

**LAPORAN PRAKTIKUM STRUKTUR
DATA**

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



Disusun Oleh :

Nama : Jundi Amru Abbas Difaullah
NIM : 103112400143

Dosen :

FAHRUDIN MUKTI WIBOWO

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

A. Dasar

Dalam bidang **ilmu komputer**, *Abstract Data Type* (ADT) atau **tipe data abstrak** merupakan model matematis yang mendeskripsikan tipe data berdasarkan **perilaku dan operasi yang dapat dilakukan terhadapnya**, bukan dari cara implementasinya. Dengan kata lain, ADT menekankan pada *apa yang dilakukan* suatu struktur data, bukan *bagaimana ia bekerja di dalamnya* (Goodrich, Tamassia, & Mount, 2011).

Konsep ADT erat kaitannya dengan **abstraksi** dan **modularitas**. Abstraksi memungkinkan pengembang untuk menyembunyikan detail teknis dari pengguna dan hanya menampilkan fungsi-fungsi dasar yang dibutuhkan (Dale & Weems, 2017). Sementara itu, modularitas berperan dalam memisahkan program menjadi beberapa bagian yang saling terhubung, seperti file header (.h) untuk deklarasi, file implementasi (.cpp) untuk isi kode, serta file utama (main.cpp) untuk pengujian sistem (Pressman & Maxim, 2019).

Selain itu, penerapan ADT juga melibatkan **manajemen memori** menggunakan **pointer**, yang memungkinkan akses langsung ke lokasi data di memori. Pendekatan ini efisien terutama dalam pengolahan data berformat **array dua dimensi** atau struktur data kompleks lainnya (Weiss, 2014).

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

Guided 1

Code *mahasiswa.h*

```
#ifndef MAHASISWA_H_INCLUDED
#define MAHASISWA_H_INCLUDED
struct mahasiswa
{
    char nim[10];
    int nilai1, nilai2;
};
void inoutMhs(mahasiswa &m);
float rata2(mahasiswa m);
#endif
```

Code *mahasiswa.cpp*

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

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

```

        cout << "input nilai 1 = ";
        cin >> (m) .nilai1;
        cout << "input nilai 2 = ";
        cin >> (m) .nilai2;
    }
    float rata2(mahasiswa m)
    {
        return float(m.nilai1 + m.nilai2) / 2;
    }

```

Code *main.cpp*

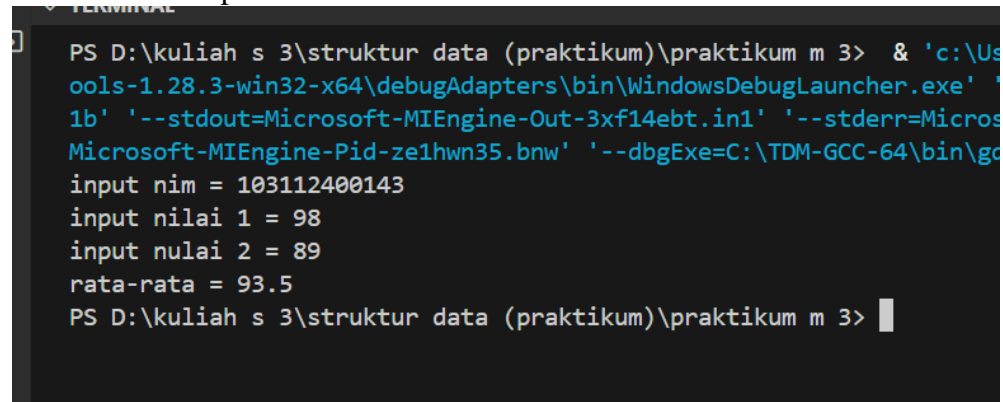
```

#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:



```

PS D:\kuliahs3\struktur data (praktikum)\praktikum m 3> & 'c:\Us
ools-1.28.3-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '
1b' '--stdout=Microsoft-MIEngine-Out-3xf14ebt.in1' '--stderr=Micros
Microsoft-MIEngine-Pid-ze1hwn35.bnw' '--dbgExe=C:\TDM-GCC-64\bin\gc
input nim = 103112400143
input nilai 1 = 98
input nilai 2 = 89
rata-rata = 93.5
PS D:\kuliahs3\struktur data (praktikum)\praktikum m 3>

```

Deskripsi:

Program ini dibuat untuk menyimpan data berupa nama, nilai pertama, dan nilai kedua, lalu menghitung rata-rata dari kedua nilai tersebut. Saat dijalankan, pengguna akan diminta memasukkan nama beserta dua nilai. Setelah itu, program yang memakai konsep ADT (Abstract Data Type) ini akan menampilkan kembali semua data yang sudah dimasukkan, lengkap dengan hasil rata-ratanya.

- C. Unguided/Tugas (berisi screenshot source code & output program disertai penjelasannya)
- Unguided 1
Code *DataMhs.h*

