

LAPORAN PRAKTIKUM STRUKTUR DATA DAN ALGORITMA

MODUL 1 TIPE DATA



Disusun Oleh :

Muhammad Hamzah Haifan Ma'ruf
231102091

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 adalah suatu memori atau media pada komputer yang digunakan untuk menampung informasi atau data sementara. Tipe data berfungsi untuk mempresentasikan jenis dari sebuah nilai yang terdapat dalam program. Berikut ini merupakan macam-macam tipe data serta penjelasannya yang terdapat pada C++ :

1. Tipe Data Primitif

Tipe data primitif adalah tipe data yang sudah ditentukan oleh sistem dan disediakan oleh banyak bahasa pemrograman.

- Int (Bilangan bulat)
- Float (Bilangan desimal)
- Char (Huruf)
- Boolean (True/False)

2. Tipe Data Abstrak

Tipe data abstrak atau yang biasa disebut Abstrak Data Type 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.

3. Tipe Data Koleksi

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

- Array => struktur data statis yang menyimpan elemen-elemen dengan tipe data yang sama.
- Vector => vector mirip seperti array yang memiliki kemampuan untuk menyimpan data dalam bentuk elemen-elemen yang alokasi memorinya dilakukan otomatis dan bersebelahan.
- Map => mirip dengan array namun dengan index yang memungkinkan untuk berupa tipe data selain integer.

B. Guided 1

Source Code

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

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

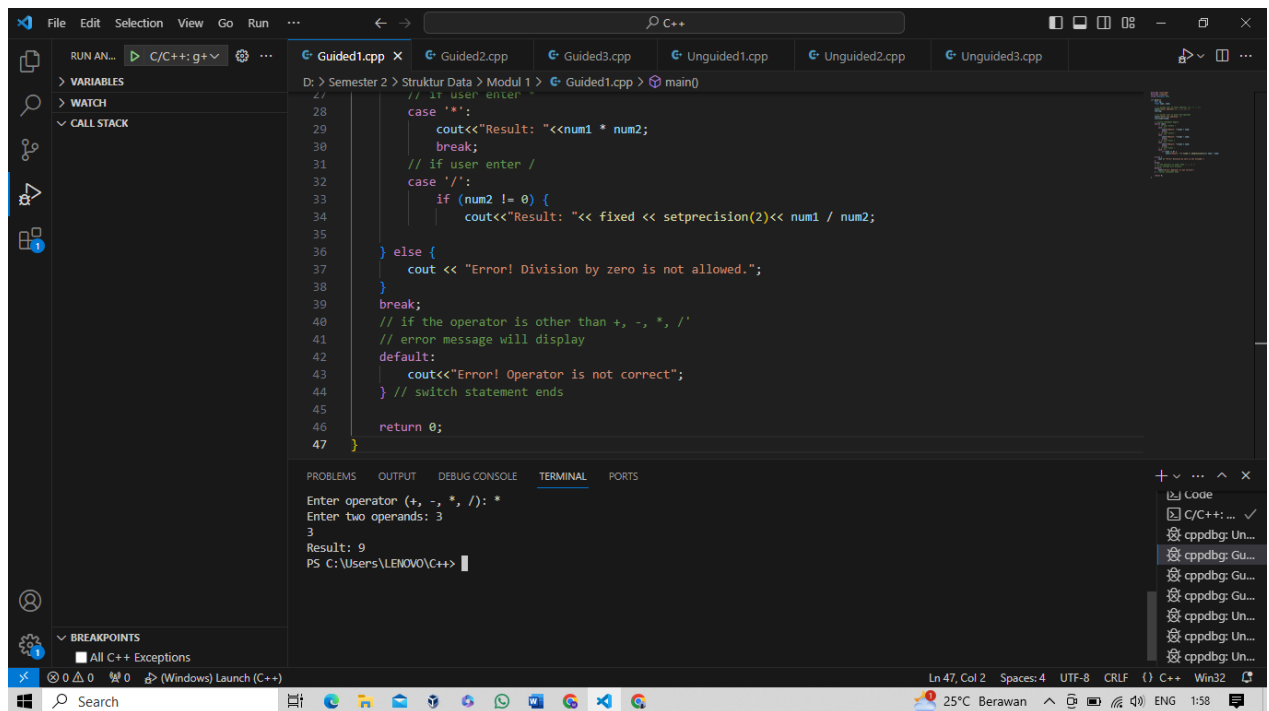
    // it allows user to enter operator i.e. +, -, *, /
    cout<<"Enter operator (+, -, *, /): ";
    cin>>op;

