==\\n" << endl;
            cout << "Masukkan bilangan pertama : ";
            cin >> a;
            cout << "Masukkan bilangan kedua : ";
            cin >> b;
            cout << "Hasil perkalian : " << a << " x " << b << " = " << perkalian (a,b);
            break;
        case 4:
            cout << "\\n== Pembagian ==\\n" << endl;

```

```

cout << "Masukkan bilangan pertama : ";
cin >> a;
cout << "Masukkan bilangan kedua : ";
cin >> b;
cout << "Hasil pembagian : " << a << " ÷ " << b << " = " << pembagian(a,b);
break;
}
}

```

## Screenshots Output

```

PS C:\Users\ASUS\OneDrive\Dokumen\semester 2> & 'c:\Users\ASUS\.vscode\extensions\ms-vscode.cpptools-1.19.6-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-glwnsn5q.25p' '--stdout=Microsoft-MIEngine-Out-lzyi2qak.0yw' '--stderr=Microsoft-MIEngine-Error-inlj1pqb.mgj' '--pid=Microsoft-MIEngine-Pid-ta2wjlr.33y' '--dbgExe=c:\mingw32\bin\gdb.exe' '--interpreter=mi'
|Program Kalkulator Sederhana|

1. Penjumlahan
2. Pengurangan
3. Perkalian
4. Pembagian

Masukkan Pilihan : 1

== Penjumlahan ==

Masukkan bilangan pertama : 2
Masukkan bilangan kedua : 3
Hasil penjumlahan : 2 + 3 = 5
PS C:\Users\ASUS\OneDrive\Dokumen\semester 2>

```

\*Tidak berjudul - Notepad

File Edit Lihat

NIM : 2311102046  
Nama : Faisal Khoiruddin  
Kelas : IF-11-B

Ln 3, Col 16 100% Windows (CRLF) UTF-8

```

PS C:\Users\ASUS\OneDrive\Dokumen\semester 2> & 'c:\Users\ASUS\.vscode\extensions\ms-vscode.cpptools-1.19.6-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-dwhdqlpo.qzp' '--stdout=Microsoft-MIEngine-Out-vse0noi. fkz' '--stderr=Microsoft-MIEngine-Error-pv5loybd.15k' '--pid=Microsoft-MIEngine-Pid-3obkv4ez.0v0' '--dbgExe=c:\mingw32\bin\gdb.exe' '--interpreter=mi'
|Program Kalkulator Sederhana|

1. Penjumlahan
2. Pengurangan
3. Perkalian
4. Pembagian

Masukkan Pilihan : 2

== Pengurangan ==

Masukkan bilangan pertama : 5
Masukkan bilangan kedua : 2
Hasil pengurangan : 5 - 2 = 3
PS C:\Users\ASUS\OneDrive\Dokumen\semester 2>

```

\*Tidak berjudul - Notepad

File Edit Lihat

NIM : 2311102046  
Nama : Faisal Khoiruddin  
Kelas : IF-11-B

Ln 3, Col 16 100% Windows (CRLF) UTF-8

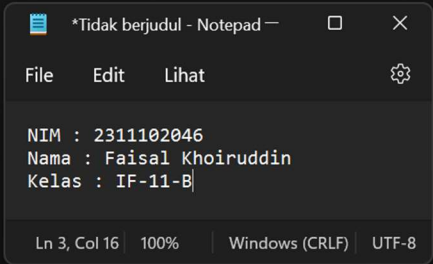
```
PS C:\Users\ASUS\OneDrive\Dokumen\semester 2> & 'c:\Users\ASUS\.vscode\extensions\ms-vscode.cpptools-1.19.6-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-vqfu3mzd.vtp' '--stdout=Microsoft-MIEngine-Out-by5sjpbo.y1d' '--stderr=Microsoft-MIEngine-Error-nxhc4oio.vzf' '--pid=Microsoft-MIEngine-Pid-nkqsm32p.it2' '--dbgExe=c:\mingw32\bin\gdb.exe' '--interpreter=mi'
|Program Kalkulator Sederhana|

1. Penjumlahan
2. Pengurangan
3. Perkalian
4. Pembagian

Masukkan Pilihan : 3

== Perkalian ==

Masukkan bilangan pertama : 4
Masukkan bilangan kedua : 2
Hasil perkalian : 4 x 2 = 8
PS C:\Users\ASUS\OneDrive\Dokumen\semester 2> |
```



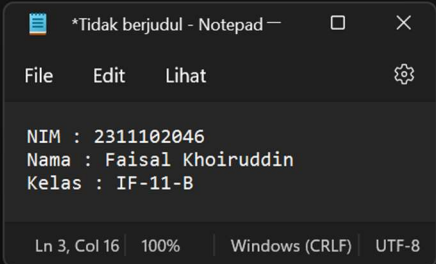
```
PS C:\Users\ASUS\OneDrive\Dokumen\semester 2> & 'c:\Users\ASUS\.vscode\extensions\ms-vscode.cpptools-1.19.6-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-dmbmh1pg.iga' '--stdout=Microsoft-MIEngine-Out-txyzmqbw.rvc' '--stderr=Microsoft-MIEngine-Error-11z4bvdh.d5b' '--pid=Microsoft-MIEngine-Pid-2lgd2fu4.iii' '--dbgExe=c:\mingw32\bin\gdb.exe' '--interpreter=mi'
|Program Kalkulator Sederhana|

