LAPORAN PRAKTIKUM
STRUKTUR DATA

MODUL III
ABSTRACT DATA TYPE (ADT)

Disusun Oleh :
NAMA : RISKY CAHAYU
NIM : 103112430121

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 dasar dalam pemrograman yang berfokus pada pendefinisian struktur data dan operasi yang dapat dilakukan terhadapnya tanpa menampilkan detail implementasi internalnya. Dengan kata lain, ADT menekankan apa yang dapat dilakukan oleh suatu tipe data, bukan bagaimana hal tersebut dilakukan.

Dalam ADT, data dan operasi yang berhubungan dikelompokkan menjadi satu kesatuan logis. Hal ini membantu programmer untuk bekerja pada level abstraksi yang lebih tinggi, sehingga program menjadi lebih mudah dipahami, dikembangkan, dan dipelihara.

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

Guided 1

Kode mahasiswa.cpp

```cpp
// Risky Cahayu
// 103112430121

#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;
}
```

Kode mahasiswa.h

```cpp
// Risky Cahayu
// 103112430121

#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
```

Kode main.cpp

```
// Risky Cahayu
// 103112430121

#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

```
PS C:\Users\ASUS\Videos\strukdat> cd "c:\Users\ASUS\Videos\strukdat\" ; if ($?) { g++ main.cpp -o main } ; if ($?) { .\main }
input nama = Cahayu
input nilai = 100
input nilai2 = 80
rata - rata =90
PS C:\Users\ASUS\Videos\strukdat>
```

Deskripsi:

Program di atas merupakan implementasi sederhana dari Abstract Data Type (ADT) menggunakan bahasa C++. Program ini mendefinisikan sebuah struktur bernama mahasiswa yang memiliki tiga anggota data, yaitu nim, nilai1, dan nilai2, untuk menyimpan informasi dasar mahasiswa. Melalui fungsi inputMhs(), pengguna diminta untuk memasukkan data mahasiswa secara interaktif lewat terminal. Fungsi rata2() kemudian digunakan untuk menghitung nilai rata-rata dari dua nilai yang telah dimasukkan. Struktu

r dan fungsi tersebut dideklarasikan di file header mahasiswa.h, diimplementasikan di file mahasiswa.cpp, dan dijalankan melalui main.cpp yang memanggil kedua fungsi tersebut. Program ini menunjukkan penerapan konsep modularitas dan enkapsulasi ADT, di mana data dan operasi terkait mahasiswa dipisahkan dalam file terstruktur agar kode lebih mudah dipahami dan dipelihara.

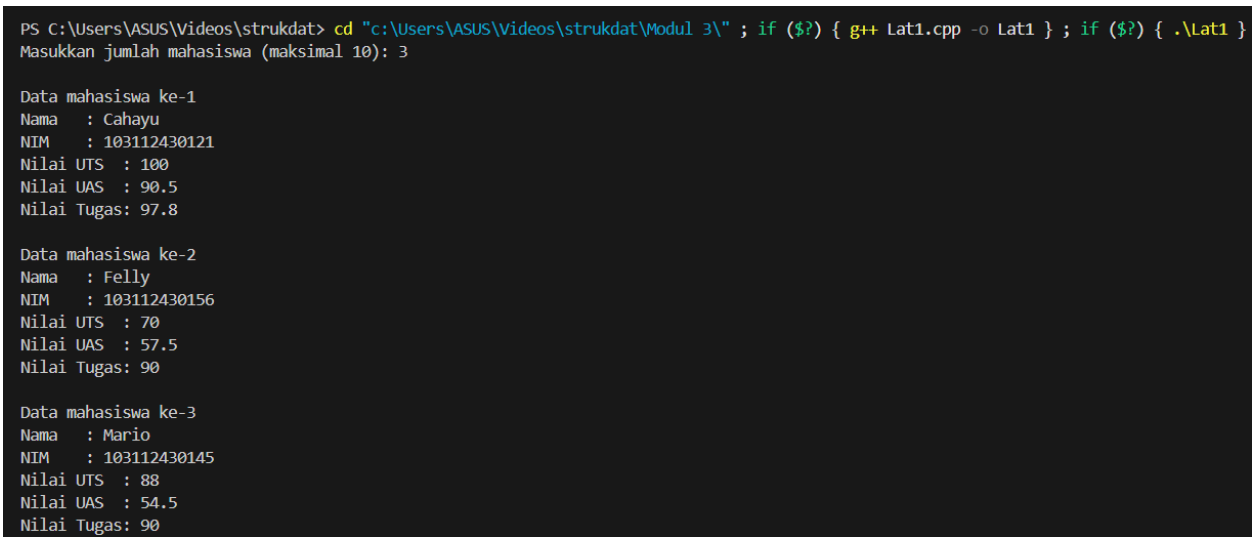
C. Unguided/Tugas (berisi screenshot source code & output program disertai penjelasannya)

Unguided 1