    // it allows user to enter the operands
    cout<<"Enter two operands: ";
    cin>>num1>>num2;

    // switch statement begins
    switch (op){
        // if user enters +
        case '+':
            cout<<"Result: "<<num1 + num2;
            break;
        // if user enters -
        case '-':
            cout<<"Result: "<<num1 - num2;
            break;
        // if user enter *
        case '*':
            cout<<"Result: "<<num1 * num2;
            break;
        // if user enter /
        case '/':
            if (num2 != 0) {
                cout<<"Result: "<< fixed << setprecision(2)<< num1 / num2;
            }
            else {
                cout << "Error! Division by zero is not allowed.";
            }
            break;
        // if the operator is other than +, -, *, /
        // error message will display
        default:
            cout<<"Error! Operator is not correct";
    } // switch statement ends

    return 0;
```

Screenshots Output :



The screenshot displays a C++ IDE with a project named "Guided1.cpp". The code implements a simple calculator with the following logic:

- It prompts the user to "Enter operator (+, -, *, /):".
- It prompts the user to "Enter two operands: 3 3".
- Based on the operator, it performs addition, subtraction, multiplication, or division.
- It includes error handling for division by zero and unrecognized operators.

The output window shows the program's execution:

```
Enter operator (+, -, *, /): *
Enter two operands: 3 3
Result: 9
PS C:\Users\LENOVO\C++>
```

Deskripsi :

Program di atas adalah sebuah kalkulator sederhana yang dibuat menggunakan bahasa pemrograman C++. Ketika program dijalankan, pengguna diminta untuk memasukkan operator matematika seperti tambah (+), kurang (-), kali (*), atau bagi (/). Kemudian, pengguna diminta untuk memasukkan dua angka yang akan dioperasikan. Setelah pengguna memasukkan operator dan dua angka, program akan melakukan operasi sesuai dengan operator yang dimasukkan. Misalnya, jika pengguna memasukkan operator tambah (+), program akan menjumlahkan dua angka yang dimasukkan pengguna. Hasil operasi tersebut kemudian ditampilkan. Program ini juga memiliki penanganan kesalahan untuk operasi pembagian. Jika pengguna memilih operator bagi (/) dan angka kedua yang dimasukkan adalah nol, maka program akan menampilkan pesan kesalahan karena pembagian dengan nol tidak dapat dilakukan. Jika pengguna memasukkan operator yang tidak dikenali, misalnya karakter selain dari +, -, *, atau /, maka program akan menampilkan pesan kesalahan bahwa operator tersebut tidak benar.

Guided 2

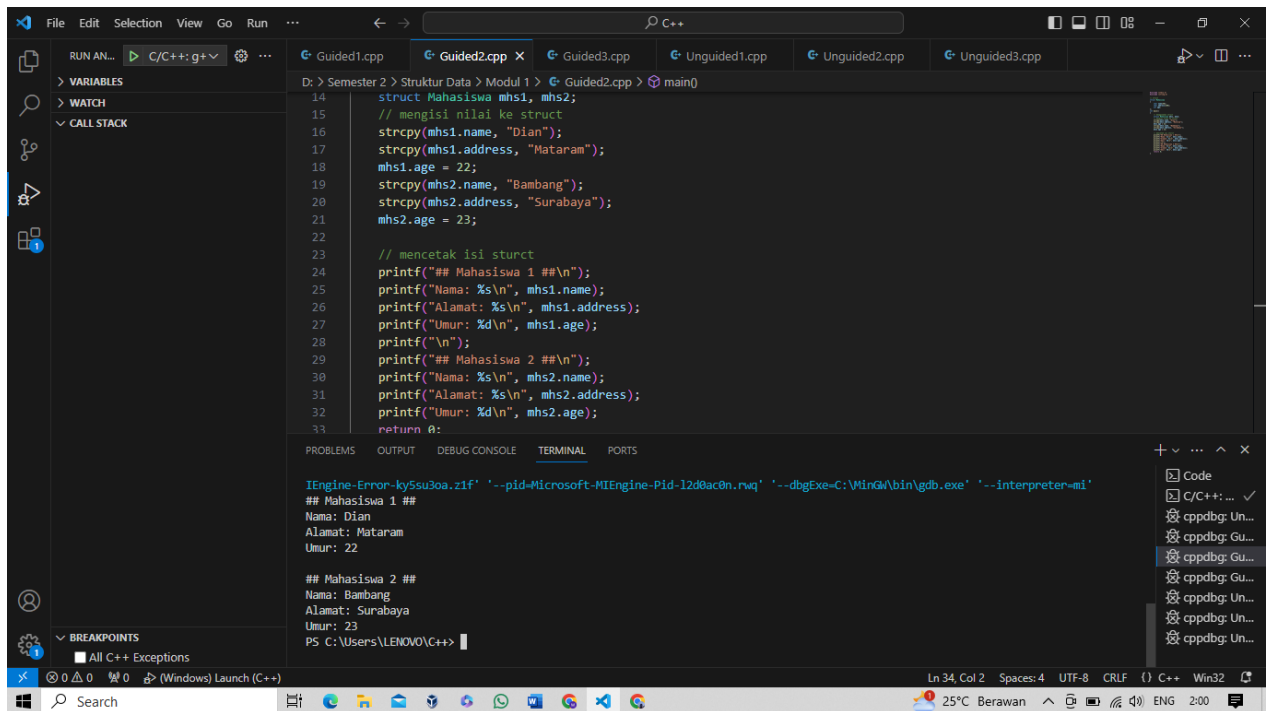
```
#include <stdio.h>
#include <string.h>

