

LAPORAN PRAKTIKUM
Modul 3
Abstrak Data Type



Disusun Oleh :
Arzario Irsyad Al Fatih/2211104032
SE 06 2

Asisten Praktikum :
Aldi Putra
Andini Nur Hidayah

Dosen Pengampu :
Wahyu Andi Saputra

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

1. Tujuan

- a. Mahasiswa mampu memahami definisi dan konsep dasar Abstract Data Type (ADT) dalam pemrograman, termasuk pemisahan antara representasi data dan operasinya.
- b. Mahasiswa mampu mengimplementasikan ADT pada program menggunakan bahasa pemrograman, dengan memisahkan deklarasi dan implementasi ADT melalui file header dan file implementasi.
- c. Mahasiswa mampu membangun program yang modular dengan memanfaatkan ADT, sehingga program lebih terstruktur, mudah dipelihara, dan dikembangkan.
- d. Mahasiswa mampu memanfaatkan ADT untuk meningkatkan abstraksi dalam program dengan memisahkan antara penggunaan data dan detail implementasi dari struktur data tersebut.
- e. Mahasiswa memahami pentingnya enkapsulasi (encapsulation) dalam ADT untuk membatasi akses langsung ke elemen-elemen data, dan hanya memperbolehkan interaksi melalui fungsi/prosedur yang telah didefinisikan.

2. Landasan Teori

- a. Abstrak Data Type

Abstrak Data Type (ADT) dalam C++ adalah konsep yang memungkinkan pengembang untuk mendefinisikan struktur data dan operasinya secara terpisah dari implementasinya. Dengan ADT, programmer dapat merancang tipe data yang kompleks, seperti stack, queue, atau linked list, yang memungkinkan manipulasi data secara efisien dan terorganisir. ADT menyediakan antarmuka yang jelas melalui fungsi atau metode, sementara rincian implementasi disembunyikan, mendukung prinsip encapsulation dalam pemrograman berorientasi objek. Ini meningkatkan modularitas dan pemeliharaan kode, karena perubahan pada implementasi tidak mempengaruhi bagian lain dari program yang menggunakan ADT tersebut.

3. Guided

- a. Abstrak Data Type

Abstrak Data Type (ADT) adalah sebuah konsep dalam pemrograman yang mendefinisikan tipe data dengan cara yang terpisah dari implementasinya. ADT menyusun struktur data dan operasi yang dapat dilakukan pada data tersebut dalam bentuk antarmuka yang jelas. Dengan ADT, pengguna hanya perlu memahami operasi yang disediakan tanpa perlu mengetahui detail bagaimana data tersebut diorganisasi atau diimplementasikan di dalam memori. Contoh umum dari ADT termasuk stack, queue, dan list. Konsep ini mendukung prinsip-prinsip pemrograman berorientasi objek, seperti encapsulation dan modularitas, sehingga memudahkan pengembangan dan pemeliharaan kode.

PROBLEMS

OUTPUT

TERMINAL

PORTS

DEBUG CONSOLE

```
PS C:\Users\toshiba\Documents\Tugas Kuliah\Semester 5\Praktikum
aktikum\STD_Arzario_Irsyad_Al_Fatih_2211104032\03_Abstrack_Data
Input NIM = 221104032
Input Nilai = 100
Input Nilai = 100
rata-rata = 100
```

```
#include<iostream>

using namespace std;

struct mahasiswa{
    char nim[20];
    int nilai1, nilai2;
};

void inputMhs(mahasiswa &m);
float rata2(mahasiswa m);

int main(){
    mahasiswa mhs;
    inputMhs(mhs);
    cout << "rata-rata = " << rata2(mhs);
    return 0;
}

void inputMhs(mahasiswa &m) {
    cout << "Input NIM = ";
    cin >> (m).nim;
    cout << "Input Nilai = ";
    cin >> (m).nilai1;
    cout << "Input Nilai = ";
    cin >> (m).nilai2;
}

float rata2(mahasiswa m){
    return(m.nilai1+m.nilai2)/2;
}
```

4. Unguided

- Buat program yang dapat menyimpan data mahasiswa (max. 10) ke dalam sebuah array dengan field nama, nim, uts, uas, tugas, dan nilai akhir. Nilai akhir diperoleh dari FUNGSI dengan rumus $0.3 \times \text{uts} + 0.4 \times \text{uas} + 0.3 \times \text{tugas}$.

