

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

**MODUL III
ABSTRACT DATA TYPE (ADT)**



Disusun Oleh :

NAMA : Muhammad Fachri Auravyano Saka

NIM : 103112430180

Dosen

FAHRUDIN MUKTI WIBOWO

**PROGRAM STUDI STRUKTUR DATA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025**

A. Dasar Teori

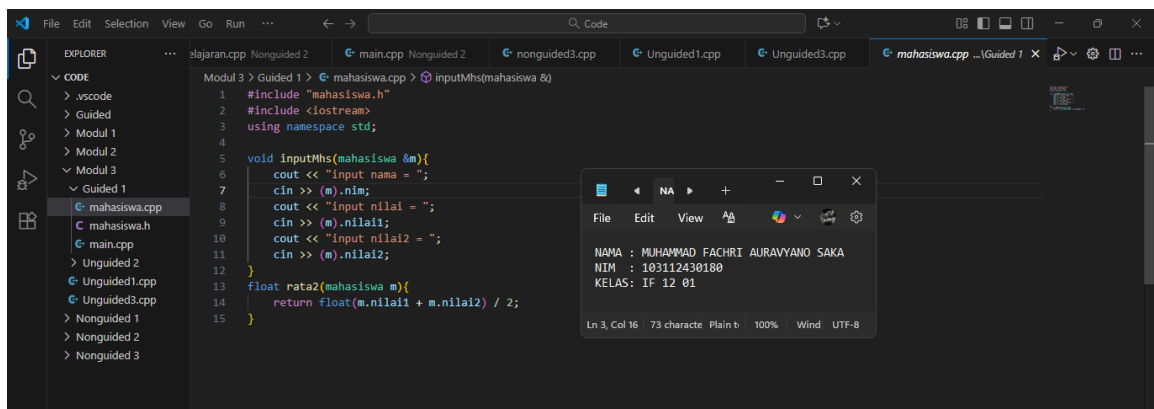
Abstract Data Type (ADT) atau Tipe Data Abstrak merupakan konsep penting dalam pemrograman yang digunakan untuk mendefinisikan tipe data beserta operasi-operasi dasar yang dapat dilakukan terhadapnya. ADT bersifat abstrak karena hanya menjelaskan apa yang dapat dilakukan, bukan bagaimana cara melakukannya. Dengan kata lain, ADT berfungsi untuk memisahkan antara spesifikasi dan implementasi suatu tipe data. Dalam ADT, biasanya terdapat beberapa jenis operasi seperti konstruktor untuk membentuk objek, selektor untuk mengakses nilai komponen, prosedur untuk mengubah nilai, serta operator relasional dan aritmatika terhadap tipe tersebut.

Dalam penerapannya, ADT biasanya dipisahkan menjadi dua bagian utama, yaitu file header (.h) yang berisi definisi tipe dan spesifikasi fungsi atau prosedur, serta file implementasi (.cpp) yang berisi realisasi dari fungsi dan prosedur tersebut. Konsep ini membuat program menjadi lebih terstruktur dan mudah dipahami. Contoh penerapan ADT yang sederhana adalah pembuatan tipe data mahasiswa yang memiliki beberapa atribut seperti NIM dan nilai, lalu dilengkapi dengan fungsi untuk menghitung rata-rata nilai.

B. Guided (berisi screenshot source code & output program disertai penjelasannya)

Guided 1

Mahasiswa.cpp



```
1 #include "mahasiswa.h"
2 #include <iostream>
3 using namespace std;
4
5 void inputMhs(mahasiswa &m){
6     cout << "input nama = ";
7     cin >> (m).nim;
8     cout << "input nilai = ";
9     cin >> (m).nilai1;
10    cout << "input nilai2 = ";
11    cin >> (m).nilai2;
12 }
13
14 float rata2(mahasiswa m){
15     return float(m.nilai1 + m.nilai2) / 2;
16 }
```

```
#include "mahasiswa.h"

#include <iostream>

using namespace std;

void inputMhs(mahasiswa &m){
    cout << "input nama = ";
    cin >> (m).nim;
    cout << "input nilai = ";
```