```
#ifndef DATAMHS_H_INCLUDED
#define DATAMHS_H_INCLUDED

#include <iostream>
#include <string>
#include <array>

using namespace std;

struct DataMhs {
    string nama;
    string nim;
    float nilaiUTS;
    float nilaiUAS;
    float nilaiTugas;
    float rataRata;
};

const int JUMLAH_MAKS = 10;
typedef array<DataMhs, JUMLAH_MAKS> KumpulanMhs;

void isiData(DataMhs &mhs);
void hitungRata(DataMhs &mhs);
void tampilData(const DataMhs &mhs);
void inputSemua(KumpulanMhs &listMhs, int &jumlah);
void tampilSemua(const KumpulanMhs &listMhs, int jumlah);

#endif
```

Code *DataMhs.cpp*

```
#include "dataMhs.h"

void isiData(DataMhs &mhs) {
    cout << "Masukkan NIM          : ";
    cin >> mhs.nim;
    cin.ignore();
    cout << "Masukkan Nama          : ";
    getline(cin, mhs.nama);
}
```

```

        cout << "Masukkan Nilai UTS    : ";
        cin >> mhs.nilaiUTS;
        cout << "Masukkan Nilai UAS    : ";
        cin >> mhs.nilaiUAS;
        cout << "Masukkan Nilai Tugas : ";
        cin >> mhs.nilaiTugas;

        hitungRata(mhs);
    }
    void hitungRata(DataMhs &mhs) {
        mhs.rataRata = (mhs.nilaiUTS * 0.3f) + (mhs.nilaiUAS * 0.4f) +
(mhs.nilaiTugas * 0.3f);
    }
    void tampilData(const DataMhs &mhs) {
        cout << "-----\n";
        cout << "NIM            : " << mhs.nim << endl;
        cout << "Nama            : " << mhs.nama << endl;
        cout << "Nilai UTS       : " << mhs.nilaiUTS << endl;
        cout << "Nilai UAS       : " << mhs.nilaiUAS << endl;
        cout << "Nilai Tugas    : " << mhs.nilaiTugas << endl;
        cout << "Nilai Akhir    : " << mhs.rataRata << endl;
    }
}

```

Code *main.cpp*

```

#include "dataMhs.h"

void inputSemua(KumpulanMhs &listMhs, int &jumlah) {
    do {
        cout << "Berapa jumlah mahasiswa (maks " << JUMLAH_MAKS << "): ";
        cin >> jumlah;
    } while (jumlah <= 0 || jumlah > JUMLAH_MAKS);

    for (int i = 0; i < jumlah; ++i) {
        cout << "\n=== Input Data Mahasiswa ke-" << i + 1 << " ===" << endl;
        isiData(listMhs[i]);
    }
}

void tampilSemua(const KumpulanMhs &listMhs, int jumlah) {
    cout << "\n=== Data Mahasiswa yang Telah Dientri ===" << endl;
    for (int i = 0; i < jumlah; ++i) {
        cout << "\nMahasiswa ke-" << i + 1 << ":" << endl;
        tampilData(listMhs[i]);
    }
}

```

```

}
int main() {
    KumpulanMhs daftarMhs;
    int totalMhs = 0;

    inputSemua(daftarMhs, totalMhs);
    tampilSemua(daftarMhs, totalMhs);

    return 0;
}

```

Screenshots Output:

```

PS D:\kuliah s 3\struktur data (praktikum)\laprak m3> g++ main.cpp dataMhs.cpp -o programMhs
PS D:\kuliah s 3\struktur data (praktikum)\laprak m3> ./programMhs
Berapa jumlah mahasiswa (maks 10): 3

=== Input Data Mahasiswa ke-1 ===
Masukkan NIM      : 10311240143
Masukkan Nama     : Jundi
Masukkan Nilai UTS : 99
Masukkan Nilai UAS : 97
Masukkan Nilai Tugas : 100

=== Input Data Mahasiswa ke-2 ===
Masukkan NIM      : 103112400142
Masukkan Nama     : Devi
Masukkan Nilai UTS : 97
Masukkan Nilai UAS : 90
Masukkan Nilai Tugas : 87