// struct
struct Mahasiswa
{
    char name[50];
    char address[100];
    int age;
};

int main()
{
    // menggunakan struct
    struct Mahasiswa mhs1, mhs2;
    // mengisi nilai ke struct
    strcpy(mhs1.name, "Dian");
    strcpy(mhs1.address, "Mataram");
    mhs1.age = 22;
    strcpy(mhs2.name, "Bambang");
    strcpy(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("\n");
    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 :



```
14 struct Mahasiswa mhs1, mhs2;
15 // mengisi nilai ke struct
16 strcpy(mhs1.name, "Dian");
17 strcpy(mhs1.address, "Mataram");
18 mhs1.age = 22;
19 strcpy(mhs2.name, "Bambang");
20 strcpy(mhs2.address, "Surabaya");
21 mhs2.age = 23;
22
23 // mencetak isi struct
24 printf("## Mahasiswa 1 ##\n");
25 printf("Nama: %s\n", mhs1.name);
26 printf("Alamat: %s\n", mhs1.address);
27 printf("Umur: %d\n", mhs1.age);
28 printf("\n");
29 printf("## Mahasiswa 2 ##\n");
30 printf("Nama: %s\n", mhs2.name);
31 printf("Alamat: %s\n", mhs2.address);
32 printf("Umur: %d\n", mhs2.age);
33 return 0;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
IEngine-Error-ky5su3oa.zif' '--pid=Microsoft-MIEngine-Pid-12d0ac8n.rwq' '--dbgExe=C:\WinG\bin\gdb.exe' '--interpreter=mi'
## Mahasiswa 1 ##
Nama: Dian
Alamat: Mataram
Umur: 22

## Mahasiswa 2 ##
Nama: Bambang
Alamat: Surabaya
Umur: 23
PS C:\Users\LENOVO\C++>
```

Deskripsi :

Program di atas merupakan contoh penggunaan struktur (struct) dalam bahasa pemrograman C. Struct digunakan untuk mengelompokkan beberapa variabel yang berbeda tipe data menjadi satu kesatuan. Dalam program ini, kita memiliki sebuah struct bernama "Mahasiswa" yang memiliki tiga anggota: "name" bertipe array of char, "address" bertipe array of char, dan "age" bertipe integer. Kemudian, di dalam fungsi main(), kita mendeklarasikan dua variabel bertipe struct Mahasiswa yaitu mhs1 dan mhs2. Kita mengisi nilai ke dalam kedua variabel tersebut menggunakan fungsi strcpy() untuk menyalin string ke dalam array char dan langsung mengisi nilai age secara langsung. Setelah itu, kita mencetak isi dari kedua variabel mhs1 dan mhs2 dengan menggunakan printf(). Hal ini memungkinkan kita untuk menampilkan informasi nama, alamat, dan umur dari masing-masing mahasiswa. Kode tersebut akan mencetak informasi kedua mahasiswa yang sudah diisi sebelumnya ke layar dengan format yang telah ditentukan.