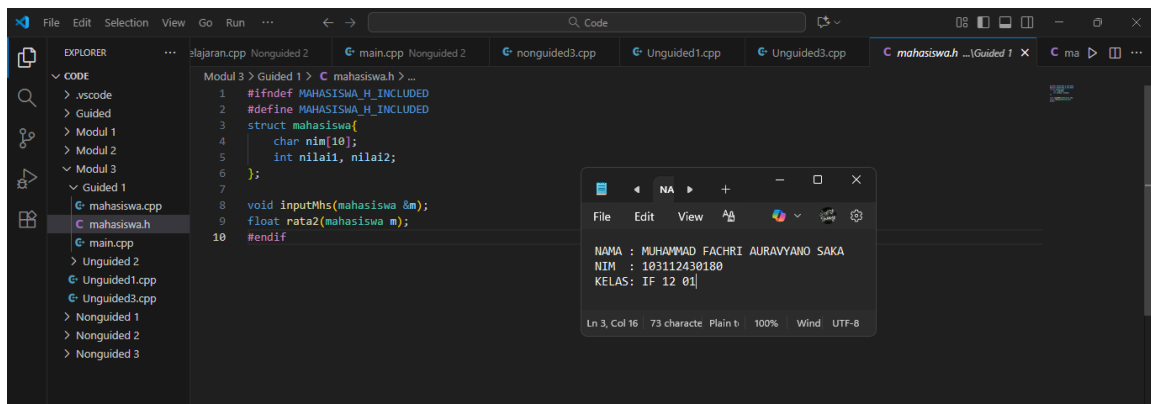
```

        cin >> (m).nilai1;
        cout << "input nilai2 = ";
        cin >> (m).nilai2;
    }

float rata2(mahasiswa m){
    return float(m.nilai1 + m.nilai2) / 2;
}

```

Mahasiswa.h



```

#ifndef MAHASISWA_H_INCLUDED
#define MAHASISWA_H_INCLUDED

struct mahasiswa{

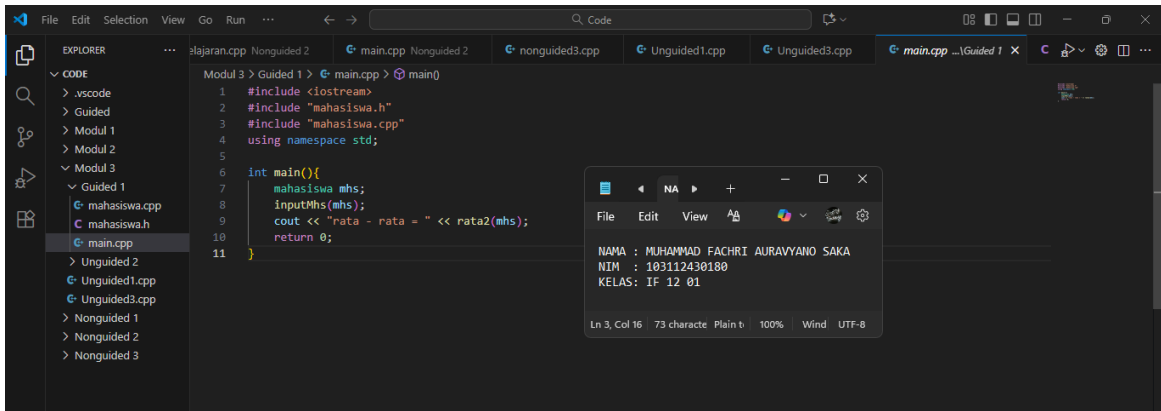
    char nim[10];

    int nilai1, nilai2;
};

void inputMhs(mahasiswa &m);
float rata2(mahasiswa m);
#endif

```

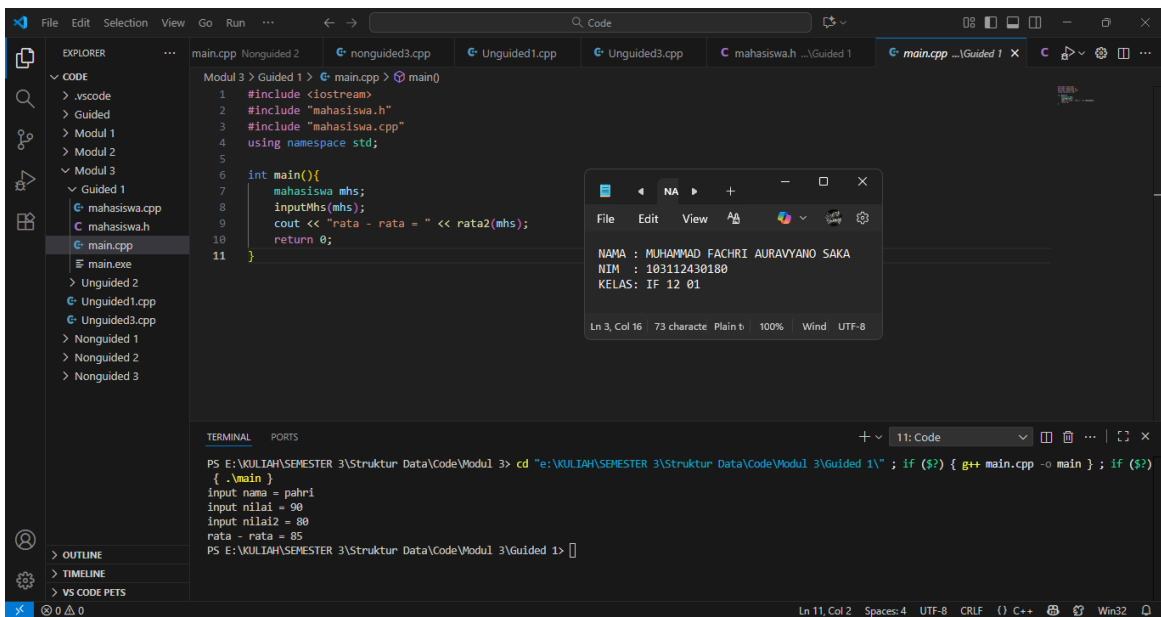
main.cpp



```
#include <iostream>
#include "mahasiswa.h"
#include "mahasiswa.cpp"
using namespace std;

int main(){
    mahasiswa mhs;
    inputMhs(mhs);
    cout << "rata - rata = " << rata2(mhs);
    return 0;
}
```

