

**LAPORAN PRAKTIKUM STRUKTUR
DATA DAN ALGORITMA**

**MODUL I
TIPE DATA**



Disusun Oleh :

Nia Novela Ariandini
2311102057

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. Ini adalah blok bangunan fundamental untuk menyimpan nilai-nilai sederhana. Contoh tipe data primitif:

- 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 (Map terasa mirip dengan array namun dengan index yang memungkinkan untuk berupa tipe data selain integer)

B. Guided

Guided 1

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

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

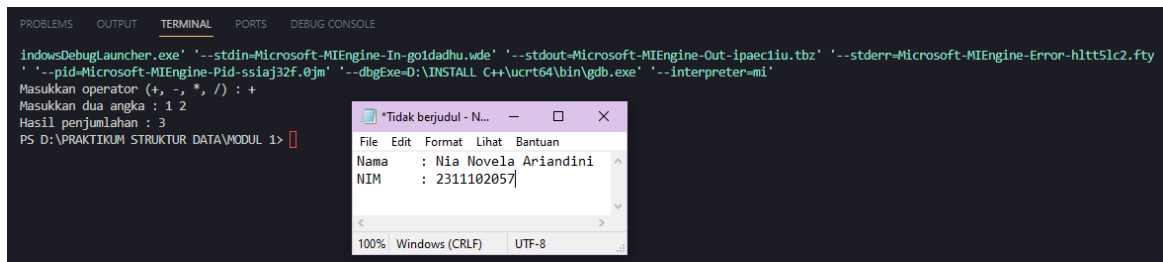
    // Meminta pengguna untuk memasukkan operator: +, -, *, /
    cout << "Masukkan operator (+, -, *, /) : ";
    cin >> op;

    // Meminta pengguna untuk memasukkan operand
    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 :



Deskripsi :

Pada awal program, variabel `op`, `num1`, dan `num2` diinisialisasi untuk menyimpan operator dan dua angka yang akan dioperasikan. Variabel `op` bertipe `char`, sedangkan `num1` dan `num2` bertipe `float`. Selanjutnya, program menggunakan `cout` untuk memberikan instruksi kepada pengguna agar memasukkan operator (+, -, *, /). Kemudian, `cin` digunakan untuk menyimpan operator yang dimasukkan oleh pengguna ke dalam variabel `op`. Program juga menggunakan `cout` untuk meminta pengguna memasukkan dua angka, dan `cin` digunakan untuk menyimpan dua angka tersebut ke dalam variabel `num1` dan `num2`. Selanjutnya, program menggunakan `switch` statement untuk mengevaluasi nilai dari variabel `op`. Setiap case dalam `switch` mengecek operator yang dimasukkan dan melakukan operasi matematika yang sesuai. Jika operator tidak sesuai dengan yang diharapkan, program akan menjalankan blok default dan menampilkan pesan error. Hasil operasi matematika ditampilkan ke layar menggunakan `cout`. Sebelum melakukan pembagian (/), program juga melakukan pemeriksaan apakah `num2` tidak sama dengan nol. Hal ini dilakukan untuk menghindari pembagian dengan nol yang tidak diperbolehkan dalam matematika.

Guided 2

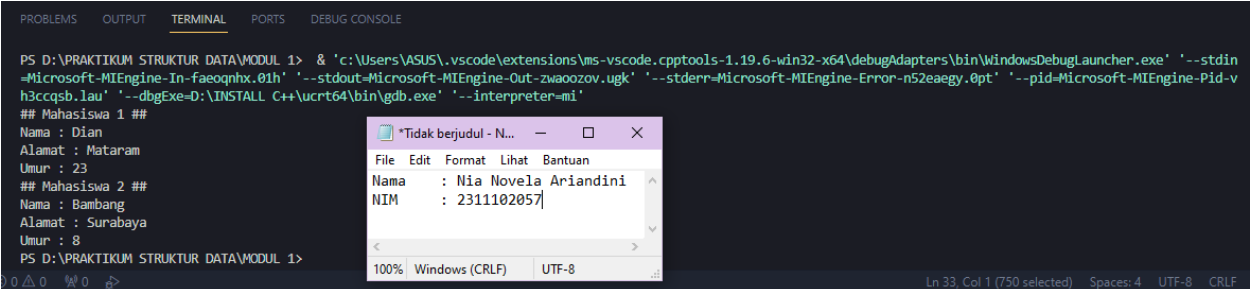
```
#include <stdio.h>  
  
//Struct  
struct Mahasiswa  
{  
    const char *name;  
    const char *address;  
    int age;  
};
```

```

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";
    mhs1.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 :