1. Penjumlahan
2. Pengurangan
3. Perkalian
4. Pembagian

Masukkan Pilihan : 4

== Pembagian ==

Masukkan bilangan pertama : 6
Masukkan bilangan kedua : 3
Hasil pembagian : 6 ÷ 3 = 2
PS C:\Users\ASUS\OneDrive\Dokumen\semester 2> |
```



## Deskripsi:

Program tersebut program menggunakan tipe data primitif yaitu program operator penjumlahan, pengurangan, perkalian, dan pembagian. Program tersebut menggunakan empat fungsi yaitu fungsi pada penjumlahan, pengurangan, perkalian, dan pembagian.

Program tersebut meminta user untuk memilih salah satu operator yang diinginkan, kemudian memasukkan bilangan pertama dan bilangan kedua. Program tersebut menampilkan hasil dari operasi kedua bilangan tersebut.

- `#include <iostream>` ➔ merupakan input output stream header yang digunakan sebagai standar input output operasi yang digunakan di c++
- `#include <stdlib.h>` ➔ header ini menyediakan banyak fungsi-fungsi dengan tujuan umum seperti aritmatika, konversi, pengurutan, pencarian, manajemen memori dinamis, pembuatan bulangan acak dan lain sebagainya.
- `#include <iomanip>` ➔ standar untuk menentukan beberapa manipulator yang

masing-masing mengambil satu argumen.

- `using namespace std;` ➔ digunakan untuk mendeklarasikan/ memberitahukan kepada compiler bahwa kita akan menggunakan semua fungsi/class/file yang terdapat dalam namespace `std`
- `double penjumlahan (double a, double b)` ➔ ➔ deklarasi nama fungsi yaitu penjumlahan dengan nama tipe data `double` dan parameter variabel `a` dan `b` dengan nama tipe data `double`
- `double jumlah = a + b;` ➔ rumus dari penjumlahan yaitu  $jumlah = a + b$
- `return jumlah;` ➔ untuk mengembalikan nilai dari variabel fungsi `jumlah`
- `double pengurangan (double a, double b)` ➔ deklarasi nama fungsi yaitu pengurangan dengan nama tipe data `double` dan parameter variabel `a` dan `b` dengan nama tipe data `double`
- `double kurang = a - b;` ➔ rumus dari perkalian yaitu  $kurang = a - b$
- `return kurang;` ➔ untuk mengembalikan nilai dari variabel fungsi `kurang`
- `double perkalian (double a, double b)` ➔ deklarasi nama fungsi yaitu perkalian dengan nama tipe data `double` dan parameter variabel `a` dan `b` dengan nama tipe data `double`
- `double kali = a * b;` ➔ rumus dari perkalian yaitu  $kali = a * b$
- `return kali;` ➔ untuk mengembalikan nilai dari variabel fungsi `kali`
- `double pembagian (double a, double b)` ➔ deklarasi nama fungsi yaitu pembagian dengan nama tipe data `double` dan parameter variabel `a` dan `b` dengan nama tipe data `double`
- `double bagi = a / b;` ➔ rumus dari perkalian yaitu  $bagi = a / b$
- `return bagi;` ➔ untuk mengembalikan nilai dari variabel fungsi `bagi`
- `int main()` ➔ merupakan fungsi utama yang akan dieksekusi saat program dijalankan.
- `int pilihan;` ➔ deklarasi variabel untuk menyimpan pilihan operasi dengan nama tipe data `integer`

- `double a, b;` ➔ deklarasi variable a dan b untuk menyimpan dua bilangan yang akan dioperasikan
- `cout << "|Program Kalkulator Sederhana|\n" << endl;` ➔ cout menampilkan judul program yaitu program kalkulator sederhana
- `cout << "1. Penjumlahan" << endl;` ➔ cout menampilkan pilihan menu pertama yaitu Penjumlahan
- `cout << "2. Pengurangan" << endl;` ➔ cout menampilkan pilihan menu kedua yaitu Pengurangan
- `cout << "3. Perkalian" << endl;` ➔ cout menampilkan pilihan menu ketiga yaitu Perkalian
- `cout << "4. Pembagian\n" << endl;` ➔ cout menampilkan pilihan menu keempat yaitu Pembagian
- `cout << "Masukkan Pilihan : ";` ➔ cout menampilkan masukkan pilihan
- `cin >> pilihan;` ➔ menginput pilihan menu dari 1 hingga 4 sesuai keinginan pengguna
- `switch (pilihan)` ➔ melakukan operasi menggunakan switch
- `case 1:` ➔ jika pengguna memilih operasi satu, maka akan melakukan operasi penjumlahan
- `cout << "\n== Penjumlahan ==\n" << endl;` ➔ cout menampilkan pernyataan judul operasi yaitu penjumlahan
- `cout << "Masukkan bilangan pertama : ";` ➔ cout menampilkan masukkan bilangan pertama dan meminta menginput bilangan pertama
- `cin >> a;` ➔ menyimpan input pengguna pada variabel a
- `cout << "Masukkan bilangan kedua : ";` ➔ cout menampilkan pernyataan masukkan bilangan kedua dan meminta menginput bilangan kedua
- `cin >> b;` ➔ menyimpan input pengguna pada variabel b
- `cout << "Hasil penjumlahan : " << a << " + " << b << " = " << penjumlahan (a,b);`  
➔ menampilkan hasil penjumlahan



- `break;` ➔ menghentikan eksekusi dan keluar dari blok switch
- case 2: ➔ jika pengguna memilih dua, maka akan melakukan operasi pengurangan
- `cout << "\n== Pengurangan ==\n" << endl;` ➔ cout menampilkan pernyataan judul operasi yaitu pengurangan
- `cout << "Masukkan bilangan pertama : ";` ➔ cout menampilkan masukkan bilangan pertama dan meminta menginput bilangan pertama
- `cin >> a;` ➔ menyimpan input pengguna pada variabel a
- `cout << "Masukkan bilangan kedua : ";` ➔ cout menampilkan pernyataan masukkan bilangan kedua dan meminta menginput bilangan kedua
- `cin >> b;` ➔ menyimpan input pengguna pada variabel b
- `cout << "Hasil pengurangan : " << a << " - " << b << " = " << pengurangan (a,b);`  
➔ menampilkan hasil pengurangan
- `break;` ➔ menghentikan eksekusi dan keluar dari blok switch
- case 3: ➔ jika pengguna memilih tiga, maka akan melakukan operasi perkalian
- `cout << "\n== Perkalian ==\n" << endl;` ➔ cout menampilkan pernyataan judul operasi yaitu perkalian
- `cout << "Masukkan bilangan pertama : ";` ➔ cout menampilkan masukkan bilangan pertama dan meminta menginput bilangan pertama
- `cin >> a;` ➔ menyimpan input pengguna pada variabel a
- `cout << "Masukkan bilangan kedua : ";` ➔ cout menampilkan pernyataan masukkan bilangan kedua dan meminta menginput bilangan kedua
- `cin >> b;` ➔ menyimpan input pengguna pada variabel b
- `cout << "Hasil perkalian : " << a << " x " << b << " = " << perkalian (a,b);` ➔ menampilkan hasil perkalian
- `break;` ➔ menghentikan eksekusi dan keluar dari blok switch
- case 4: ➔ jika pengguna memilih empat, maka akan melakukan operasi pembagian

- `cout << "\n== Pembagian ==\n" << endl;` ➔ `cout` menampilkan pernyataan judul operasi yaitu pembagian
- `cout << "Masukkan bilangan pertama : ";` ➔ `cout` menampilkan masukkan bilangan pertama dan meminta menginput bilangan pertama
- `cin >> a;` ➔ menyimpan input pengguna pada variabel `a`
- `cout << "Masukkan bilangan kedua : ";` ➔ `cout` menampilkan pernyataan masukkan bilangan kedua dan meminta menginput bilangan kedua
- `cin >> b;` ➔ menyimpan input pengguna pada variabel `b`
- `cout << "Hasil pembagian : " << a << " ÷ " << b << " = " << pembagian (a,b);` ➔ menampilkan hasil pembagian
- `break;` ➔ menghentikan eksekusi dan keluar dari blok `switch`

**Kesimpulan dari tipe data primitif.** Tipe data primitif yaitu tipe data yang sudah ditentukan oleh bahasa pemrograman. Dalam bahasa pemrograman C++ tipe data primitif digunakan untuk menyimpan data yang sederhana. Beberapa contoh tipe data primitif yaitu : integer yaitu tipe data untuk menyimpan bilangan bulat, float yaitu tipe data untuk menyimpan bilangan desimal, char yaitu tipe data untuk menyimpan data sebuah huruf, dan boolean yaitu tipe data untuk menyimpan data berupa true atau false.

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

### a. Struct

- Struct merupakan salah satu fitur bahasa pemrograman C/C++, merupakan keyword yang memungkinkan kita untuk membuat sebuah deklarasi untuk membuat pengelompokan variabel dengan tipe data yang berbeda.
- Struct memungkinkan membuat deklarasi yang di dalamnya dapat terdapat banyak variabel, hasil deklarasi ini akan menghasilkan *user-defined data type* yaitu *tipe structure* yang berguna untuk mendirikan object.
- Struct sebenarnya hanya deklarasi untuk membuat sebuah tipe data baru yang didirikan oleh programmer sebagai data structure. Data structure tersebut akan digunakan sebagai pembuatan object, pembuatan object itu

dapat dilakukan didalam deklarasi struct atau diluar deklarasi struct.

- Deklarasi struct sama sekali tidak memakan memori, struct bisa dikatakan hanyalah sebuah rancangan, deklarasi struct dibuat untuk membuat tipe data yang dibuat oleh programmer.
- Pada C++, Struct sama dengan Class kecuali bahwa anggotanya secara public default.

#### b. Class

- *Class* adalah salah satu dari konsep OOP yang digunakan untuk membungkus data abstraksi *procedural* sebagai deskripsi tergeneralisir atau rancangan dari sebuah *object* untuk mendefinisikan atau menggambarkan isi dan tingkah laku sebagai entitas dari *object*.
- *Class* memiliki kemampuan lebih daripada Struct, seperti dimungkinkan untuk memuat method, inheritance dan lain-lain.
- Terdapat banyak fasilitas yang disediakan oleh *class*, yaitu dapat menampung *membervariabel*, *function*, *constructor*, *desctructor*, *overloading* dan lain-lain. Diluar definisi *class* kita juga dimungkinkan untuk membuat relalasi seperti *inheritance* dan *overriding*.
- Pada *class* kita juga dimungkinkan untuk mendirikan *function prototype*, dan mendefinisikanya di luar definisi *class*. Untuk memberi definisi pada *function* tersebut kita dapat melakukannya dengan menggunakan “*Scope Resolution Operator ::*“, yang diletakan di antara nama *class* dan nama *member function*.
- Pada Class akses default bersifat pribadi

#### ▪ Class