Guided 3

```

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

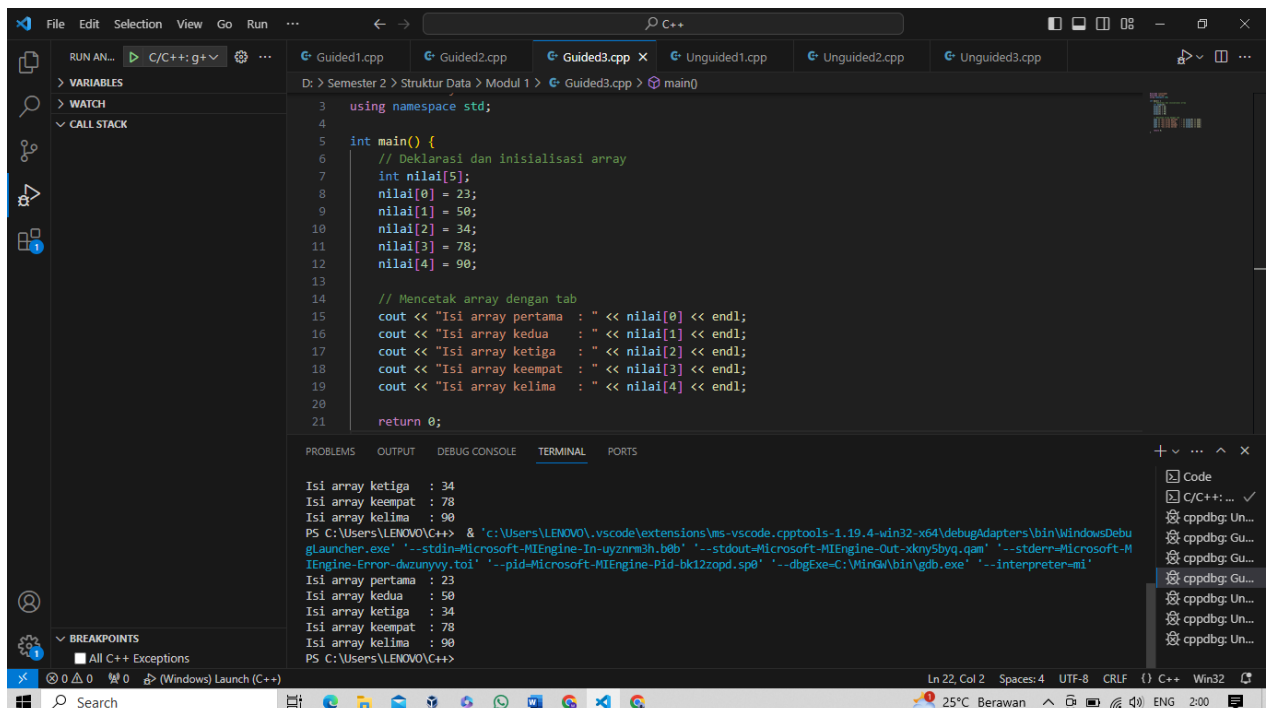
int main() {
    // Deklarasi dan inisialisasi array
    int nilai[5];
    nilai[0] = 23;
    nilai[1] = 50;
    nilai[2] = 34;
    nilai[3] = 78;
    nilai[4] = 90;

    // Mencetak array dengan tab
    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 :



Deskripsi :

Program di atas adalah contoh penggunaan array dalam bahasa pemrograman C++ untuk menyimpan dan mencetak beberapa nilai secara berurutan. Pertama-tama, program mendeklarasikan sebuah array bernama "nilai" yang dapat menampung 5 elemen bertipe integer. Kemudian, nilai-nilai dari array tersebut diinisialisasi dengan nilai yang spesifik. Setiap elemen array diisi dengan nilai tertentu menggunakan indeksinya, mulai dari nilai[0] hingga nilai[4]. Setelah nilai-nilai array diinisialisasi, program mencetak isi dari masing-masing elemen array ke layar menggunakan objek "cout" dari header iostream. Pencetakan dilakukan dengan menyertakan indeks dari setiap elemen array dan nilai yang sesuai dengan indeks tersebut. Ketika program dijalankan, hasilnya adalah mencetak isi array pertama hingga array kelima ke layar dengan format yang telah ditentukan.

C. Unguided 1

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

//fungsi untuk menghitung keliling persegi
double hitungkelilingpersegi_091(double sisi) {
    return sisi * 4;
}

//fungsi untuk menghitung luas persegi
double hitungluaspersegi_091(double sisi) {
    return sisi * sisi;
}

int main() {
    double sisi;

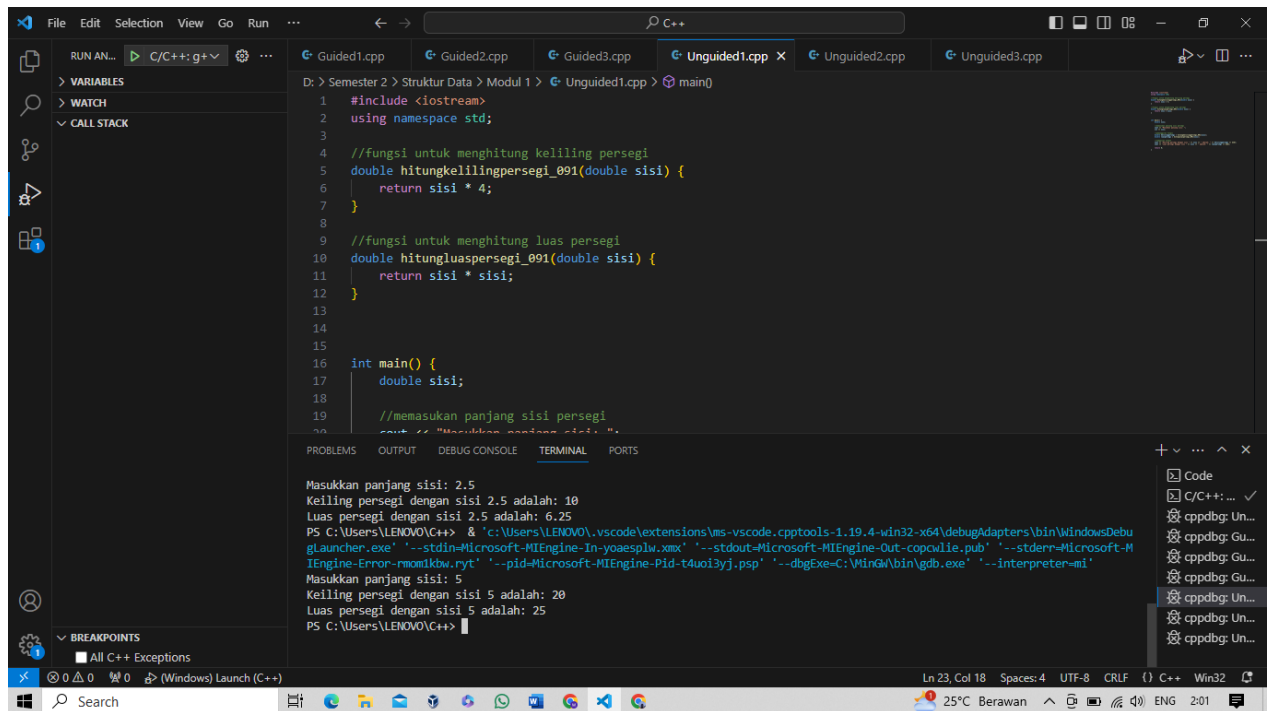
    //memasukan panjang sisi persegi
    cout << "Masukkan panjang sisi: ";
    cin >> sisi;

    //pemanggilan fungsi
    double kelilingpersegi = hitungkelilingpersegi_091(sisi);
    double luaspersegi = hitungluaspersegi_091(sisi);

    //tampilan output
    cout << "Keliling persegi dengan sisi " << sisi << " adalah: "
    << kelilingpersegi << endl;
    cout << "Luas persegi dengan sisi " << sisi << " adalah: " <<
    luaspersegi << endl;

    return 0;
}
```