```

PS D:\PRAKTIKUM STRUKTUR DATA\MODUL 1> & 'c:\Users\ASUS\.vscode\extensions\ms-vscode.cpptools-1.19.6-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin
=Microsoft-MIEngine-In-faeoqhcx.01h' '--stdout=Microsoft-MIEngine-Out-zwaozov.ugk' '--stderr=Microsoft-MIEngine-Error-n52eaegy.0pt' '--pid=Microsoft-MIEngine-Pid-v
h3ccqsb.lau' '--dbgExe=D:\INSTALL C++\ucrt64\bin\gdb.exe' '--interpreter=mi'
## Mahasiswa 1 ##
Nama : Dian
Alamat : Mataram
Umur : 22
## Mahasiswa 2 ##
Nama : Bambang
Alamat : Surabaya
Umur : 23
PS D:\PRAKTIKUM STRUKTUR DATA\MODUL 1>

```

Deskripsi :

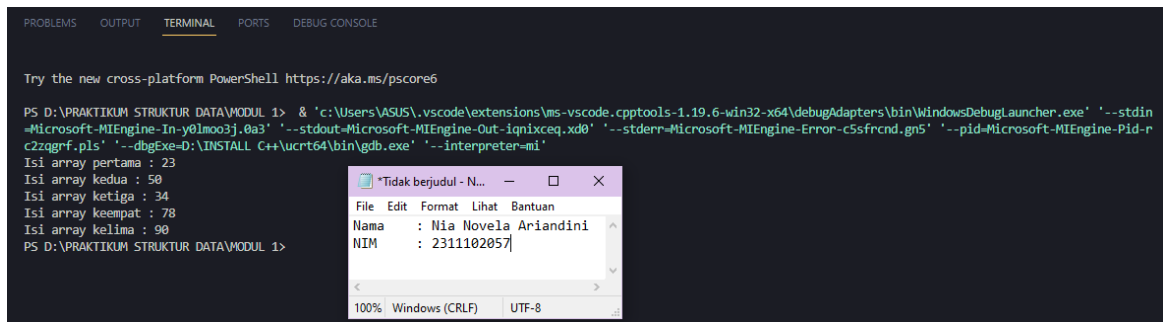
Program ini mendefinisikan sebuah struct bernama Mahasiswa, yang memiliki tiga anggota: name (nama mahasiswa), address (alamat mahasiswa), dan age (usia mahasiswa). Dalam fungsi main, dua objek struct Mahasiswa, yaitu mhs1 dan mhs2, dideklarasikan. Nilai-nilai untuk setiap objek diisi dengan menggunakan operator '='. Kemudian, informasi tentang masing-masing mahasiswa dicetak ke layar menggunakan fungsi printf. Setiap informasi (nama, alamat, dan usia) diakses melalui operator '.'. Dengan demikian, program ini menggunakan struct untuk mengorganisir dan mencetak informasi mahasiswa secara sederhana. Program mencetak nama, alamat, dan usia masing-masing mahasiswa ke layar.

Guided 3