```
#include <iostream>
#include <string>

using namespace std;

// Membuat class dengan nama 'Mahasiswa'
class Data_diri {
public: // Mengatur akses ke public
```

```

string namalengkap; // Membuat atribut 'nama'
string jeniskelamin; // Membuat atribut 'jurusan'
string tempattanggalLahir; // Membuat atribut 'angkatan'
string alamatlengkap;
string telp;
string agama;
string tinggibadan;
string warnakulit;
string kewarganegaraan;
string suku;

// Membuat metode untuk menampilkan detail mahasiswa
void tampilkanDetail() {
    cout << "Nama lengkap" : " << namalengkap << "\n";
    cout << "Jenis kelamin" : " << jeniskelamin << "\n";
    cout << "Tempat & Tanggal Lahir" : " << tempattanggalLahir << "\n";
    cout << "Alamat lengkap" : " << alamatlengkap << "\n";
    cout << "No. Telp" : " << telp << "\n";
    cout << "Agama" : " << agama << "\n";
    cout << "Tinggi badan" : " << tinggibadan << "\n";
    cout << "Warna kulit" : " << warnakulit << "\n";
    cout << "Kewarganegaraan" : " << kewarganegaraan << "\n";
    cout << "Suku" : " << suku << "\n";
}
};

int main() {
    // Membuat objek dari class 'Mahasiswa'
    Data_diri org1;
    org1.namalengkap = "Andi Budiman";
    org1.jeniskelamin = "Laki - Laki";
    org1.tempattanggalLahir = "Semarang, 2 Januari 2000";
    org1.alamatlengkap = "Jl. Jend. Soedirman No.2 Karang Jambu, Kab. Semarang, Jawa Tengah";
    org1.telp = "081132932746";
    org1.agama = "Islam";
    org1.tinggibadan = "170";
    org1.warnakulit = "Sawo matang";
    org1.kewarganegaraan = "Indonesia";
    org1.suku = "Jawa";

