LAPORAN PRAKTIKUM
STRUKTUR DATA

MODUL 3
ABSTRACT DATA TYPE

Disusun Oleh :

NAMA : M.IRFAN ADIB

NIM : 103112400257

Dosen

FAHRUDIN MUKTI WIBOWO

PROGRAM STUDI STRUKTUR DATA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

A. Dasar Teori

Abstract Data Type (ADT) adalah model matematis yang mendefinisikan tipe data secara abstrak berdasarkan perilaku dan operasinya, berfokus pada apa yang dapat dilakukan oleh tipe data tersebut, alih-alih bagaimana cara implementasinya. Sebuah ADT terdiri dari definisi tipe data dan serangkaian operasi dasar (primitif) seperti konstruktor, selektor, mutator, serta operator relasional dan aritmatika. Dalam implementasi, ADT mempromosikan modularitas dengan memisahkan spesifikasi (definisi tipe dan deklarasi fungsi di file header .h) dari realisasi/implementasi fungsi (di file source .c atau .cpp), yang membuat kode menjadi lebih terstruktur, mudah dikelola, dan memungkinkan perubahan implementasi tanpa memengaruhi program utama yang menggunakannya.

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

Guided 1

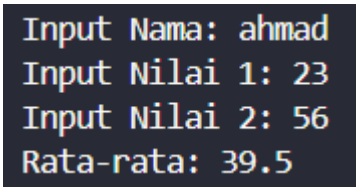
```
#ifndef MAHASISWA_H_INCLUDED
#define MAHASISWA_H_INCLUDED
struct mahasiswa
{
    char nim[10];
    int nilai1, nilai2;
};
void inputMahasiswa(mahasiswa &m);
float rerata(mahasiswa m);
#endif
```

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

void inputMahasiswa(mahasiswa &m)
{
    cout << "Input Nama: ";
    cin >> (m.nim);
    cout << "Input Nilai 1: ";
    cin >> (m.nilai1);
    cout << "Input Nilai 2: ";
    cin >> (m.nilai2);
}

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

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

int main()
{
    mahasiswa mhs;
```

```
    inputMahasiswa(mhs);
    cout << "Rata-rata: " << rerata(mhs) << endl;
    return 0;
}
```

Screenshots Output



Deskripsi:

Kode tersebut mengimplementasikan sebuah Abstract Data Type (ADT) sederhana bernama mahasiswa dalam bahasa C++, yang mewakili data mahasiswa dengan Nomor Induk Mahasiswa (nim) dan dua nilai (nilai1, nilai2). ADT ini didefinisikan dalam header file mahasiswa.h yang memuat spesifikasi struct dan prototipe dua operasi: inputMahasiswa (untuk mengisi data) dan rerata (untuk menghitung rata-rata nilai). Realisasi (implementasi) dari kedua operasi tersebut diletakkan dalam source file terpisah, dan program utama (dalam fungsi main) kemudian menggunakan ADT ini dengan cara mendeklarasikan objek mhs, memanggil operasi inputMahasiswa untuk mengisi datanya, dan menampilkan hasilnya menggunakan operasi rerata.

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

Unguided 1

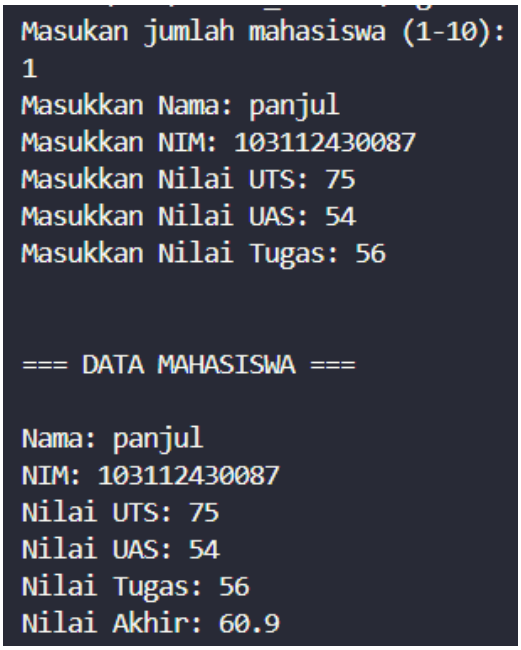
```
#ifndef MAHASISWA_H_INCLUDED
#define MAHASISWA_H_INCLUDED
#include <string>

using namespace std;

struct Mahasiswa
{
    string nama, nim;
    int nilaiUts, nilaiUas, nilaiTugas;
    float nilaiAkhir;
};

void inputMahasiswa(Mahasiswa dataMhs[], int jumlahMhs);
float nilaiAkhir(Mahasiswa mhs);
#endif
```