```cpp
// Risky Cahayu
// 103112430121

#include <iostream>
using namespace std;

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

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

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

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

    if (n > 10) {
        cout << "Jumlah mahasiswa melebihi batas!" << endl;
        return 0;
    }

    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;

        // Hitung nilai akhir
```

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

    cout << "\n--- Data 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 << "UTS          : " << mhs[i].uts << endl;
        cout << "UAS          : " << mhs[i].uas << endl;
        cout << "Tugas        : " << mhs[i].tugas << endl;
        cout << "Nilai Akhir  : " << mhs[i].nilaiAkhir << endl;
    }

    return 0;
}
```

Screenshots Output



```
--- Data Mahasiswa ---

Mahasiswa ke-1
Nama         : Cahayu
NIM          : 103112430121
UTS          : 100
UAS          : 90.5
Tugas        : 97.8
Nilai Akhir  : 95.54

Mahasiswa ke-2
Nama         : Felly
NIM          : 103112430156
UTS          : 70
UAS          : 57.5
Tugas        : 90
Nilai Akhir  : 71

Mahasiswa ke-3
Nama         : Mario
NIM          : 103112430145
UTS          : 88
UAS          : 54.5
Tugas        : 90
Nilai Akhir  : 75.2
PS C:\Users\ASUS\Videos\strukdat\Modul 3>
```

Deskripsi:

Program di atas merupakan implementasi sederhana dalam bahasa C++ yang menggunakan konsep Abstract Data Type (ADT) melalui struktur struct bernama Mahasiswa. Program berfungsi untuk menginput dan menghitung nilai akhir beberapa mahasiswa berdasarkan nilai UTS, UAS, dan tugas dengan bobot masing-masing 30%, 40%, dan 30%. Fungsi hitungNilaiAkhir() digunakan untuk melakukan perhitungan nilai akhir tiap mahasiswa, sementara data mahasiswa disimpan dalam array mhs yang mampu menampung hingga sepuluh entri. Setelah seluruh data dimasukkan, program akan menampilkan kembali informasi setiap mahasiswa beserta nilai akhirnya. Dengan penggunaan struct dan fungsi terpisah, program ini mencerminkan prinsip ADT yaitu pemisahan antara data (atribut mahasiswa) dan operasi yang dilakukan terhadap data (perhitungan nilai akhir) sehingga kode menjadi lebih terstruktur, mudah dipahami, dan efisien.

