

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

**MODUL 3
ABSTRACT DATA TYPE (ADT)**



Disusun Oleh :

NAMA : LAHRA BUDI SAPUTRA

NIM : 103112430054

Dosen :

FAHRUDIN MUKTI WIBOWO

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

A. Dasar Teori

Abstract Data Type (ADT) dalam Pemrograman Modular

Abstract Data Type (ADT) adalah sebuah konsep fundamental yang memodelkan tipe data secara matematis dengan memisahkan spesifikasi (*apa yang dapat dilakukan*) dari implementasi (*bagaimana melakukannya*). Sebuah ADT didefinisikan sebagai sebuah TYPE dan sekumpulan PRIMITIF (operasi dasar), termasuk Konstruktor, Selector, dan operasi lain seperti aritmatika dan relasional. Prinsip utama ADT adalah modularitas dan penyembunyian informasi (*information hiding*). Dalam pemrograman, hal ini direalisasikan dengan memisahkan definisi TYPE dan deklarasi primitif ke dalam *header file* (.h) (Spesifikasi), dan realisasi kode (implementasi) primitif ke dalam *source file* (.c atau .cpp) (Body). Pemisahan ini memungkinkan program lain menggunakan ADT melalui *interface* yang jelas tanpa perlu mengetahui detail internalnya, sehingga memudahkan pemeliharaan dan modifikasi kode.

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

Guided 1

```
/*Nama Lengkap : Lahra budi Saputra
NIM          : 103112430054
Kelas       : IF 12-06*/
#include "mahasiswa.h"
#include <iostream>
using namespace std;

void inputMhs(mahasiswa &m)
{
    cout << "Input Nama: ";
    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;
}
```

Deskripsi:

File Mahasiswa.cpp menyediakan Realisasi atau Body dari ADT Mahasiswa. Setelah meng-include Mahasiswa.h, file ini berisi implementasi kode program yang sesungguhnya untuk setiap primitif yang dideklarasikan. Di sinilah logika fungsi inputMhs (untuk menerima input dari pengguna) dan fungsi rata2 (untuk menghitung rata-rata nilai) diwujudkan . Tujuan dari file ini adalah untuk menyembunyikan detail bagaimana operasi tersebut dilakukan, mendukung prinsip penyembunyian informasi.

Guided 2

```
/*Nama Lengkap : Lahra budi Saputra  
NIM : 103112430054  
Kelas : IF 12-06*/  
#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
```

Deskripsi:

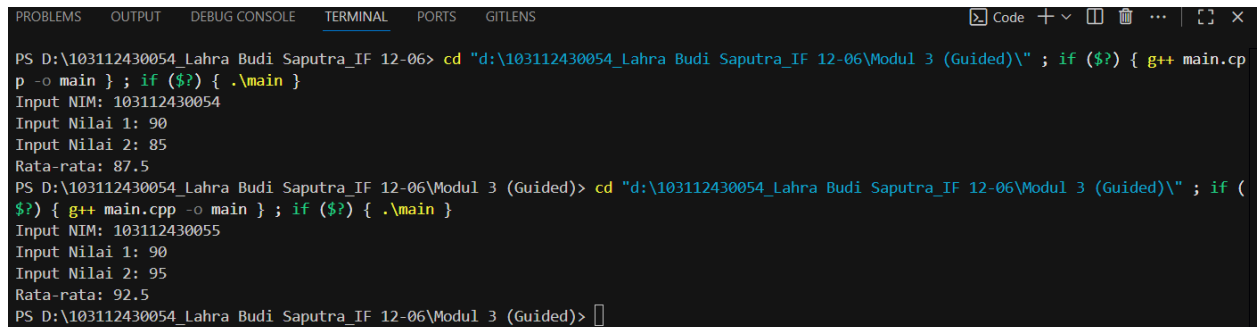
File Mahasiswa.h berfungsi sebagai Spesifikasi atau Interface dari Abstract Data Type (ADT) Mahasiswa. File ini mendefinisikan apa yang dapat dilakukan dengan data. Ia berisi deklarasi struktur data mahasiswa (sebagai TYPE ADT) yang mengemas komponen data (nim, nilai1, nilai2) . Selain itu, ia mendeklarasikan prototipe (atau header) dari fungsi-fungsi primitif ADT, yaitu void inputMhs(mahasiswa &m) dan float rata2(mahasiswa m), yang menentukan operasi yang diizinkan pada data.