```
#include "mahasiswa.h"
#include <iostream>
#include <string>

using namespace std;

void inputMahasiswa(Mahasiswa dataMhs[], int jumlahMhs) {
    for (int i = 0; i < jumlahMhs; i++) {
        cout << "Masukkan Nama: ";
        getline(cin, dataMhs[i].nama);
        cout << "Masukkan NIM: ";
        cin >> dataMhs[i].nim;
        cout << "Masukkan Nilai UTS: ";
        cin >> dataMhs[i].nilaiUts;
        cout << "Masukkan Nilai UAS: ";
        cin >> dataMhs[i].nilaiUas;
        cout << "Masukkan Nilai Tugas: ";
        cin >> dataMhs[i].nilaiTugas;
        cout << endl;
```

```
        dataMhs[i].nilaiAkhir = nilaiAkhir(dataMhs[i]);
        cin.ignore();
    }

}

float nilaiAkhir(Mahasiswa dataMhs)
{
    return (0.3 * dataMhs.nilaiUts) + (0.4 * dataMhs.nilaiUas) +
(0.3 * dataMhs.nilaiTugas);
}
```

```
#include "mahasiswa.h"
#include <iostream>

using namespace std;

int main()
{
    Mahasiswa dataMhs[10];
    int jumlahMhs;
    cout << "Masukan jumlah mahasiswa (1-10): " << endl;
    cin >> jumlahMhs;
    if (jumlahMhs < 1 || jumlahMhs > 10) {
        cout << "Jumlah mahasiswa harus antara 1 sampai 10." <<
endl;
        return 1;
    }

    cin.ignore();
    inputMahasiswa(dataMhs, jumlahMhs);
    cout << "\n=== DATA MAHASISWA ===\n";
    for (int i = 0; i < jumlahMhs; i++) {
        cout << "\nNama: " << dataMhs[i].nama << endl;
        cout << "NIM: " << dataMhs[i].nim << endl;
        cout << "Nilai UTS: " << dataMhs[i].nilaiUts << endl;
        cout << "Nilai UAS: " << dataMhs[i].nilaiUas << endl;
        cout << "Nilai Tugas: " << dataMhs[i].nilaiTugas << endl;
        cout << "Nilai Akhir: " << dataMhs[i].nilaiAkhir << endl;
    }
}
```

Screenshot Output



Deskripsi:

Program C++ ini mengelola data mahasiswa sebagai sebuah Abstract Data Type (ADT) bernama Mahasiswa, yang definisinya (spesifikasi struktur data dan prototipe fungsi) dipisahkan dalam header file. ADT ini menyimpan nama, NIM, nilai UTS, UAS, dan Tugas. Program utama meminta pengguna memasukkan jumlah mahasiswa (maksimum 10), kemudian memanggil fungsi inputMahasiswa untuk mengumpulkan data lengkap untuk setiap mahasiswa. Selama proses input, program menggunakan fungsi nilaiAkhir untuk menghitung dan menyimpan rata-rata nilai dengan bobot 30% UTS, 40% UAS, dan 30% Tugas, yang mengilustrasikan prinsip modularitas ADT, sebelum akhirnya menampilkan semua data mahasiswa yang telah diolah.

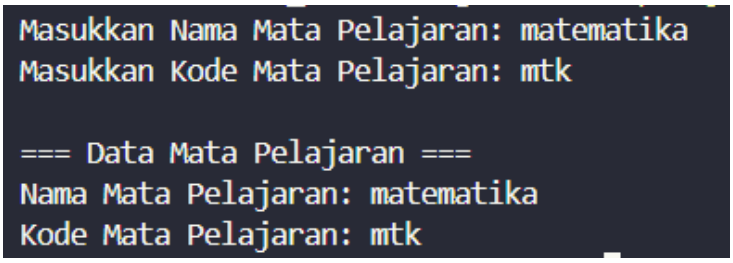
Unguided 2

```
#ifndef PELAJARAN_H_INCLUDED
#define PELAJARAN_H_INCLUDED
#include <string>

using namespace std;

struct Pelajaran {
    string namaMapel, kodeMapel;
};

