

LAPORAN PRAKTIKUM
Modul 3
Abstrak Data Type



Disusun Oleh :
Satria Ariq Adelard Dompas/2211104033SE 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 dapat memahami definisi dan konsep dasar Abstract Data Type (ADT) dalam pemrograman, termasuk pemisahan antara bagaimana data direpresentasikan dan bagaimana operasinya dilakukan.
- b. Mahasiswa mampu mengimplementasikan ADT dalam program menggunakan bahasa pemrograman dengan memisahkan deklarasi dan implementasinya melalui file header dan file implementasi.
- c. Mahasiswa dapat membangun program yang bersifat modular dengan memanfaatkan ADT, sehingga struktur program menjadi lebih rapi, mudah dikelola, dan dapat dikembangkan lebih lanjut.
- d. Mahasiswa mampu meningkatkan tingkat abstraksi dalam program melalui pemanfaatan ADT dengan memisahkan antara penggunaan data dan detail teknis implementasi struktur data tersebut.
- e. Mahasiswa memahami pentingnya prinsip enkapsulasi dalam ADT, yang berfungsi untuk membatasi akses langsung ke data dan memungkinkan interaksi hanya melalui fungsi atau prosedur yang telah ditentukan.

2. Landasan Teori

a. Abstrak Data Type

Abstrak Data Type (ADT) dalam C++ adalah konsep yang memungkinkan pengembang mendefinisikan struktur data beserta operasinya secara terpisah dari implementasinya. Melalui ADT, programmer dapat merancang tipe data kompleks seperti stack, queue, atau linked list, yang memfasilitasi pengolahan data secara efisien dan terstruktur. ADT menyediakan antarmuka yang jelas melalui fungsi atau metode, sambil menyembunyikan detail implementasi untuk mendukung prinsip enkapsulasi dalam pemrograman berorientasi objek. Hal ini meningkatkan modularitas dan kemudahan pemeliharaan kode, karena perubahan pada implementasi tidak akan memengaruhi bagian lain dari program yang menggunakan ADT tersebut.

3. Guided

a. Abstrak Data Type

Abstrak Data Type (ADT) adalah konsep dalam pemrograman yang memisahkan definisi tipe data dari implementasinya. ADT merancang struktur data beserta operasi-operasi yang dapat dilakukan melalui antarmuka yang jelas. Dengan pendekatan ini, pengguna cukup memahami fungsi dan operasi yang disediakan tanpa perlu mengetahui detail tentang bagaimana data tersebut dikelola atau disusun dalam memori. Contoh-contoh umum ADT meliputi stack, queue, dan list. Konsep ini sejalan dengan prinsip pemrograman berorientasi objek, seperti enkapsulasi dan modularitas, sehingga memudahkan proses pengembangan dan pemeliharaan kode.

```
OUTPUT  DEBUG CONSOLE  PROBLEMS  TERMINAL  PORTS  COMMENTS
PS D:\Praktikum Struktur Data\Pertemuan3> cd 'd:\Praktikum Struktur Data\Pertemuan3\GUIDED\output'
PS D:\Praktikum Struktur Data\Pertemuan3\GUIDED\output> & .\modul3.exe
Input NIM = 2211104033
Input Nilai = 100
Input Nilai = 97
rata-rata = 98
PS D:\Praktikum Struktur Data\Pertemuan3\GUIDED\output> |
```

```
GUIDED > C++ modul3.cpp > ...
1  #include<iostream>
2
3  using namespace std;
4
5  struct mahasiswa{
6      char nim[20];
7      int nilai1,nilai2;
8  };
9
10 void inputMhs(mahasiswa &m);
11 float rata2(mahasiswa m);
12
13 int main(){
14     mahasiswa mhs;
15     inputMhs(mhs);
16     cout << "rata-rata = " << rata2(mhs);
17     return 0;
18 }
19
20 void inputMhs(mahasiswa &m) {
21     cout << "Input NIM = ";
22     cin >> (m).nim;
23     cout << "Input Nilai = ";
24     cin >> (m).nilai1;
25     cout << "Input Nilai = ";
26     cin >> (m).nilai2;
27 }
28
29 float rata2(mahasiswa m){
30     return(m.nilai1+m.nilai2)/2;
31 }
```

4. Unguided

- a. 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;
}
```

```
OUTPUT  DEBUG CONSOLE  PROBLEMS  TERMINAL  PORTS  COMMENTS

Nilai UAS : 67
Nilai Tugas: 35

Data mahasiswa ke-8
Nama      : Edi
NIM       : 2211104021
Nilai UTS : 90
Nilai UAS : 75
Nilai Tugas: 83

Data Mahasiswa:
=====
No  Nama          NIM          UTS      UAS      Tugas  Nilai Akhir
=====
1   Satria        2211104033   90       100      98      96.40
2   Ade           2211104044   80.00    90.00    77.00    83.10
3   Arzario       2211104022   90.00    89.00    69.00    83.30
4   Bambang      2211104034   83.00    74.00    79.00    78.20
5   Kurcil        2211104099   89.00    90.00    45.00    76.20
6   Aji           2211104038   53.00    325.00   67.00    166.00
7   Yogi          2211104056   78.00    67.00    35.00    60.70
8   Edi           2211104021   90.00    75.00    83.00    81.90
PS D:\Praktikum Struktur Data\Pertemuan3\UNGUIDED\output>
```

- 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”!

```
UNGUIDED > h soal-02-pelajaran.h > ...
1  #ifndef PELAJARAN_H
2  #define PELAJARAN_H
3
4  #include <string>
5  using namespace std;
6
7  struct pelajaran {
8      string namaMapel;
9      string kodeMapel;
10 };
11
12 pelajaran create_pelajaran(string namapel, string kodepel);
13
14 void tampil_pelajaran(pelajaran pel);
15
16 #endif
```

```

UNGUIDED > C++ soal-02-main.cpp > main()
1  #include <iostream>
2  #include "soal-02-pelajaran.h"
3  #include "soal-02-pelajaran.cpp"
4
5  using namespace std;
6
7  int main() {
8      string namapel = "Struktur Data";
9      string kodepel = "STD";
10
11     pelajaran pel = create_pelajaran(namelapel, kodepel);
12
13     tampil_pelajaran(pel);
14
15     return 0;
16 }

```

```

UNGUIDED > C++ soal-02-pelajaran.cpp > tampil_pelajaran(pelajaran)
1  #include "soal-02-pelajaran.h"
2  #include <iostream>
3  using namespace std;
4
5  pelajaran create_pelajaran(string namapel, string kodepel) {
6      pelajaran pel;
7      pel.namaMapel = namapel;
8      pel.kodeMapel = kodepel;
9      return pel;
10 }
11
12 void tampil_pelajaran(pelajaran pel) {
13     cout << "nama pelajaran : " << pel.namaMapel << endl;
14     cout << "kode pelajaran : " << pel.kodeMapel << endl;
15 }

```

```

PS D:\Praktikum Struktur Data\Pertemuan3\UNGUIDED\output> & .\'soal-02-main.exe'
nama pelajaran : Struktur Data
kode pelajaran : STD
PS D:\Praktikum Struktur Data\Pertemuan3\UNGUIDED\output>

```

- 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 D:\Praktikum Struktur Data\Pertemuan3\UNGUIDED\output> cd 'd:\Praktikum Struktur Data\Pertemuan3\UNGUIDED\output'
PS D:\Praktikum Struktur Data\Pertemuan3\UNGUIDED\output> & .\'soal-03.exe'
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):

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
PS D:\Praktikum Struktur Data\Pertemuan3\UNGUIDED\output> █
```