

**LAPORAN PRAKTIKUM STRUKTUR
DATA DAN ALGORITMA**

**MODUL I
TIPE DATA**



DISUSUN OLEH :
BINTANG YUDHISTIRA
2311102052

Dosen
WAHYU ANDI SAPUTA, S.Pd., M.Eng

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

A. Dasar Teori

Tipe data adalah klasifikasi variabel untuk menentukan bentuk keadaan data yang akan disimpan dalam memori dan jenis operasi apa yang dapat dilakukan oleh variabel tersebut. Tipe data adalah konsep penting dalam pemrograman. Tipe data mengacu pada jenis nilai yang dapat disimpan dan diolah oleh suatu program. Setiap tipe data memiliki karakteristik dan batasan tertentu yang mempengaruhi cara data tersebut dapat digunakan. Tipe data digunakan untuk menentukan jenis nilai atau value yang akan disimpan ke dalam memori, dan yang akan di proses dalam program. Tipe Data yang akan dipelajari :

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

Tipe data Primitif

Tipe data primitif di bahasa pemrograman C++ adalah tipe data yang sudah terdefinisi secara bawaan dalam bahasa pemrograman tersebut. Tipe data primitif biasanya digunakan untuk menyimpan nilai yang sederhana dan tidak kompleks seperti bilangan bulat, bilangan pecahan, karakter, dan nilai boolean. Berikut adalah daftar tipe data primitif yang umum digunakan di bahasa pemrograman C++ :

1. `int` : tipe data ini digunakan untuk menyimpan bilangan bulat seperti 1, 2, 3, dan seterusnya.
2. `float` : tipe data ini digunakan untuk menyimpan bilangan pecahan seperti 2.5, 3.14, dan seterusnya.
3. `char` : tipe data ini digunakan untuk menyimpan karakter seperti 'a', 'b', 'c', dan seterusnya.
4. `bool` : tipe data ini digunakan untuk menyimpan nilai boolean yang hanya memiliki dua nilai yaitu `true` dan `false`.

Tipe data Abstrak

Tipe data abstrak (TDA) atau lebih dikenal dalam bahasa Inggris sebagai Abstract data type (ADT) merupakan model matematika yang merujuk pada sejumlah bentuk struktur data yang memiliki kegunaan atau perilaku yang serupa; atau suatu tipe data dari suatu bahasa pemrograman yang memiliki semantik yang serupa. Tipe data abstrak umumnya didefinisikan tidak secara langsung, melainkan hanya melalui operasi matematis tertentu sehingga membutuhkan penggunaan tipe data tersebut meski dengan risiko kompleksitas yang lebih tinggi atas operasi tersebut. Tipe data abstrak adalah tipe data yang ditentukan oleh pengguna. Contohnya, Pengguna (coder/programmer) dapat mendefinisikan kelas di C++ atau struktur. Biasanya C++ menyediakan tipe data ini yaitu: kelas, struktur, union, enumerasi, dan typedef.

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 :

1. Array adalah struktur data yang menyimpan elemen-elemen dengan tipe data yang sama dalam urutan tertentu. Setiap elemen diakses dengan menggunakan indeks numerik.
2. Vector adalah struktur data dinamis yang menyimpan koleksi elemen dengan tipe data yang sama dalam urutan tertentu. Vector memperluas ukurannya secara otomatis saat diperlukan.
3. Map adalah struktur data yang memetakan kunci ke nilai. Setiap elemen dalam map terdiri dari sepasang kunci-nilai yang unik.

