

# **LAPORAN PRAKTIKUM**

## **MODUL I TIPE DATA**



**Disusun oleh:  
Baharuddin Barkah Pratama  
NIM: 2311102321**

**Dosen Pengampu:**

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

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS INFORMATIKA  
INSTITUT TEKNOLOGI TELKOM PURWOKERTO  
PURWOKERTO  
2023**

# **BAB I**

## **TUJUAN PRAKTIKUM**

1. Mahasiswa dapat mempelajari tipe data primitive, 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 kompiller dapat mengetahui bagaimana sebuah data akan digunakan. Adapun tipe data yang akan dipelajari, sebagai berikut :

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

#### **TIPE DATA PRIMITIF**

Tipe data primitif adalah tipe data yang sudah ditentukan oleh sistem, tipe data primitif ini disediakan oleh banyak bahasa pemrograman, perbedaannya terletak pada jumlah bit yang dialokasikan untuk setiap bit pada tipe data primitif tergantung pada bahasa pemrograman, compiler dan sistem operasinya. Contoh tipe data primitif adalah :

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

#### **TIPE DATA ABSTRAK**

Tipe data abstrak atau yang biasa disebut Abstrak Data Tipe (ADT) merupakan tipe data yang dibentuk oleh programmer itu sendiri. Pada tipe data abstrak bisa berisi banyak tipe data, jadi nilainya bisa lebih dari satu dan beragam tipe data. Fitur Class adalah fitur Object Oriented Program (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](https://learn.microsoft.com) perbedaan antara Struct dan Class adalah pada akses defaultnya dimana Struct bersifat public dan Class bersifat private.

#### **TIPE DATA KOLEKSI**

Tipe data koleksi (Collection Data Type) adalah tipe data yang digunakan untuk mengelompokkan dan menyimpan beberapa nilai atau objek secara bersamaan. Tipe data koleksi memungkinkan Anda menyimpan, mengelola, dan mengakses sejumlah besar data dengan cara yang terstruktur. Ada beberapa tipe data koleksi yang umum digunakan dalam pemrograman, dan di antaranya adalah:

- a. Array : Array adalah struktur data statis yang menyimpan elemen-elemen dengan tipe data yang sama. Elemen-elemen tersebut dapat diakses dengan menggunakan indeks. Array memiliki ukuran tetap yang ditentukan saat deklarasi.
- b. Vector : Vector adalah Standard Template Library (STL) jika di dalam C/C++ memiliki bentuk `std::vector` . Umumnya, vector mirip seperti array yang memiliki kemampuan untuk menyimpan data dalam bentuk elemen-elemen yang alokasi memorinya dilakukan otomatis dan bersebelahan. Kemampuan vector bukan hanya pada jumlah elemen yang dinamis, vector pada C/C++ juga dilengkapi dengan fitur-fitur pelengkap seperti element access, iterators, capacity, modifiers
- c. Map : Map terasa mirip dengan array namun dengan index yang memungkinkan untuk berupa tipe data selain integer. Pada map, indeks tersebut diberi nama “key”. Pada `std::map` digunakan Self-Balancing Tree khususnya Red-Black Tree.