Screenshots Output



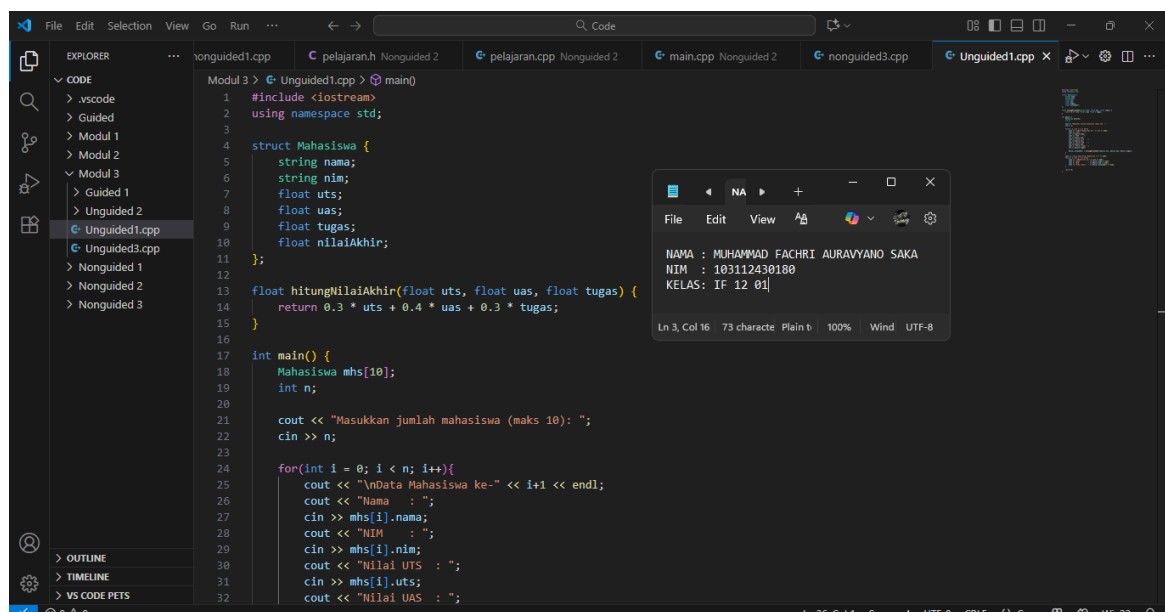
Deskripsi

Code C++ ini dirancang untuk melakukan tugas sederhana: menerima input data seorang

