

**LAPORAN PRAKTIKUM STRUKTUR DATA DAN
PEMROGRAGAMAN**

**MODUL 1
TIPE DATA**



Disusun oleh :

ZEFANYA.B.T.TARIGAN

2311102028

IF-11-A

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 elemen dasar untuk menyimpan informasi dalam program C++. Tipe data dibutuhkan agar kompiler dapat mengetahui bagaimana sebuah data akan digunakan. Adapun tipe data yang akan dipelajari, sebagai berikut :

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

TIPE DATA PRIMITIF

Tipe data primitif adalah tipe data dasar yang tersedia secara bawaan, 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/en-us/cpp/faq/struct-vs-class) perbedaan antara Struct dan Class adalah pada akses defaultnya dimana Struct bersifat public dan Class bersifat private.

TIPE DATA KOLEKSI

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

- a. Array : Array adalah struktur data statis yang menyimpan elemen-elemen dengan tipe data yang sama. Elemen-elemen tersebut dapat diakses dengan menggunakan indeks. Array memiliki ukuran tetap yang ditentukan saat deklarasi.
- b. Vector : Vector adalah Standard Template Library (STL) jika di dalam C/C++ memiliki bentuk `std::vector` . Umumnya, vector mirip seperti array yang memiliki kemampuan untuk menyimpan data dalam bentuk elemen-elemen yang alokasi memorinya dilakukan

otomatis dan bersebelahan. Kemampuan vector bukan hanya pada jumlah elemen yang dinamis, vector pada C/C++ juga dilengkapi dengan fitur-fitur pelengkap seperti element access, iterators, capacity, modifiers

- c. Map : Map terasa mirip dengan array namun dengan index yang memungkinkan untuk berupa tipe data selain integer. Pada map, indeks tersebut diberi nama “key”. Pada `std::map` digunakan Self-Balancing Tree khususnya Red-Black Tree.

BAB III

GUIDED

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

```
D: > struktur data > E day2.cpp > main()
1  #include <iostream>
2  using namespace std;
3  // Main program
4  int main()
5  {
6      char op;
7      float num1, num2;
8      // It allows user to enter operator i.e.,+,-,*,/
9      cin >> op;
10     // It allow user to enter the operands
11     cin >> num1 >> num2;
12     // Switch statement begins
13     switch (op)
14     {
15         // If user enter +
16         case '+':
17             cout << num1 + num2;
18             break;
19         // If user enter -
20         case '-':
21             cout << num1 - num2;
22             break;
23         // If user enter *
24         case '*':
25             cout << num1 * num2;
26             break;
27         // If user enter /
28         case '/':
29             cout << num1 / num2;
30             break;
31         default:
32             cout << "Invalid operator";
33     }
34 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\jj> cd "d:\struktur data\" ; if ($?) { g++ day2.cpp -o day2 } ; if ($?) { .\day2 }
+
6
7
13
PS D:\struktur data> |
```

DESKRIPSI PROGRAM

Di program tersebut mendeklarasikan variable **op** yaitu karakter sebagai penyimpan operator aritmatika yang dipilih pengguna (+, -, *, /). Terus program akan meminta pengguna untuk memasukkan operator aritmatika dan nilai dua bilangan. Di program ini menggunakan switch untuk memeriksa nilai operator dan melakukan operasi yang sesuai, jika operator ialah '+', 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 = "Pragos";
mhs1.address = "Mataram";
mhs1.age = 100;
mhs2.name = "Magos";
mhs2.address = "Jepang";
mhs2.age = 48;