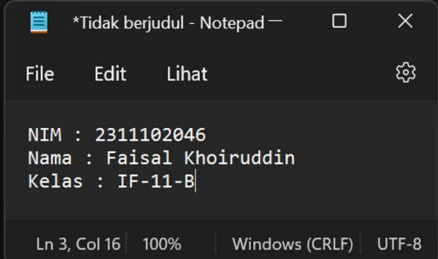
    // Menampilkan detail mahasiswa menggunakan metode 'tampilkanDetail'
    org1.tampilkanDetail();

    return 0;
}

```

## Screenshots Output

```
PS C:\Users\ASUS\OneDrive\Dokumen\semester 2> & 'c:\Users\ASUS\.vscode\extensions\ms-vscode.cpptools-1.19.6-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-0qe4xzvc.pft' '--stdout=Microsoft-MIEngine-Out-cyeols5g.1ua' '--stderr=Microsoft-MIEngine-Error-0oyy1wmv.nez' '--pid=Microsoft-MIEngine-Pid-cofgduwx.lbi' '--dbgExe=c:\mingw32\bin\gdb.exe' '--interpreter=mi'
Nama lengkap      : Andi Budiman
Jenis kelamin     : Laki - Laki
Tempat & Tanggal Lahir : Semarang, 2 Januari 2000
Alamat lengkap    : Jl. Jend. Soedirman No.2 Karang Jambu, Kab. Semarang, Jawa Tengah
No. Telp          : 081132932746
Agama             : Islam
Tinggi badan      : 170
Warna kulit       : Sawo matang
Kewarganegaraan   : Indonesia
Suku              : Jawa
PS C:\Users\ASUS\OneDrive\Dokumen\semester 2>
```



### Deskripsi:

Program tersebut merupakan contoh dari program data abstrak yaitu program Class yang menampilkan data diri mahasiswa. Program tersebut merupakan program sederhana yang menggunakan OOP (Object Oriented Programming) untuk membuat sebuah class. Output dari program tersebut yaitu menampilkan data diri dari mahasiswa.

- `#include <iostream>` ➔ merupakan input output stream header yang digunakan sebagai standar input output operasi yang digunakan di c++
- `#include <string>` ➔ mengimpor library string yang berisi fungsi untuk mengelola string
- `using namespace std;` ➔ digunakan untuk mendeklarasikan/ memberitahukan kepada compiler bahwa kita akan menggunakan semua fungsi/class/file yang terdapat dalam namespace std
- `class Data_diri` ➔ membuat kelas dengan nama Data\_diri
- `public:` ➔ mengatur akses ke publik
- `string namalengkap;` ➔ membuat variabel namalengkap dengan tipe data string yang merupakan member variable dari class tersebut
- `string jeniskelamin;` ➔ membuat variabel jeniskelamin dengan tipe data string yang merupakan member variable dari class tersebut
- `string tempattanggalLahir;` ➔ membuat variabel tempattanggalLahir dengan tipe data string yang merupakan member variable dari class tersebut
- `string alamatlengkap;` ➔ membuat variabel alamatlengkap dengan tipe data string yang merupakan member variable dari class tersebut
- `string telp;` ➔ membuat variabel telp dengan tipe data string yang merupakan member variable dari class tersebut

- `string agama;` ➔ membuat variabel agama dengan tipe data string yang merupakan member variable dari class tersebut
- `string tinggibadan;` ➔ membuat variabel tinggibadan dengan tipe data string yang merupakan member variable dari class tersebut
- `string warnakulit;` ➔ membuat variabel warnakulit dengan tipe data string yang merupakan member variable dari class tersebut
- `string kewarganegaraan;` ➔ membuat variabel kewarganegaraan dengan tipe data string yang merupakan member variable dari class tersebut
- `string suku;` ➔ membuat variabel suku dengan tipe data string yang merupakan member variable dari class tersebut
- `void tampilkanDetail()` ➔ membuat metode untuk menampilkan detail mahasiswa dan mendefinisikan fungsi `tampilkanDetail` yang tidak mengembalikan nilai
- `cout << "Nama lengkap : " << namalengkap << "\n";` ➔ menampilkan string pernyataan nama lengkap dan nilai dari variabel `namalengkap`
- `cout << "Jenis kelamin : " << jeniskelamin << "\n";` ➔ menampilkan string pernyataan Jenis kelamin dan nilai dari variabel `jeniskelamin`
- `cout << "Tempat & Tanggal Lahir : " << tempattanggalLahir << "\n";` ➔ menampilkan string pernyataan Tempat & Tanggal Lahir : dan nilai dari variabel `tempattanggalLahir`
- `cout << "Alamat lengkap : " << alamatlengkap << "\n";` ➔ menampilkan string pernyataan Alamat lengkap dan nilai dari variabel `alamatlengkap`
- `cout << "No. Telp : " << telp << "\n";` ➔ menampilkan string pernyataan No. Telp dan nilai dari variabel `telp`
- `cout << "Agama : " << agama << "\n";` ➔ menampilkan string pernyataan Agama dan nilai dari variabel `agama`
- `cout << "Tinggi badan : " << tinggibadan << "\n";` ➔ menampilkan string pernyataan Tinggi badan dan nilai dari `tinggibadan`
- `cout << "Warna kulit : " << warnakulit << "\n";` ➔ menampilkan string pernyataan Warna kulit dan nilai dari `warnakulit`
- `cout << "Kewarganegaraan : " << kewarganegaraan << "\n";` ➔ menampilkan string pernyataan Kewarganegaraan dan nilai dari `kewarganegaraan`
- `cout << "Suku : " << suku << "\n";` ➔ menampilkan string pernyataan Suku dan nilai dari `suku`
- `int main()` ➔ merupakan fungsi utama yang akan dieksekusi saat program dijalankan.

- Data\_diri org1; ➔ Membuat objek dari class Data\_diri
- org1.namalengkap = "Andi Budiman"; ➔ mengisi member namalengkap pada objek org1
- org1.jeniskelamin = "Laki - Laki"; ➔ mengisi member jeniskelamin pada objek org1
- org1.tempattanggalLahir = "Semarang, 2 Januari 2000"; ➔ mengisi member tempattanggalLahir pada objek org1
- org1.alamatlengkap = "Jl. Jend. Soedirman No.2 Karang Jambu, Kab. Semarang, Jawa Tengah"; ➔ mengisi member alamatlengkap pada objek org1
- org1.telp = "081132932746"; ➔ mengisi member telp pada objek org1
- org1.agama = "Islam"; ➔ mengisi member agama pada objek org1
- org1.tinggibadan = "170"; ➔ mengisi member tinggibadan pada objek org1
- org1.warnakulit = "Sawo matang"; ➔ mengisi member tinggibadan pada objek org1
- org1.kewarganegaraan = "Indonesia"; ➔ mengisi member tinggibadan pada objek org1
- org1.suku = "Jawa"; ➔ mengisi member tinggibadan pada objek org1
- org1.tampilkanDetail(); ➔ Menampilkan detail mahasiswa menggunakan metode tampilkanDetail
- return 0; ➔ program akan mengembalikan (return) nilai 0 ke operating sistem yang menjalankan program tersebut

#### ▪ Struct