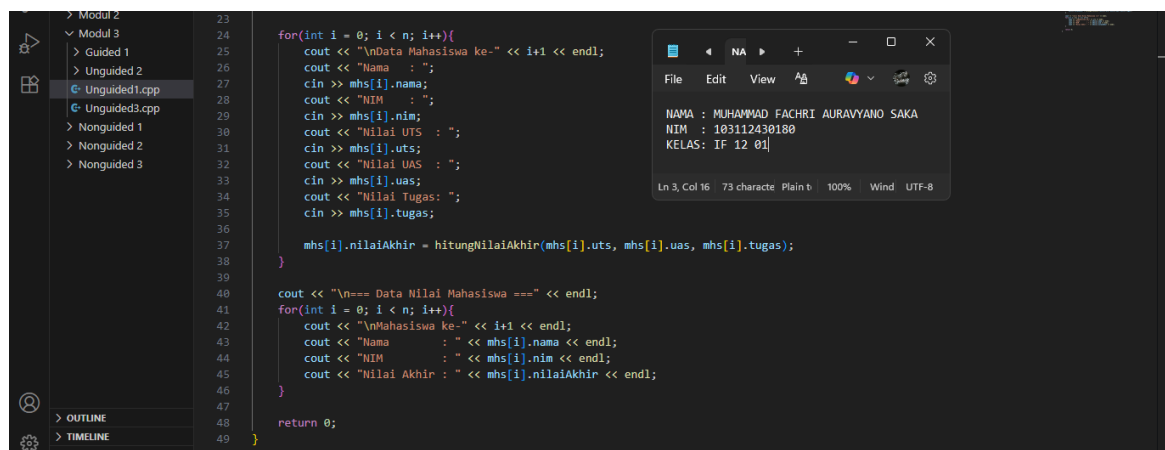
mahasiswa dan menghitung nilai rata-rata dari dua nilai yang dimasukkan. Programnya dibagi menjadi tiga file untuk kerapian. File mahasiswa.h berfungsi sebagai "daftar isi" atau header, yang mendeklarasikan struktur data mahasiswa (untuk menyimpan NIM dan dua nilai) serta dua fungsi utama. File mahasiswa.cpp berisi implementasi atau "penjelasan" dari fungsi-fungsi tersebut, yaitu inputMhs untuk meminta pengguna memasukkan data dan rata2 untuk menghitung rata-rata. Terakhir, file main.cpp adalah program utama yang menjalankan semuanya secara berurutan: pertama memanggil fungsi inputMhs untuk mengisi data, lalu memanggil fungsi rata2 untuk menghitung, dan akhirnya menampilkan hasilnya ke layar.

C. Unguided/Tugas (berisi screenshot source code & output program disertai penjelasannya)

Unguided 1



```
Modul 3 > Unguided1.cpp > main()
1 #include <iostream>
2 using namespace std;
3
4 struct Mahasiswa {
5     string nama;
6     string nim;
7     float uts;
8     float uas;
9     float tugas;
10    float nilaiAkhir;
11};
12
13 float hitungNilaiAkhir(float uts, float uas, float tugas) {
14     return 0.3 * uts + 0.4 * uas + 0.3 * tugas;
15 }
16
17 int main() {
18     Mahasiswa mhs[10];
19     int n;
20
21     cout << "Masukkan jumlah mahasiswa (maks 10): ";
22     cin >> n;
23
24     for(int i = 0; i < n; i++){
25         cout << "\nData Mahasiswa ke-" << i+1 << endl;
26         cout << "Nama    : ";
27         cin >> mhs[i].nama;
28         cout << "NIM     : ";
29         cin >> mhs[i].nim;
30         cout << "Nilai UTS : ";
31         cin >> mhs[i].uts;
32         cout << "Nilai UAS : ";
```



```
33         cin >> mhs[i].uas;
34         cout << "Nilai Tugas : ";
35         cin >> mhs[i].tugas;
36
37         mhs[i].nilaiAkhir = hitungNilaiAkhir(mhs[i].uts, mhs[i].uas, mhs[i].tugas);
38     }
39
40     cout << "\n== Data Nilai Mahasiswa ==> << endl;
41     for(int i = 0; i < n; i++){
42         cout << "\nMahasiswa ke-" << i+1 << endl;
43         cout << "Nama      : " << mhs[i].nama << endl;
44         cout << "NIM       : " << mhs[i].nim << endl;
45         cout << "Nilai Akhir : " << mhs[i].nilaiAkhir << endl;
46     }
47
48     return 0;
49 }
50
```

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

```

struct Mahasiswa {
    string nama;
    string nim;
    float uts;
    float uas;
    float tugas;
    float nilaiAkhir;
};

float hitungNilaiAkhir(float uts, float uas, float tugas)
{
    return 0.3 * uts + 0.4 * uas + 0.3 * tugas;
}

int main() {
    Mahasiswa mhs[10];
    int n;

    cout << "Masukkan jumlah mahasiswa (maks 10): ";
    cin >> n;

    for(int i = 0; i < n; i++){
        cout << "\nData Mahasiswa ke-" << i+1 << endl;
        cout << "Nama      : ";
        cin >> mhs[i].nama;
        cout << "NIM       : ";
        cin >> mhs[i].nim;
        cout << "Nilai UTS   : ";
        cin >> mhs[i].uts;
        cout << "Nilai UAS   : ";
    }
}

