

**LAPORAN PRAKTIKUM STRUKTUR
DATA**

**MODUL
ABSTRACT DATA TYPE (ADT)**



Disusun Oleh :

NAMA : Ayu Setyaning Tyas

NIM : 103112430119

Dosen

FAHRUDIN MUKTI WIBOWO

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

A. Dasar Teori

Dalam ilmu komputer, tipe data abstrak (ADT) adalah model matematika untuk tipe data, yang didefinisikan oleh perilakunya (semantik) dari sudut pandang *pengguna* data, khususnya dalam hal nilai yang mungkin, operasi yang mungkin pada data tipe ini, dan perilaku operasi ini. teori Abstract Data Type (ADT) dan konsep modularitas. ADT mengajarkan kita untuk menyembunyikan detail rumit dari pengguna, cukup membuat tipe data sendiri (struct) dan fungsi-fungsi dasarnya, yang dikenal sebagai abstraksi. Proses ini dipermudah oleh modularitas, yaitu membagi kode menjadi tiga *file* terpisah .h untuk daftar perintah, .cpp untuk isi kode, dan main.cpp untuk pengujian. Selain itu, program juga mencakup manajemen memori, yaitu cara kita menggunakan pointer untuk menunjuk langsung ke lokasi data di memori agar kita bisa mengubah nilai data asli secara efisien, terutama saat bekerja dengan data berformat Array Dua Dimensi.

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

Code *mahasiswa.cpp*

```
#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;
}

```

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:\C++\Modul 3> cd "d:\C++\Modul 3\" ; if ($?) { g++ main.cpp -o main } ; if ($?) { .\main }
input nama = ayu
input nilai = 99
input nilai2 = 100
rata-rata = 99.5
PS D:\C++\Modul 3>

```

Deskripsi:

Program ini merupakan program yang digunakan untuk mendata nama, nilai1 dan nilai2, dan rata-rata dari kedua nilai tersebut. User akan di minta menginput nama beserta kedua nilai lalu program yang di jalankan menggunakan ADT tersebut akan meng output semua yang sudah di input namun dengan tambahan nilai rata-rata dari kedua nilai yang di input user.

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

Unguided 1

Code *mahasiswa2.h*

```
#ifndef MAHASISWA_H_INCLUDED
#define MAHASISWA_H_INCLUDED

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

using namespace std;

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

const int MAX_MHS = 10;
typedef array<Mahasiswa, MAX_MHS> ArrMahasiswa;
void InputDataMhs(Mahasiswa &m);
void HitungNilaiAkhir(Mahasiswa &m);
void TampilMhs(Mahasiswa m);
void InputArrayMhs(ArrMahasiswa &arr, int &n);
void TampilArrayMhs(const ArrMahasiswa &arr, int n);

#endif
```

Code *mahasiswa2.cpp*

```
#include "Mahasiswa2.h"
```

```

void InputDataMhs(Mahasiswa &m) {
    cout << "Input NIM: ";
    cin >> m.nim;
    cout << "Input Nama: ";
    cin.ignore();
    getline(cin, m.nama);
    cout << "Input Nilai UTS: ";
    cin >> m.uts;
    cout << "Input Nilai UAS: ";
    cin >> m.uas;
    cout << "Input Nilai Tugas: ";
    cin >> m.tugas;
    HitungNilaiAkhir(m);
}

void HitungNilaiAkhir(Mahasiswa &m) {
    m.nilai_akhir = (0.3 * m.uts) + (0.4 * m.uas) + (0.3 * m.tugas);
}

void TampilMhs(Mahasiswa m) {
    cout << "NIM\t\t: " << m.nim << endl;
    cout << "Nama\t\t: " << m.nama << endl;
    cout << "Nilai UTS\t: " << m.uts << endl;
    cout << "Nilai UAS\t: " << m.uas << endl;
    cout << "Nilai Tugas\t: " << m.tugas << endl;
    cout << "Nilai Akhir\t: " << m.nilai_akhir << endl;
}

```

Code *main.cpp*

```

#include "Mahasiswa2.h"

void InputArrayMhs(ArrMahasiswa &arr, int &n) {
    do {

```

```

        cout << "Masukkan jumlah mahasiswa (max " << MAX_MHS << "): ";
        cin >> n;
    } while (n <= 0 || n > MAX_MHS);