B. Guided

Guided 1 : Tipe data Primitif

```
#include <iostream>

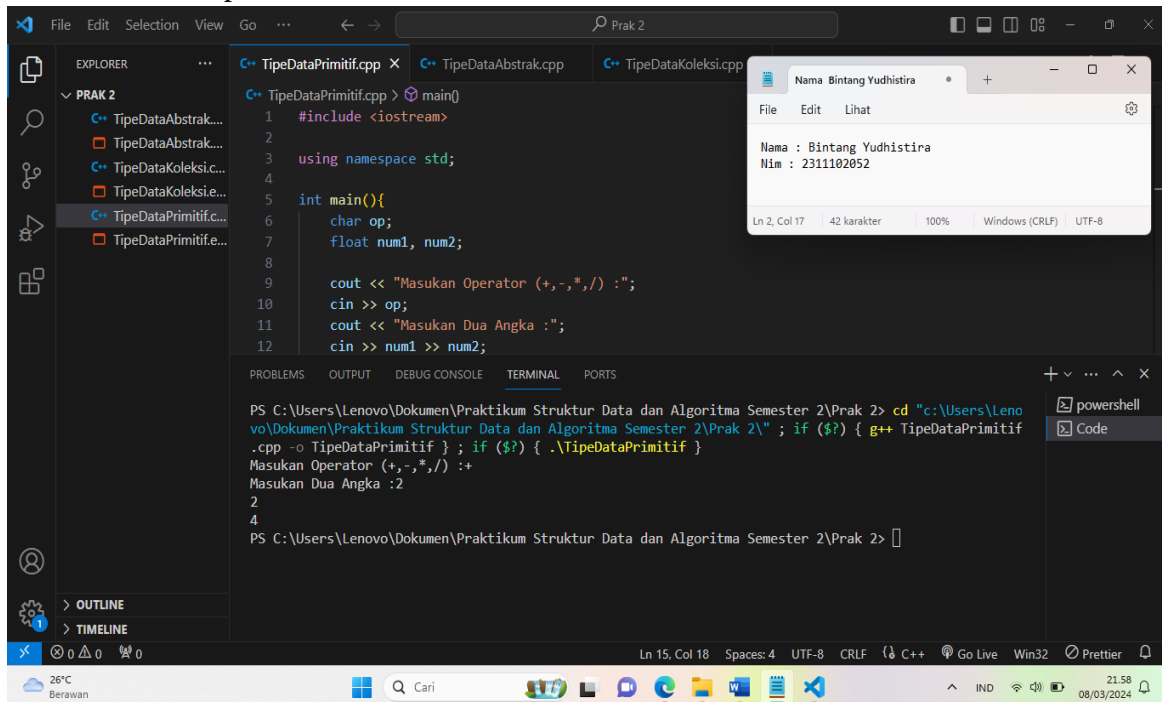
using namespace std;

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

    cout << "Masukan Operator (+,-,*,/) : ";
    cin >> op;
    cout << "Masukan Dua Angka : ";
    cin >> num1 >> num2;

    switch(op){
        case '+':
            cout << num1 + num2;
            break;
        case '-':
            cout << num1 - num2;
            break;
        case '*':
            cout << num1 * num2;
            break;
        case '/':
            cout << num1 / num2;
            break;
        default :
            cout << "Error! operator is not correct";
    }
    return 0;
}
```

Screenshots Output



The screenshot shows a C++ IDE with the following components:

- EXPLORER:** Shows a project named 'Prak 2' with files: TipeDataAbstrak..., TipeDataAbstrak..., TipeDataKoleksi..., TipeDataKoleksi..., TipeDataPrimitif..., and TipeDataPrimitif...
- Code Editor:** Displays the code for 'TipeDataPrimitif.cpp'. The code includes `<iostream>`, uses the `std` namespace, and defines a `main()` function. Inside `main()`, it declares `char op;` and `float num1, num2;`. It prompts the user to enter an operator and two numbers, then performs a calculation based on the operator.
- TERMINAL:** Shows the command prompt output. The user runs the command `cd "c:\Users\Lenovo\Documents\Praktikum Struktur Data dan Algoritma Semester 2\Prak 2\"` and then `g++ TipeDataPrimitif.cpp -o TipeDataPrimitif`. The program prompts for an operator and two numbers, and the user enters `+`, `2`, and `4`. The program then outputs the result of the addition.

Deskripsi:

Deklarasi Variabel: Program mendeklarasikan tiga variabel, yaitu `op` (operator) dengan tipe data `char`, `num1` dan `num2` (dua angka) dengan tipe data `float`.

Input: Program mencetak pesan "Masukan Operator (+,-,*,/) : " dan pengguna diminta untuk memasukkan operator matematika. Kemudian, program mencetak pesan "Masukan Dua Angka : " dan pengguna diminta untuk memasukkan dua angka.

Switch Case: Program menggunakan struktur kontrol `switch` untuk mengevaluasi nilai dari variabel `op` (operator) yang dimasukkan pengguna. Berdasarkan nilai `op`, program akan memilih case yang sesuai:

Jika operator adalah `+`, maka program akan mencetak hasil penjumlahan `num1 + num2`.

Jika operator adalah `-`, maka program akan mencetak hasil pengurangan `num1 - num2`.

Jika operator adalah `*`, maka program akan mencetak hasil perkalian `num1 * num2`.

Jika operator adalah `/`, maka program akan mencetak hasil pembagian `num1 / num2`.

Jika operator tidak cocok dengan salah satu case di atas, maka program akan mencetak pesan "Error! operator is not correct".