```

```

        cin >> mhs[i].uas;

        cout << "Nilai Tugas: ";

        cin >> mhs[i].tugas;

        mhs[i].nilaiAkhir = hitungNilaiAkhir(mhs[i].uts,
mhs[i].uas, mhs[i].tugas);

    }

    cout << "\n=== Data Nilai Mahasiswa ===" << endl;
    for(int i = 0; i < n; i++){

        cout << "\nMahasiswa ke-" << i+1 << endl;

        cout << "Nama          : " << mhs[i].nama << endl;

        cout << "NIM             : " << mhs[i].nim << endl;

        cout << "Nilai Akhir : " << mhs[i].nilaiAkhir <<
endl;

    }

    return 0;
}

```

Screenshot Output

The screenshot shows the Visual Studio Code interface with a C++ program running in the terminal. The program calculates the final grade of three students based on their UTS, UAS, and Tugas scores. The output displays the student data and their calculated final grades.

```

PS E:\VULIAH\SEMESTER 3\Struktur Data\Code\Modul 3> cd "e:\VULIAH\SEMESTER 3\Struktur Data\Code\Modul 3\"; if ($?) { g++ Unguided1.cpp -o Unguided1 }; if ($?) { .\Unguided1 }

Masukkan jumlah mahasiswa (maks 10): 3

Data Mahasiswa ke-1
Nama      : pahr1
NIM       : 123
Nilai UTS : 90
Nilai UAS : 80
Nilai Tugas: 70

Data Mahasiswa ke-2
Nama      : budi
NIM       : 321
Nilai UTS : 95
Nilai UAS : 85
Nilai Tugas: 75

Data Mahasiswa ke-3
Nama      : toni
NIM       : 456
Nilai UTS : 10
Nilai UAS : 20
Nilai Tugas: 30

=== Data Nilai Mahasiswa ===

Mahasiswa ke-1
Nama      : pahr1
NIM       : 123
Nilai Akhir : 80

Mahasiswa ke-2
Nama      : budi
NIM       : 321
Nilai Akhir : 85

Mahasiswa ke-3
Nama      : toni
NIM       : 456
Nilai Akhir : 20

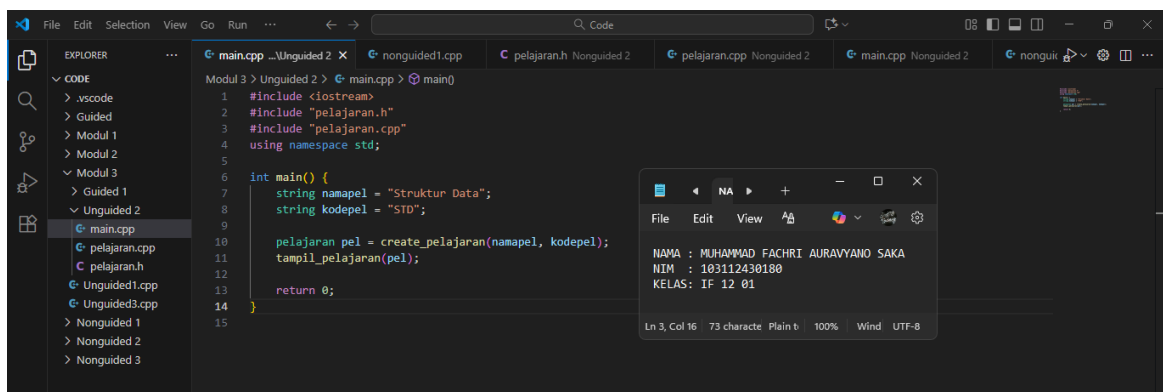
```

Deskripsi

Program pertama dibuat untuk menyimpan data beberapa mahasiswa dengan jumlah maksimal sepuluh orang. Setiap mahasiswa memiliki data berupa nama, NIM, nilai UTS, UAS, dan tugas. Program ini juga memiliki fungsi khusus untuk menghitung nilai akhir berdasarkan rumus $0.3 \times \text{UTS} + 0.4 \times \text{UAS} + 0.3 \times \text{Tugas}$. Data mahasiswa disimpan di dalam array yang berisi beberapa elemen bertipe struct, sehingga setiap mahasiswa memiliki tempat penyimpanan tersendiri. Setelah semua data diinput, program akan menampilkan kembali nama, NIM, dan nilai akhir dari setiap mahasiswa. Program ini menunjukkan penerapan dasar struktur data dan fungsi dalam C++ secara sederhana.