```
#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 D:\PRAKTIKUM STRUKTUR DATA\MODUL 1> & 'c:\Users\VASUS\.vscode\extensions\ms-vscode.cpptools-1.19.6-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin
=Microsoft-MIEngine-In-y0lmoo3j.0a3' '--stdout=Microsoft-MIEngine-Out-ignixceq.xd0' '--stderr=Microsoft-MIEngine-Error-c5sfrcond.gn5' '--pid=Microsoft-MIEngine-Pid-r
c2zqgrf.pls' '--dbgExe=D:\INSTALL C++\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 D:\PRAKTIKUM STRUKTUR DATA\MODUL 1>
```

Deskripsi :

Program di atas menggunakan array untuk menyimpan dan mencetak nilai-nilai. Pada awalnya, kita mendeklarasikan array bernama nilai yang memiliki kapasitas untuk menyimpan 5 nilai bertipe integer. Setelah itu, setiap elemen dalam array diinisialisasi dengan nilai tertentu menggunakan indeks array.

Selanjutnya, kita menggunakan perintah cout untuk mencetak isi array sesuai dengan indeksinya. Proses ini diulangi untuk setiap indeks array, sehingga kita dapat melihat nilai yang disimpan di dalam array.

Hasil dari program ini adalah mencetak nilai-nilai array ke layar, memberikan gambaran sederhana tentang cara mengakses dan menggunakan elemen-elemen array sesuai indeksinya. Dengan menggunakan array, program dapat secara efisien menyimpan sejumlah nilai dalam satu struktur data, memudahkan dalam pengelolaan dan pengaksesan informasi.

D. Unguided/Tugas

Unguided 1

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

// Fungsi untuk menghitung luas persegi panjang
double hitungluaspersegipanjang(double panjang, double lebar)
{
    return panjang * lebar;
}

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

int main()
{
    cout << "==== Menghitung Luas Persegi Panjang dan Volume Balok =====> endl;

    double panjang, lebar, tinggi;

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

    cout << "Masukkan lebar persegi panjang : ";
    cin >> lebar;

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

    double luas = hitungluaspersegipanjang(panjang, lebar);
```

```

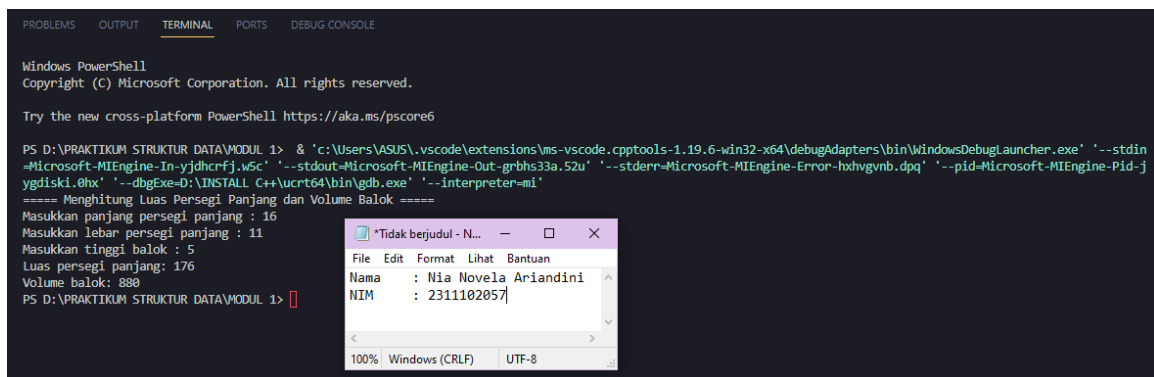
    cout << "Luas persegi panjang: " << luas << endl;

    double volume = hitungvolumebalok(panjang, lebar, tinggi);
    cout << "Volume balok: " << volume << endl;

    return 0;
}

```

Screenshots Output :



Deskripsi :

Program di atas digunakan untuk menghitung luas persegi panjang dan volume balok berdasarkan input panjang, lebar, dan tinggi yang diberikan oleh pengguna. Program ini terdiri dari dua fungsi, yaitu `hitungluaspersegipanjang` untuk menghitung luas persegi panjang dan `hitungvolumebalok` untuk menghitung volume balok. Setiap fungsi menerima parameter berupa panjang, lebar, dan tinggi, dan mengembalikan hasil perhitungan sesuai dengan rumus yang sesuai.

Fungsi utama (main) dimulai dengan mencetak judul program menggunakan `cout`. Selanjutnya, variabel panjang, lebar, dan tinggi dideklarasikan untuk menyimpan input dari pengguna. Selanjutnya, program meminta pengguna untuk memasukkan nilai panjang, lebar, dan tinggi persegi panjang serta balok menggunakan `cout`. Input dari pengguna kemudian disimpan ke dalam variabel yang telah dideklarasikan menggunakan `cin`. Kemudian, program memanggil fungsi `hitungluaspersegipanjang` dengan parameter panjang dan lebar, menyimpan hasilnya dalam variabel `luas`, dan mencetak luas persegi panjang ke layar menggunakan `cout`. Selanjutnya, program memanggil fungsi `hitungvolumebalok` dengan parameter panjang, lebar, dan tinggi, menyimpan hasilnya dalam variabel `volume`, dan mencetak volume balok ke layar menggunakan `cout`.