Unguided 2

File main.cpp

```cpp
// Risky Cahayu
// 103112430121

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

int main() {
    string namatkul = "Struktur Data";
    string kodematkul = "STD";

    // Membuat objek pelajaran
```

```
    pelajaran pel = create_pelajaran(namatkul, kodematkul);

    // Menampilkan hasil
    tampil_pelajaran(pel);

    return 0;
}
```

File pelajaran.cpp

```
// Risky Cahayu
// 103112430121

#include <iostream>
#include "pelajaran.h"

// Membuat data pelajaran
pelajaran create_pelajaran(string namatkul, string kodematkul) {
    pelajaran p;
    p.namaMatkul = namatkul;
    p.kodeMatkul = kodematkul;
    return p;
}

// Menampilkan data pelajaran
void tampil_pelajaran(pelajaran pel) {
    cout << "nama pelajaran : " << pel.namaMatkul << endl;
    cout << "nilai : " << pel.kodeMatkul << endl;
}
```

File pelajaran.h

```
// Risky Cahayu
// 103112430121

#ifndef PELAJARAN_H
#define PELAJARAN_H

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

// Tipe data pelajaran
struct pelajaran {
    string namaMatkul;
    string kodeMatkul;
};

// Fungsi dan prosedur yang akan diimplementasikan di pelajaran.cpp
```

```
pelajaran create_pelajaran(string namatkul, string kodematkul);
void tampil_pelajaran(pelajaran pel);

#endif
```

Screenshots Output

```
PS C:\Users\ASUS\Videos\pelajaran> g++ main.cpp pelajaran.cpp -o main
>> .\main
nama pelajaran : Struktur Data
nilai : STD
```

Deskripsi:

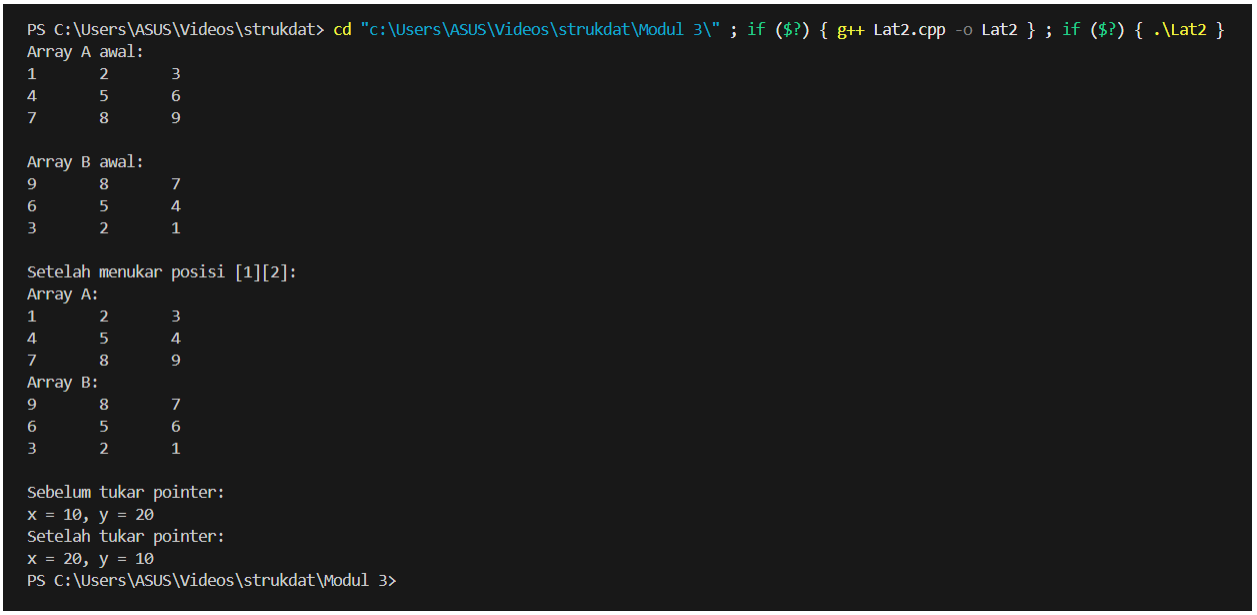
Program di atas merupakan implementasi konsep Abstract Data Type (ADT) dalam bahasa C++ dengan menggunakan tiga bagian utama, yaitu main.cpp, pelajaran.cpp, dan pelajaran.h. File pelajaran.h berfungsi sebagai header file yang mendefinisikan struktur data pelajaran dengan dua atribut, yaitu namaMatkul dan kodeMatkul, serta mendeklarasikan dua fungsi utama, create_pelajaran() dan tampil_pelajaran(). File pelajaran.cpp berisi implementasi fungsi tersebut, di mana create_pelajaran() digunakan untuk membuat objek pelajaran dengan nilai atribut yang diberikan, sedangkan tampil_pelajaran() menampilkan data mata kuliah ke layar. Sementara itu, file main.cpp berperan sebagai program utama yang membuat objek pelajaran berdasarkan input variabel string, lalu menampilkan hasilnya melalui fungsi yang sudah didefinisikan. Program ini menunjukkan penerapan prinsip modularitas dan enkapsulasi, di mana data dan operasinya dipisahkan secara terstruktur sesuai konsep ADT.