Unguided 2

Main.cpp



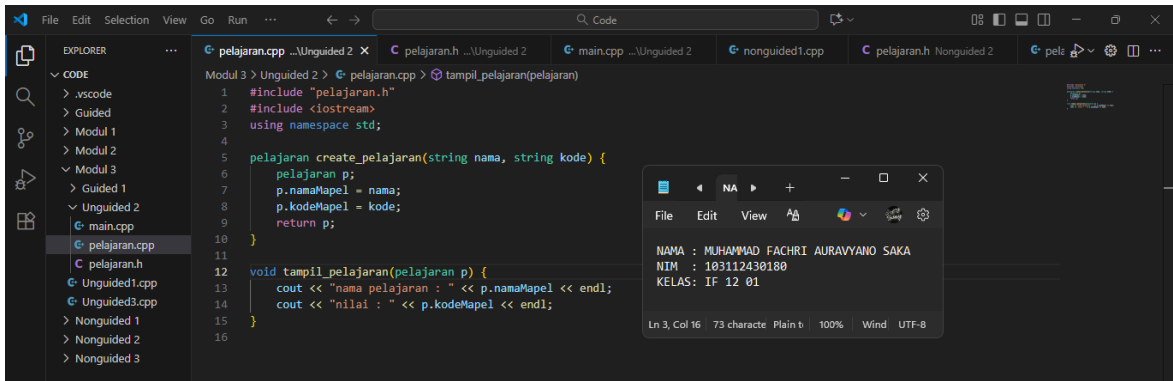
```
#include <iostream>
#include "pelajaran.h"
#include "pelajaran.cpp"
using namespace std;

int main() {
    string namapel = "Struktur Data";
    string kodepel = "STD";

    pelajaran pel = create_pelajaran(namapel, kodepel);
    tampil_pelajaran(pel);

    return 0;
}
```

Pelajaran.cpp



```
#include "pelajaran.h"

#include <iostream>

using namespace std;

pelajaran create_pelajaran(string nama, string kode) {

    pelajaran p;

    p.namaMapel = nama;

    p.kodeMapel = kode;

    return p;

}

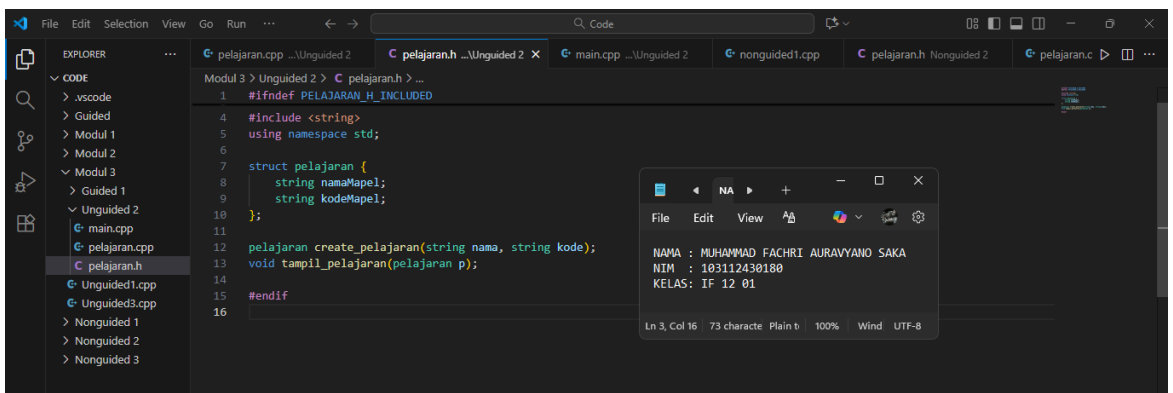
void tampil_pelajaran(pelajaran p) {

    cout << "nama pelajaran : " << p.namaMapel << endl;

    cout << "nilai : " << p.kodeMapel << endl;

}
```

Pelajaran.h



```
#ifndef PELAJARAN_H_INCLUDED
```

```

#define PELAJARAN_H_INCLUDED

#include <string>
using namespace std;

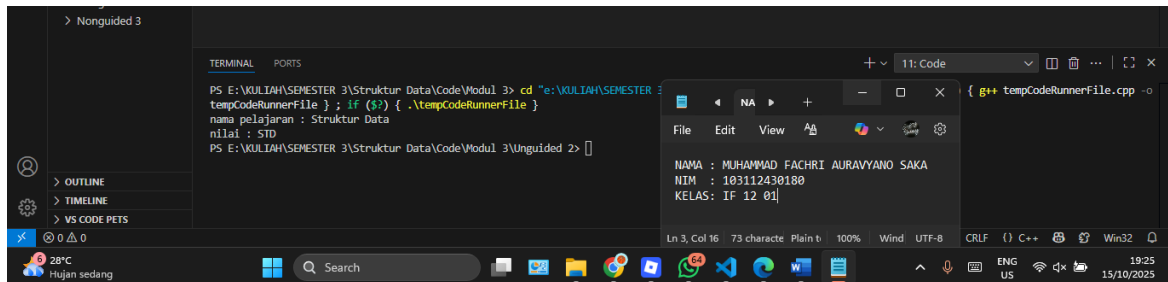
struct pelajaran {
    string namaMapel;
    string kodeMapel;
};

pelajaran create_pelajaran(string nama, string kode);
void tampil_pelajaran(pelajaran p);

#endif