Guided 3

```
/*Nama Lengkap : Lahra budi Saputra  
NIM : 103112430054  
Kelas : IF 12-06*/  
#include <iostream>  
#include "mahasiswa.h"  
#include "mahasiswa.cpp"  
using namespace std;  
  
int main()  
{  
    mahasiswa mhs;  
    inputMhs(mhs);  
    cout << "Rata-rata: " << rata2(mhs) << endl;  
    return 0;  
}
```

```
}
```

Screenshots Output



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
PS D:\103112430054_Lahra Budi Saputra_IF 12-06> cd "d:\103112430054_Lahra Budi Saputra_IF 12-06\Modul 3 (Guided)\\" ; if ($?) { g++ main.cpp -o main } ; if ($?) { .\main }
Input NIM: 103112430054
Input Nilai 1: 90
Input Nilai 2: 85
Rata-rata: 87.5
PS D:\103112430054_Lahra Budi Saputra_IF 12-06\Modul 3 (Guided)> cd "d:\103112430054_Lahra Budi Saputra_IF 12-06\Modul 3 (Guided)\\" ; if ($?) { g++ main.cpp -o main } ; if ($?) { .\main }
Input NIM: 103112430055
Input Nilai 1: 90
Input Nilai 2: 95
Rata-rata: 92.5
PS D:\103112430054_Lahra Budi Saputra_IF 12-06\Modul 3 (Guided)> 
```

Deskripsi:

File main.cpp adalah Program Driver yang berperan sebagai pengguna akhir dari ADT tersebut. File ini hanya perlu meng-include Mahasiswa.h untuk mengakses spesifikasi ADT. Di dalamnya, objek mhs bertipe mahasiswa dibuat, dan fungsi-fungsi primitif ADT (inputMhs dan rata2) kemudian dipanggil untuk memanipulasi dan menampilkan hasil tanpa perlu include Mahasiswa.cpp atau mengetahui detail kodenya . Ini adalah demonstrasi akhir dari modularitas dan abstraksi ADT.

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

Unguided 1

```
/*Nama Lengkap : Lahra budi Saputra
NIM : 103112430054
Kelas : IF 12-06*/
#include <iostream>
#include <string>

using namespace std;

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

// Rumus:  $0.3 \times uts + 0.4 \times uas + 0.3 \times tugas$ 
float hitungNilaiAkhir(float nilai_uts, float nilai_uas, float nilai_tugas) {
    return (0.3 * nilai_uts) + (0.4 * nilai_uas) + (0.3 * nilai_tugas);
}

void dataMhs(Mahasiswa &mhs) {
    cin.ignore();

    cout << "Masukkan Nama Lengkap : ";
    getline(cin, mhs.nama);

    cout << "Masukkan NIM      : ";
    cin >> mhs.nim;

    cout << "Masukkan Nilai UTS   : ";
    cin >> mhs.uts;

    cout << "Masukkan Nilai UAS    : ";
    cin >> mhs.uas;

    cout << "Masukkan Nilai Tugas  : ";
    cin >> mhs.tugas;

    // Hitung dan simpan Nilai Akhir
    mhs.nilai_akhir = hitungNilaiAkhir(mhs.uts, mhs.uas, mhs.tugas);
}
```

```

void tampilkanData(const Mahasiswa mhs[], int jumlah) {
    cout <<
    "\n=====
    ===\n";
    cout << "No\tNIM\tNama\tNilai Akhir\n";
    cout << "-----\n";

    for (int i = 0; i < jumlah; i++) {
        cout << i + 1 << "\t"
            << mhs[i].nim << "\t"
            << mhs[i].nama << "\t\t"
            << mhs[i].nilai_akhir << endl;
    }
    cout <<
    "=====
    ===\n";
}

int main() {
    Mahasiswa list[10];
    int jumlah = 0;
    char lagi;

    cout << "====> INPUT DATA MAHASISWA <====\n";

    do {
        if (jumlah >= 10) {
            cout << "\n Tidak dapat menambah data lagi (maks. 10).\n";
            break;
        }

        dataMhs(list[jumlah]);
        jumlah++;

