

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

**MODUL III
ABSTRACT DATA TYPE**



Disusun Oleh :

Nama: Dina Nadhyfa

NIM : 103112430052

Dosen

FAHRUDIN MUKTI WIBOWO

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025**

A. Dasar Teori

Abstract Data Type (ADT) adalah suatu model matematis untuk tipe data yang didefinisikan secara konseptual berdasarkan perilakunya, yang mencakup sekumpulan nilai, operasi yang mungkin pada data tersebut, dan perilaku dari operasi-operasi ini, tanpa menentukan bagaimana data disimpan atau bagaimana operasi diimplementasikan (struktur data konkritnya). Konsep fundamental ini mempromosikan abstraksi data dan penyembunyian informasi (information hiding), memungkinkan pengguna tipe data untuk berinteraksi dengannya hanya melalui antarmuka publik yang disediakan seperti fungsi atau metode, sehingga memisahkan desain logis (apa yang dilakukan) dari implementasi fisik (bagaimana dilakukan). Penerapan ADT, seperti Stack, Queue, atau List, ke dalam sebuah bahasa pemrograman dikenal sebagai struktur data.

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

```
// 103112430052_Dina Nadhyfa
#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;
}
```

```
#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
```

```
#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:\COOLYEAH\SEMESTER 3\Strukdat> cd "d:\COOLYEAH\SEMESTER 3\Strukdat\" ; if ($?) { g++ main.cpp -o main } ; if ($?) { .\main }
input nama = dunni
input nilai = 95
input nilai2 = 90
rata-rata = 92.5
PS D:\COOLYEAH\SEMESTER 3\Strukdat> █
```

Deskripsi:

Program di atas terdiri dari tiga bagian utama yaitu:

1. mahasiswa.h berisi deklarasi struktur dan fungsi
2. mahasiswa.cpp berisi definisi fungsi
3. main.cpp berisi fungsi main() untuk menjalankan program

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

Unguided 1

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

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

// fungsi untuk menghitung nilai akhir
float nilaiakhir(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 (max 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 << "UTS    : ";
        cin >> mhs[i].uts;
        cout << "UAS    : ";
        cin >> mhs[i].uas;
        cout << "Tugas  : ";
        cin >> mhs[i].tugas;

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

    cout << "\n=== Daftar Nilai Mahasiswa ===" << endl;
    for (int i = 0; i < n; i++) {
        cout << mhs[i].nama << " (" << mhs[i].nim << ") = " <<
mhs[i].nilaiAkhir << endl;
    }

    return 0;
}

```

Screenshots Output

```

PS D:\COOLYEAH\SEMESTER 3\Strukdat\Modul 3-soal1> cd "d:\COOLYEAH\SEMESTER 3\Strukdat\Modul 3-soal1\" ; if ($?) { g++ Soal1.cpp -o Soal1 } ; if ($?)
{ .\Soal1 }
Masukkan jumlah mahasiswa (max 10): 2

Data mahasiswa ke-1
Nama   : Dina
NIM    : 103112430052
UTS    : 90
UAS    : 89
Tugas  : 85

Data mahasiswa ke-2
Nama   : Dani
NIM    : 103112430055
UTS    : 90
UAS    : 90
Tugas  : 90

=== Daftar Nilai Mahasiswa ===
Dina (103112430052) = 88.1
Dani (103112430055) = 90
PS D:\COOLYEAH\SEMESTER 3\Strukdat\Modul 3-soal1>

```

Deskripsi:

Program ini digunakan untuk menyimpan data beberapa mahasiswa (maksimal 10 orang) dan menghitung nilai akhir masing-masing mahasiswa berdasarkan nilai UTS, UAS, dan Tugas. Rumus nilai akhir yang digunakan:

$$\text{Nilai Akhir} = 0.3 \times UTS + 0.4 \times UAS + 0.3 \times Tugas$$

Unguided 2

Soal2-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 namapel, string kodepel);
void tampil_pelajaran(pelajaran pel);

#endif
```

Soal2-pelajaran.cpp

```
#include <iostream>
#include "Soal2-pelajaran.h"
using namespace std;

pelajaran create_pelajaran(string namapel, string kodepel) {
    pelajaran p;
    p.namaMapel = namapel;
    p.kodeMapel = kodepel;
    return p;
}

void tampil_pelajaran(pelajaran pel) {
    cout << "nama pelajaran : " << pel.namaMapel << endl;
    cout << "nilai : " << pel.kodeMapel << endl;
}
```