```

Screenshots Output



Deskripsi:

Code C++ kamu kali ini bertujuan untuk mengelola data pelajaran, yang terdiri dari nama dan kode mata pelajaran. Strukturnya sangat baik, terbagi menjadi tiga file yang masing-masing punya tugas spesifik. File pelajaran.h adalah header yang mendefinisikan "cetakan" atau struct untuk data pelajaran dan mendeklarasikan fungsi-fungsi yang akan digunakan. File pelajaran.cpp berisi detail implementasi dari fungsi create_pelajaran yang bertugas membuat objek pelajaran baru dan tampil_pelajaran yang berfungsi untuk menampilkannya. Terakhir, main.cpp menjadi "otak" dari program, di mana kamu mendefinisikan data mata pelajaran "Struktur Data" dengan kode "STD", lalu memanggil kedua fungsi tadi untuk membuat dan menampilkan data tersebut ke layar.

Unguided 3

The screenshot shows the Visual Studio Code editor with the file `Unguided3.cpp` open. The code implements a swap function and a main function. The swap function is defined as `tukarPosisi` and `tukarPointer`. The main function initializes two 3x3 arrays, `A` and `B`, and calls the swap functions to swap elements and pointers.

```
1 #include <iostream>
2 using namespace std;
3
4 void tampilArray(int arr[3][3]) {
5     for (int i = 0; i < 3; i++) {
6         for (int j = 0; j < 3; j++) {
7             cout << arr[i][j] << " ";
8         }
9         cout << endl;
10    }
11 }
12
13 void tukarPosisi(int arr1[3][3], int arr2[3][3], int baris, int kolom) {
14     int temp;
15     temp = arr1[baris][kolom];
16     arr1[baris][kolom] = arr2[baris][kolom];
17     arr2[baris][kolom] = temp;
18 }
19
20 void tukarPointer(int *p1, int *p2) {
21     int temp;
22     temp = *p1;
23     *p1 = *p2;
24     *p2 = temp;
25 }
26
27 int main() {
28     int A[3][3] = { {1,2,3}, {4,5,6}, {7,8,9} };
29     int B[3][3] = { {9,8,7}, {6,5,4}, {3,2,1} };
30     int *ptr1, *ptr2;
31     int x = 10, y = 20;
32 }
```

The screenshot shows the Visual Studio Code editor with the file `Unguided3.cpp` open. The code implements a swap function and a main function. The main function initializes two 3x3 arrays, `A` and `B`, and calls the swap functions to swap elements and pointers. The output of the program is displayed in the console window.

```
33 cout << "Array A:" << endl;
34 tampilArray(A);
35 cout << "\nArray B:" << endl;
36 tampilArray(B);
37
38 cout << "\nMenukar posisi elemen (baris=1, kolom=2)" << endl;
39 tukarPosisi(A, B, 1, 2); // posisi baris ke-1 kolom ke-2
40
41 cout << "\nArray A setelah ditukar:" << endl;
42 tampilArray(A);
43 cout << "\nArray B setelah ditukar:" << endl;
44 tampilArray(B);
45
46 // pointer
47 ptr1 = &x;
48 ptr2 = &y;
49
50 cout << "\nSebelum tukar pointer:" << endl;
51 cout << "x = " << x << ", y = " << y << endl;
52
53 tukarPointer(ptr1, ptr2);
54
55 cout << "Setelah tukar pointer:" << endl;
56 cout << "x = " << x << ", y = " << y << endl;
57
58 return 0;
59 }
60 }
```

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

void tampilArray(int arr[3][3]) {
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            cout << arr[i][j] << " ";
        }
        cout << endl;
    }
}
```