Pelajaran tambahPelajaran(Pelajaran inputPelajaran);
void tampilkanPelajaran(Pelajaran inputPelajaran);
#endif
```

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

using namespace std;

Pelajaran tambahPelajaran(Pelajaran inputPelajaran) {
    cout << "Masukkan Nama Mata Pelajaran: ";
    getline(cin, inputPelajaran.namaMapel);
    cout << "Masukkan Kode Mata Pelajaran: ";
    getline(cin, inputPelajaran.kodeMapel);
    return inputPelajaran;
}

void tampilkanPelajaran(Pelajaran inputPelajaran) {
    cout << endl;
    cout << "=== Data Mata Pelajaran ===" << endl;
    cout << "Nama Mata Pelajaran: " << inputPelajaran.namaMapel <<
endl;
    cout << "Kode Mata Pelajaran: " << inputPelajaran.kodeMapel <<
endl;
}
```

```cpp
#include <iostream>
#include <string>
#include "pelajaran.h"

using namespace std;

int main() {
    Pelajaran pelajaran;
    Pelajaran pel = tambahPelajaran(pelajaran);
    tampilkanPelajaran(pel);
    return 0;
}
```

Screenshot Output



Deskripsi:

Program C++ ini mengimplementasikan Abstract Data Type (ADT) sederhana bernama Pelajaran, yang spesifikasinya didefinisikan dalam header file (pelajaran.h) berisi struktur Pelajaran (untuk menyimpan namaMapel dan kodeMapel) serta deklarasi fungsi operasi. Implementasi operasionalnya, yaitu fungsi tambahPelajaran (untuk menerima input nama dan kode mata pelajaran) dan tampilkanPelajaran (untuk mencetak data), berada di source file terpisah. Pada fungsi main, program mendeklarasikan objek Pelajaran, memanggil tambahPelajaran untuk mengisi datanya, dan kemudian memanggil tampilkanPelajaran untuk menyajikan hasilnya, menunjukkan pemisahan yang jelas antara antarmuka dan implementasi dalam pendekatan ADT.

Unguided 3

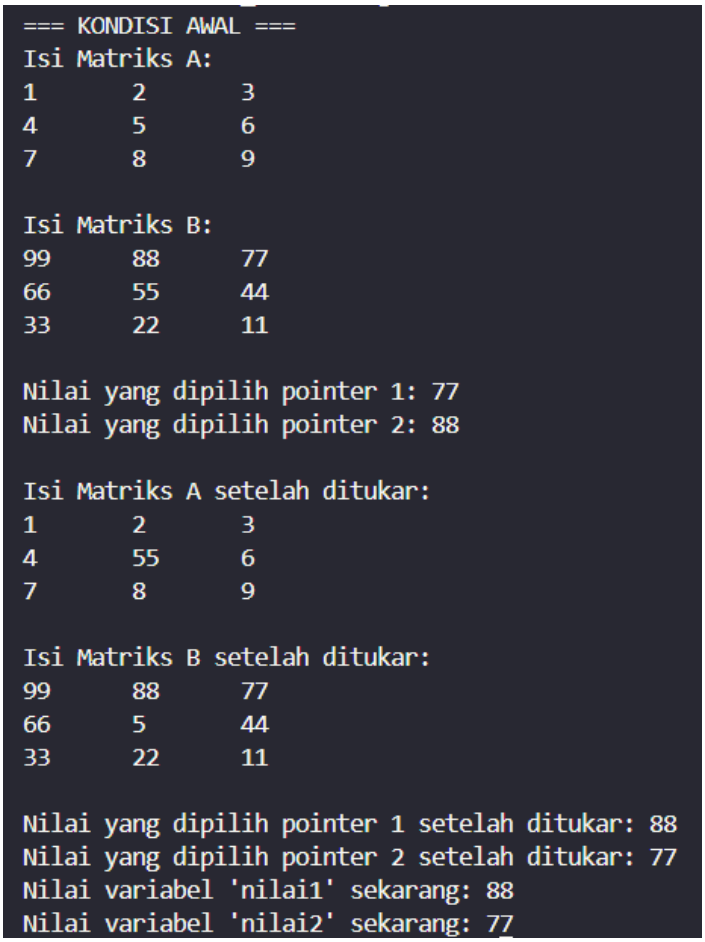
```h
#ifndef SWITCHDATA_H_INCLUDED
#define SWITCHDATA_H_INCLUDED
#define MAX 3

void tampilkanArray(int arr[MAX][MAX]);
void tukarElemenArray(int arr1[MAX][MAX], int arr2[MAX][MAX], int
baris, int kolom);
void tukarNilaiPointer(int* pointer1, int* pointer2);
#endif
```

```cpp
#include <iostream>
#include "switchData.h"

using namespace std;

void tampilkanArray(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 tukarElemenArray(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 tukarNilaiPointer(int* ptr1, int* ptr2) {
    int temp = *ptr1;
    *ptr1 = *ptr2;
    *ptr2 = temp;
}
```