Screenshots Output :



```
D:\Semester 2 > Struktur Data > Modul 1 > Unguided1.cpp > main()
1  #include <iostream>
2  using namespace std;
3
4  //fungsi untuk menghitung keliling persegi
5  double hitungkelilingpersegi_091(double sisi) {
6      return sisi * 4;
7  }
8
9  //fungsi untuk menghitung luas persegi
10 double hitungluaspersegi_091(double sisi) {
11     return sisi * sisi;
12 }
13
14
15
16 int main() {
17     double sisi;
18
19     //memasukkan panjang sisi persegi
20     cout << "Masukkan panjang sisi: ";
21
22     cin >> sisi;
23
24     double kelilingpersegi = hitungkelilingpersegi_091(sisi);
25     double luaspersegi = hitungluaspersegi_091(sisi);
26
27     cout << "Keliling persegi dengan sisi " << sisi << " adalah: " << kelilingpersegi << endl;
28     cout << "Luas persegi dengan sisi " << sisi << " adalah: " << luaspersegi << endl;
29 }

Masukkan panjang sisi: 2.5
Keliling persegi dengan sisi 2.5 adalah: 10
Luas persegi dengan sisi 2.5 adalah: 6.25
PS C:\Users\LENOVO\C++> & "c:\Users\LENOVO\.vscode\extensions\ms-vscode.cpptools-1.19.4-win32-x64\debug\adapters\bin\WindowsDebu
gLauncher.exe" "--stdin=Microsoft-MIEngine-In-yoasplw.xmx" "--stdout=Microsoft-MIEngine-Out-copwllie.pub" "--stderr=Microsoft-M
IEngine-Error-rwom1kbw.ryt" "--pid=Microsoft-MIEngine-Pid-t4uol3yj.psp" "--dbgExe=C:\Windows\WinSxS\x-wwg\gdb.exe" "--interpreter=mi"
Masukkan panjang sisi: 5
Keliling persegi dengan sisi 5 adalah: 20
Luas persegi dengan sisi 5 adalah: 25
PS C:\Users\LENOVO\C++>
```