```
#include <stdio.h>

// Membuat struct dengan nama 'Mahasiswa'
struct BiodataMhs {
    const char *nim;
    const char *namalngkp; // Membuat atribut 'nama'
    const char *namapgln;
    const char *TTL;
    int umur;
    const char *ank_ke;
    const char *jnsklmn;
    const char *agama;
    const char *telp;
    const char *kwrgn;
    const char *hobi;
    const char *alamat; // Membuat atribut 'jurusan'
```

```

    const char *angkatan; // Membuat atribut 'angkatan'
};

int main() {
    // Membuat objek dari struct 'Mahasiswa'
    struct BiodataMhs Mhs1;
    Mhs1.nim = "2190230073";
    Mhs1.namalngkp = "Toni Wijaya";
    Mhs1.namapgl = "Toni";
    Mhs1.TTL = "Surakarta, 2 Agustus 2001";
    Mhs1.umur = 23;
    Mhs1.ank_ke = "2 dari dua bersaudara";
    Mhs1.jnsklmn = "Laki-Laki";
    Mhs1.agama = "Islam";
    Mhs1.tlp = "081913128091";
    Mhs1.kwrgn = "Indonesia";
    Mhs1.hobi = "Menggambar";
    Mhs1.alamat = "Jl. Yosodipuro No.2 Kec. Banjarsari, Kota Surakarta, Jawa Tengah";
    Mhs1.angkatan = "2021";

    // Menampilkan detail mahasiswa
    printf("-----Biodata Mahasiswa-----\n");
    printf("\nNIM          : %s\n", Mhs1.nim);
    printf("Nama lengkap      : %s\n", Mhs1.namalngkp);
    printf("Nama panggilan     : %s\n", Mhs1.namapgl);
    printf("Tempat & Tanggal Lahir : %s\n", Mhs1.TTL);
    printf("Umur              : %d\n", Mhs1.umur);
    printf("Anak ke-          : %s\n", Mhs1.ank_ke);
    printf("Jenis kelamin      : %s\n", Mhs1.jnsklmn);
    printf("Agama            : %s\n", Mhs1.agama);
    printf("No. Telepon        : %s\n", Mhs1.tlp);
    printf("Kewarganegaraan    : %s\n", Mhs1.kwrgn);
    printf("Hobi              : %s\n", Mhs1.hobi);
    printf("Alamat            : %s\n", Mhs1.alamat);
    printf("Angkatan          : %s\n", Mhs1.angkatan);

    // Mencetak alamat memori dari name dan address
    printf("\nAlamat memori dari biodata Mahasiswa 1:\n");
    printf("NIM          : %p\n", Mhs1.nim);
    printf("Nama lengkap : %p\n", Mhs1.namalngkp);
    printf("Nama panggilan : %p\n", Mhs1.namapgl);
    printf("Tempat & Tanggal Lahir : %p\n", Mhs1.TTL);
    printf("Anak ke-      : %p\n", Mhs1.ank_ke);
    printf("Jenis kelamin : %p\n", Mhs1.jnsklmn);
    printf("Agama         : %p\n", Mhs1.agama);
    printf("No. Telepon    : %p\n", Mhs1.tlp);
    printf("Kewarganegaraan : %p\n", Mhs1.kwrgn);

```