// 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: %s\n",mhs2.age);

return 0;
}

```

SCREENSHOOT PROGRAM

```

D:\> struktur data > day2_guided2.cpp > main()
11 int main()
12 {
13     struct Mahasiswa mhs1, mhs2;
14     // mengisi nilai ke struct
15     mhs1.name = "Pragos";
16     mhs1.address = "Mataram";
17     mhs1.age = 100;
18     mhs2.name = "Magos";
19     mhs2.address = "Jepang";
20     mhs2.age = 48;
21
22     // mencetak isi struct
23     printf("## Mahasiswa 1 ##\n");
24     printf("Nama : %s\n",mhs1.name);
25     printf("Alamat: %s\n",mhs1.address);
26     printf("Umur : %d\n",mhs1.age);
27     printf("## Mahasiswa 2 ##\n");
28     printf("Nama : %s\n",mhs2.name);
29     printf("Alamat: %s\n",mhs2.address);
30     printf("Umur: %s\n",mhs2.age);
31     return 0;
32 }
33
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\jj> cd "d:\struktur data"; if ($?) { g++ day2_guided2.cpp -o day2_guided2 }; if ($?) { .\day2_guided2 }
## Mahasiswa 1 ##
Nama : Pragos
Alamat: Mataram
Umur : 100
## Mahasiswa 2 ##
Nama : Magos
Alamat: Jepang
PS D:\struktur data>

```

DESKRIPSI PROGRAM

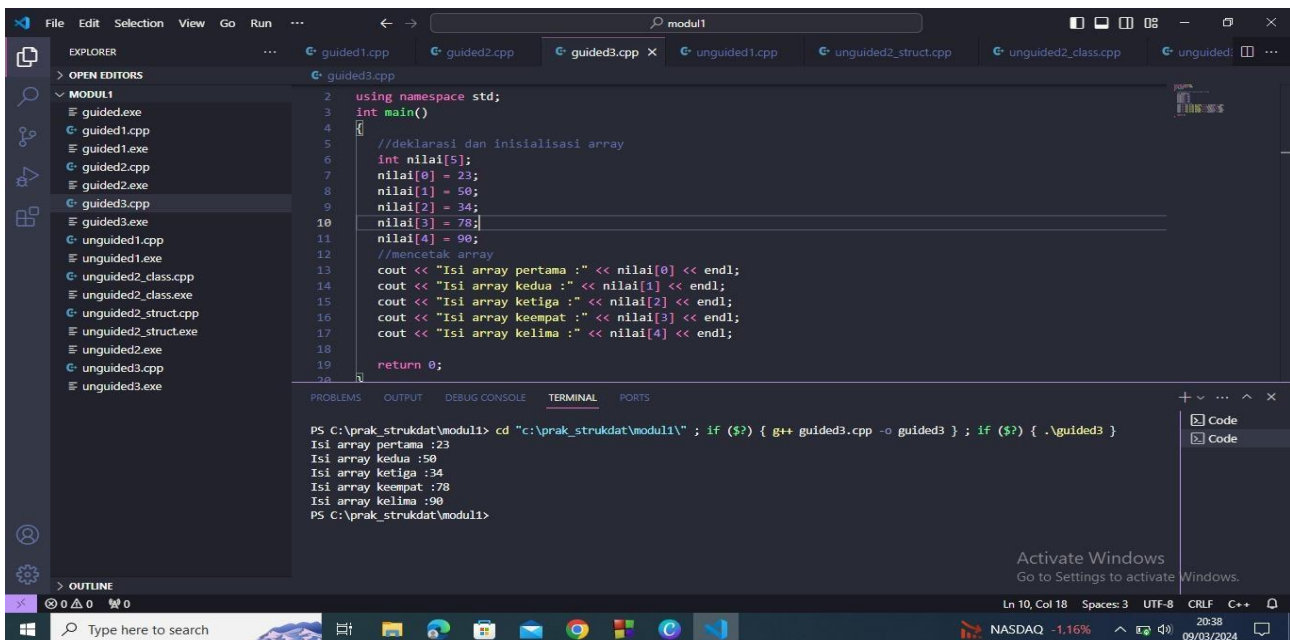
Baris include <stdio.h> menyertakan Pustaka standar input/output yang diperlukan untuk operasi seperti print f() dan scanf(). Terus disini program mendefinisikan struktu “Mahasiswa” dengan tiga anggota yaitu name, pointer ke string yang berisi alamat mahasiswa, terus address, pointer ke string yang berisi 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 OUTPUT



The screenshot shows the Visual Studio IDE with the 'guided3.cpp' file open. The code defines an array 'nilai' of size 5 and initializes it with values 23, 50, 34, 78, and 90. It then prints each element of the array. The terminal window at the bottom shows the command to compile and run the program, followed by the output: 'Isi array pertama :23', 'Isi array kedua :50', 'Isi array ketiga :34', 'Isi array keempat :78', and 'Isi array kelima :90'.