## BAB III

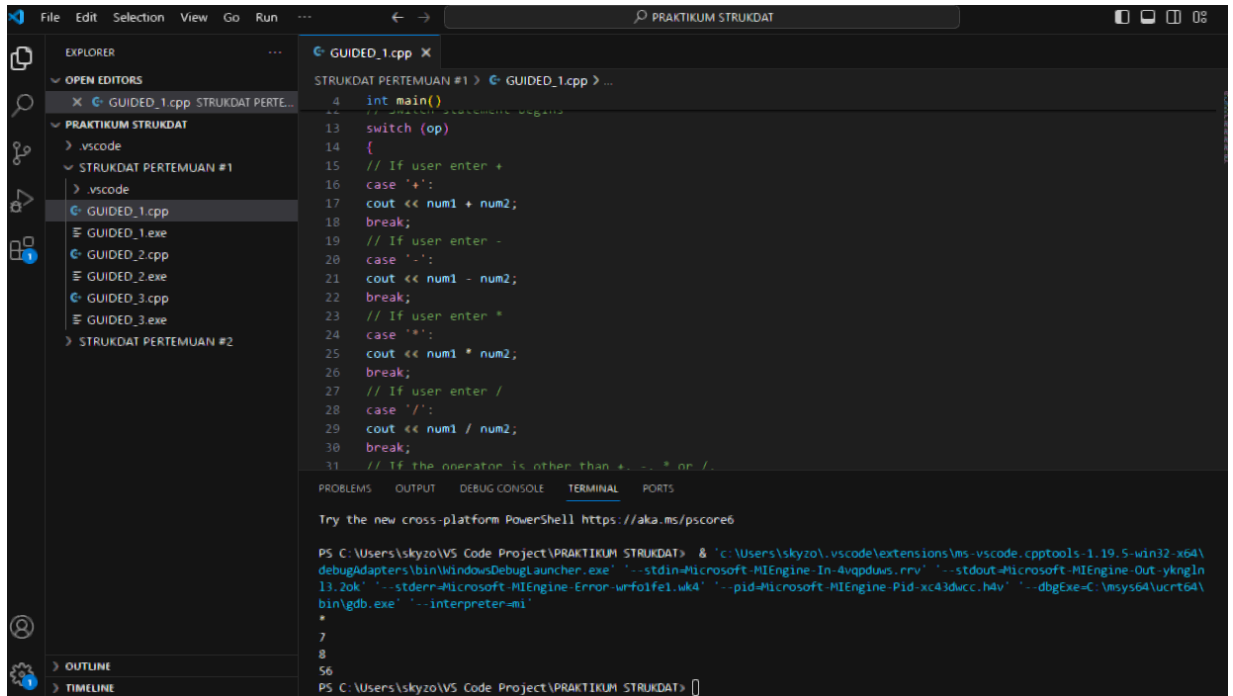
### GUIDED

#### Guided 1

##### Source code

```
#include <iostream>
using namespace std;
// Main program
int main()
{
    char op;
    float num1, num2;
    // It allows user to enter operator i.e. +, -, *, /
    cin >> op;
    // It allow user to enter the operands
    cin >> num1 >> num2;
    // Switch statement begins
    switch (op)
    {
        // If user enter
        + case '+':
            cout << num1 + num2;
            break;
        // If user enter
        - case '-':
            cout << num1 - num2;
            break;
        // If user enter
        * case '*':
            cout << num1 * num2;
            break;
        // If user enter
        / case '/':
            cout << num1 / num2;
            break;
        // If the operator is other than +, -, * or /,
        // error message will display
        default:
            cout << "Error! operator is not correct";
    } // switch statement ends
    return 0;
}
```

##### Screenshoot program



The screenshot shows the Visual Studio Code interface with a C++ file named `GUIDED_1.cpp` open. The file is part of a workspace named `PRAKTIKUM STRUKDAT`. The code implements a simple calculator using a `switch` statement to handle different operators. The terminal at the bottom shows the command prompt and the execution of the program.

```
4 int main()
5 {
6     // If user enter +
7     case '+':
8         cout << num1 + num2;
9         break;
10    // If user enter -
11    case '-':
12        cout << num1 - num2;
13        break;
14    // If user enter *
15    case '*':
16        cout << num1 * num2;
17        break;
18    // If user enter /
19    case '/':
20        cout << num1 / num2;
21        break;
22    // If the operator is other than +, -, * or /
23    default:
24        cout << "Invalid operator";
25    }
26    return 0;
27 }
```

Terminal output:

```
PS C:\Users\skyzo\VS Code Project\PRAKTIKUM STRUKDAT> & 'c:\Users\skyzo\.vscode\extensions\ms-vscode.cpptools-1.19.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-4vqpdwus.rnv' '--stdout=Microsoft-MIEngine-Out-ykngIn13.2ok' '--stderr=Microsoft-MIEngine-Error-wrfo1fe1.wk4' '--pid=Microsoft-MIEngine-Pid-xc43dwcc.h4v' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
*
7
8
56
PS C:\Users\skyzo\VS Code Project\PRAKTIKUM STRUKDAT>
```

## DESKRIPSI PROGRAM

Di program ini mendeklarasikan variable **op** yaitu karakter untuk menyimpan operator aritmatika yang dipilih pengguna (+, -, \*, /). Terus program akan meminta pengguna untuk memasukan operator aritmatika dan nilai dua bilangan. Di program ini menggunakan switch untuk memeriksa nilai operator dan melakukan operasi yang sesuai, jika operator adalah '+' hasil penjumlahan num1 dan num2 akan ditampilkan. Jika operator adalah '-' hasil pengurangan num1 dan num2 akan ditampilkan. Jika operator adalah '\*' hasil pengurangan num1 dan num2 akan ditampilkan. Jika operator adalah '/' hasil pengurangan num1 dan num2 akan ditampilkan

## Guided 2

### Source Code:

```

#include <stdio.h>
//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 screenshot shows the Visual Studio Code interface with a C++ project named 'PRAKTIKUM STRUKDAT'. The Explorer panel on the left shows the project structure, including files like GUIDED\_1.cpp, GUIDED\_2.cpp, and their corresponding executables. The main editor window displays the source code of GUIDED\_2.cpp, which is the same code shown in the first block. The Output panel at the bottom shows the program's execution results, which match the expected output from the code.