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

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

float hitungNilaiAkhir(float uts, float uas, float tugas) {
    return 0.3 * uts + 0.4 * uas + 0.3 * tugas;
}

int main() {
    const int MAKS_MAHASISWA = 10;
    Mahasiswa mahasiswa[MAKS_MAHASISWA];
    int jumlahMahasiswa;

    cout << "Masukkan jumlah mahasiswa (maksimal 10): ";
    cin >> jumlahMahasiswa;

    if (jumlahMahasiswa > MAKS_MAHASISWA) {
        cout << "Jumlah mahasiswa melebihi batas maksimal (10). Program dihentikan." << endl;
        return 1;
    }

    for (int i = 0; i < jumlahMahasiswa; i++) {
        cout << "\nData mahasiswa ke-" << (i + 1) << endl;
        cout << "Nama      : ";
        cin.ignore();
        getline(cin, mahasiswa[i].nama);
        cout << "NIM       : ";
        getline(cin, mahasiswa[i].nim);
        cout << "Nilai UTS : ";
        cin >> mahasiswa[i].uts;
        cout << "Nilai UAS : ";
        cin >> mahasiswa[i].uas;
        cout << "Nilai Tugas: ";
        cin >> mahasiswa[i].tugas;

        mahasiswa[i].nilaiAkhir = hitungNilaiAkhir(mahasiswa[i].uts, mahasiswa[i].uas, mahasiswa[i].tugas);
    }

    cout << "\nData Mahasiswa: \n";
    cout << "===== \n";
    cout << left << setw(5) << "No" << setw(20) << "Nama" << setw(15) << "NIM" << setw(10) << "UTS" << setw(10) <<
    "UAS" << setw(10) << "Tugas" << setw(10) << "Nilai Akhir" << endl;
    cout << "===== \n";
    for (int i = 0; i < jumlahMahasiswa; i++) {
        cout << left << setw(5) << (i + 1)
        << setw(20) << mahasiswa[i].nama
        << setw(15) << mahasiswa[i].nim
        << setw(10) << mahasiswa[i].uts
        << setw(10) << mahasiswa[i].uas
        << setw(10) << mahasiswa[i].tugas
        << setw(10) << fixed << setprecision(2) << mahasiswa[i].nilaiAkhir << endl;
    }

    return 0;
}
```

```
PS C:\Users\toshiba\Documents\Tugas Kuliah\Semester 5\Praktikum\STD_Arzario_Irsyad_Al_syad_Al_Fatih_2211104032\03_Abstrack_Data_Type\UNGUIDED\" ; if ($?) { g++ soal-01.cpp
Masukkan jumlah mahasiswa (maksimal 10): 3
```

Data mahasiswa ke-1

Nama : Rio
NIM : 2211104032
Nilai UTS : 100
Nilai UAS : 100
Nilai Tugas: 100

Data mahasiswa ke-2

Nama : Yogi
NIM : 2211104061
Nilai UTS : 99
Nilai UAS : 99
Nilai Tugas: 99

Data mahasiswa ke-3

Nama : Aji
NIM : 2211104049
Nilai UTS : 98
Nilai UAS : 98
Nilai Tugas: 98

Data Mahasiswa:

No	Nama	NIM	UTS	UAS	Tugas	Nilai Akhir
1	Rio	2211104032	100	100	100	100.00
2	Yogi	2211104061	99.00	99.00	99.00	99.00
3	Aji	2211104049	98.00	98.00	98.00	98.00

- b. Buatlah ADT pelajaran sebagai berikut di dalam file “pelajaran.h”:
Buatlah implementasi ADT pelajaran pada file “pelajaran.cpp”
Cobalah hasil implementasi ADT pada file “main.cpp”!