```
PS C:\prak_strukdat\modul1> cd "c:\prak_strukdat\modul1" ; if ($?) { g++ guided3.cpp -o guided3 } ; if ($?) { .\guided3 }
Isi array pertama :23
Isi array kedua :50
Isi array ketiga :34
Isi array keempat :78
Isi array kelima :90
PS C:\prak_strukdat\modul1>
```

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

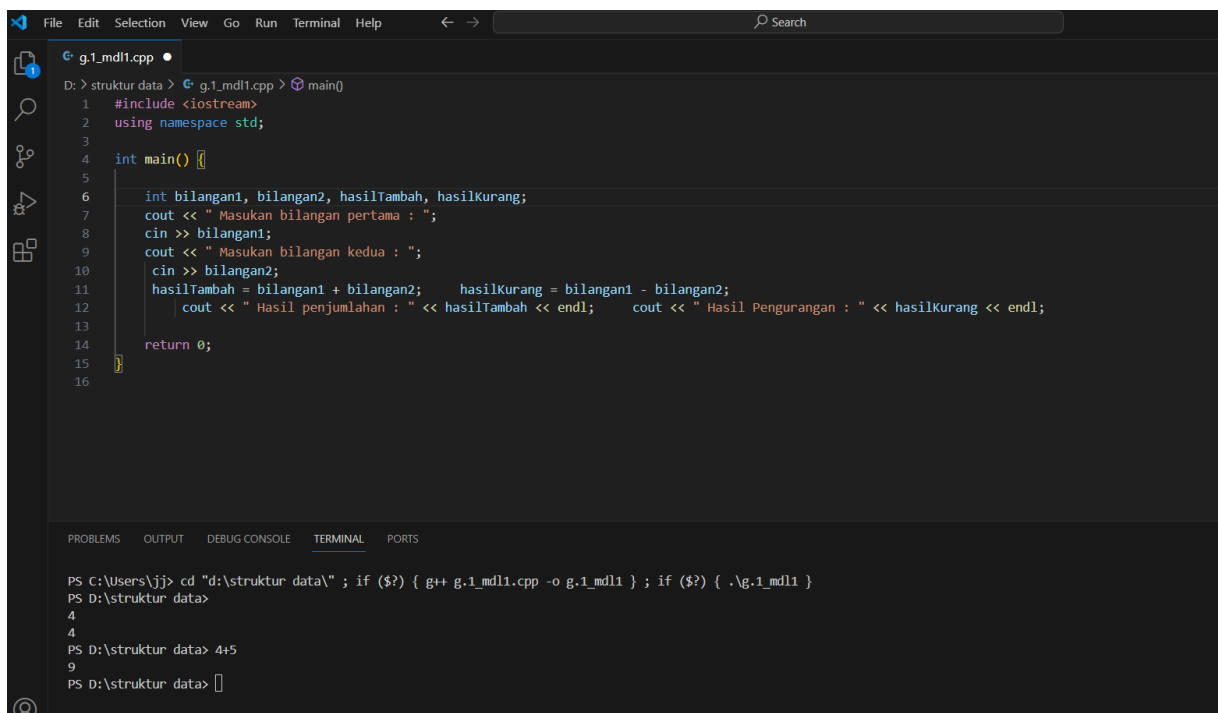
```
#include <iostream>
using namespace std;