```

SD.itzg' '--stderr=Microsoft.MIEngine.Error-matsjbyv.qnw' '--pid=Microsoft.MIEngine-Pid-pufimwz.wkc' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
## Mahasiswa 1 ##
Nama: Dian
Alamat: Mataram
Umur: 22
## Mahasiswa 2 ##
Nama: Bambang
Alamat: Surabaya
Umur: 23
PS C:\Users\skyzo\VS Code Project\PRAKTIKUM STRUKDAT>

```

## DESKRIPSI PROGRAM

Baris include <stdio.h> menyertakan Pustaka standar input/output yang diperlukan untuk operasi seperti print f() dan scanf(). Terus disini program mendefinisikan struktu “Mahasiswa” dengan tiga anggota yaitu name, pointer ke string yang berisi alamat mahasiswa, terus address, pointer ke string yang berisi alamat mahasiswa, terus yg terakhir ada age, tipe data integer untuk menyimpan usia mahasiswa. Baris struct Mahasiswa mhs1, mhs2; mendeklarasikan dua variable mhs1 dan mhs2 bertipe “mahasiswa”. Program akan mengisi nilai anggota struktur mhs1 dan mhs2 dengan nama, alamat, dan usia masing-masing mahasiswa. Di baris printf() untuk mencetak data anggota struktur mhs1 dan mhs2 ke layar.

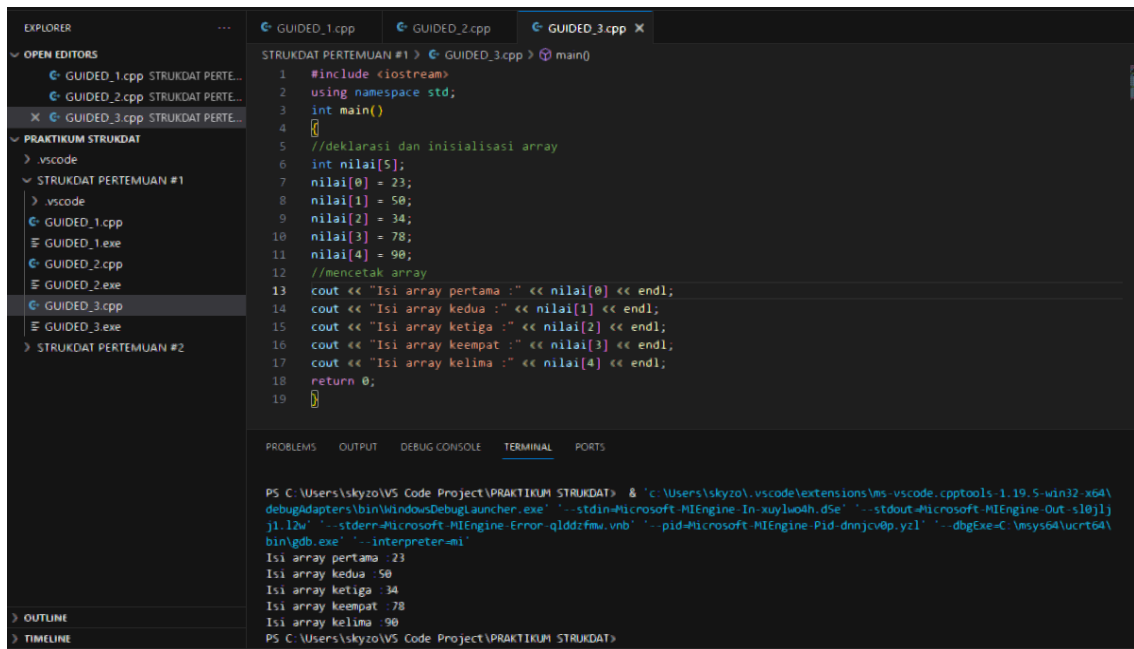
## Guided 3

### Source Code:

```
#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;
}
```

## Screenshoot program





The screenshot shows the Visual Studio Code interface with three open files: GUIDED\_1.cpp, GUIDED\_2.cpp, and GUIDED\_3.cpp. The active file, GUIDED\_3.cpp, contains the following C++ code:

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

The bottom panel shows the terminal output of the program:

```
PS C:\Users\skyzo\VS Code Project\PRAKTIKUM STRUKDAT> & "c:\Users\skyzo\.vscode\extensions\ms-vscode.cpptools-1.19.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe" "--stdin=Microsoft-MIEngine-In-xuylw0h.dse" "--stdout=Microsoft-MIEngine-Out-s10j1j.jl.12w" "--stderr=Microsoft-MIEngine-Error-qlddzfmv.vnb" "--pid=Microsoft-MIEngine-Pid-dnnjcv0p.yz1" "--dbgExe=C:\msys64\ucrt64\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\skyzo\VS Code Project\PRAKTIKUM STRUKDAT>
```

## DESKRIPSI PROGRAM

Program ini mendeklarasikan array nilai dengan 5 elemen dan menginisialisasi setiap elemen dengan nilai integer yang berbeda. Kemudian, program menggunakan loop for untuk mencetak nilai setiap elemen array ke layar. Program ini menunjukkan cara sederhana untuk bekerja dengan array dalam bahasa C++ dan dapat dimodifikasi untuk kebutuhan yang lebih kompleks, seperti menambahkan lebih banyak elemen, membaca data dari input pengguna, atau melakukan operasi lain pada elemen array.

## BAB IV

### UNGUIDED

#### Unguided 1

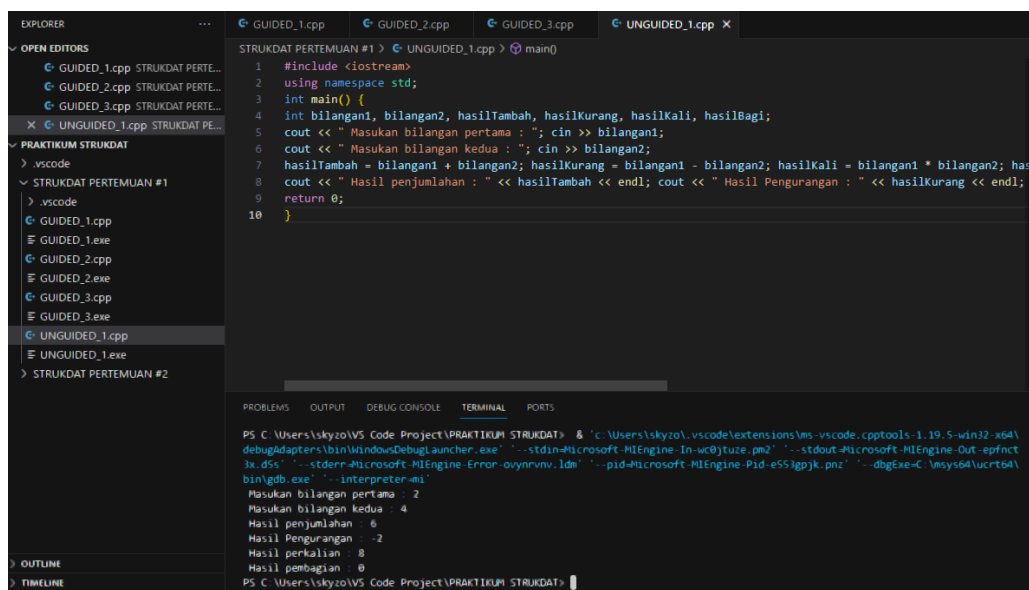
#### Source Code:

```
#include <iostream> using namespace std;
int main() {
    int bilangan1, bilangan2, hasilTambah, hasilKurang, hasilKali, hasilBagi;
    cout << " Masukan bilangan pertama : "; cin >> bilangan1;
    cout << " Masukan bilangan kedua : "; cin >> bilangan2;
    hasilTambah = bilangan1 + bilangan2; hasilKurang = bilangan1 - bilangan2;
    hasilKali = bilangan1 * bilangan2; hasilBagi = bilangan1 / bilangan2;

    cout << " Hasil penjumlahan : " << hasilTambah << endl; cout << " Hasil
    Pengurangan : " << hasilKurang << endl; cout << " Hasil perkalian : " <<
    hasilKali << endl; cout << " Hasil pembagian : " << hasilBagi << endl;