Kesimpulan Tipe Data Primitif : Tipe data primitif merupakan sebuah tipe data yang sudah ditentukan oleh sistem, disediakan oleh banyak bahasa pemrograman, dan merupakan blok bangunan fundamental untuk menyimpan nilai-nilai yang bersifat sederhana.

Unguided 2

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

// Struct untuk merepresentasikan informasi alat tulis
struct AlatTulis {
    string nama;
    int harga;
    int stok;
};

// Class untuk merepresentasikan jasa di toko alat tulis
class Jasa {
private:
    string nama;
    int waktu;
    int harga;

public:
    // Konstruktor untuk inisialisasi objek Jasa
    Jasa(string namaJasa, int waktuJasa, int hargaJasa) :
        nama(namaJasa), waktu(waktuJasa), harga(hargaJasa) {}

    // Method untuk menampilkan informasi jasa
    void tampilkanInfo() {
        cout << "\nNama Jasa    : " << nama << endl;
        cout << "Durasi jasa : " << waktu << " menit" << endl;
        cout << "Harga          : Rp. " << harga << endl << endl;
    }
};

int main()
{
    cout << "\n==== Penggunaan Struct dan Class Toko Alat
Tulis =====> << endl;
    // Contoh Penggunaan Struct untuk Alat Tulis
    AlatTulis pensil, kertas;

    pensil.nama = "Pensil 2B";
    pensil.harga = 2000;
    pensil.stok = 50;
```

```

    kertas.nama = "Kertas A4";
    kertas.harga = 5000;
    kertas.stok = 100;

    cout << "\nNama Alat Tulis : " << pensil.nama << endl
          << "Harga           : Rp." << pensil.harga <<
endl
          << "Stok           : " << pensil.stok << endl;

    cout << "\nNama Alat Tulis : " << kertas.nama << endl
          << "Harga           : Rp." << kertas.harga <<
endl
          << "Stok           : " << kertas.stok << endl;

    // Contoh Penggunaan Class untuk Jasa
    Jasa pengiriman("Pengiriman Barang", 45, 25000);
    Jasa laminasi("Laminasi Dokumen", 15, 10000);
    pengiriman.tampilkanInfo();
    laminasi.tampilkanInfo();

    return 0;
}

```

Screenshots Output :

```

PROBLEMS  OUTPUT  TERMINAL  PORTS  DEBUG CONSOLE

mhiygi4.n4t' '--dbgExe=D:\INSTALL C++\ucrt64\bin\gdb.exe' '--interpreter=mi'

===== Penggunaan Struct dan Class Toko Alat Tulis =====

Nama Alat Tulis : Pensil 2B
Harga           : Rp.2000
Stok            : 50

Nama Alat Tulis : Kertas A4
Harga           : Rp.5000
Stok            : 100

Nama Jasa       : Pengiriman Barang
Durasi jasa    : 45 menit
Harga          : Rp. 25000

Nama Jasa       : Laminasi Dokumen
Durasi jasa    : 15 menit
Harga          : Rp. 10000

PS D:\PRAKTIKUM STRUKTUR DATA\MODUL 1>

```

Deskripsi :

Program di atas digunakan dengan memanfaatkan struktur (struct) dan kelas (class) untuk menyusun dan merepresentasikan informasi alat tulis serta jasa dalam konteks toko. Dalam bagian pertama, kita menggunakan struktur (struct) yang bernama AlatTulis untuk merepresentasikan informasi alat tulis. Struktur ini memiliki anggota berupa nama, harga, dan stok. Melalui objek-objek seperti pensil dan kertas, kita dapat mengakses dan mengisi informasi terkait masing-masing alat tulis.

Selanjutnya, program mencetak informasi alat tulis, seperti nama, harga, dan stok, ke layar menggunakan perintah cout. Cara mengakses informasi ini adalah dengan menggunakan notasi titik, seperti pensil.nama, pensil.harga, dan pensil.stok. Dalam bagian kedua, kita menggunakan kelas (class) yang dinamai Jasa untuk merepresentasikan jenis-jenis jasa yang ditawarkan di toko alat tulis. Kelas ini memiliki konstruktor untuk inisialisasi objek dengan nama, waktu, dan harga. Method tampilkanInfo digunakan untuk mencetak informasi jasa ke layar. Contoh penggunaan kelas ini ditunjukkan melalui objek-objek seperti pengiriman dan laminasi. Objek-objek ini diinisialisasi menggunakan konstruktor, dan kemudian method tampilkanInfo dipanggil untuk menampilkan informasi jasa ke layar.

A. Fungsi dari 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. Fitur class adalah fitur Object Oriented Program pada bahasa C++.

B. Fungsi dari Struct

Data struktur, atau Struct dalam pemrograman C++, adalah sekumpulan data yang dikelompokkan ke dalam sebuah “tabel” yang memiliki elemen-elemen yang saling terkait. Struct digunakan apabila data yang ingin dikelompokkan memiliki tipe data yang berbeda atau membuat struktur dari suatu record.

Unguided 3

