

LAPORAN PRAKTIKUM
Modul 3
“Abstract Data Type (ADT)”



Disusun Oleh:
Rengganis Tantri Pramudita - 2311104065
S1 SE0702

Dosen :
Wahyu Andy Saputra

PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY
PURWOKERTO
2024

1. Tujuan

Memahami konsep Abstract Data Type (ADT) dan penggunaannya dalam pemrograman.

2. Landasan Teori

ADT adalah TYPE dan sekumpulan PRIMITIF (operasi dasar) terhadap TYPE tersebut. Selain itu, dalam sebuah ADT yang lengkap, disertakan pula definisi invarian dari TYPE dan aksioma yang berlaku. ADT merupakan definisi STATIK.

ADT terdiri dari dua komponen utama: data dan operasi. Data merepresentasikan informasi yang disimpan, sementara operasi adalah fungsi-fungsi yang dapat digunakan untuk memanipulasi data tersebut. Dengan menggunakan ADT, programmer dapat bekerja pada tingkat abstraksi yang lebih tinggi, yang memungkinkan pengembangan software yang lebih modular, mudah dipelihara, dan dapat digunakan kembali. Contoh umum dari ADT meliputi stack, queue, list, dan tree, yang masing-masing memiliki operasi spesifik yang terkait dengannya.

3. Guided

Code



```
1 #include <iostream>
2
3 using namespace std;
4
5 struct mahasiswa{
6     char nim[10];
7     int nilai1, nilai2;
8 };
9
10 void inputMhs(mahasiswa m){
11     float rata2(mahasiswa m);
12 }
13
14 int main() {
15     mahasiswa mhs;
16     inputMhs(mhs);
17     cout << "rata-rata = " << rata2(mhs);
18     return 0;
19 }
20
21 void inputMhs(mahasiswa m){
22     cout << "input nim = ";
23     cin >> m.nim;
24     cout << "input nilai = ";
25     cin >> m.nilai1;
26     cout << "input nilai = ";
27     cin >> m.nilai2;
28 }
29
30 float rata2(mahasiswa m){
31     return (m.nilai1+m.nilai2)/2;
32 }
```

Keterangan

1. **struct mahasiswa {...}:** Mendefinisikan sebuah struktur bernama mahasiswa yang memiliki tiga anggota:
 - *nim[10]*: array karakter untuk menyimpan NIM mahasiswa
 - *nilai1* dan *nilai2*: variabel integer untuk menyimpan dua nilai mahasiswa

2. **void inputMhs(mahasiswa &m);** Deklarasi fungsi untuk input data mahasiswa. Menggunakan referensi (&) agar dapat mengubah data asli.
3. **float rata2(mahasiswa m);** Deklarasi fungsi untuk menghitung rata-rata nilai mahasiswa.
4. **int main() {...}**: Fungsi utama program. Di sini:
 - Objek *mhs* dari tipe mahasiswa dibuat
 - Fungsi *inputMhs()* dipanggil untuk mengisi data
 - Hasil rata-rata dihitung dan ditampilkan

Outputnya

```

input nim = 3311104065
input nilai = 100
input nilai = 95
rata-rata = 97
Process returned 0 (0x0) execution time : 9.828 s
Press any key to continue.
  
```

4. Unguided

1. 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 $0.3 \cdot \text{uts} + 0.4 \cdot \text{uas} + 0.3 \cdot \text{tugas}$.

Jawab:

kode

```

1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 struct Mahasiswa {
6     string nama;
7     int nim;
8     float uts, uas, tugas, akhir;
9 };
10
11 const int MAX = 10;
12
13 int main() {
14     Mahasiswa mhs[MAX];
15     int i;
16
17     for (i = 0; i < MAX; i++) {
18         cout << "Masukkan data mahasiswa ke-" << i+1 << endl;
19         cout << "Nama: ";
20         getline(mhs[i].nama);
21         cout << "NIM: ";
22         int nim;
23         while (nim < 0 || nim > 999999999) {
24             cin >> nim;
25             cout << "NIM harus antara 0 dan 999999999\n";
26         }
27         mhs[i].nim = nim;
28         cout << "Uts: ";
29         float uts;
30         while (uts < 0 || uts > 100) {
31             cin >> uts;
32             cout << "Uts harus antara 0 dan 100\n";
33         }
34         mhs[i].uts = uts;
35         cout << "Uas: ";
36         float uas;
37         while (uas < 0 || uas > 100) {
38             cin >> uas;
39             cout << "Uas harus antara 0 dan 100\n";
40         }
41         mhs[i].uas = uas;
42         cout << "Tugas: ";
43         float tugas;
44         while (tugas < 0 || tugas > 100) {
45             cin >> tugas;
46             cout << "Tugas harus antara 0 dan 100\n";
47         }
48         mhs[i].tugas = tugas;
49         mhs[i].akhir = 0.3 * mhs[i].uts + 0.4 * mhs[i].uas + 0.3 * mhs[i].tugas;
50         cout << "Nilai Akhir: " << mhs[i].akhir << endl;
51     }
52
53     cout << "Data Mahasiswa:\n";
54     for (i = 0; i < MAX; i++) {
55         cout << "Nama: " << mhs[i].nama << ", NIM: " << mhs[i].nim << ", Uts: " << mhs[i].uts << ", Uas: " << mhs[i].uas << ", Tugas: " << mhs[i].tugas << ", Akhir: " << mhs[i].akhir << endl;
56     }
57
58     return 0;
59 }
  
```

Keterangan

- struct **Mahasiswa** untuk menyimpan data mahasiswa dengan field nama,

nim, uts, uas, tugas, dan nilai_akhir.

- Membuat array **daftarMhs** dengan maksimum 10 mahasiswa.
- Menggunakan fungsi **hitungNilaiAkhir()** untuk menghitung nilai akhir dengan rumus yang diberikan.
- Memungkinkan pengguna untuk memasukkan data hingga 10 mahasiswa atau berhenti kapan saja dengan mengetik 'selesai' untuk nama.
- Menampilkan daftar mahasiswa beserta nilai akhir mereka.

Outputnya

```

C:\Users\ictipernanaka>
Number data mahasiswa no-1 (Nama, jenis kelamin) urutur nama (jika sudah selesai):
Nama: ranggaria
UMR: 2011000000
Nilai UTS: 80
Nilai UAS: 90
Nilai Tugas: 80
Masukkan data mahasiswa no-2 (Nama, jenis kelamin) urutur nama (jika sudah selesai):
Nama: tiarna
UMR: 2011000000
Nilai UTS: 80
Nilai UAS: 90
Nilai Tugas: 100
Masukkan data mahasiswa no-3 (Nama, jenis kelamin) urutur nama (jika sudah selesai):
Nama: naura
UMR: 2011000000
Nilai UTS: 80
Nilai UAS: 90
Nilai Tugas: 80
Masukkan data mahasiswa no-4 (Nama, jenis kelamin) urutur nama (jika sudah selesai):
Nama: afrelal
Input Mahasiswa:
Mahasiswa no-1:
Nama: ranggaria
UMR: 2011000000
Nilai Akhir: PG.1
Mahasiswa no-2:
Nama: tiarna
UMR: 2011000000
Nilai Akhir: PG.1
Mahasiswa no-3:
Nama: naura
UMR: 2011000000
Nilai Akhir: PG.0
Mahasiswa no-4:
Nama: afrelal
UMR: 2011000000
Nilai Akhir: PG.0
PUSKAS RETURNED 0 (000) : execution time : 07.750 s
PUSKAS why say to continue..

```

2. Buatlah ADT pelajaran sebagai berikut di dalam file “pelajaran.h”:

Kode

[illegible]

Buatlah implementasi ADT pelajaran pada file “pelajaran.cpp”

kode

[illegible]

Cobalah hasil implementasi ADT pada file “main.cpp”

Kode

5. Kesimpulan

Praktikum modul 3 tentang Abstract Data Type (ADT) memberikan pemahaman mendalam tentang konsep penting dalam struktur data dan pemrograman. Melalui praktikum ini, mahasiswa dapat memahami bahwa ADT adalah model matematis untuk struktur data yang mendefinisikan tipe data berdasarkan perilakunya, bukan implementasinya. Hal ini memungkinkan pemisahan yang jelas antara spesifikasi (antarmuka) dan implementasi, yang merupakan prinsip kunci dalam pengembangan perangkat lunak modern. Praktikum ini juga menekankan pentingnya modularitas dan reusabilitas dalam pengembangan perangkat lunak. Mahasiswa belajar bagaimana ADT memungkinkan abstraksi data, menyembunyikan detail implementasi dari pengguna, yang mengarah pada kode yang lebih bersih dan mudah dipelihara.