    return 0;
}
```

#### Screenshoot program



## DESKRIPSI PROGRAM

Program ini meminta pengguna untuk memasukkan dua bilangan bulat, yaitu `bilangan1` dan `bilangan2`, menggunakan perintah `cin`. Setelah menerima input, program melakukan empat operasi aritmetika dasar, yaitu penjumlahan, pengurangan, perkalian, dan pembagian, menggunakan operator `+`, `-`, `*`, dan `/`. Hasil dari setiap operasi disimpan dalam variabel `hasilTambah`, `hasilKurang`, `hasilKali`, dan `hasilBagi`. Selanjutnya, program menampilkan hasil-hasil tersebut menggunakan perintah `cout`. Program ini memberikan output berupa hasil penjumlahan, pengurangan, perkalian, dan pembagian dari dua bilangan yang dimasukkan oleh pengguna.

## KESIMPULAN

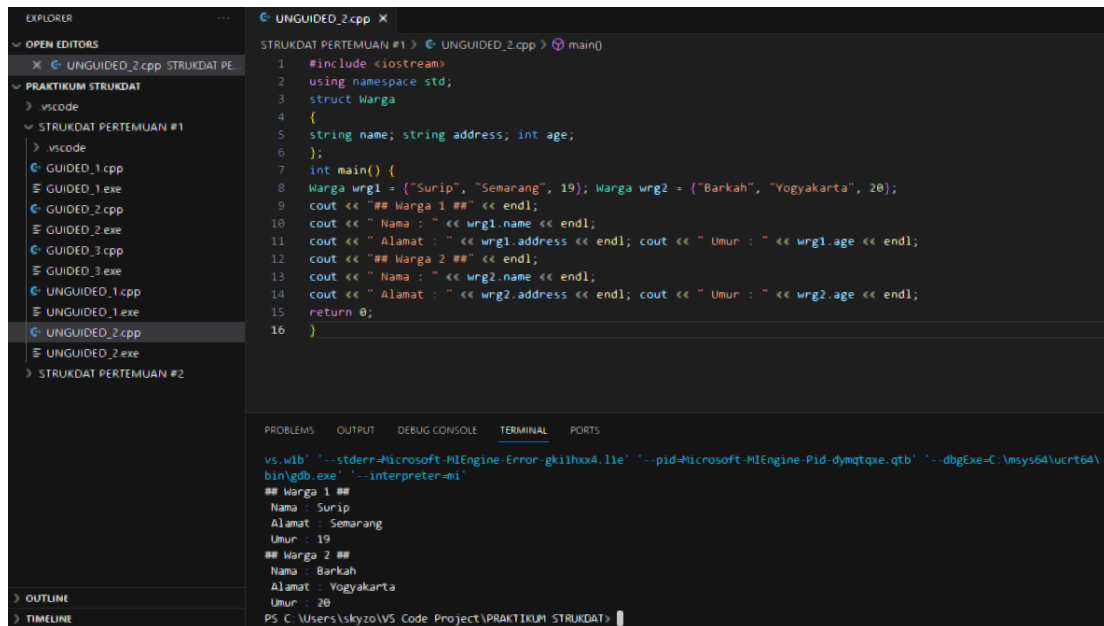
Tipe data primitif adalah tipe data bawaan yang tersedia dalam bahasa pemrograman C++. Tipe data ini digunakan untuk mendefinisikan variabel yang menyimpan nilai-nilai sederhana seperti angka, karakter, dan nilai logika. contohnya ada `bool`, `char`, `int`, `float`, `double`, dan `void`.

## Unguided 2

### Source Code:

```
#include <iostream>
using namespace std;
struct Warga
{
    string name; string address; int age;
};
int main() {
    Warga wrng1 = {"Surip", "Semarang", 19}; Warga wrng2 = {"Barkah", "Yogyakarta", 20};
    cout << "## Warga 1 ##" << endl;
    cout << " Nama : " << wrng1.name << endl;
    cout << " Alamat : " << wrng1.address << endl; cout << " Umur : " << wrng1.age << endl;
    cout << "## Warga 2 ##" << endl;
    cout << " Nama : " << wrng2.name << endl;
    cout << " Alamat : " << wrng2.address << endl; cout << " Umur : " << wrng2.age << endl;
    return 0;
}
```

## Screenshoot program



```
EXPLORER
  OPEN EDITORS
    UNGUIDED_2.cpp, STRUKDAT PE
  PRAKTIKUM STRUKDAT
    vscode
    STRUKDAT PERTEMUAN #1
      vscode
      GUIDED_1.cpp
      GUIDED_1.exe
      GUIDED_2.cpp
      GUIDED_2.exe
      GUIDED_3.cpp
      GUIDED_3.exe
      UNGUIDED_1.cpp
      UNGUIDED_1.exe
      UNGUIDED_2.cpp
      UNGUIDED_2.exe
    STRUKDAT PERTEMUAN #2

STRUKDAT PERTEMUAN #1 > UNGUIDED_2.cpp > main()
1  #include <iostream>
2  using namespace std;
3  struct Warga
4  {
5      string name; string address; int age;
6  };
7  int main() {
8      Warga wrg1 = {"Surip", "Semarang", 19}; Warga wrg2 = {"Barkah", "Yogyakarta", 20};
9      cout << "## Warga 1 ##" << endl;
10     cout << " Nama : " << wrg1.name << endl;
11     cout << " Alamat : " << wrg1.address << endl; cout << " Umur : " << wrg1.age << endl;
12     cout << "## Warga 2 ##" << endl;
13     cout << " Nama : " << wrg2.name << endl;
14     cout << " Alamat : " << wrg2.address << endl; cout << " Umur : " << wrg2.age << endl;
15     return 0;
16 }
```