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

void TampilArrayMhs(const ArrMahasiswa &arr, int n) {
    for (int i = 0; i < n; ++i) {
        cout << "\n--- Mahasiswa ke-" << i + 1 << " ---" << endl;
        TampilMhs(arr[i]);
    }
}

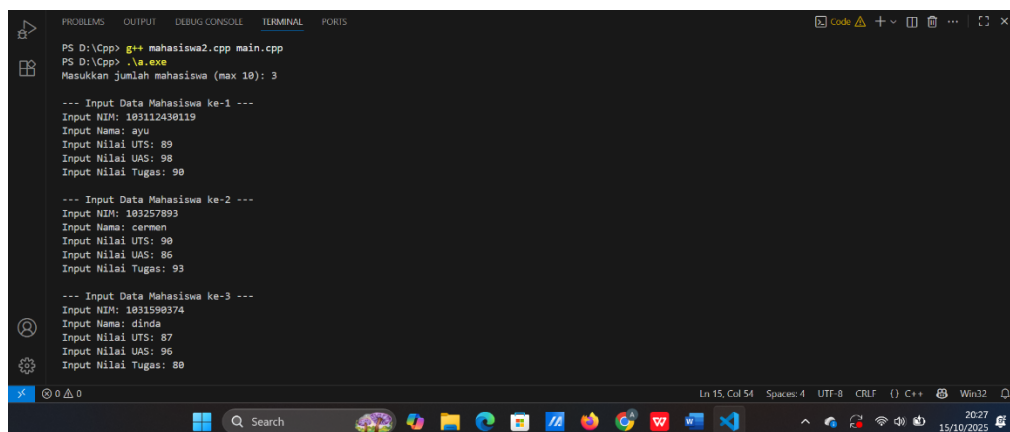
int main() {
    ArrMahasiswa data_mahasiswa;
    int jumlah_mhs = 0;

    InputArrayMhs(data_mahasiswa, jumlah_mhs);
    TampilArrayMhs(data_mahasiswa, jumlah_mhs);

    return 0;
}

```

Screenshots Output



```

PS D:\C++> g++ mahasiswa2.cpp main.cpp
PS D:\C++> .\exe.exe
Masukkan jumlah mahasiswa (max 10): 3

--- Input Data Mahasiswa ke-1 ---
Input NIM: 103112430119
Input Nama: ayu
Input Nilai UTS: 89
Input Nilai UAS: 98
Input Nilai Tugas: 90

--- Input Data Mahasiswa ke-2 ---
Input NIM: 103257893
Input Nama: cermen
Input Nilai UTS: 90
Input Nilai UAS: 86
Input Nilai Tugas: 93

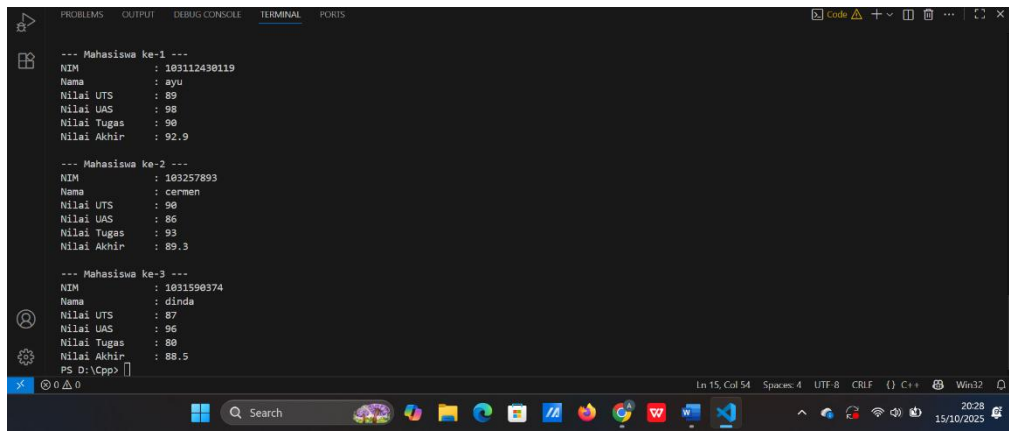
--- Input Data Mahasiswa ke-3 ---
Input NIM: 1031590374
Input Nama: dinda
Input Nilai UTS: 87
Input Nilai UAS: 96
Input Nilai Tugas: 80

```

Deskripsi:

Program ini merupakan program yang digunakan untuk mendata nilai dari mahasiswa. User akan menginput NIM, Nama, nilai UTS, nilai UAS, dan nilai Tugas dari mahasiswa lalu program akan mendata semua inputan itu setelah itu semua nilai akan di hitung menggunakan rumus $(0.3 * m.uts) + (0.4 * m.uas) + (0.3 * m.tugas)$ untuk mendapatkan nilai Akhir.

Unguided 2