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

int main() {
    int matriksA[3][3] = {
        {1, 2, 3},
        {4, 5, 6},
        {7, 8, 9}
    };

    int matriksB[3][3] = {
        {99, 88, 77},
        {66, 55, 44},
        {33, 22, 11}
    };

    int nilai1 = 77;
    int nilai2 = 88;
    int* pointer1 = &nilai1;
    int* pointer2 = &nilai2;

    cout << "=== KONDISI AWAL ===" << endl;
    cout << "Isi Matriks A:" << endl;
    tampilkanArray(matriksA);
    cout << "\nIsi Matriks B:" << endl;
    tampilkanArray(matriksB);

    cout << "\nNilai yang dipilih pointer 1: " << *pointer1 <<
endl;
    cout << "Nilai yang dipilih pointer 2: " << *pointer2 << endl;

    tukarElemenArray(matriksA, matriksB, 1, 1);

    cout << "\nIsi Matriks A setelah ditukar:" << endl;
    tampilkanArray(matriksA);
    cout << "\nIsi Matriks B setelah ditukar:" << endl;
    tampilkanArray(matriksB);

    tukarNilaiPointer(pointer1, pointer2);
```

```
    cout << "\nNilai yang dipilih pointer 1 setelah ditukar: " <<
*pointer1 << endl;
    cout << "Nilai yang dipilih pointer 2 setelah ditukar: " <<
*pointer2 << endl;
    cout << "Nilai variabel 'nilai1' sekarang: " << nilai1 <<
endl;
    cout << "Nilai variabel 'nilai2' sekarang: " << nilai2 <<
endl;
    return 0;
}
```

Screenshots Output



Deskripsi:

Program C++ ini mengimplementasikan serangkaian fungsi untuk manipulasi dan pertukaran data, didefinisikan sebagai modul ADT yang dipisahkan dalam header file (SWITCHDATA_H_INCLUDED). Fungsi utamanya meliputi tampilkanArray untuk mencetak array dua dimensi 3×3, tukarElemenArray untuk menukar satu elemen pada posisi tertentu antara dua array dua dimensi, dan tukarNilaiPointer untuk menukar nilai yang ditunjuk oleh dua pointer integer. Dalam fungsi main, program menginisialisasi dua matriks 3×3 (matriksA dan matriksB) dan dua variabel integer (nilai1, nilai2) yang ditunjuk oleh pointer. Program kemudian mendemonstrasikan kedua metode penukaran: menukar elemen di indeks [1][1] antara matriks A dan B menggunakan tukarElemenArray, dan menukar nilai variabel nilai1 dan nilai2 secara tidak langsung melalui pointer menggunakan tukarNilaiPointer, sekaligus menunjukkan bagaimana fungsi ini memodifikasi nilai asli yang ditunjuk oleh pointer.

D. Kesimpulan

Penggunaan Abstract Data Type (ADT) secara mendasar memodifikasi metodologi pemrograman, menghasilkan kode yang lebih modular dan terstruktur. Prinsip ini tercapai melalui pemisahan yang jelas antara spesifikasi (seperti definisi struktur data dan prototipe fungsi, biasanya ditempatkan di file header .h) dan implementasi kode yang sebenarnya (file sumber .cpp). Pemisahan ini sangat meningkatkan kemudahan pengelolaan dan pemahaman program. Studi praktik ini menggarisbawahi bahwa ADT, baik untuk data sederhana maupun struktur kompleks (misalnya, data mahasiswa atau mata pelajaran), memungkinkan program utama (atau driver) untuk berinteraksi dengan data hanya melalui serangkaian operasi yang terdefinisi dengan baik, tanpa perlu memahami detail internal mengenai bagaimana operasi tersebut bekerja.

E. Referensi

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). *Introduction to Algorithms* (4th ed.). The MIT Press.

Stroustrup, B. (2014). *Programming: Principles and Practice Using C++* (2nd ed.). Addison-Wesley Professional.