Deskripsi :

Program di atas adalah program C++ sederhana yang digunakan untuk menghitung keliling dan luas persegi berdasarkan panjang sisi yang dimasukkan oleh pengguna. Pertama, terdapat dua fungsi yang didefinisikan di luar fungsi main(). Fungsi-fungsi tersebut bertujuan untuk menghitung keliling dan luas persegi. Fungsi 'hitungkelilingpersegi_091(double sisi)' digunakan untuk menghitung keliling persegi, sementara fungsi 'hitungluaspersegi_091(double sisi)' digunakan untuk menghitung luas persegi. Kedua fungsi tersebut mengambil panjang sisi sebagai parameter, dan hasil perhitungan kembali sebagai nilai balik. Di dalam fungsi main(), program pertama-tama meminta pengguna untuk memasukkan panjang sisi persegi melalui input menggunakan perintah 'cout << "Masukkan panjang sisi: ";' dan 'cin >> sisi;'. Nilai yang dimasukkan oleh pengguna disimpan dalam variabel 'sisi'. Kemudian, program memanggil fungsi-fungsi yang telah didefinisikan sebelumnya untuk menghitung keliling dan luas persegi. Hasil perhitungan disimpan dalam variabel 'kelilingpersegi' dan 'luaspersegi'. Terakhir, program mencetak hasil perhitungan keliling dan luas persegi ke layar menggunakan perintah 'cout'. Ketika program dijalankan, pengguna diminta untuk memasukkan panjang sisi persegi, dan program akan menghitung keliling dan luasnya berdasarkan input pengguna, serta mencetak hasilnya ke layar.

Kesimpulan:

Keseluruhan, tipe data primitif, input/output, fungsi, dan operator aritmatika merupakan komponen-komponen dasar yang penting dalam pengembangan perangkat lunak, dan

mereka digunakan secara efektif dalam codingan tersebut untuk menghitung keliling dan luas persegi.

Unguided 2

```
#include <stdio.h>
#include <string>
#include <iostream>

using namespace std;

//contoh struct
struct menu {
    string nama_091;
    int harga, jumlah;
};

//contoh class
class reservasi
{
private:
    string nama_091;
    int waktu;
    int harga;

public:
    //inisialisasi objek reservasi
    reservasi(string namareservasi, int waktureservasi, int
hargareservasi) {
        nama_091 = namareservasi;
        waktu = waktureservasi;
        harga = hargareservasi;
    }

    //menampilkan informasi reservasi
    void tampilkanInfo() {
        cout << "Nama Pemain : " << nama_091 << endl;
        cout << "Durasi Main : " << waktu << " jam " << endl;
        cout << "Harga : Rp. " << harga << endl << endl;
    }
};

int main() {

    //penggunaan struct
    menu minuman, makanan;
```

```

minuman.nama_091 = "Es Kelapa Sawit";
minuman.jumlah = 4;
minuman.harga = 25000;

cout << "Nama Minuman : " << minuman.nama_091 << endl
    << "Jumlah : " << minuman.jumlah << endl
    << "Harga : Rp." << minuman.harga << endl << endl;

//penggunaan class
reservasi billiard("Stev", 10, 500000);
reservasi vip("Jhon", 5, 250000);
billiard.tampilkanInfo();
vip.tampilkanInfo();
}

```

Screenshots Output :

The screenshot shows the Visual Studio Code IDE with a C++ project. The code editor displays the same code as the previous block. The terminal window at the bottom shows the output of the program:

```

Nama Minuman : Es Kelapa Sawit
Jumlah : 4
Harga : Rp.25000

Nama Pemain : Stev
Durasi Main : 10 jam
Harga : Rp. 500000

Nama Pemain : Jhon
Durasi Main : 5 jam
Harga : Rp. 250000

```

Deskripsi :