int main() {

    int bilangan1, bilangan2,
    hasil Tambah,
    hasilKurang;
    cout << " Masukan
    bilangan pertama : ";
    cin >> bilangan1;
    cout << " Masukan
    bilangan kedua : ";
    cin >> bilangan2;
    hasilTambah =
    bilangan1 + bilangan2;
    hasilKurang = bilangan1 -
    bilangan2;
    cout << " Hasil
    penjumlahan : " <<
    hasilTambah << endl;
    cout << " Hasil
    Pengurangan : " <<
    hasilKurang << endl;

    return 0;
}
```

SCREENSHOOT CODE



The screenshot displays a code editor with a dark theme. The editor window shows the C++ source code for a program that calculates the sum and difference of two numbers. The code is as follows:

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5
6     int bilangan1, bilangan2, hasilTambah, hasilKurang;
7     cout << " Masukan bilangan pertama : ";
8     cin >> bilangan1;
9     cout << " Masukan bilangan kedua : ";
10    cin >> bilangan2;
11    hasilTambah = bilangan1 + bilangan2;    hasilKurang = bilangan1 - bilangan2;
12    cout << " Hasil penjumlahan : " << hasilTambah << endl;    cout << " Hasil Pengurangan : " << hasilKurang << endl;
13
14    return 0;
15 }
16
```

Below the editor, the terminal window shows the command to compile and run the program:

```
PS C:\Users\jj> cd "d:\struktur data\"; if ($?) { g++ g_1_md11.cpp -o g_1_md11 }; if ($?) { .\g_1_md11 }
PS D:\struktur data>
4
4
PS D:\struktur data> 4+5
9
PS D:\struktur data>
```

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 hasil tambah dan hasil kurang. Selanjutnya, program menampilkan hasil-hasil tersebut menggunakan perintah cout. Program ini memberikan output berupa hasil penjumlahan dan pengurangan dari dua bilangan yang dimasukkan oleh pengguna.

KESIMPULAN

Tipe data primitif ialah sebuah tipe data bawaan yang tersedia dalam bahasa pemrograman C++. Tipe data ini digunakan untuk mendefinisikan sebuah 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 “ STRUCT ”

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

struct Warga{

    string name;
    string address;
    int age;
};

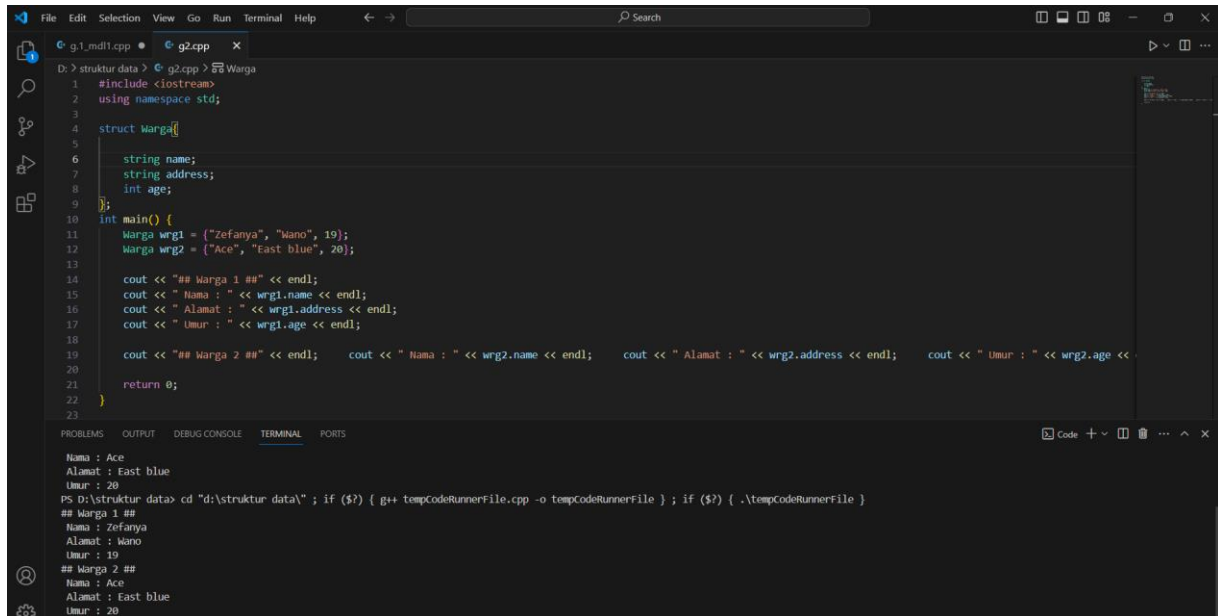
int main() {
    Warga wrg1 =
{"Zefanya", "Wano", 19};
    Warga wrg2 = {"Ace",
"East blue", 20};