```
#include <iostream>
#include <map> // Library supaya bisa menggunakan fungsi map
dibawah

using namespace std;

int main() {

    // Deklarasi map dengan key int dan value string
    map<int, string> kopi_1611;

    // Menambahkan elemen-elemen ke dalam map
    kopi_1611[1] = "Gula Aren";
    kopi_1611[2] = "Caramel Machiato";
    kopi_1611[3] = "Moccahino";
    kopi_1611[4] = "Kopi Tubruk";
    kopi_1611[5] = "Americano";
```

```

        cout << "==== Selamat Datang di Kopi Tuku! =====> <<
endl;
        for (int i = 1; i <= kopi_1611.size(); ++i) { // Looping
        untuk menampilkan semua elemen map
                cout << "No." << i << " " << kopi_1611[i] << endl; //
        Menampilkan elemen map
        }

        return 0;
}

```

Screenshots Output :

Deskripsi :

Script di atas menggunakan C++ dan memanfaatkan library map untuk membuat dan mengelola map (peta) dengan key bertipe int dan value bertipe string. Dalam program ini, kita mendeklarasikan map yang dinamai kopi_1611 untuk menyimpan informasi jenis kopi yang ditawarkan di suatu tempat kopi. Setiap jenis kopi diberikan nomor urut dan nama, kemudian ditambahkan ke dalam map menggunakan sintaks kopi_1611[key] = value; dimana key adalah bertipe int dan value bertipe string. Pada akhirnya, program menggunakan loop for untuk menampilkan seluruh jenis kopi beserta nomor urutnya. Hasilnya, program mencetak judul "==== Selamat Datang di Kopi Tuku! =====> ke layar dan kemudian menampilkan nomor dan nama kopi dari map tersebut. Pendekatan ini memberikan gambaran bagaimana kita dapat menggunakan map untuk mengorganisir dan mengakses informasi dalam format key-value dengan mudah.

Perbedaan Array dengan Map

1. Indeks : Array harus bilangan bulat non-negatif & mulai berurutan dari 0 ; Map dapat berupa nilai berapa pun & tidak harus berurutan.
2. Ukuran : Array harus dideklarasikan dengan ukuran yang tetap ; Map dapat dideklarasikan dengan ukuran yang dinamis.
3. Tipe Data : Array hanya dapat menggunakan tipe data primitif ; Map dapat menggunakan tipe data apapun.

4. Penggunaan : Array digunakan untuk menyimpan data yang berurutan ; Map digunakan untuk menyimpan data yang tidak berurutan dan perlu diakses berdasarkan key.

E. Kesimpulan

Tipe data C++ merupakan elemen penting dalam pemrograman yang menentukan jenis nilai yang dapat disimpan variabel dan operasi yang dapat dilakukan pada nilai tersebut. Memahami tipe data dengan baik membantu kita sebagai programmer dalam menulis program yang lebih efisien, akurat, dan mudah dibaca. Tipe data primitif merupakan sebuah tipe data yang sudah ditentukan oleh sistem, disediakan oleh banyak bahasa pemrograman, dan merupakan blok bangunan fundamental untuk menyimpan nilai-nilai yang bersifat sederhana.

F. Referensi

- [1] Khoirudin, Algoritma & Struktur Data dengan Python 3. Semarang: Universitas Semarang Press, 2019.
- [2] Meidyan Permata Putri, et al., Algoritma dan Struktur Data. Bandung: Widina Bhakti Persada Bandung, 2022.
- [3] L Sitoris, Algoritma dan Pemrograman: Academia edu, 2015.