```
vs_wib' '--stderr=Microsoft-MIEngine-Error-gk11hox4.11e' '--pid=Microsoft-MIEngine-Pid-dymqtqxe.qtb' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
## Warga 1 ##
Nama : Surip
Alamat : Semarang
Umur : 19
## Warga 2 ##
Nama : Barkah
Alamat : Yogyakarta
Umur : 20
PS C:\Users\skyzo\VS Code Project\PRAKTIKUM STRUKDAT>
```

## DESKRIPSI / FUNGSI PROGRAM

Program tersebut merupakan sebuah program C++ sederhana yang menggunakan struktur (struct) untuk merepresentasikan informasi mengenai warga. Struktur tersebut bernama "Warga" dan memiliki tiga anggota data: "name" untuk menyimpan nama warga, "address" untuk menyimpan alamat warga, dan "age" untuk menyimpan usia warga. Dalam fungsi utama (main), dua variabel dari tipe data struct "Warga" dideklarasikan, yaitu "wrg1" dan "wrg2", yang masing-masing diinisialisasi dengan data nama, alamat, dan usia warga. Selanjutnya, program menampilkan informasi masing-masing warga seperti nama, alamat, dan usia menggunakan perintah cout. Fungsi utama dari struct adalah untuk mengorganisir dan mengelompokkan data secara terstruktur, sehingga dapat membentuk suatu entitas yang lebih kompleks. Jadi fungsi penggunaan struct pada program tersebut adalah untuk menciptakan struktur data yang lebih terstruktur dan memudahkan representasi informasi terkait warga.

## Source “Class”:

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

class Warga {
public:
    string name;
    string address;
    int age;

    Warga(string n, string addr, int a) : name(n), address(addr), age(a) {}

    void displayInfo() {
        cout << "Nama      : " << name << endl;
        cout << "Alamat    : " << address << endl;
        cout << "Umur      : " << age << endl;
    }
};

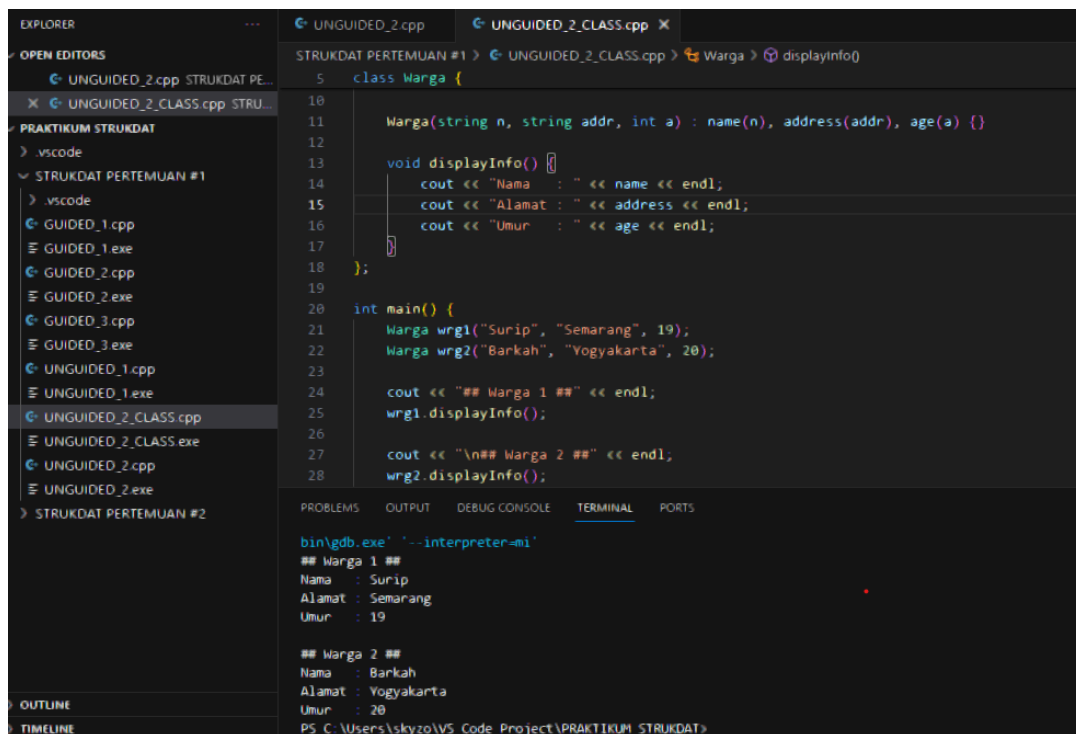
int main() {
    Warga wrng1("Surip", "Semarang", 19);
    Warga wrng2("Barkah", "Yogyakarta", 20);

    cout << "## Warga 1 ##" << endl;
    wrng1.displayInfo();

    cout << "\n## Warga 2 ##" << endl;
    wrng2.displayInfo();

    return 0;
}
```