Unguided 3

```
// Risky Cahayu
// 103112430121

#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] << "\t";
        }
        cout << endl;
    }
}

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

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

    cout << "Array A awal:" << endl;
    tampilArray(A);
    cout << "\nArray B awal:" << endl;
    tampilArray(B);
]
    tukarPosisi(A, B, 1, 2);

    cout << "\nSetelah menukar posisi [1][2]:" << endl;
    cout << "Array A:" << endl;
    tampilArray(A);
    cout << "Array B:" << endl;
    tampilArray(B);

    int x = 10, y = 20;
    int *ptr1 = &x;
    int *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;
}
```

Screenshots Output



Deskripsi:

Program di atas merupakan implementasi dasar penggunaan array dua dimensi dan pointer dalam bahasa C++. Program ini mendefinisikan tiga fungsi utama, yaitu tampilArray() untuk menampilkan isi array 2D berukuran 3x3, tukarPosisi() untuk menukar elemen pada posisi tertentu antara dua array 2D, serta tukarPointer() untuk menukar nilai dari dua variabel menggunakan pointer. Di dalam fungsi main(), terdapat dua array, yaitu A dan B, yang masing-masing berisi nilai berbeda. Program menampilkan isi awal kedua array, kemudian menukar elemen pada posisi [1][2] (baris ke-2 kolom ke-3) antara A dan B, lalu menampilkan hasil setelah pertukaran. Selain itu, program juga memperlihatkan contoh penggunaan pointer dengan menukar nilai dua variabel x dan y melalui fungsi tukarPointer(). Secara keseluruhan, program ini menunjukkan konsep dasar manipulasi data pada array dua dimensi dan penggunaan pointer untuk pertukaran nilai secara efisien dalam C++.

D. Kesimpulan

Program tersebut menunjukkan penerapan Abstract Data Type (ADT) dalam C++ dengan memisahkan antara definisi tipe data, implementasi fungsi, dan program utama ke dalam file yang berbeda. Melalui pendekatan ini, data dan operasi yang berkaitan dikemas dalam satu kesatuan logis yang terstruktur, sehingga program menjadi lebih modular, mudah dipahami, serta mudah dikembangkan. Selain itu, penggunaan header file dan fungsi terpisah memperlihatkan bagaimana enkapsulasi diterapkan untuk menjaga agar detail implementasi tersembunyi dari pengguna, sesuai dengan prinsip dasar ADT dalam pemrograman berorientasi objek.

E. Referensi

Abstract data type — Wikipedia menyediakan definisi ADT sebagai model matematika untuk data types, yang didefinisikan dari sudut pandang pengguna data yaitu nilai-nilai yang mungkin, operasi-operasi yang dapat dilakukan, dan perilaku dari operasi tersebut; berbeda dengan data structure yang adalah representasi konkret data. https://en.wikipedia.org/wiki/Abstract_data_type

Data type — Halaman ini juga menyebutkan kaitan antara tipe data abstrak (ADT) dan struktur data serta menjelaskan bahwa ADT mendefinisikan operasi dan perilaku, bukan implementasi https://en.wikipedia.org/wiki/Data_type