Program kemudian mengembalikan nilai 0, menandakan bahwa program berakhir dengan sukses.

Guided 2 : Tipe data Abstrak

```
#include <stdio.h>

// struct
struct Mahasiswa {
    const char *name;
    const char *address;
    int age;
};

int main(){
    // menggunakan struct
    struct Mahasiswa mhs1, mhs2;

    // mengisi nilai 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;
}
```

Screenshot Output

The screenshot displays the Visual Studio Code interface. The Explorer pane on the left shows a project named 'PRAK 2' with files 'TipeDataAbstrak.cpp', 'TipeDataKoleksi.cpp', and 'TipeDataPrimitif.cpp'. The main editor shows the code in 'TipeDataAbstrak.cpp', which includes a struct definition and a main function that prints student information. The Output pane at the bottom shows the execution results of the program.

```
// 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;
```

Output:

```
vo\Dokumen\Praktikum Struktur Data dan Algoritma Semester 2\Prak 2> ; if ($?) { g++ TipeDataAbstrak.
cpp -o TipeDataAbstrak } ; if ($?) { .\TipeDataAbstrak }
## Mahasiswa 1 ##
Nama: Dian
Alamat: Mataram
Umur: 22
## Mahasiswa 2 ##
Nama: Bambang
Alamat: Surabaya
Umur: 23
PS C:\Users\Lenovo\Dokumen\Praktikum Struktur Data dan Algoritma Semester 2\Prak 2>
```

Deskripsi:

Deklarasi Struct Mahasiswa: Program mendeklarasikan sebuah struct dengan nama Mahasiswa, yang memiliki tiga anggota yaitu name (nama mahasiswa), address (alamat mahasiswa), dan age (umur mahasiswa).

Fungsi Main: Program kemudian mendeklarasikan dua variabel bertipe Mahasiswa yaitu mhs1 dan mhs2.

Inisialisasi Nilai Struct: Nilai-nilai untuk setiap anggota struct diisi untuk kedua variabel mhs1 dan mhs2.

Mencetak Isi Struct: Program mencetak informasi mengenai masing-masing mahasiswa (nama, alamat, dan umur) menggunakan printf().

Kembali Nilai 0: Program mengembalikan nilai 0 untuk menandakan bahwa program telah berakhir dengan sukses.

Guided 3 : Tipe data Koleksi

```
#include <iostream>

using namespace std;

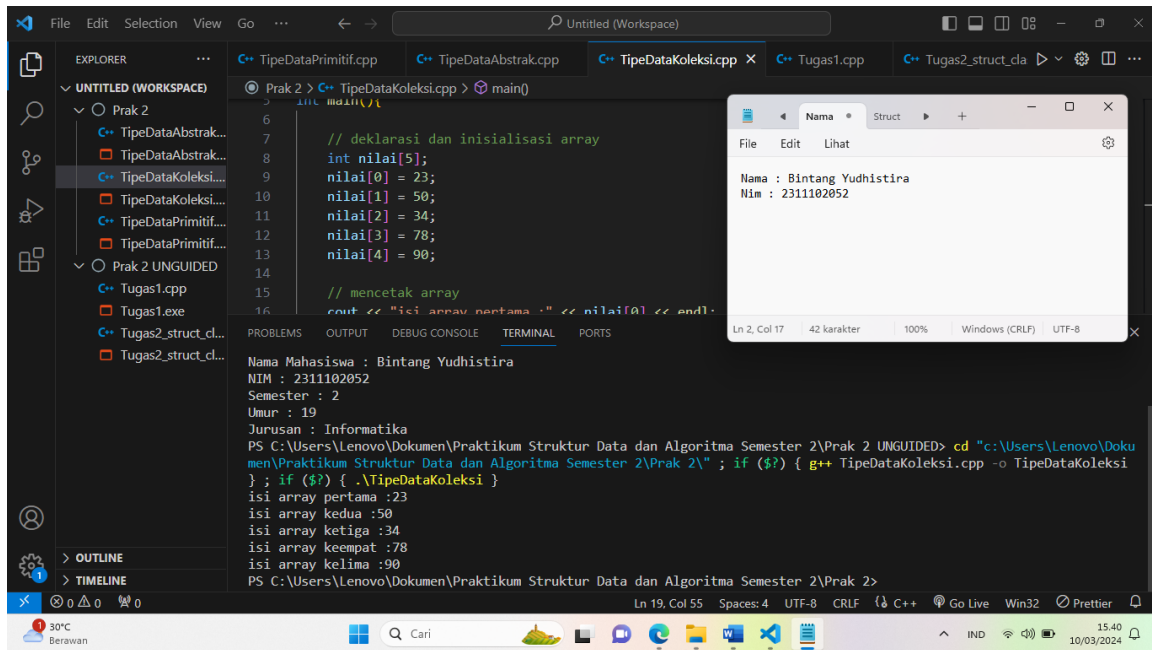
int main(){

    // 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;
}
```


