

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

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



Disusun Oleh :
NAMA : KHOIRUL ADDIFA
NIM : 103112400172

Dosen
FAHRUDIN MUKTI WIBOWO

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

A. Dasar Teori

ADT merupakan tipe data yang didefinisikan oleh pengguna, yang memiliki sekumpulan operasi dasar (primitif) untuk memanipulasi datanya tanpa memperlihatkan detail implementasinya. ADT mencakup konsep seperti konstruktor, selector, mutator, dan destruktor untuk membentuk, mengakses, serta mengubah data dengan cara terstruktur. Dalam pemrograman C++, ADT biasanya diimplementasikan melalui pemisahan antara spesifikasi (file header .h) dan realisasi (file implementasi .cpp), sehingga program menjadi modular, mudah dibaca, dan dapat digunakan kembali. Konsep ini melatih mahasiswa memahami bagaimana data dan operasi diorganisasikan secara abstrak sebelum diimplementasikan secara konkret dalam kode. Guided (berisi screenshot source code & output program disertai penjelasannya)

Guided 1

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

// Struktur data mahasiswa
struct Mahasiswa {
    string nama;
    string nim;
    float uts;
    float uas;
    float tugas;
    float nilaiAkhir;
};

// Fungsi untuk menghitung nilai akhir
float hitungNilaiAkhir(float uts, float uas, float tugas) {
    return (0.3 * uts) + (0.4 * uas) + (0.3 * tugas);
}

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

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

    // Validasi jumlah
    if (n > 10) n = 10;

    // Input data mahasiswa
    for (int i = 0; i < n; i++) {
        cout << "\nData mahasiswa ke-" << i + 1 << endl;
        cout << "Nama : ";
    }
}
```

```

        cin.ignore(); // menghapus newline sebelumnya
        getline(cin, mhs[i].nama);
        cout << "NIM    : ";
        getline(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 menggunakan fungsi
        mhs[i].nilaiAkhir = hitungNilaiAkhir(mhs[i].uts, mhs[i].uas,
mhs[i].tugas);
    }

    // Tampilkan hasil
    cout << "\n=====";
    cout << "Daftar Nilai Mahasiswa\n";
    cout << "=====";
    cout << "No  Nama\t\tNIM\t\tNilai Akhir\n";
    cout << "-----\n";
    for (int i = 0; i < n; i++) {
        cout << i + 1 << ". " << mhs[i].nama << "\t\t" << mhs[i].nim
            << "\t\t" << mhs[i].nilaiAkhir << endl;
    }

    return 0;
}

```

Screenshots Output

The screenshot shows a terminal window with the following output:

```

PROBLEMS OUTPUT DEBUG CONSOLE PORTS TERMINAL

PS D:\Tel-U\semester 3\struktur data\laprak3> cd "d:\Tel-U\semester 3\struktur data\laprak3"
; if ($?) { g++ soal1.cpp -o soal1 } ; if ($?) { ./soal1 }
Masukkan jumlah mahasiswa (maks 10): 2

Data mahasiswa ke-1
Nama   : jaka
NIM    : 2401
Nilai UTS : 80
Nilai UAS : 90
Nilai Tugas : 85

Data mahasiswa ke-2
Nama   : joko
NIM    : 2402
Nilai UTS : 70
Nilai UAS : 80
Nilai Tugas : 75

=====
Daftar Nilai Mahasiswa
=====
No  Nama           NIM      Nilai Akhir
-----
1. jaka          2401      85.5
2. joko          2402      75.5
PS D:\Tel-U\semester 3\struktur data\laprak3>

```

Deskripsi:

Program ini bertujuan untuk menyimpan data maksimal 10 mahasiswa menggunakan array struktur yang memiliki field nama, NIM, UTS, UAS, tugas, dan nilai akhir. Nilai akhir dihitung menggunakan fungsi dengan rumus $0.3 \times \text{UTS} + 0.4 \times \text{UAS} + 0.3 \times \text{Tugas}$. Melalui soal ini, mahasiswa belajar mengelola data terstruktur serta menerapkan fungsi untuk melakukan perhitungan otomatis berdasarkan nilai input.

Guided 2

```
Pelajaran.h
#ifndef PELAJARAN_H
#define PELAJARAN_H
#include <string>
using namespace std;

struct pelajaran {
    string namaMapel;
    string kodeMapel;
};

// fungsi untuk membuat data pelajaran baru
pelajaran create_pelajaran(string namaPel, string kodePel);

// prosedur untuk menampilkan data pelajaran
void tampil_pelajaran(pelajaran pel);

#endif

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

// implementasi fungsi create_pelajaran
pelajaran create_pelajaran(string namaPel, string kodePel) {
    pelajaran p;
    p.namaMapel = namaPel;
    p.kodeMapel = kodePel;
    return p;
}