=== Input Data Mahasiswa ke-3 ===
Masukkan NIM      : 103112400144
Masukkan Nama     : Wayah
Masukkan Nilai UTS : 89
Masukkan Nilai UAS : 96
Masukkan Nilai Tugas : 88

=== Data Mahasiswa yang Telah Dientri ===

Mahasiswa ke-1:
-----
NIM      : 10311240143
Nama     : Jundi
Nilai UTS : 99
Nilai UAS : 97
Nilai Tugas : 100
Nilai Akhir : 98.5

Mahasiswa ke-2:
-----
NIM      : 103112400142
Nama     : Devi
Nilai UTS : 97
Nilai UAS : 90
Nilai Tugas : 87
Nilai Akhir : 91.2

Mahasiswa ke-3:
-----
NIM      : 103112400144
Nama     : Wayah
Nilai UTS : 89
Nilai UAS : 96
Nilai Tugas : 88
Nilai Akhir : 91.5
PS D:\kuliah s 3\struktur data (praktikum)\laprak m3>

```

Deskripsi :

Program ini dibuat buat nyimpen dan ngolah data nilai mahasiswa. Jadi nanti pengguna bakal diminta masukin NIM, nama, sama nilai UTS, UAS, dan tugas. Setelah semuanya diisi, program bakal nyimpen data itu terus ngitung nilai akhirnya pakai rumus $(0.3 * \text{nilaiUTS}) + (0.4 * \text{nilaiUAS}) + (0.3 * \text{nilaiTugas})$. Hasil akhirnya, semua data mahasiswa bakal ditampilin lengkap bareng nilai akhir yang udah dihitung otomatis sama programnya.

- Unguided 2
Code *pelajaran.h*

```
#ifndef PELAJARAN_H
#define PELAJARAN_H

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

struct Pelajaran {
    string namaMapel;
    string kodeMapel;
};

Pelajaran buatPelajaran(string nama, string kode);
void tampilPelajaran(Pelajaran p);

#endif
```

Code *prlajaran.cpp*

```
#include "pelajaran.h"

Pelajaran buatPelajaran(string nama, string kode) {
    Pelajaran p;
    p.namaMapel = nama;
    p.kodeMapel = kode;
    return p;
}

void tampilPelajaran(Pelajaran p) {
    cout << "Matkul : " << p.namaMapel << endl;
    cout << "Kode matkul : " << p.kodeMapel << endl;
}
```

Code *main.cpp*

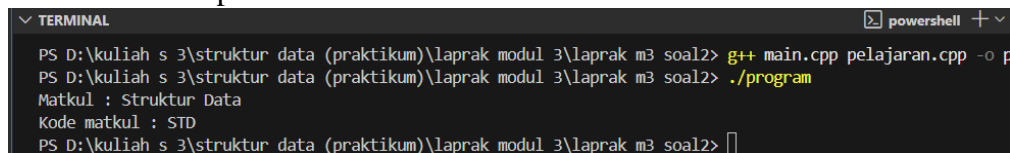
```
#include <iostream>
#include "pelajaran.h"
using namespace std;

int main() {
    string namaMapel = "Struktur Data";
    string kodeMapel = "STD";

    Pelajaran p = buatPelajaran(namaMapel, kodeMapel);
    tampilPelajaran(p);

    return 0;
}
```

Screenshots Output:



```
PS D:\kuliah s 3\struktur data (praktikum)\laprak modul 3\laprak m3 soal2> g++ main.cpp pelajaran.cpp -o p
PS D:\kuliah s 3\struktur data (praktikum)\laprak modul 3\laprak m3 soal2> ./program
Matkul : Struktur Data
Kode matkul : STD
PS D:\kuliah s 3\struktur data (praktikum)\laprak modul 3\laprak m3 soal2> █
```

Deskripsi :

Program ini dibuat pakai konsep ADT (Abstract Data Type) buat nampilin data mata pelajaran. Jadi, program bakal nyimpen nama pelajaran sama kode pelajarannya, terus pas dijalankan, dua data itu langsung ditampilin di layar. Programnya simpel tapi nunjukin gimana cara kerja ADT buat ngatur data biar lebih rapi dan terstruktur.

- Unguided 3
Code *tukmat.h*

```
#ifndef TUKMAT_H
#define TUKMAT_H

#include <iostream>
using namespace std;
const int BARIS = 3;
const int KOLOM = 3;

void tampilMatriks(int matriks[BARIS][KOLOM], string nama);
void tukarElemen(int matriksA[BARIS][KOLOM], int matriksB[BARIS][KOLOM], int
baris, int kolom);
```



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

#endif
```