Screenshot Output



The screenshot shows a Visual Studio IDE with a C++ project named 'Prak 2'. The main file, 'TipeDataKoleksi.cpp', contains the following code:

```
1 // main()
2 int main()
3 {
4     // deklarasi dan inisialisasi array
5     int nilai[5];
6     nilai[0] = 23;
7     nilai[1] = 50;
8     nilai[2] = 34;
9     nilai[3] = 78;
10    nilai[4] = 90;
11
12    // mencetak array
13    cout << "isi array pertama : " << nilai[0] << endl;
14
15    return 0;
16 }
```

The output window shows the following text:

```
Nama : Bintang Yudhistira
Nim : 2311102052
```

The terminal window shows the command prompt output:

```
PS C:\Users\Lenovo\Documents\Praktikum Struktur Data dan Algoritma Semester 2\Prak 2 UNGUIDED> cd "c:\Users\Lenovo\Documents\Praktikum Struktur Data dan Algoritma Semester 2\Prak 2\"; if ($?) { g++ TipeDataKoleksi.cpp -o TipeDataKoleksi.exe }
isi array pertama :23
isi array kedua :50
isi array ketiga :34
isi array keempat :78
isi array kelima :90
PS C:\Users\Lenovo\Documents\Praktikum Struktur Data dan Algoritma Semester 2\Prak 2>
```

Deskripsi:

Kodingan di atas adalah program C++ sederhana yang menggunakan array untuk menyimpan dan mencetak nilai-nilai.

1. Deklarasi dan Inisialisasi Array: Array nilai dideklarasikan sebagai array integer dengan ukuran 5, kemudian setiap elemen array diinisialisasi dengan nilai tertentu menggunakan indeks array.
2. Mencetak Array: Setiap elemen array dicetak menggunakan cout. Indeks array digunakan untuk mengakses nilai pada setiap elemen array. Program ini mencetak nilai-nilai array dari indeks 0 hingga 4.
3. Program kemudian mengembalikan nilai 0, menandakan bahwa program berakhir dengan sukses.

C. 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>

using namespace std;

int fungsi1(int a, int b){
    int angka = a + b;
    cout << angka << endl;