```

printf("Hobi          : %p\n", Mhs1.hobi);
printf("Alamat       : %p\n", Mhs1.alamat);
printf("Angkatan     : %p\n", Mhs1.angkatan);

return 0;
}

```

## Screenshots Output

```

PS C:\Users\ASUS\OneDrive\Dokumen\semester 2> & 'c:\Users\ASUS\.vscode\extensions\ms-vscode.cpptools-1.19.7-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-lj3g0jrv.pef' '--stdout=Microsoft-MIEngine-Out-4gtxhe3y.eu5' '--stderr=Microsoft-MIEngine-Error-q2eivw50.yrj' '--pid=Microsoft-MIEngine-Pid-axekwzmc.wux' '--dbgExe=c:\mingw32\bin\gdb.exe' '--interpreter=mi'
-----Biodata Mahasiswa-----
NIM                : 2190230073
Nama lengkap      : Toni Wijaya
Nama panggilan    : Toni
Tempat & Tanggal Lahir : Surakarta, 2 Agustus 2001
Umur              : 23
Anak ke-          : 2 dari dua bersaudara
Jenis kelamin     : Laki-Laki
Agama            : Islam
No. Telepon       : 081913128091
Kewarganegaraan   : Indonesia
Hobi              : Menggambar
Alamat           : Jl. Yosodipuro No.2 Kec. Banjarsari, Kota Surakarta, Jawa Tengah
Angkatan         : 2021

```

```

Alamat memori dari biodata Mahasiswa 1:
NIM                : 00404044
Nama lengkap      : 0040404F
Nama panggilan    : 00404058
Tempat & Tanggal Lahir : 00404060
Anak ke-          : 0040407A
Jenis kelamin     : 00404090
Agama            : 0040409A
No. Telepon       : 004040A0
Kewarganegaraan   : 004040AD
Hobi              : 004040B7
Alamat           : 004040C4
Angkatan         : 00404105
PS C:\Users\ASUS\OneDrive\Dokumen\semester 2>

```

## Deskripsi:

Program tersebut merupakan contoh dari program data abstrak yaitu program Struct yang menampilkan biodata mahasiswa. Program tersebut sederhana yang menggunakan struct untuk membuat sebuah struktur data BiodataMhs yang berisi beberapa member. Output dari program tersebut yaitu menampilkan biodata mahasiswa.

- `#include <stdio.h>` ➔ merupakan pre-processor untuk mengakses file yang bernama `stdio.h` agar codingan kita dapat mengakses data-data yang ada disitu
- `struct BiodataMhs` ➔ membuat struct dengan nama `BiodataMhs`
- `const char *nim;` ➔ membuat member `nim` dengan tipe data pointer ke `char`
- `const char *namaIngk;` ➔ membuat member `namaIngk` dengan tipe data pointer ke `char`

- `const char *namapgl;` ➔ membuat member `namapgl` dengan tipe data pointer ke `char`
- `const char *TTL;` ➔ membuat member `TTL` dengan tipe data pointer ke `char`
- `int umur;` ➔ membuat member `umur` dengan nama tipe data integer
- `const char *ank_ke;` ➔ membuat member `ank_ke` dengan tipe data pointer ke `char`
- `const char *jnsklmn;` ➔ membuat member `jnsklmn` dengan tipe data pointer ke `char`
- `const char *agama;` ➔ membuat member `agama` dengan tipe data pointer ke `char`
- `const char* tlp;` ➔ membuat member `tlp` dengan tipe data pointer ke `char`
- `const char *kwrng;` ➔ membuat member `kwrng` dengan tipe data pointer ke `char`
- `const char *hobi;` ➔ membuat member `hobi` dengan tipe data pointer ke `char`
- `const char *alamat;` ➔ membuat member `alamat` dengan tipe data pointer ke `char`
- `const char *angkatan;` ➔ membuat member `angkatan` dengan tipe data pointer ke `char`
- `int main()` ➔ merupakan fungsi utama yang akan dieksekusi saat program dijalankan.
- `struct BiodataMhs Mhs1;` ➔ membuat objek dari `struct BiodataMhs`
- `Mhs1.nim = "2190230073";` ➔ mengisi member `nim` pada objek `Mhs1`
- `Mhs1.namalngkp = "Toni Wijaya";` ➔ mengisi member `namalngkp` pada objek `Mhs1`
- `Mhs1.namapgl = "Toni";` ➔ mengisi member `namapgl` pada objek `Mhs1`
- `Mhs1.TTL = "Surakarta, 2 Agustus 2001";` ➔ mengisi member `namapgl` pada objek `Mhs1`
- `Mhs1.umur = 23;` ➔ mengisi member `umur` pada objek `Mhs1`
- `Mhs1.ank_ke = "2 dari dua bersaudara";` ➔ mengisi member `ank_ke` pada objek `Mhs1`
- `Mhs1.jnsklmn = "Laki-Laki";` ➔ mengisi member `jnsklmn` pada objek `Mhs1`
- `Mhs1.agama = "Islam";` ➔ mengisi member `agama` pada objek `Mhs1`
- `Mhs1.tlp = "081913128091";` ➔ mengisi member `tlp` pada objek `Mhs1`
- `Mhs1.kwrng = "Indonesia";` ➔ mengisi member `kwrng` pada objek `Mhs1`
- `Mhs1.hobi = "Menggambar";` ➔ mengisi member `hobi` pada objek `Mhs1`

- `Mhs1.alamat = "Jl. Yosodipuro No.2 Kec. Banjarsari, Kota Surakarta, Jawa Tengah"; ➔` mengisi member alamat pada objek `Mhs1`
- `Mhs1.angkatan = "2021"; ➔` mengisi member alamat pada objek `Mhs1`
- `printf("-----Biodata Mahasiswa-----\n"); ➔` menampilkan judul program Biodata Mahasiswa
- `printf("\nNIM : %s\n", Mhs1.nim); ➔` menampilkan string "NIM : " dan nilai dari member `nim` pada objek `Mhs1`
- `printf("Nama lengkap : %s\n", Mhs1.namalngkp); ➔` menampilkan string " Nama lengkap : " dan nilai dari member `namalngkp` pada objek `Mhs1`
- `printf("Nama panggilan : %s\n", Mhs1.namapgl); ➔` menampilkan string " Nama panggilan : " dan nilai dari member `namapgl` pada objek `Mhs1`
- `printf("Tempat & Tanggal Lahir : %s\n", Mhs1.TTL); ➔` menampilkan string " Tempat & Tanggal Lahir : " dan nilai dari member `TTL` pada objek `Mhs1`
- `printf("Umur : %d\n", Mhs1.umur); ➔` menampilkan string " Umur: " dan nilai dari member `umur` pada objek `Mhs1`
- `printf("Anak ke- : %s\n", Mhs1.ank_ke); ➔` menampilkan string " Anak ke- : " dan nilai dari member `ank_ke` pada objek `Mhs1`
- `printf("Jenis kelamin : %s\n", Mhs1.jnsklmn); ➔` menampilkan string " Jenis kelamin : " dan nilai dari member `jnsklmn` pada objek `Mhs1`
- `printf("Agama : %s\n", Mhs1.agama); ➔` menampilkan string " Agama: " dan nilai dari member `agama` pada objek `Mhs1`
- `printf("No. Telepon : %s\n", Mhs1.tlp); ➔` menampilkan string " No. Telepon: " dan nilai dari member `tlp` pada objek `Mhs1`
- `printf("Kewarganegaraan : %s\n", Mhs1.kwrgn); ➔` menampilkan string " Kewarganegaraan : " dan nilai dari member `kwrgn` pada objek `Mhs1`
- `printf("Hobi : %s\n", Mhs1.hobi); ➔` menampilkan string " Hobi : " dan nilai dari member `hobi` pada objek `Mhs1`
- `printf("Alamat : %s\n", Mhs1.alamat); ➔` menampilkan string " Alamat : " dan nilai dari member `alamat` pada objek `Mhs1`
- `printf("Angkatan : %s\n", Mhs1.angkatan); ➔` menampilkan string " Angkatan : " dan nilai dari member `angkatan` pada objek `Mhs1`
- `printf("\nAlamat memori dari biodata Mahasiswa 1:\n"); ➔` menampilkan string judul `nAlamat memori dari biodata Mahasiswa 1:`
- `printf("NIM : %p\n", Mhs1.nim); ➔` menampilkan string "NIM : " dan alamat memori dari member `nim` pada objek `Mhs1`

- `printf("Nama lengkap : %p\n", Mhs1.namalngkp);` ➔ menampilkan string " Nama lengkap : " dan alamat memori dari member `namalngkp` pada objek `Mhs1`
- `printf("Nama panggilan : %p\n", Mhs1.namapgl);` ➔ menampilkan string " Nama panggilan : " dan alamat memori dari member `namapgl` pada objek `Mhs1`
- `printf("Tempat & Tanggal Lahir : %p\n", Mhs1.TTL);` ➔ menampilkan string " Tempat & Tanggal Lahir : " dan alamat memori dari member `TTL` pada objek `Mhs1`
- `printf("Anak ke- : %p\n", Mhs1.ank_ke);` ➔ menampilkan string " Anak ke- : " dan alamat memori dari member `ank_ke` pada objek `Mhs1`
- `printf("Jenis kelamin : %p\n", Mhs1.jnsklmn);` ➔ menampilkan string " Jenis kelamin : " dan alamat memori dari member `jnsklmn` pada objek `Mhs1`
- `printf("Agama : %p\n", Mhs1.agama);` ➔ menampilkan string " Agama: " dan alamat memori dari member `agama` pada objek `Mhs1`
- `printf("No. Telepon : %p\n", Mhs1.tlp);` ➔ menampilkan string " No. Telepon: " dan alamat memori dari member `tlp` pada objek `Mhs1`
- `printf("Kewarganegaraan : %p\n", Mhs1.kwrgn);` ➔ menampilkan string " Kewarganegaraan: " dan alamat memori dari member `kwrgn` pada objek `Mhs1`
- `printf("Hobi : %p\n", Mhs1.hobi);` ➔ menampilkan string " Hobi: " dan alamat memori dari member `hobi` pada objek `Mhs1`
- `printf("Alamat : %p\n", Mhs1.alamat);` ➔ menampilkan string " Alamat: " dan alamat memori dari member `alamat` pada objek `Mhs1`
- `printf("Angkatan : %p\n", Mhs1.angkatan);` ➔ menampilkan string " Angkatan: " dan alamat memori dari member `angkatan` pada objek `Mhs1`
- `return 0;` ➔ program akan mengembalikan (return) nilai 0 ke operating sistem yang menjalankan program tersebut

### Unguided 3 :

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

dengan map

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

using namespace std;
int main() {
    // Membuat map dengan key berupa string dan value berupa int
```

```

map<string, int> umur;

// Menambahkan 10 pasangan key-value ke dalam map
umur["Ali"] = 18;
umur["Banu"] = 34;
umur["Celine"] = 20;
umur["Dodi"] = 24;
umur["Edi"] = 45;
umur["Farah"] = 25;
umur["Gani"] = 27;
umur["Hakan"] = 29;
umur["Irma"] = 30;
umur["Jaka"] = 31;

// Menampilkan semua elemen map
for(auto i = umur.begin(); i != umur.end(); i++) {

    cout << "Nama : " << i->first << ", Umur : " << i->second << endl;
}

return 0;
}