Code *tukmat.cpp*

```
#include "tukmat.h"

void tampilMatriks(int matriks[BARIS][KOLOM], string nama) {
    cout << "Isi Matriks " << nama << ":" << endl;
    for (int i = 0; i < BARIS; i++) {
        for (int j = 0; j < KOLOM; j++) {
            cout << matriks[i][j] << "\t";
        }
        cout << endl;
    }
}

void tukarElemen(int matriksA[BARIS][KOLOM], int matriksB[BARIS][KOLOM], int
baris, int kolom) {
    int sementara = matriksA[baris][kolom];
    matriksA[baris][kolom] = matriksB[baris][kolom];
    matriksB[baris][kolom] = sementara;
    cout << "Elemen di baris " << baris << " kolom " << kolom << " sudah
ditukar!" << endl;
}

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

Code *main.cpp*

```
#include "tukmat.h"

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

    int matriksB[BARIS][KOLOM] = {
```

```

        {70, 30, 60},
        {20, 50, 90},
        {40, 80, 10}
    };
    int nilaiX = 199;
    int nilaiY = 123;

    int *ptrX = &nilaiX;
    int *ptrY = &nilaiY;

    cout << "=== Sebelum Ditukar ===" << endl;
    tampilMatriks(matriksA, "A");
    tampilMatriks(matriksB, "B");
    cout << "Nilai X: " << nilaiX << endl;
    cout << "Nilai Y: " << nilaiY << endl;

    tukarElemen(matriksA, matriksB, 1, 1);
    tukarPointer(ptrX, ptrY);

    cout << "\n=== Setelah Ditukar ===" << endl;
    tampilMatriks(matriksA, "A");
    tampilMatriks(matriksB, "B");
    cout << "Nilai X sekarang: " << nilaiX << endl;
    cout << "Nilai Y sekarang: " << nilaiY << endl;

    return 0;
}

```

Screenshots Output :

```

PS D:\kuliah s 3\struktur data (praktikum)\laprak modul 3\laprak m3 soal3> g++ main.cpp tukmat.cpp -o program
PS D:\kuliah s 3\struktur data (praktikum)\laprak modul 3\laprak m3 soal3> ./program
=== Sebelum Ditukar ===
Isi Matriks A:
5      3      4
8      2      9
7      1      6
Isi Matriks B:
70     30     60
20     50     90
40     80     10
Nilai X: 199
Nilai Y: 123
Elemen di baris 1 kolom 1 sudah ditukar!

=== Setelah Ditukar ===
Isi Matriks A:
5      3      4
8      50     9
7      1      6
Isi Matriks B:
70     30     60
20     2      90
40     80     10
Nilai X sekarang: 123
Nilai Y sekarang: 199
PS D:\kuliah s 3\struktur data (praktikum)\laprak modul 3\laprak m3 soal3>

```

Deskripsi :

Program ini dibuat untuk latihan penggunaan array 2 dimensi dan pointer. Program menampilkan dua matriks 3x3 dan menukar satu elemen dari kedua matriks serta menukar nilai dua variabel menggunakan pointer. Semua perubahan ditampilkan sebelum dan sesudah penukaran agar hasilnya langsung terlihat.

D. Kesimpulan

Dari Latihan praktikum dan laprak ini, bisa disimpulkan bahwa latihan membuat program Mahasiswa, ADT Pelajaran, hingga Array 2D dan Pointer sangat membantu untuk memahami cara menulis kode yang rapi dan terstruktur. Dengan memisahkan definisi data di file .h dan implementasi fungsi di file .cpp, program jadi lebih modular dan mudah diperbaiki. Selain itu, penggunaan pointer mengajarkan cara komputer mengelola alamat memori, sementara array dan matriks mempermudah penyimpanan serta manipulasi banyak data sekaligus. Secara keseluruhan, praktikum ini membiasakan kita berpikir terstruktur, memahami alur data, dan membuat program yang lebih profesional dan mudah dikembangkan.

E. Referensi

- TutorialPoint. (n.d.). *C++ programming language features*. Diakses 29 Oktober 2025, dari <https://www.tutorialspoint.com/cplusplus-programming-language-features>
- TutorialPoint. (n.d.). *C++ overview*. Diakses 29 Oktober 2025, dari https://www.tutorialspoint.com/cplusplus/cpp_overview.htm
- GeeksforGeeks. (n.d.). *Operators in C++*. Diakses 29 Oktober 2025, dari <https://www.geeksforgeeks.org/cpp/operators-in-cpp/>