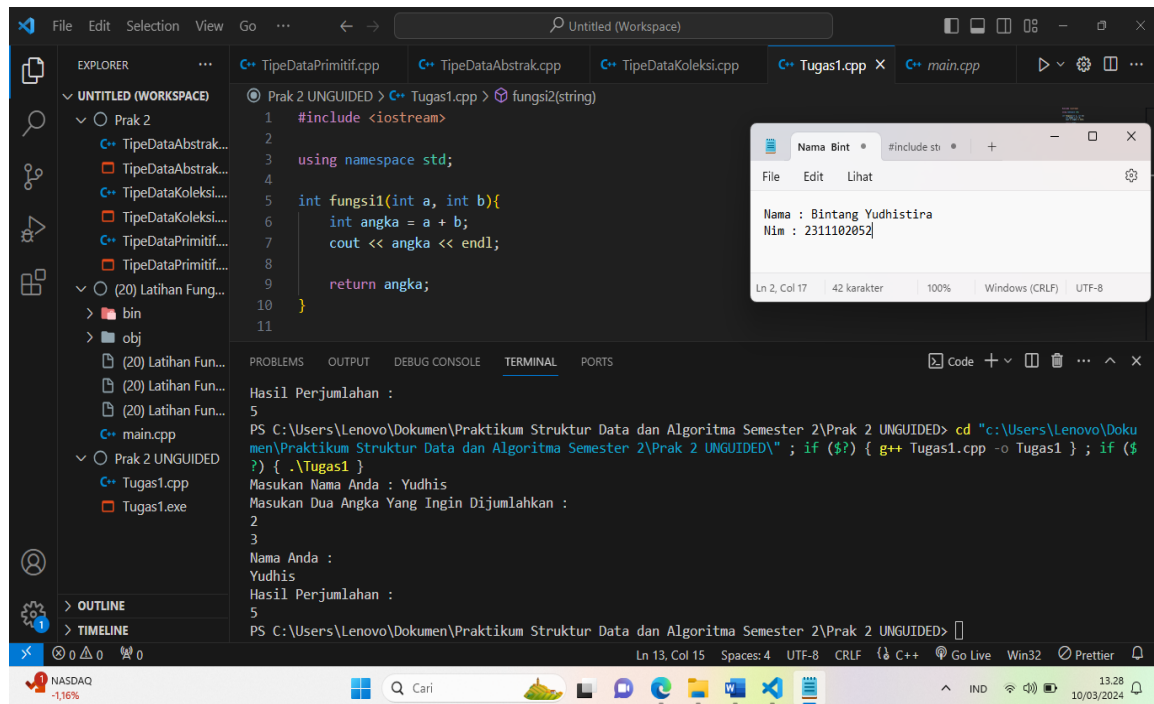
    return angka;
}

string fungsi2(string a){
    cout << a;
    return a;
}

int main (){

    int a, b;
    string nama;
    cout << "Masukan Nama Anda : ";
    cin >> nama;
    cout << "Masukan Dua Angka Yang Ingin Dijumlahkan : " << endl;
    cin >> a >> b;
    cout << "Nama Anda : " << endl;
    fungsi2(nama);
    cout << endl;
    cout << "Hasil Perjumlahan : " << endl;
    fungsi1(a,b);
    return 0;
}
```

Screenshots Output



```
Prak 2 UNGUIDED > C++ Tugas1.cpp > fungsi2(string)
1  #include <iostream>
2
3  using namespace std;
4
5  int fungsi1(int a, int b){
6      int angka = a + b;
7      cout << angka << endl;
8
9      return angka;
10 }
11
```

```
PS C:\Users\Lenovo\Dokumen\Praktikum Struktur Data dan Algoritma Semester 2\Prak 2 UNGUIDED> cd "c:\Users\Lenovo\Dokumen\Praktikum Struktur Data dan Algoritma Semester 2\Prak 2 UNGUIDED\"; if ($?) { g++ Tugas1.cpp -o Tugas1 }; if ($?) { .\Tugas1 }
Masukan Nama Anda : Yudhis
Masukan Dua Angka Yang Ingin Dijumlahkan :
2
3
Nama Anda :
Yudhis
Hasil Penjumlahan :
5
PS C:\Users\Lenovo\Dokumen\Praktikum Struktur Data dan Algoritma Semester 2\Prak 2 UNGUIDED>
```

Deskripsi:

Program ini merupakan program sederhana dalam bahasa C++ yang meminta pengguna memasukkan nama dan dua angka menggunakan fungsi, dimana fungsinya terdiri dari dua fungsi, fungsi pertama untuk menjumlahkan dua bilangan dan fungsi kedua untuk menampilkan string nama. Kemudian program akan menampilkan nama pengguna dan hasil penjumlahan dari dua angka tersebut di fungsi utama yaitu fungsi MAIN.

Fungsi Utama (main): Ini adalah tempat dimulainya eksekusi program. Langkah-langkahnya adalah:

- Mendeklarasikan variabel a dan b untuk menyimpan dua angka yang dimasukkan oleh pengguna.
- Mendeklarasikan variabel nama untuk menyimpan nama yang dimasukkan oleh pengguna.
- Meminta pengguna memasukkan nama dan dua angka.
- Mencetak nama pengguna menggunakan fungsi fungsi2.
- Mencetak hasil penjumlahan dari dua angka tersebut menggunakan fungsi fungsi1.
- Mengembalikan nilai 0 untuk menandakan bahwa program telah berakhir dengan sukses.

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

```
#include <iostream>

using namespace std;

//menggunakan struct
struct dewasiswa{

    string nama;

    string nim;

    int semester;

};

//menggunakan class
class DataDiri {

private:

    int umur;

    string jurusan;

public:

    DataDiri(int _umur, string _jurusan){

        umur = _umur;

        jurusan = _jurusan;

    }

    void display(){

        cout << "Umur : " << umur << endl;

        cout << "Jurusan : " << jurusan << endl;