## Screenshoot program



## DESKRIPSI / FUNGSI PROGRAM

Program ini mendefinisikan sebuah class bernama "Warga" yang memiliki tiga atribut data: "name" (nama warga), "address" (alamat warga), dan "age" (usia warga). Dalam class Warga, terdapat sebuah constructor untuk inisialisasi objek Warga dengan nilai-nilai awal yang diberikan pada saat pembuatan objek. Selain itu, terdapat metode "displayInfo()" yang digunakan untuk menampilkan informasi warga.

Dalam fungsi utama (main), dua objek Warga, yaitu "wrg1" dan "wrg2", dibuat dengan menggunakan constructor. Program kemudian menampilkan informasi masing-masing warga dengan memanggil metode "displayInfo()" pada objek tersebut. Keseluruhan program memberikan contoh penggunaan class dan objek untuk mengorganisir dan mengakses data terkait warga dengan cara yang lebih terstruktur. Penggunaan class pada program tersebut memberikan kerangka kerja yang jelas dan terstruktur untuk merepresentasikan dan mengelola informasi terkait warga, memanfaatkan konsep pemrograman berorientasi objek (OOP) untuk meningkatkan keterbacaan, modularitas, dan pemeliharaan program.

## Unguided 3

### Source Code:

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

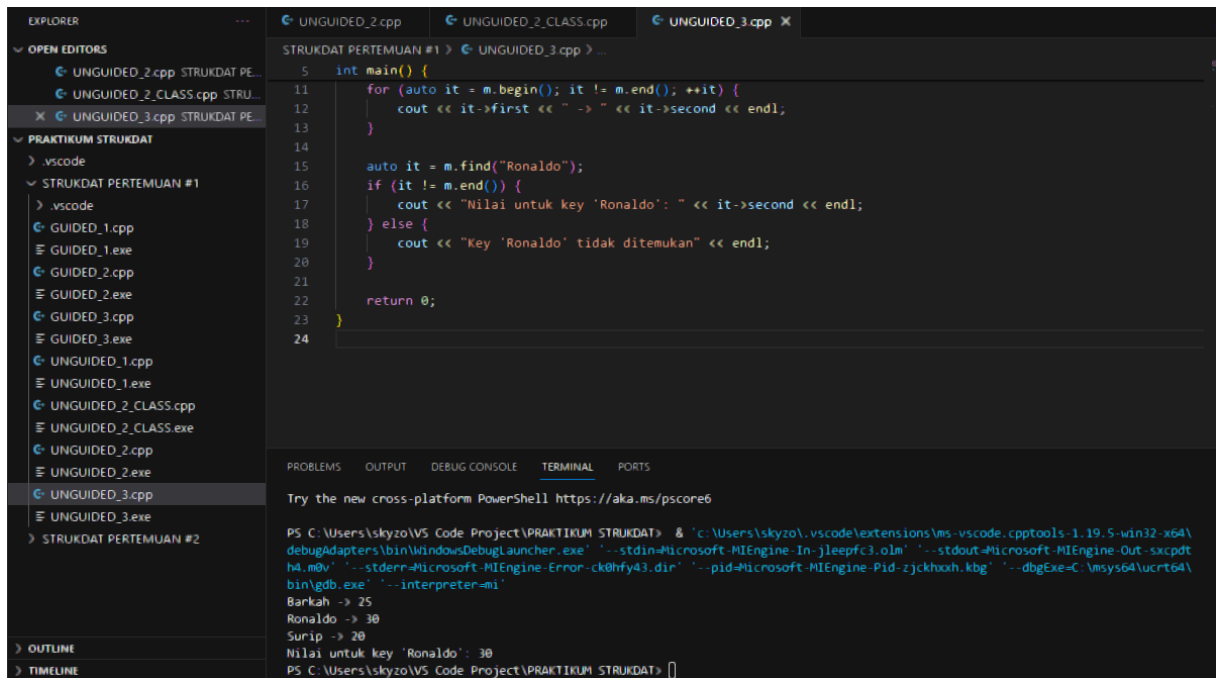
int main() {
    map<string, int> m;
    m["Surip"] = 20;
    m["Barkah"] = 25;
    m["Ronaldo"] = 30;