```
--- Mahasiswa ke-1 ---
NIM      : 103112430119
Nama     : ayu
Nilai UTS : 89
Nilai UAS : 98
Nilai Tugas : 90
Nilai Akhir : 92.9

--- Mahasiswa ke-2 ---
NIM      : 103257893
Nama     : carmen
Nilai UTS : 90
Nilai UAS : 86
Nilai Tugas : 93
Nilai Akhir : 89.3

--- Mahasiswa ke-3 ---
NIM      : 1031590374
Nama     : dinda
Nilai UTS : 87
Nilai UAS : 96
Nilai Tugas : 80
Nilai Akhir : 88.5
```

Code *pelajar.h*

```
#ifndef PELAJARAN_H_INCLUDED
#define PELAJARAN_H_INCLUDED

#include <iostream>
#include <string>
using namespace std;
struct pelajar
{
    string namamapel;
    string kodemapel;
};

pelajar create_pelajar(string namapel, string kodepel);

void tampil_pelajaran(pelajar pel);
#endif
```

Code *pelajar.cpp*

```
#include "pelajar.h"

pelajar create_pelajar(string namapel, string kodepel)
```

```

{
    pelajar pel;
    pel.namamapel = namapel;
    pel.kodemapel = kodepel;
    return pel;
}

void tampil_pelajaran(pelajar pel)
{
    cout << "Nama Pelajaran: " << pel.namamapel << endl;
    cout << "Nilai: " << pel.kodemapel << endl;
}

```

Code *main.cpp*

```

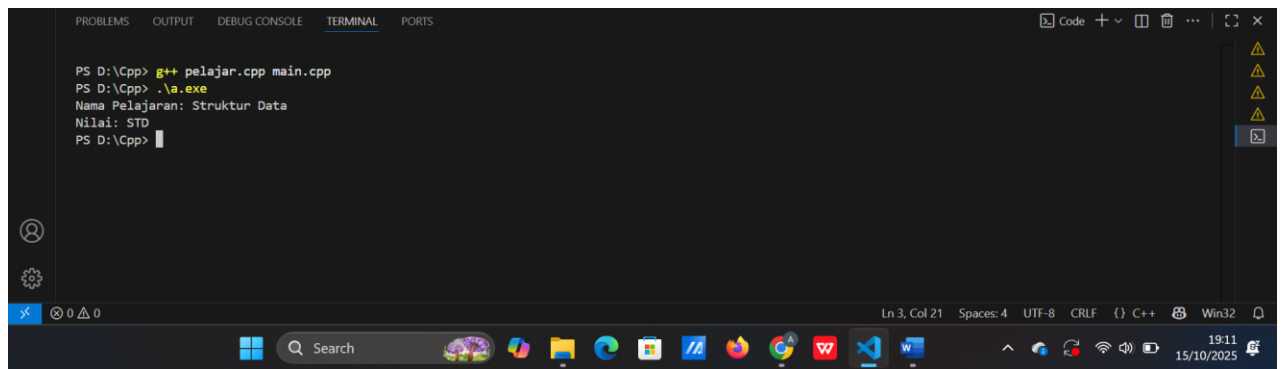
#include <iostream>
#include "pelajar.h"
using namespace std;

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

    pelajar pel = create_pelajar(namapel, kodepel);
    tampil_pelajaran(pel);
    return 0;
}

```

Screenshoot Output



```
PS D:\Cpp> g++ pelajar.cpp main.cpp
PS D:\Cpp> .\a.exe
Nama Pelajaran: Struktur Data
Nilai: STD
PS D:\Cpp>
```

Deskripsi:

Program ini merupakan program yang dibuat menggunakan ADT untuk menampilkan Nama Pelajaran dan kode mata Pelajaran(nilai).

Unguided 3