Soal2-main.cpp

```
#include <iostream>
#include "Soal2-pelajaran.h"
using namespace std;

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

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

    return 0;
}
```

```
}
```

Screenshots Output

```
PS D:\COOLYEAH\SEMESTER 3\Strukdat\Modul 3 - soal 2> g++ Soal2-main.cpp Soal2-pelajaran.cpp -o Soal2
PS D:\COOLYEAH\SEMESTER 3\Strukdat\Modul 3 - soal 2> .\Soal2.exe
nama pelajaran : Struktur Data
nilai : STD
PS D:\COOLYEAH\SEMESTER 3\Strukdat\Modul 3 - soal 2> |
```

Deskripsi:

Program ini dibuat untuk menerapkan konsep Abstract Data Type (ADT) pada bahasa C++. Program menyimpan dan menampilkan data pelajaran (mata kuliah), yaitu:

1. Nama Mata Pelajaran
2. Kode Mata Pelajaran

Unguided 3

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

// Fungsi untuk menampilkan isi array 3x3
void tampilArray(int A[3][3]) {
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            cout << A[i][j] << " ";
        }
        cout << endl;
    }
}

// Fungsi untuk menukar elemen array di posisi tertentu
void tukarPosisi(int A[3][3], int B[3][3], int baris, int kolom) {
    int temp = A[baris][kolom];
    A[baris][kolom] = B[baris][kolom];
    B[baris][kolom] = temp;
}

// Fungsi untuk menukar nilai yang ditunjuk dua pointer
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}};
int *p1, *p2;

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

// Tukar elemen array di posisi [1][1] (baris ke-2 kolom ke-2)
tukarPosisi(A, B, 1, 1);

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

// Contoh pointer
int x = 10, y = 20;
p1 = &x;
p2 = &y;

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

tukarPointer(p1, p2);

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

return 0;
}

```

Screenshots Output

```

PS D:\COOLYEAH\SEMESTER 3\Strukdat\Modul 3 - soal 2> cd "d:\COOLYEAH\SEMESTER 3\Strukdat\Modul 3 - soal 3\" ; if ($?) { g++ Soal3.cpp -o Soal3 } ; if ($?) { .
\Soal3 }
Array A:
1 2 3
4 5 6
7 8 9
Array B:
9 8 7
6 5 4
3 2 1

Setelah tukar posisi [1][1]:
Array A:
1 2 3
4 5 6
7 8 9
Array B:
9 8 7
6 5 4
3 2 1

Sebelum tukar pointer:
x = 10, y = 20
Setelah tukar pointer:
x = 20, y = 10
PS D:\COOLYEAH\SEMESTER 3\Strukdat\Modul 3 - soal 3>

```

Deskripsi:

Program ini bertujuan untuk menunjukkan cara kerja array dua dimensi (3x3), memahami penggunaan pointer di C++, menampilkan isi array, menukar isi elemen array pada posisi tertentu, dan menukar isi dua variabel menggunakan pointer.

E. Kesimpulan

Melalui ketiga latihan pada Modul 3.2, dapat disimpulkan bahwa konsep Abstract Data Type (ADT) membantu membuat program menjadi lebih terstruktur, modular, dan mudah dikelola. Pada latihan pertama, mahasiswa belajar menyimpan dan mengolah data dalam bentuk array of struct, serta memanfaatkan fungsi untuk menghitung nilai akhir. Latihan kedua memperkenalkan pembuatan ADT pelajaran dengan pemisahan antara file header, implementasi, dan program utama, sehingga program lebih rapi dan reusable. Sedangkan latihan ketiga memperkuat pemahaman tentang fungsi, array dua dimensi, dan pointer, termasuk cara menukar nilai baik dalam array maupun antar variabel melalui alamat memori. Secara keseluruhan, modul ini melatih kemampuan berpikir terstruktur dalam merancang dan mengimplementasikan tipe data serta operasinya secara efisien di C++.

F. Referensi

<https://share.google/ZYsGA1BZUF70pWuNK>

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