Program di atas adalah contoh penggunaan struktur (struct) dan kelas (class) dalam bahasa pemrograman C++. Dalam program tersebut, digunakan dua struktur dan satu kelas untuk merepresentasikan data terkait menu dan reservasi. Struktur pertama, yaitu 'menu', digunakan untuk menyimpan informasi tentang menu, seperti nama, jumlah, dan harga. Di dalam 'main()' fungsi, struktur ini digunakan untuk membuat variabel 'minuman' yang menyimpan informasi tentang satu jenis minuman. Struktur kedua, yaitu 'reservasi', digunakan untuk menyimpan informasi tentang reservasi, seperti nama pemain, durasi main, dan harga. Di dalamnya terdapat fungsi 'tampilkanInfo()' yang digunakan untuk

mencetak informasi reservasi ke layar. Kelas `reservasi` digunakan untuk membuat objek `billiard` dan `vip`, yang merepresentasikan dua jenis reservasi. Objek-objek ini diinisialisasi dengan menggunakan konstruktor yang menerima tiga argumen: nama pemain, durasi main, dan harga. Kemudian, program mencetak informasi tentang minuman dan reservasi ke layar menggunakan perintah `cout`. Secara keseluruhan, program tersebut digunakan untuk memahami konsep dasar dalam C++ seperti struktur, kelas, dan objek, serta cara mengakses dan menggunakan data dalam struktur dan kelas.

Struct:

Struct digunakan untuk mengelompokkan beberapa variabel yang berbeda tipe data menjadi satu kesatuan. Dalam codingan tersebut, struktur `menu` digunakan untuk mengorganisir informasi tentang menu, seperti nama, jumlah, dan harga. Ini memungkinkan pengelompokan data yang terkait dalam satu unit, sehingga memudahkan dalam penggunaan dan manipulasi data terkait menu.

Class:

Kelas digunakan untuk membuat tipe data baru yang menyatukan data dan fungsi-fungsi yang berkaitan dengan data tersebut. Dalam codingan tersebut, kelas `reservasi` digunakan untuk merepresentasikan informasi tentang reservasi dalam permainan. Kelas ini memiliki atribut-atribut seperti nama pemain, durasi main, dan harga, serta fungsi `tampilkanInfo()` untuk menampilkan informasi reservasi ke layar. Dengan menggunakan kelas, kita dapat membuat objek-objek yang merepresentasikan instance khusus dari tipe data reservasi tersebut, yang dapat diakses dan dimanipulasi sesuai kebutuhan.

Unguided 3

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

using namespace std;

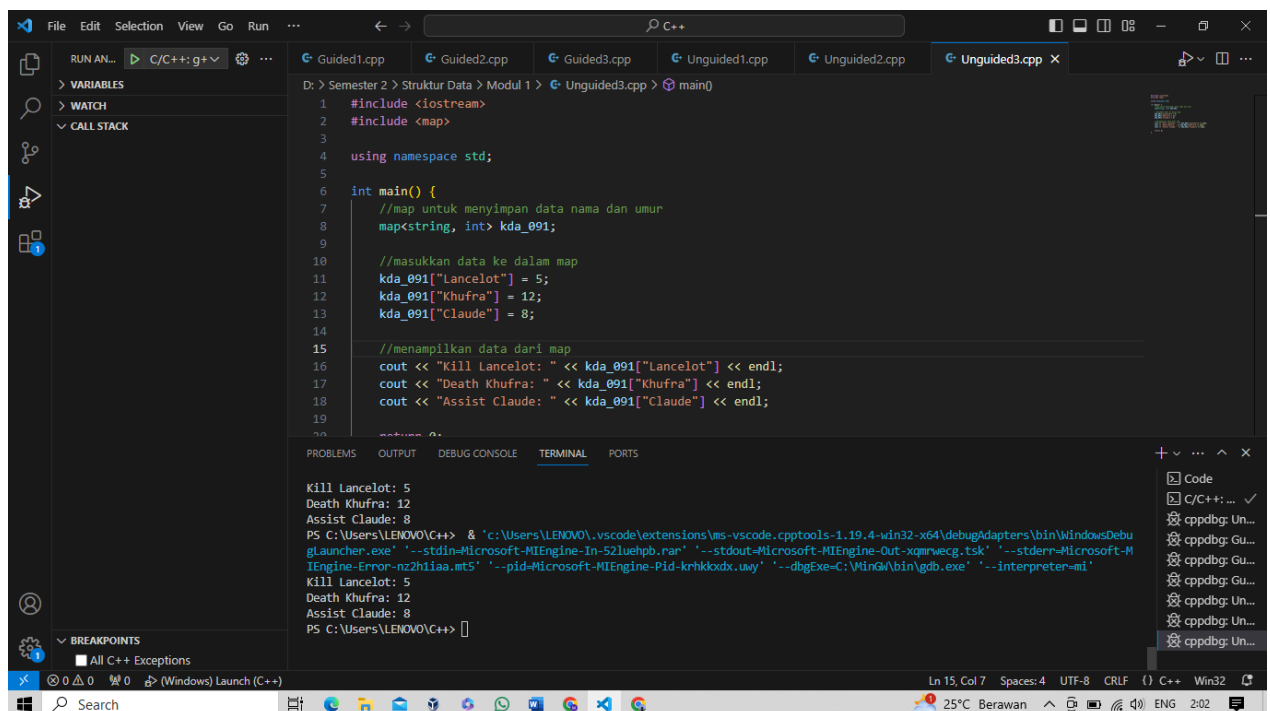
int main() {
    //map untuk menyimpan data nama dan umur
    map<string, int> kda_091;