```

## Screenshots Output

```

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

PS C:\Users\ASUS\OneDrive\Dokumen\semester 2> & 'c:\Users\ASUS\.vscode\extensions\ms-vscode.cpptools-1.19.7-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-2gy2oul2.qwx' '--stdout=Microsoft-MIEngine-Out-jz412b44.odu' '--stderr=Microsoft-MIEngine-Error-gmvoyoh3.sgt' '--pid=Microsoft-MIEngine-Pid-pnh3xnsh.xbd' '--dbgExe=c:\mingw32\bin\gdb.exe' '--interpreter=mi'
Nama : Ali, Umur : 18
Nama : Banu, Umur : 34
Nama : Celine, Umur : 20
Nama : Dodi, Umur : 24
Nama : Edi, Umur : 45
Nama : Farah, Umur : 25
Nama : Gani, Umur : 27
Nama : Hakan, Umur : 29
Nama : Irma, Umur : 30
Nama : Jaka, Umur : 31
PS C:\Users\ASUS\OneDrive\Dokumen\semester 2>

```

\*Tidak berjudul - Notepad

File Edit Lihat

NIM : 2311102046  
 Nama : Faisal Khoiruddin  
 Kelas : IF-11-B

Ln 3, Col 16 | 100% | Windows (CRLF) | UTF-8

## Deskripsi:

Program tersebut merupakan contoh dari program data koleksi yaitu program Map. Program tersebut merupakan program sederhana yang menggunakan map untuk menyimpan dan menampilkan data umur dari beberapa orang. Program tersebut menampilkan nama dan umur setiap orang.