    cout << "## Warga 1
##" << endl;
    cout << " Nama : " <<
wrg1.name << endl;
    cout << " Alamat : "
<< wrg1.address << endl;
    cout << " Umur : " <<
wrg1.age << endl;

    cout << "## Warga 2
##" << endl;
    cout <<
" Nama : " << wrg2.name
<< endl;
    cout << "
Alamat : " <<
wrg2.address << endl;
    cout << " Umur : " <<
wrg2.age << endl;

    return 0;
}
```

SCREENSHOOT PROGRAM



```
1 #include <iostream>
2 using namespace std;
3
4 struct Warga{
5
6     string name;
7     string address;
8     int age;
9 };
10
11 int main() {
12     Warga wrg1 = {"Zefanya", "Mano", 19};
13     Warga wrg2 = {"Ace", "East blue", 20};
14
15     cout << "## Warga 1 ##" << endl;
16     cout << " Nama : " << wrg1.name << endl;
17     cout << " Alamat : " << wrg1.address << endl;
18     cout << " Umur : " << wrg1.age << endl;
19
20     cout << "## Warga 2 ##" << endl;
21     cout << " Nama : " << wrg2.name << endl;
22     cout << " Alamat : " << wrg2.address << endl;
23     cout << " Umur : " << wrg2.age << endl;
24
25     return 0;
26 }
```

Terminal Output:

```
Nama : Ace
Alamat : East blue
Umur : 20

## Warga 1 ##
Nama : Zefanya
Alamat : Mano
Umur : 19

## Warga 2 ##
Nama : Ace
Alamat : East blue
Umur : 20
```

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 yang anda inginkan, alamat, dan usia seorang warga. Selanjutnya, program menampilkan informasi masing-masing warga seperti nama, alamat, dan usia menggunakan perintah cout. Fungsi utama dari struct ialah sebagai mengorganisir dan mengelompokkan sebuah 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;

    // Constructor untuk inisialisasi objek Warga
```

```

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

// Metode untuk menampilkan informasi warga
void displayInfo() {
    cout << "Nama    : " << name << endl;
    cout << "Alamat  : " << address << endl;
    cout << "Umur    : " << age << endl;
}
};

int main() {
    // Membuat objek Warga menggunakan constructor
    Warga wrg1("Zefanya", "Wano", 19);
    Warga wrg2("Ace", "East blue", 20);

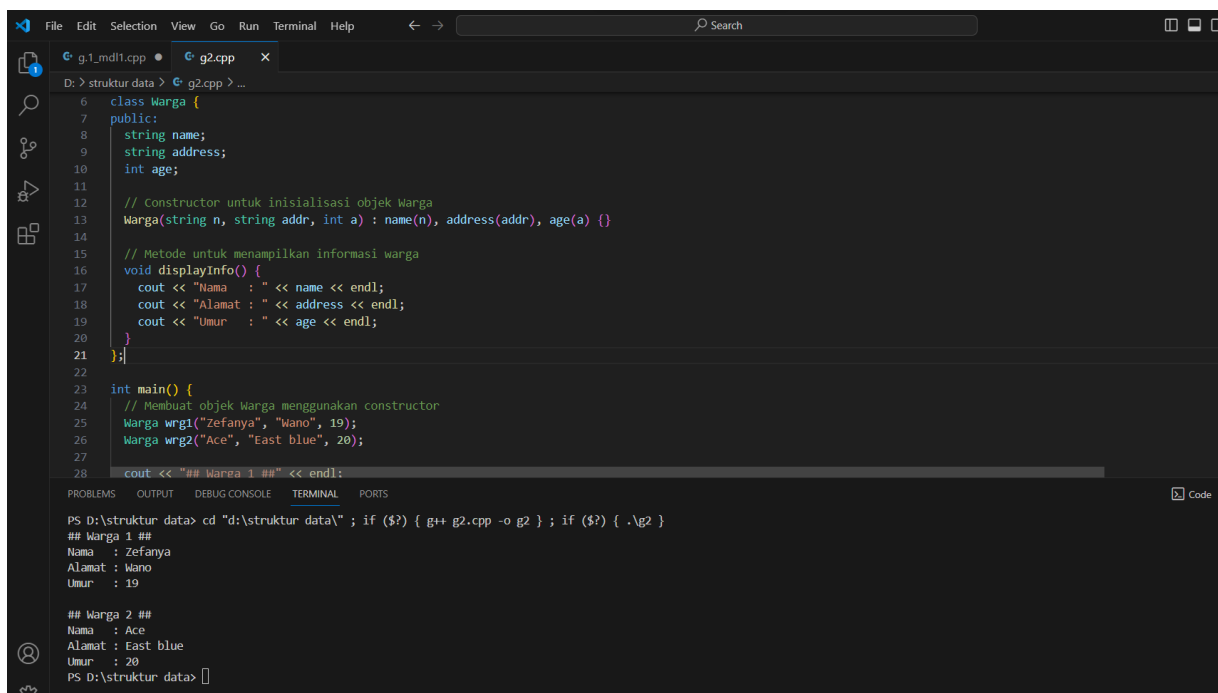
    cout << "## Warga 1 ##" << endl;
    wrg1.displayInfo();

    cout << "\n## Warga 2 ##" << endl;
    wrg2.displayInfo();

    return 0;
}

```

SCREENSHOOT PROGRAM



The screenshot shows a Visual Studio IDE with a C++ project. The code editor displays the same code as the first block. The terminal window at the bottom shows the output of the program, which matches the expected output from the code.

```

PS D:\struktur data> cd "d:\struktur data\"; if ($?) { g++ g2.cpp -o g2 }; if ($?) { .\g2 }
## Warga 1 ##
Nama    : Zefanya
Alamat  : Wano
Umur    : 19

## Warga 2 ##
Nama    : Ace
Alamat  : East blue
Umur    : 20
PS D:\struktur data>

```

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.

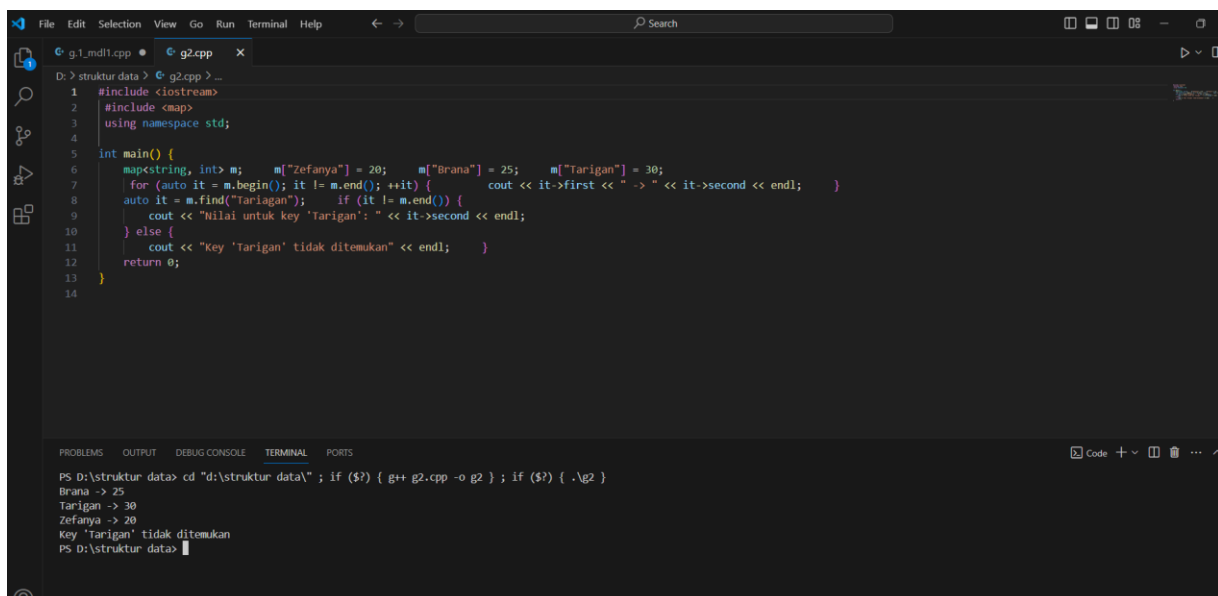
Dalam fungsi utama (main), dua objek Warga, yaitu "wrg1" dan "wrg2", dibuat dengan menggunakan constructor. Program kemudian menampilkan informasi masing-masing warga dengan memanggil metode "displayInfo()" pada objek tersebut. Keseluruhan program memberikan contoh penggunaan class dan objek untuk mengorganisir 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.