    //masukkan data ke dalam map
    kda_091["Lancelot"] = 5;
    kda_091["Khufra"] = 12;
    kda_091["Claude"] = 8;

    //menampilkan data dari map
    cout << "Kill Lancelot: " << kda_091["Lancelot"] << endl;
    cout << "Death Khufra: " << kda_091["Khufra"] << endl;
    cout << "Assist Claude: " << kda_091["Claude"] << endl;

    return 0;
}
```

Screenshots Output :



Deskripsi :

Program di atas menggunakan struktur data map dalam bahasa pemrograman C++ untuk menyimpan informasi tentang kill, death, dan assist dari beberapa karakter dalam suatu permainan. Pertama, program mendeklarasikan sebuah map dengan nama `kda_091`. Map ini memiliki key berupa string, yang merupakan nama karakter, dan value berupa integer, yang merupakan nilai kill/death/assist dari karakter tersebut. Kemudian, data kill, death, dan assist dari beberapa karakter dimasukkan ke dalam map menggunakan operator `[]`. Setiap karakter memiliki key unik yang terkait dengan nilai kill/death/assist yang sesuai. Selanjutnya, program mencetak informasi tentang kill, death, dan assist dari beberapa karakter ke layar menggunakan perintah `cout`. Hal ini dilakukan dengan mengakses nilai dalam map menggunakan key-nya. Dengan menggunakan struktur data map, program dapat menyimpan dan mengakses informasi yang terstruktur dengan mudah, seperti statistik pemain dalam suatu permainan, yang dapat digunakan untuk analisis dan pemrosesan lebih lanjut.

Array:

- Array adalah struktur data yang menyimpan elemen-elemen dengan tipe data yang sama dan diindeks secara berurutan.
- Dalam konteks codingan di atas, array tidak digunakan. Namun, secara umum, jika data harus diakses dengan indeks yang berurutan dan tipe datanya sama, array merupakan pilihan yang baik.
- Contoh: `int nilai[5];` akan mendeklarasikan array `nilai` yang dapat menampung 5 bilangan bulat berturut-turut.

Map:

- Map adalah struktur data yang menyimpan pasangan key-value, di mana setiap key unik dan terkait dengan satu nilai tertentu.
- Dalam codingan di atas, map digunakan untuk menyimpan informasi tentang kill, death, dan assist dari beberapa karakter dalam suatu permainan. Setiap karakter (key) memiliki nilai kill/death/assist (value) yang terkait.
- Kelebihan map adalah fleksibilitasnya dalam menyimpan data dengan kunci yang tidak harus berurutan dan tipe data key dan value yang berbeda-beda.
- Contoh: `map<string, int> kda_091;` akan mendeklarasikan map `kda_091` yang memiliki key berupa string (nama karakter) dan value berupa integer (nilai kill/death/assist).

D. Kesimpulan

Setiap bahasa pemrograman itu pasti memiliki banyak tipe data, jadi kalian perlu mempelajari setiap tipe data tersebut supaya bisa menggunakannya. Sebelum masuk ke tipe data dari bahasa program tersebut, kita perlu tahu dulu mengenai tipe data tersebut. Tipe data itu adalah cara kita untuk memberitahu komputer untuk mengelompokkan data berdasarkan apa yang dipahami komputer. Bahasa pemrograman yang merupakan evolusi

dari bahasa pemrograman C ini sebenarnya memiliki 2 tipe data lho. Jadi ada 2 tipe data yakni tipe data sederhana atau Primitive Data Types dan juga tipe data kompleks atau Non Primitive Data Types. Berikut ini beberapa data yang ada dari setiap kategori tipe data yang tersedia pada bahasa pemrograman tersebut.

E. Referensi

<https://eprints.uad.ac.id/32726/1/Dasar%20Pemrograman%20Bahasa%20C++.pdf>

<https://course-net.com/blog/c-adalah/>