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

main.cpp
#include <iostream>
#include <string>
#include "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

```

PROBLEMS    OUTPUT    DEBUG CONSOLE    PORTS    TERMINAL +
PS D:\Tel-U\semester 3\struktur data\laprak3> g++ main.cpp pelajaran.cpp
PS D:\Tel-U\semester 3\struktur data\laprak3> .\a.exe
nama pelajaran : Struktur Data
nilai : STD
PS D:\Tel-U\semester 3\struktur data\laprak3>

```

Deskripsi:

Soal ini mengajarkan penerapan konsep ADT (Abstract Data Type) dengan membuat tipe data baru bernama pelajaran yang memiliki atribut namaMapel dan kodeMapel. Program dipecah menjadi tiga file, yaitu pelajaran.h untuk deklarasi tipe dan fungsi, pelajaran.cpp untuk implementasi fungsi, dan main.cpp untuk menjalankan program utama. Tujuannya agar mahasiswa memahami cara memisahkan spesifikasi dan implementasi sesuai prinsip modularitas dalam ADT.

Guided 3

```

#include <iostream>
using namespace std;

// Fungsi untuk menampilkan isi array 2D
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;
    }
}

// Fungsi untuk menukar isi 2 array 2D pada posisi tertentu
void tukarArrayPosisi(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;
}

// Fungsi untuk menukar isi variabel yang ditunjuk oleh 2 pointer
integer
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 x = 10, y = 20;
    int *ptr1 = &x;
    int *ptr2 = &y;

    cout << "Array A sebelum ditukar:\n";
    tampilArray(A);
    cout << "\nArray B sebelum ditukar:\n";
    tampilArray(B);

    // Menukar isi array pada posisi tertentu
    tukarArrayPosisi(A, B, 1, 2); // menukar elemen baris ke-1 kolom ke-2

    cout << "\nArray A setelah ditukar pada posisi (1,2):\n";
    tampilArray(A);
    cout << "\nArray B setelah ditukar pada posisi (1,2):\n";
    tampilArray(B);

    // Menukar isi variabel lewat pointer
    cout << "\nNilai sebelum ditukar: x = " << x << ", y = " << y <<
endl;
    tukarPointer(ptr1, ptr2);
    cout << "Nilai setelah ditukar: x = " << x << ", y = " << y << endl;
```

```
    return 0;  
}
```

Screenshots Output

```
PROBLEMS OUTPUT DEBUG CONSOLE PORTS TERMINAL  
PS D:\Tel-U\semester 3\struktur data\laprak3> cd "d:\Tel-U\semester 3\struktur data\laprak3"  
; if ($?) { g++ soal3.cpp -o soal3 } ; if ($?) { ./soal3 }  
Array A sebelum ditukar:  
1 2 3  
4 5 6  
7 8 9  
  
Array B sebelum ditukar:  
9 8 7  
6 5 4  
3 2 1  
  
Array A setelah ditukar pada posisi (1,2):  
1 2 3  
4 5 4  
7 8 9  
  
Array B setelah ditukar pada posisi (1,2):  
9 8 7  
6 5 6  
3 2 1  
  
Nilai sebelum ditukar: x = 10, y = 20  
Nilai setelah ditukar: x = 20, y = 10  
PS D:\Tel-U\semester 3\struktur data\laprak3>
```

Deskripsi:

Soal ini berfokus pada manipulasi array 2D dan pointer dalam C++. Program menggunakan dua array 2D berukuran 3x3 serta dua pointer integer, kemudian menerapkan beberapa fungsi: menampilkan isi array, menukar isi dua array pada posisi tertentu, dan menukar isi variabel yang ditunjuk oleh dua pointer. Latihan ini bertujuan agar mahasiswa memahami penggunaan fungsi, parameter, serta konsep pointer dan operasi pertukaran data dalam struktur dua dimensi.

Kesimpulan

Dalam praktikum modul 3 ini, disimpulkan bahwa konsep **Abstract Data Type (ADT)** dalam C++ digunakan untuk membangun program yang lebih terstruktur, modular, dan mudah dipelihara. Program pertama menunjukkan cara mengelola data mahasiswa menggunakan array dan fungsi untuk menghitung nilai akhir. Program kedua menerapkan konsep ADT dengan memisahkan deklarasi tipe dan fungsi ke dalam file header (.h) serta implementasi ke file terpisah (.cpp). Program ketiga memperlihatkan penggunaan array 2D dan pointer melalui fungsi untuk menukar nilai antar elemen dan variabel. Secara keseluruhan, praktikum ini melatih pemahaman mahasiswa dalam membangun program modular berbasis ADT serta penerapan fungsi dan pointer dalam pengolahan data.

B. Referensi

- Deitel, H. M., & Deitel, P. J. (2014). C++ How to Program (9th Edition). Pearson Education.
- Malik, D. S. (2011). Data Structures Using C++ (2nd Edition). Course Technology, Cengage Learning.