UNGUIDED 3 SOURCE CODE

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

int main() {
    map<string, int>
m;    m["Zefanya"]
= 20;    m["Brana"]
= 25;
m["Tarigan"] = 30;
    for (auto it =
m.begin(); it !=
m.end(); ++it) {
cout << it->first <<
" -> " << it->second
<< endl;    }
    auto it =
m.find("Tariagan");
if (it != m.end()) {
    cout <<
"Nilai untuk key
'Tarigan': " << it-
>second << endl;
    } else {
        cout << "Key
'Tarigan' tidak
ditemukan" << endl;
    }
    return 0;
}
```

SCREENSHOOT PROGRAM



The screenshot displays the Visual Studio Code editor with a C++ file named g2.cpp. The code defines a map with three entries: "Zefanya" (20), "Brana" (25), and "Tarigan" (30). It then iterates through the map to print each key-value pair. Additionally, it uses the find method to check for a key "Tariagan" (note the typo in the code) and prints a message if it is not found. The terminal output shows the execution of the program, displaying the map contents and the message "Key 'Tarigan' tidak ditemukan".

```
1 #include <iostream>
2 #include <map>
3 using namespace std;
4
5 int main() {
6     map<string, int> m;    m["Zefanya"] = 20;    m["Brana"] = 25;    m["Tarigan"] = 30;
7     for (auto it = m.begin(); it != m.end(); ++it) {    cout << it->first << " -> " << it->second << endl;    }
8     auto it = m.find("Tariagan");    if (it != m.end()) {
9         cout << "Nilai untuk key 'Tarigan': " << it->second << endl;
10    } else {
11        cout << "Key 'Tarigan' tidak ditemukan" << endl;    }
12    return 0;
13 }
14
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\struktur data> cd "d:\struktur data\" ; if ($?) { g++ g2.cpp -o g2 } ; if ($?) { .\g2 }
Brana -> 25
Tarigan -> 30
Zefanya -> 20
Key 'Tarigan' tidak ditemukan
PS D:\struktur data>
```

DESKRIPSI PROGRAM

Dalam program ini, struktur data map berguna untuk menghubungkan nama (string) dengan usia (int) dari beberapa individu. Tiga entri dimasukkan ke dalam map, dengan "Zefanya" memiliki usia 20, "Brana" dengan usia 25, dan "Tarigan" dengan usia 30. Terus program ini 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 "Tarigan" 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 lebih cocok untuk situasi di mana indeks berurutan dan jumlah elemen yang diketahui di awal, sementara map itu lebih fleksibel dan berguna ketika kita perlu mengaitkan nilai dengan kunci yang dapat bervariasi.

BAB V

KESIMPULAN

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