```
#ifndef PELAJARAN_H
#define PELAJARAN_H

#include <string>
using namespace std;

struct pelajaran {
    string namaMapel;
    string kodeMapel;
};

pelajaran create_pelajaran(string namapel, string kodepel);

void tampil_pelajaran(pelajaran pel);

#endif
```

```

#include <iostream>
#include "soal-02-pelajaran.h"
#include "soal-02-pelajaran.cpp"

using namespace std;

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

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

    return 0;
}

```

```

#include "soal-02-pelajaran.h"
#include <iostream>
using namespace std;

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

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

```

```

PS C:\Users\toshiba\Documents\Tugas Kuliah\Semester 5\Praktikum\STD_Arzari
syad_Al_Fatih_2211104032\03_Abstrack_Data_Type\UNGUIDED\> if ($?) { g++
nama pelajaran : Struktur Data
nilai : STD

```

- c. Buatlah program dengan ketentuan :
- 2 buah array 2D integer berukuran 3x3 dan 2 buah pointer integer
 - fungsi/prosedur yang menampilkan isi sebuah array integer 2D
 - fungsi/prosedur yang akan menukarkan isi dari 2 array integer 2D pada posisi tertentu

- fungsi/prosedur yang akan menukarkan isi dari variabel yang ditunjuk oleh 2 buah pointer

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

void tampilArray(int array[3][3]) {
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            cout << array[i][j] << " ";
        }
        cout << endl;
    }
}

void tukarArray(int array1[3][3], int array2[3][3], int baris, int kolom)
{
    if (baris < 3 && kolom < 3) {
        int temp = array1[baris][kolom];
        array1[baris][kolom] = array2[baris][kolom];
        array2[baris][kolom] = temp;
    } else {
        cout << "Indeks baris atau kolom di luar batas!" << endl;
    }
}

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

int main() {
    int array1[3][3] = {
        {1, 2, 3},
        {4, 5, 6},
        {7, 8, 9}
    };

    int array2[3][3] = {
        {9, 8, 7},
        {6, 5, 4},
        {3, 2, 1}
    };

    int a = 10, b = 20;
    int *p1 = &a;
    int *p2 = &b;

    cout << "Array 1 sebelum pertukaran: " << endl;
    tampilArray(array1);
    cout << "Array 2 sebelum pertukaran: " << endl;
    tampilArray(array2);

    tukarArray(array1, array2, 1, 1);

    cout << "\nArray 1 setelah pertukaran pada posisi (1,1): " << endl;
    tampilArray(array1);
    cout << "Array 2 setelah pertukaran pada posisi (1,1): " << endl;
    tampilArray(array2);

    cout << "\nNilai p1 dan p2 sebelum pertukaran: " << endl;
    cout << "p1: " << *p1 << " p2: " << *p2 << endl;

    tukarPointer(p1, p2);

    cout << "\nNilai p1 dan p2 setelah pertukaran: " << endl;
    cout << "p1: " << *p1 << " p2: " << *p2 << endl;

    return 0;
}
```

```
PS C:\Users\toshiba\Documents\Tugas Kuliah\Semester 5\Praktikum\STD_Arzari  
syad_Al_Fatih_2211104032\03_Abstrack_Data_Type\UNGUIDED\" ; if ($?) { g++
```

Array 1 sebelum pertukaran:

1 2 3

4 5 6

7 8 9

Array 2 sebelum pertukaran:

9 8 7

6 5 4

3 2 1

Array 1 setelah pertukaran pada posisi (1,1):

1 2 3

4 5 6

7 8 9

Array 2 setelah pertukaran pada posisi (1,1):

9 8 7

6 5 4

3 2 1

Nilai p1 dan p2 sebelum pertukaran:

p1: 10 p2: 20

Nilai p1 dan p2 setelah pertukaran:

p1: 20 p2: 10