```
#include <iostream>

#include <string>

using namespace std;

const int JUMLAH_BARIS = 3;
const int JUMLAH_KOLOM = 3;

void TampilkanMatriks(const int Matriks[JUMLAH_BARIS][JUMLAH_KOLOM], const
string& Nama) {
    cout << "Isi Matriks " << Nama << ":" << endl;
    for (int i = 0; i < JUMLAH_BARIS; ++i) {
        for (int j = 0; j < JUMLAH_KOLOM; ++j) {
            cout << Matriks[i][j] << "\t";
        }
        cout << endl;
    }
}

void TukarElemen(int MatriksA[JUMLAH_BARIS][JUMLAH_KOLOM], int
MatriksB[JUMLAH_BARIS][JUMLAH_KOLOM], int Baris, int Kolom) {
    int temp = MatriksA[Baris][Kolom];
```

```

    MatriksA[Baris][Kolom] = MatriksB[Baris][Kolom];

    MatriksB[Baris][Kolom] = temp;

    cout << " (Asik berhasil! Elemen pada Baris ke-" << Baris << " Kolom ke-" << Kolom << "
    ditukar.)" << endl;
}

void TukarNilaiPointer(int *PointerSatu, int *PointerDua) {
    int temp = *PointerSatu;
    *PointerSatu = *PointerDua;
    *PointerDua = temp;
}

int main() {
    int Matriks_A[JUMLAH_BARIS][JUMLAH_KOLOM] = {{5, 3, 4}, {8, 2, 9}, {7, 1, 6}};
    int Matriks_B[JUMLAH_BARIS][JUMLAH_KOLOM] = {{70, 30, 60}, {20, 50, 90}, {40,
80, 10}};

    int Nilai_X = 199;
    int Nilai_Y = 123;
    int *Ptr_Satu = &Nilai_X;
    int *Ptr_Dua = &Nilai_Y;

    cout << "===== sebelum di tukar =====" << endl;
    TampilkanMatriks(Matriks_A, "A");
    cout << endl;
    TampilkanMatriks(Matriks_B, "B");
    cout << endl;

    cout << "Nilai Variabel X: " << Nilai_X << " (Ditunjuk Ptr 1)" << endl;
    cout << "Nilai Variabel Y: " << Nilai_Y << " (Ditunjuk Ptr 2)" << endl;

    cout << "\n===== setelah di tukar =====" << endl;

```

```

TampilkanMatriks(Matriks_A, "A");

cout << endl;

TampilkanMatriks(Matriks_B, "B");

cout << endl;


cout << "Nilai Variabel X sekarang: " << Nilai_X << endl;

cout << "Nilai Variabel Y sekarang: " << Nilai_Y << endl;


return 0;

}

```

Screenshots Output

```

PS D:\Cpp> cd "d:\Cpp\" ; if ($?) { g++ unguide3.cpp -o unguide3 } ; if ($?) { .\unguide3 }
===== sebelum di tukar =====
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 Variabel X: 199 (Ditunjuk Ptr 1)
Nilai Variabel Y: 123 (Ditunjuk Ptr 2)

===== setelah di tukar =====
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 Variabel X sekarang: 123
Nilai Variabel Y sekarang: 199

```

Deskripsi:

Programam ini merupakan program untuk mengimplementasikan array 2D dan pointer yang digunakan untuk menukar nilai variable menggunakan pointer dan mencetak dua buah matriks 2D menggunakan Array.

D. Kesimpulan

Seluruh program C++ yang di atas mulai dari data Mahasiswa, ADT Pelajaran, hingga Array 2D dan Pointer memiliki tujuan utama untuk melatih membuat program yang rapi dan tertata. Semua Program di atas dibuat dalam bentuk pisahan kode menjadi bagian-bagian kecil yang terstruktur (*modular*) dengan membuat "kotak data" khusus (*ADT*) di file .h dan kode kerjanya di file .cpp, yang sangat berguna agar program besar

mudah diperbaiki. Selain itu, juga mempelajari cara komputer mengatur alamat data (*Pointer*) dan mengelola kumpulan data yang banyak (*Array* dan *Matriks*).

E. Referensi

https://en.wikipedia.org/wiki/Abstract_data_type

<https://www.tutorialspoint.com/cplusplus-programming-language-features>

https://www.tutorialspoint.com/cplusplus/cpp_overview.htm

<https://www.geeksforgeeks.org/cpp/operators-in-cpp/>