        cout << "\nTambah data lagi? (y/t): ";
        cin >> lagi;
    } while (lagi == 'y' || lagi == 'Y');

    tampilkanData(list, jumlah);
    return 0;
}

```

Screenshots Output:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
PS D:\103112430054_Lahra Budi Saputra_IF 12-06> cd "d:\103112430054_Lahra Budi Saputra_IF 12-06\Modul 3 (Unguided)\\" ; if ($?) { g++ Soal1.cpp -o
Soal1 } ; if ($?) { .\Soal1 }
====> INPUT DATA MAHASISWA <====

Masukkan Nama Lengkap : lahra budi
Masukkan NIM : 103112430054
Masukkan Nilai UTS : 90
Masukkan Nilai UAS : 90
Masukkan Nilai Tugas : 90

Tambah data lagi? (y/t): y
Masukkan Nama Lengkap : diah ayu
Masukkan NIM : 103112430055
Masukkan Nilai UTS : 90
Masukkan Nilai UAS : 98
Masukkan Nilai Tugas : 78

Tambah data lagi? (y/t): y
Masukkan Nama Lengkap : lutfi arrafi
Masukkan NIM : 103112430056
Masukkan Nilai UTS : 98
Masukkan Nilai UAS : 87
Masukkan Nilai Tugas : 87

Tambah data lagi? (y/t): y
Masukkan Nama Lengkap : bagas alfanto
Masukkan NIM : 103112430057
Masukkan Nilai UTS : 90
Masukkan Nilai UAS : 87
Masukkan Nilai Tugas : 88

Masukkan Nama Lengkap : dafa alabiyu
Masukkan NIM : 103112430058
Masukkan Nilai UTS : 88
Masukkan Nilai UAS : 88
Masukkan Nilai Tugas : 78

Tambah data lagi? (y/t): y
Masukkan Nama Lengkap : septian saputra
Masukkan NIM : 103112430059
Masukkan Nilai UTS : 88
Masukkan Nilai UAS : 89
Masukkan Nilai Tugas : 96

Tambah data lagi? (y/t): y
Masukkan Nama Lengkap : galan saputra
Masukkan NIM : 103112430060
Masukkan Nilai UTS : 88
Masukkan Nilai UAS : 89
Masukkan Nilai Tugas : 90

Tambah data lagi? (y/t): y
Masukkan Nama Lengkap : selyn saputri
Masukkan NIM : 103112430061
Masukkan Nilai UTS : 89
Masukkan Nilai UAS : 90
Masukkan Nilai Tugas : 97

Tambah data lagi? (y/t): y
Masukkan Nama Lengkap : rere hutri
Masukkan NIM : 103112430062
Masukkan Nilai UTS : 88
Masukkan Nilai UAS : 78
Masukkan Nilai Tugas : 79

Tambah data lagi? (y/t): y
Masukkan Nama Lengkap : jokowi dodo
Masukkan NIM : 103112430063
Masukkan Nilai UTS : 89
Masukkan Nilai UAS : 78
Masukkan Nilai Tugas : 89

Tambah data lagi? (y/t): t

=====
No      NIM      Nama      Nilai Akhir
-----
1      103112430054      lahra budi      90
2      103112430055      diah ayu      89.6
3      103112430056      lutfi arrafi      90.3
4      103112430057      bagas alfanto      88.2
5      103112430058      dafa alabiyu      85
6      103112430059      septian saputra      90.8
7      103112430060      galan saputra      89
8      103112430061      selyn saputri      91.8
9      103112430062      rere hutri      81.3
10     103112430063      jokowi dodo      84.6
=====
PS D:\103112430054_Lahra Budi Saputra_IF 12-06\Modul 3 (Unguided)>

```

Deskripsi:

Program di atas mengimplementasikan konsep Abstract Data Type (ADT) untuk mengelola data mahasiswa secara modular dan terstruktur. ADT Mahasiswa didefinisikan oleh struct Mahasiswa, yang berisi field seperti nama, nim, uts, uas, tugas, dan nilai_akhir. Program ini memisahkan fungsionalitas menjadi tiga primitif (operasi dasar): fungsi hitungNilaiAkhir bertugas menghitung nilai akhir dengan rumus $0.3 \times UTS + 0.4 \times UAS + 0.3 \times Tugas$; prosedur dataMhs menangani semua input data mahasiswa dan memanggil hitungNilaiAkhir untuk menyimpan hasilnya; dan prosedur tampilkanData berfungsi untuk mencetak semua data mahasiswa yang tersimpan dalam format tabel. Fungsi utama (main) bertindak sebagai driver program, menggunakan array list[10] untuk menampung data, dan menjalankan loop do-while yang secara berulang memanggil dataMhs untuk mengumpulkan data hingga mencapai batas maksimum 10, atau hingga pengguna memilih untuk berhenti, sebelum akhirnya memanggil tampilkanData untuk menampilkan keseluruhan hasil.

Unguided 2

(pelajaran.h)

```
#ifndef PELAJARAN_H_INCLUDED
#define PELAJARAN_H_INCLUDED

#include <string>

struct pelajaran {
    std::string namaMapel;
    std::string kodeMapel;
};

pelajaran create_pelajaran(std::string namapel, std::string kodepel);

void tampil_pelajaran(pelajaran pel);

#endif
```

Deskripsi Singkat : File pelajaran.h bertindak sebagai Spesifikasi ADT (Interface) , yang mendeklarasikan TYPE data melalui struct pelajaran dan memuat prototipe dua fungsi primitif: create_pelajaran (sebagai konstruktor) dan tampil_pelajaran (sebagai prosedur tampilan).

(pelajaran.cpp)

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

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 << "kode: " << pel.kodeMapel << endl;
}
```

Deskripsi Singkat : File pelajaran.cpp menyediakan Realisasi ADT (Body), di mana kode sumber untuk kedua fungsi primitif ditulis. Fungsi create_pelajaran menginisialisasi dan mengembalikan objek pelajaran baru, sementara tampil_pelajaran menangani pencetakan detail objek ke konsol.

(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);

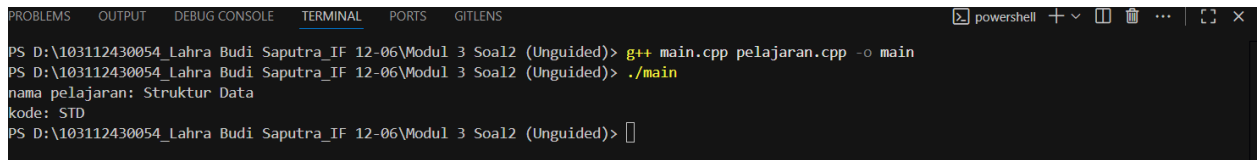
    tampil_pelajaran(pel);

    return 0;
}
```

Deskripsi Singkat : File pelajaran.cpp menyediakan Realisasi ADT (Body), di mana kode sumber untuk kedua fungsi primitif ditulis. Fungsi create_pelajaran menginisialisasi dan mengembalikan objek pelajaran baru, sementara tampil_pelajaran menangani pencetakan detail objek ke konsol.

Ketiga file kode tersebut (pelajaran.h, pelajaran.cpp, dan main.cpp) menunjukkan implementasi Abstract Data Type (ADT) Pelajaran secara modular

Screenshots Output:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
PS D:\103112430054_Lahra Budi Saputra_IF 12-06\Modul 3 Soal2 (Unguided)> g++ main.cpp pelajaran.cpp -o main
PS D:\103112430054_Lahra Budi Saputra_IF 12-06\Modul 3 Soal2 (Unguided)> ./main
nama pelajaran: Struktur Data
kode: STD
PS D:\103112430054_Lahra Budi Saputra_IF 12-06\Modul 3 Soal2 (Unguided)> |
```

Unguided 3

```
#include <iostream>

using namespace std;

// --- FUNGSI/PROSEDUR PRIMITIF ---

// Prosedur menampilkan isi array 2D
void tampilkanArray(const int arr[3][3]) {
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            cout << arr[i][j] << " ";
        }
        cout << endl;
    }
    cout << endl;
}

// Prosedur menukar elemen array 2D pada posisi tertentu
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;
}

// Prosedur menukar nilai yang ditunjuk oleh 2 pointer
void tukarNilaiPointer(int* pA, int* pB) {
    int temp = *pA;
    *pA = *pB;
    *pB = temp;
}