    }

};
```

```
int main(){

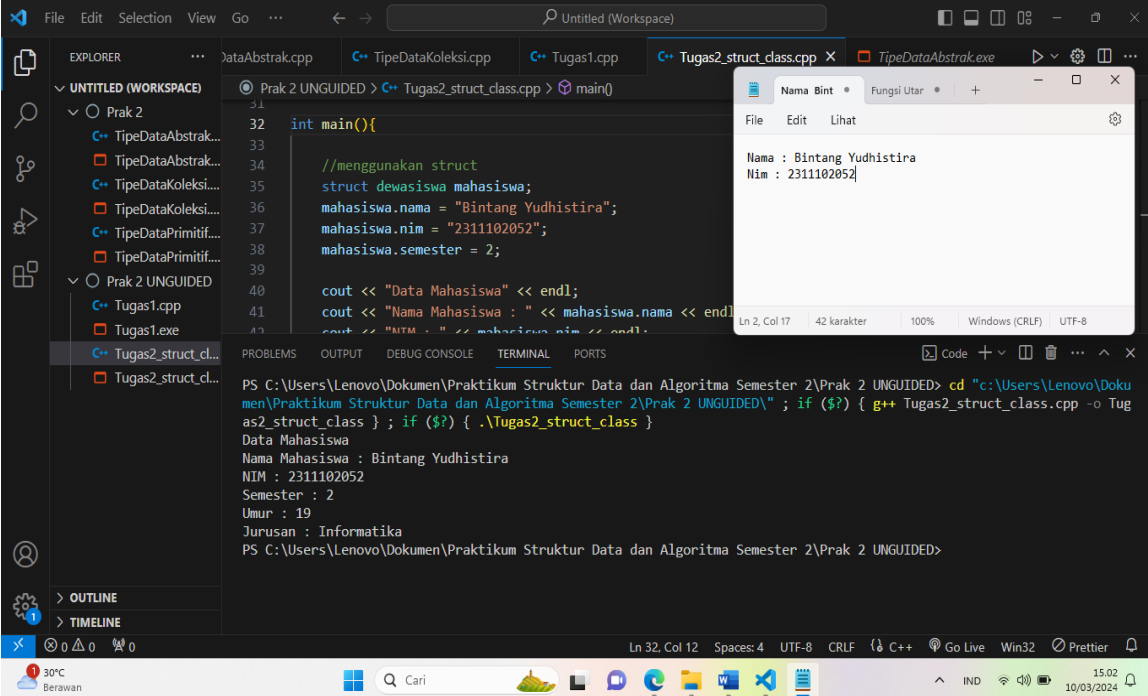
    //menggunakan struct
    struct dewasiswa mahasiswa;
    mahasiswa.nama = "Bintang Yudhistira";
    mahasiswa.nim = "2311102052";
    mahasiswa.semester = 2;

    cout << "Data Mahasiswa" << endl;
    cout << "Nama Mahasiswa : " << mahasiswa.nama << endl;
    cout << "NIM : " << mahasiswa.nim << endl;
    cout << "Semester : " << mahasiswa.semester << endl;

    //menggunakan class
    DataDiri mhs(19, "Informatika");
    mhs.display();

    return 0;
}
```

Screenshot Output



The screenshot shows a C++ IDE with a workspace named 'UNTITLED (WORKSPACE)'. The Explorer panel on the left shows a project named 'Prak 2 UNGUIDED' with files 'TipeDataAbstrak.cpp', 'TipeDataKoleksi.cpp', 'Tugas1.cpp', and 'Tugas2_struct_class.cpp'. The main editor displays the code for 'Tugas2_struct_class.cpp', which defines a struct 'dewasiswa' and a 'main()' function. The 'main()' function creates a 'dewasiswa' struct, initializes its fields with 'Bintang Yudhistira', '2311102052', and '2', and then prints the data. The Output panel at the bottom shows the execution results, which match the values in the struct. A small window titled 'Nama Bint' is also visible, showing the same data.

```
int main(){
    //menggunakan struct
    struct dewasiswa mahasiswa;
    mahasiswa.nama = "Bintang Yudhistira";
    mahasiswa.nim = "2311102052";
    mahasiswa.semester = 2;

    cout << "Data Mahasiswa" << endl;
    cout << "Nama Mahasiswa : " << mahasiswa.nama << endl;
    cout << "NIM : " << mahasiswa.nim << endl;
    cout << "Semester : " << mahasiswa.semester << endl;
}
```

Output:

```
Data Mahasiswa
Nama Mahasiswa : Bintang Yudhistira
NIM : 2311102052
Semester : 2
Umur : 19
Jurusan : Informatika
```

Deskripsi:

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.

Struct adalah 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. Structure adalah kumpulan dari beberapa variabel dalam satu nama. Variabel-variabel yang ada di dalam structure dapat memiliki tipe data yang sama atau berbeda. Dengan menggunakan structure, kita dapat membentuk tipe data baru yang merupakan gabungan dari beberapa variabel. Berikut adalah cara penulisan structure.

Pada contoh program yang saya buat:

Struct dewasiswa: Mendefinisikan sebuah struct yang disebut dewasiswa yang memiliki tiga atribut: nama, nim, dan semester.

Class DataDiri: Mendefinisikan sebuah class yang disebut DataDiri yang memiliki dua atribut private: umur dan jurusan. Terdapat juga sebuah konstruktor untuk menginisialisasi nilai umur dan jurusan, serta sebuah fungsi display() untuk menampilkan data diri mahasiswa.

Fungsi main() :

-Membuat sebuah objek mahasiswa menggunakan struct dewasiswa dan menginisialisasi nilai atributnya.

-Menampilkan data mahasiswa menggunakan objek mahasiswa.

-Membuat sebuah objek mhs menggunakan class DataDiri dengan menggunakan konstruktor untuk menginisialisasi nilai atributnya, kemudian menampilkan data diri mahasiswa menggunakan fungsi display().

Dengan menggunakan struct dan class, program dapat mengelola dan menampilkan informasi tentang mahasiswa dengan cara yang terstruktur dan mudah dipahami.

Struct digunakan untuk menyimpan beberapa data yang berhubungan secara terkait, sementara class digunakan untuk mengelompokkan data dan fungsi-fungsi yang berhubungan menjadi satu kesatuan yang lebih terorganisir.

3. Buat dan jelaskan program menggunakan fungsi map dan jelaskan perbedaan dari array dengan map.

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

using namespace std;

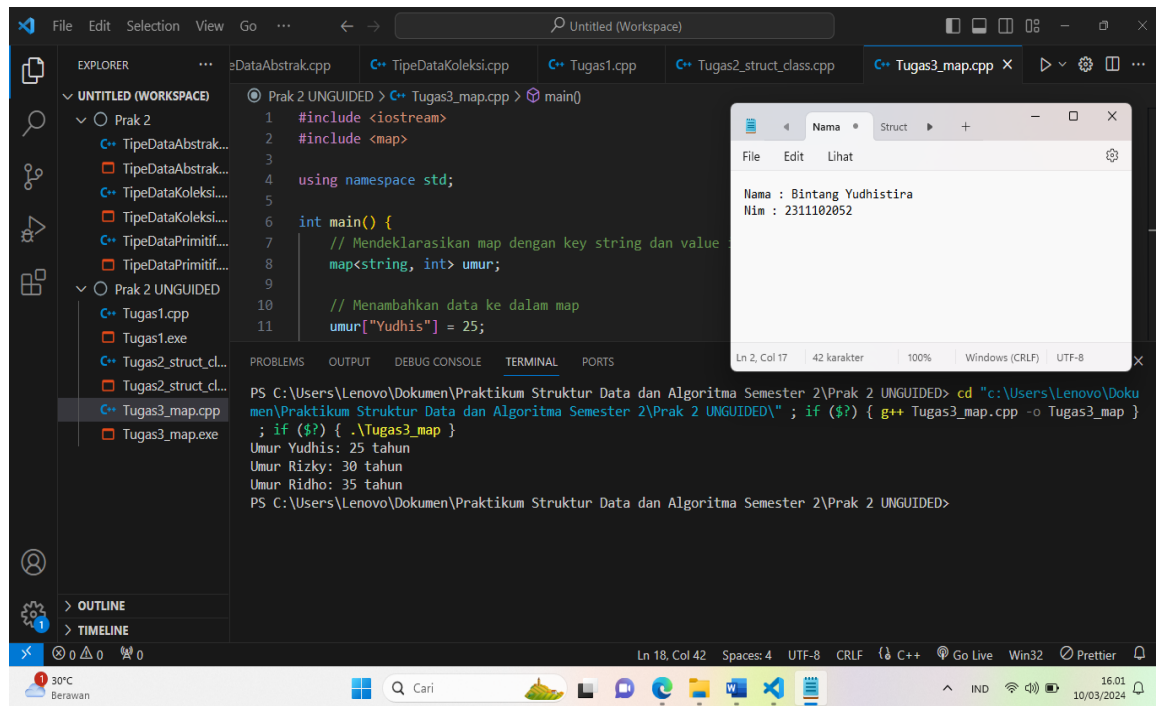
int main() {
    // Mendeklarasikan map dengan key string dan value integer
    map<string, int> umur;

    // Menambahkan data ke dalam map
    umur["Yudhis"] = 25;
    umur["Rizky"] = 30;
    umur["Ridho"] = 35;

    // Mengakses dan mencetak nilai pada map
    cout << "Umur Yudhis: " << umur["Yudhis"] << " tahun" << endl;
    cout << "Umur Rizky: " << umur["Rizky"] << " tahun" << endl;
    cout << "Umur Ridho: " << umur["Ridho"] << " tahun" << endl;

    return 0;
}
```