```

void tukarPosisi(int arr1[3][3], int arr2[3][3], int
baris, int kolom) {
    int temp;
    temp = arr1[baris][kolom];
    arr1[baris][kolom] = arr2[baris][kolom];
    arr2[baris][kolom] = temp;
}

void tukarPointer(int *p1, int *p2) {
    int temp;
    temp = *p1;
    *p1 = *p2;
    *p2 = temp;
}

int main() {
    int A[3][3] = { {1,2,3}, {4,5,6}, {7,8,9} };
    int B[3][3] = { {9,8,7}, {6,5,4}, {3,2,1} };
    int *ptr1, *ptr2;
    int x = 10, y = 20;

    cout << "Array A:" << endl;
    tampilArray(A);
    cout << "\nArray B:" << endl;
    tampilArray(B);

    cout << "\nMenukar posisi elemen (baris=1, kolom=2)"
<< endl;

    tukarPosisi(A, B, 1, 2); // posisi baris ke-1 kolom
ke-2

```

```

    cout << "\nArray A setelah ditukar:" << endl;
    tampilArray(A);

    cout << "\nArray B setelah ditukar:" << endl;
    tampilArray(B);

    // pointer
    ptr1 = &x;
    ptr2 = &y;

    cout << "\nSebelum tukar pointer:" << endl;
    cout << "x = " << x << ", y = " << y << endl;

    tukarPointer(ptr1, ptr2);

    cout << "Setelah tukar pointer:" << endl;
    cout << "x = " << x << ", y = " << y << endl;

    return 0;
}

```

Screenhot Output

The screenshot shows the output of a C++ program in a terminal window. The program demonstrates a pointer swap function. The output is as follows:

```

PS E:\KULIAH\SEMESTER 3\Struktur Data\Code\Modul 3\Unguided 2> cd "e:\KULIAH\SEMESTER 3\Struktur Data\Code\Modul 3\" ; if ($?) { g++ Unguided3.cpp -o Unguided3
} ; if ($?) { .\Unguided3 }
Array A:
1 2 3
4 5 6
7 8 9

Array B:
9 8 7
6 5 4
3 2 1

Menukar posisi elemen (baris=1, kolom=2)

Array A setelah ditukar:
1 2 3
4 5 4
7 8 9

Array B setelah ditukar:
9 8 7
6 5 6
3 2 1

Sebelum tukar pointer:
x = 10, y = 20
Setelah tukar pointer:
x = 20, y = 10
PS E:\KULIAH\SEMESTER 3\Struktur Data\Code\Modul 3>

```

A small window in the foreground displays the following information:

```

NAMA : MUHAMMAD FACHRI AURAVYANO SAKA
NIM : 103112430180
KELAS: IF 12 01

```

The status bar at the bottom of the window indicates: Ln 3, Col 16, 73 character, Plain text, 100%, Wind, UTF-8.

Deskripsi

Program kedua dibuat untuk mempraktikkan penggunaan array dua dimensi dan pointer dalam bahasa C++. Dalam program ini terdapat dua buah array 3x3 yang masing-masing berisi angka, serta dua variabel yang diakses melalui pointer. Program memiliki beberapa fungsi, yaitu fungsi untuk menampilkan isi array 2D, fungsi untuk menukar elemen dari dua array pada posisi tertentu, dan fungsi untuk menukar nilai dari dua variabel yang ditunjuk oleh pointer. Setelah dilakukan proses pertukaran, hasilnya ditampilkan ke layar agar perubahan data dapat terlihat. Program ini membantu memahami konsep dasar penggunaan array 2D, pointer, serta cara menukar data menggunakan fungsi dan parameter.

D. Kesimpulan

Berdasarkan praktikum yang telah dilaksanakan, dapat disimpulkan bahwa Abstract Data Type (ADT) adalah sebuah konsep fundamental dalam pemrograman yang sangat efektif untuk membuat kode menjadi lebih terstruktur dan modular. Melalui implementasi pada program Guided dan tugas Unguided, konsep pemisahan antara spesifikasi (deklarasi dalam file .h) dan implementasi (definisi dalam file .cpp) berhasil diterapkan dengan baik. Praktikum ini membuktikan bahwa dengan ADT, kita dapat membuat tipe data baru seperti mahasiswa dan pelajaran yang membungkus data beserta operasinya, sehingga lebih mudah dikelola dan digunakan kembali. Selain itu, pengerjaan tugas juga memperkuat pemahaman tentang konsep-konsep pendukung seperti penggunaan struct, array, dan pointer yang krusial dalam membangun dan memanipulasi struktur data yang lebih kompleks.

E. Referensi

Schmidt, D. C., ADTs, D. O. U., & ADTs, B. I. C++ Support for Abstract Data Types.

Anggoro, W. (2018). C++ Data Structures and Algorithms: Learn how to write efficient code to build scalable and robust applications in C++. Packt Publishing Ltd.