// --- PROGRAM DRIVER (MAIN) ---

int main() {
    // 1. Deklarasi 2 array 2D 3x3 dan inisialisasi
    int ArrayA[3][3] = {
        {10, 20, 30},
        {40, 50, 60},
        {70, 80, 90}
    };
}
```

```

int ArrayB[3][3] = {
    {1, 2, 3},
    {4, 5, 6},
    {7, 8, 9}
};

// 2. Deklarasi 2 pointer integer dan inisialisasi variabel target
int *ptr1, *ptr2;
int nilaiX = 1000, nilaiY = 2000;

ptr1 = &nilaiX;
ptr2 = &nilaiY;

cout << "=== KONDISI AWAL ===" << endl;
cout << "Array A:" << endl;
tampilkanArray(ArrayA);
cout << "Array B:" << endl;
tampilkanArray(ArrayB);
cout << "Nilai X (Ptr1): " << *ptr1 << ", Nilai Y (Ptr2): " << *ptr2 << endl <<
endl;

// --- OPERASI PERTUKARAN ---

// 1. Tukar elemen Array 2D pada posisi [0][0]
tukarElemenArray(ArrayA, ArrayB, 0, 0);

// 2. Tukar Nilai Pointer
tukarNilaiPointer(ptr1, ptr2);

cout << "=== KONDISI AKHIR ===" << endl;
cout << "Array A:" << endl;
tampilkanArray(ArrayA);
cout << "Array B:" << endl;
tampilkanArray(ArrayB);
cout << "Nilai X (Ptr1): " << *ptr1 << ", Nilai Y (Ptr2): " << *ptr2 << endl;

return 0;
}

```

Screenshots Output

```
PS D:\103112430054_Lahra Budi Saputra_IF 12-06> cd "d:\103112430054_Lahra Budi Saputra_IF 12-06\Modul 3 (Unguided)\\" ; if ($?) { g++ Soal3.cpp -o Soal3 } ; if ($?) { .\Soal3 }
===> Sebelum Pertukaran <===
Array A:
10 20 30
40 50 60
70 80 90

Array B:
1 2 3
4 5 6
7 8 9

Pointer 1: 1000, Pointer 2: 2000

===> Sesudah Pertukaran <===
Array A:
1 20 30
40 50 60
70 80 90

Array B:
10 2 3
4 5 6
7 8 9

Pointer 1: 2000, Pointer 2: 1000
PS D:\103112430054_Lahra Budi Saputra_IF 12-06\Modul 3 (Unguided)>
```

Deskripsi:

Program di atas mendemonstrasikan manipulasi array dua dimensi (2D) dan pointer melalui penggunaan prosedur modular. Program utama (main) mendeklarasikan dua buah array integer 3x3 (ArrayA dan ArrayB) dan dua variabel integer (nilaiX, nilaiY) yang alamatnya ditunjuk oleh pointer (ptr1, ptr2). Tiga prosedur primitif digunakan: tampilkanArray mencetak isi array 2D ke layar; tukarElemenArray menukarkan nilai elemen pada posisi tertentu antara ArrayA dan ArrayB; dan tukarNilaiPointer menukarkan nilai yang disimpan dalam variabel yang ditunjuk oleh ptr1 dan ptr2. Setelah menampilkan kondisi awal, program melakukan dua operasi utama menukarkan elemen pada posisi [0][0] kedua array (yaitu 10 dan 1), dan menukarkan nilai nilaiX dan nilaiY (1000 dan 2000)—sebelum menampilkan hasil akhir dari semua variabel.

D. Kesimpulan

Dari modul ini, dapat disimpulkan bahwa ADT merupakan konsep penting dalam pemrograman yang memisahkan antara spesifikasi dan implementasi suatu struktur data. Melalui ADT, programmer dapat mendefinisikan tipe data beserta operasi-operasinya tanpa harus memperhatikan bagaimana data tersebut direpresentasikan di memori. Pendekatan ini meningkatkan abstraksi, modularitas, dan kemudahan pemeliharaan program.

E. Referensi

Jouannaud, J. P., Okada, M. (1997). Abstract Data Type Systems. Theoretical Computer Science, 173(2), 349-391.

Astesiano, E., Reggio, G., & Tarlecki, A. (1955). Recent Trends in Data Type Specification (ADT COMPASS 1994).