Screenshot Output



The screenshot shows a C++ IDE with the following components:

- EXPLORER:** Shows the project structure with files like `TipeDataAbstrak.cpp`, `TipeDataKoleksi.cpp`, `Tugas1.cpp`, `Tugas2_struct_class.cpp`, and `Tugas3_map.cpp`.
- EDITOR:** Displays the code for `Tugas3_map.cpp`. The code includes `<iostream>` and `<map>`, uses the `std` namespace, and defines a `main()` function. Inside `main()`, a `map<string, int> umur;` is declared, and three entries are added: `umur["Yudhis"] = 25;`, `umur["Rizky"] = 30;`, and `umur["Ridho"] = 35;`.
- TERMINAL:** Shows the command to compile and run the program: `cd "c:\Users\Lenovo\Documents\Praktikum Struktur Data dan Algoritma Semester 2\Prak 2 UNGUIDED\" ; if ($?) { g++ Tugas3_map.cpp -o Tugas3_map } ; if ($?) { .\Tugas3_map }`. The output shows the ages: `Umur Yudhis: 25 tahun`, `Umur Rizky: 30 tahun`, and `Umur Ridho: 35 tahun`.
- OUTPUT:** A small window titled "Nama" displays the output: `Nama : Bintang Yudhistira` and `Nim : 2311102052`.

Deskripsi:

Array adalah tipe data terstruktur yang berguna untuk menyimpan sejumlah data yang bertipe sama. Array dapat didefinisikan dengan indeks awal dan indeks akhirnya, dan setiap elemen dalam array dapat diakses menggunakan indeks. Array dapat didefinisikan secara statik atau dinamik, dan indeks array selalu dimulai dari 0.

Map, sama seperti array, adalah tipe data terstruktur yang berguna untuk menyimpan data. Namun, map memiliki fitur yang lebih canggih dibandingkan array. Map memiliki index yang memungkinkan untuk berupa tipe data selain integer, yang mirip dengan dictionary di Python. Dalam penggunaan praktis, array lebih cocok untuk menyimpan data bertipe sama yang akan diakses secara berurutan, sedangkan map lebih cocok untuk menyimpan data dengan key yang berbeda, seperti nama-nama dan nilai-nilai.

Pada contoh kodingan yang saya buat :

Deklarasi Map: Program mendeklarasikan sebuah map dengan key bertipe string dan value bertipe integer menggunakan `map<string, int>`.

Menambahkan Data ke Map: Tiga pasangan key-value ditambahkan ke dalam map umur menggunakan operator `[]`. Setiap key dipetakan ke value yang sesuai, yaitu umur masing-masing orang.

Mengakses dan Mencetak Nilai Map: Program mengakses nilai pada map menggunakan key dan mencetaknya menggunakan `cout`. Nilai umur dari setiap orang ditampilkan dengan pesan yang sesuai.

Dengan menggunakan map, kita dapat dengan mudah memetakan key ke value dan mengakses nilai-nilai tersebut sesuai dengan key-nya.

D. Kesimpulan

Kesimpulannya, tipe data dalam pemrograman memainkan peran kunci dalam mengorganisir dan mengelola data dalam sebuah program. Tipe data primitif seperti int, float, char, dan boolean cocok untuk menyimpan nilai sederhana dan menjalankan operasi dasar. Sementara itu, tipe data abstrak dan koleksi seperti array, struct, dan map memungkinkan pengelompokan data yang lebih kompleks dan terstruktur.

Setiap struktur data memiliki karakteristik yang berbeda dan digunakan sesuai dengan kebutuhan program. Penggunaan yang tepat dari tipe data dan struktur data dapat meningkatkan efisiensi, keterbacaan, dan kinerja program. Oleh karena itu, pemahaman yang baik tentang tipe data dan struktur data penting bagi pengembang untuk membangun program yang baik dan efektif.

E. Referensi

Fadielse.(2020, 12 Februari).*Class dan Struct*.

Diakses pada 10 Maret 2024, dari

[Class dan Struct - KodingKita](#)

Microsoft(2023,13 oktober).*Kelas dan Struktur (c++)*.

Diakses pada 10 Maret 2024, dari

[Kelas dan Struktur \(C++\) | Microsoft Learn](#)