    for (auto it = m.begin(); it != m.end(); ++it) {
        cout << it->first << " -> " << it->second << endl;
    }

    auto it = m.find("Ronaldo");
    if (it != m.end()) {
        cout << "Nilai untuk key 'Ronaldo': " << it->second << endl;
    } else {
        cout << "Key 'Ronaldo' tidak ditemukan" << endl;
    }

    return 0;
}
```

## Screenshoot program



The screenshot shows a Visual Studio Code editor with a C++ project named 'PRAKTIKUM STRUKDAT'. The Explorer sidebar on the left shows the project structure, including files like 'UNGUIDED\_1.cpp', 'UNGUIDED\_2.cpp', and 'UNGUIDED\_3.cpp'. The main editor window displays the code for 'UNGUIDED\_3.cpp', which defines a map 'm' with three entries: 'Surip' (20), 'Barkah' (25), and 'Ronaldo' (30). The 'main' function iterates through the map, printing each key-value pair. It then uses 'm.find("Ronaldo")' to check for a specific key. Since 'Ronaldo' exists, it prints the corresponding value '30'. The bottom panel shows the 'TERMINAL' output, which displays the program's execution results: 'Barkah -> 25', 'Ronaldo -> 30', 'Surip -> 20', and 'Nilai untuk key 'Ronaldo': 30'.

```
5  int main() {
11      for (auto it = m.begin(); it != m.end(); ++it) {
12          cout << it->first << " -> " << it->second << endl;
13      }
14
15      auto it = m.find("Ronaldo");
16      if (it != m.end()) {
17          cout << "Nilai untuk key 'Ronaldo': " << it->second << endl;
18      } else {
19          cout << "Key 'Ronaldo' tidak ditemukan" << endl;
20      }
21
22      return 0;
23  }
24
```

Try the new cross-platform PowerShell <https://aka.ms/powershell>

PS C:\Users\skizo\VS Code Project\PRAKTIKUM STRUKDAT> & 'c:\Users\skizo\.vscode\extensions\ms-vscode.cpptools-1.19.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-jleefc3.0lm' '--stdout=Microsoft-MIEngine-Out-sxcpt h4.m0v' '--stderr=Microsoft-MIEngine-Error-ck0hfy43.dir' '--pid=Microsoft-MIEngine-Pid-zjckhoah.kbg' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'

Barkah -> 25  
Ronaldo -> 30  
Surip -> 20  
Nilai untuk key 'Ronaldo': 30  
PS C:\Users\skizo\VS Code Project\PRAKTIKUM STRUKDAT>

## DESKRIPSI PROGRAM

Dalam program ini, struktur data map digunakan untuk menghubungkan nama (string) dengan usia (int) dari beberapa individu. Tiga entri dimasukkan ke dalam map, dengan "Surip" memiliki usia 20, "Barkah" dengan usia 25, dan "Ronaldo" dengan usia 30. Terus program melakukan iterasi melalui map menggunakan loop for, dan untuk setiap entri, menampilkan nama dan usianya di layar. Selanjutnya, program mencoba menemukan nilai yang terkait dengan kunci "Ronaldo" menggunakan fungsi find(). Jika kunci tersebut ditemukan, program akan menampilkan nilai yang sesuai; sebaliknya, jika tidak ditemukan, program memberikan pesan bahwa kunci tersebut tidak ditemukan.

## PERBEDAAN

Perbedaannya yaitu array cocok untuk situasi di mana indeks berurutan dan jumlah elemen diketahui di awal, sementara map lebih fleksibel dan berguna ketika kita perlu mengaitkan nilai dengan kunci yang dapat bervariasi.

## **BAB V**

### **KESIMPULAN**

Memahami berbagai jenis tipe data sangat penting untuk menulis program C++ yang efektif. Tipe data primitive cocok digunakan untuk menyimpan nilai sederhana contohnya ada bool, char, int, float, double, void. Tipe data abstrak untuk menyembunyikan detail implementasi dan menyediakan operasi yang dapat dilakukan pada data contohnya ada kelas, struct. Tipe data koleksi digunakan untuk menyimpan kumpulan-kumpulan data. Juga menyediakan cara untuk mengorganisir dan mengakses data secara efisien. Contohnya ada vector, list, map, array.

### **DAFTAR PUSTAKA**

Karumanchi, N. (2016). Data Structures and algorithms made easy: Concepts, problems, Interview Questions. CareerMonk Publications. TylerMSFT. (n.d.). Collections (C++/CX). diakses dari <https://learn.microsoft.com/en-us/cpp/cppcx/collections-c-cx?view=msvc-170>