- `#include <iostream>` ➔ merupakan input output stream header yang digunakan sebagai standar input output operasi yang digunakan di c++
- `#include <map>` ➔ berfungsi untuk mengimpor library map yang berisi fungsi untuk mengelola map

- `using namespace std;` ➔ digunakan untuk mendeklarasikan/ memberitahukan kepada compiler bahwa kita akan menggunakan semua fungsi/class/file yang terdapat dalam namespace std
- `int main()` ➔ merupakan fungsi utama yang akan dieksekusi saat program dijalankan.
- `map<string, int> umur;` ➔ membuat map dengan key berupa string dan value berupa int
- `umur["Ali"] = 18;` ➔ menambahkan pasangan key-value "Ali"-18 ke dalam map umur
- `umur["Banu"] = 34;` ➔ menambahkan pasangan key-value " Banu"-34 ke dalam map umur
- `umur["Celine"] = 20;` ➔ menambahkan pasangan key-value "Celine"-20 ke dalam map umur
- `umur["Dodi"] = 24;` ➔ Menambahkan pasangan key-value "Dodi"-24 ke dalam map umur
- `umur["Edi"] = 45;` ➔ Menambahkan pasangan key-value "Edi"-45 ke dalam map umur
- `umur["Farah"] = 25;` ➔ Menambahkan pasangan key-value "Farah"-25 ke dalam map umur
- `umur["Gani"] = 27;` ➔ enambahkan pasangan key-value "Gani"-27 ke dalam map umur
- `umur["Hakan"] = 29;` ➔ Menambahkan pasangan key-value "Hakan"-29 ke dalam map umur
- `umur["Irma"] = 30;` ➔ Menambahkan pasangan key-value "Irma"-30 ke dalam map umur
- `umur["Jaka"] = 31;` ➔ Menambahkan pasangan key-value "Jaka"-31 ke dalam map umur
- `for(auto i = umur.begin(); i != umur.end(); i++)` ➔ menampilkan semua elemen map dan perulangan for melalui setiap elemen map umur
- `cout << "Nama : " << i->first << ", Umur : " << i->second << endl;` ➔ cout menampilkan string "Nama : ", key dari elemen map, string ", Umur : ", dan value dari elemen map
- `return 0;` ➔ program akan mengembalikan (return) nilai 0 ke operating sistem yang menjalankan program tersebut

Perbedaan dari array dengan map yaitu :

Array	Map
Array adalah kumpulan elemen dari tipe data yang sama	Map adalah struktur hash dari pasangan kunci dan nilai.
Indeks daftar merupakan bilangan bulat mulai dari 0	Kunci Map dapat berupa tipe data apa pun
Elemen diakses melalui indeks	Elemen diakses melalui nilai kunci
Array dapat berupa 1D, 2D atau multidimensi	Map dapat berupa multimap, Unordered Multimap, Unordered map, dll
Ukuran array harus ditentukan selama deklarasi array.	Ukuran peta bersifat dinamis.

#### D. Kesimpulan

Tipe data yaitu sebuah pengklasifikasian data berdasarkan jenis data tersebut. Beberapa tipe data yaitu:

1. Tipe data primitif yaitu tipe data yang sudah ditentukan oleh bahasa pemrograman, tipe data primitif 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 yaitu :

- a. Int. Int yaitu tipe data yang digunakan untuk menyimpan bilangan bulat.
  - b. Float. Float merupakan tipe data yang digunakan untuk menyimpan bilangan desimal.
  - c. Char. Char berfungsi untuk menyimpan data berupa sebuah huruf.
  - d. hanya memiliki dua nilai yaitu true(benar) dan false(salah).
2. Tipe data abstrak merupakan tipe data yang dibentuk oleh programmer itu sendiri. 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. Perbedaan antara Struct dan Class yaitu pada akses default Struct bersifat publik dan akses default Class bersifat privat.
  3. Tipe Data Koleksi. Tipe data koleksi yaitu tipe data yang digunakan untuk mengelompokkan dan menyimpan beberapa nilai atau objek secara bersamaan. Ada beberapa tipe data koleksi yang umum digunakan dalam pemrograman yaitu:
    - a. Array. Array yaitu struktur data statis yang menyimpan elemen-elemen dengan

tipe data yang sama.

- b. Vector . Vector yaitu Standard Template Library (STL) jika di dalam C/C++ memiliki bentuk `std::vector`. 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.

#### E. Referensi

BobTylerMSFT. (n.d.).<iomanip>. diakses dari

<https://learn.microsoft.com/id-id/cpp/standard-library/iomanip?view=msvc-170>

Fajar. Penjelasan dan Cara Mendirikan Struct. diakses dari

<https://www.belajarcpp.com/tutorial/cpp/struct>

BobTylerMSFT.(n.d.).struct (C++). diakses dari

<https://learn.microsoft.com/id-id/cpp/cpp/struct-cpp?view=msvc-170>

Fajar. Penjelasan dan Cara Pembuatan Class dan Object.diakses dari

<https://www.belajarcpp.com/tutorial/cpp/class/>

BobTylerMSFT. (n.d.). Kelas dan Struktur (C++). diakses dari

<https://learn.microsoft.com/en-us/cpp/cpp/classes-and-structs-cpp?view=msvc-170>