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

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

Outputny

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*uts+0.4*uas+0.3*tugas.

Jawab:

kode

```
Southern X Properties X

| Description of the Control of the Contr
```

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

```
Troobstocksperiorman by

Parameter data managemen we-1 (atam bertin 'select)' series name jibe south select)'

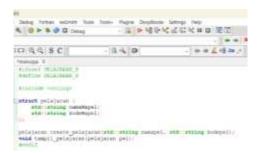
Name: renggaris

Niled Trops: No

Name: toles: N
```

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

Kode



Buatlah implementasi ADT pelajaran pada file "pelajaran.cpp"

kode

Cobalah hasil implementasi ADT pada file "main.cpp"

Kode

```
#inches "pelajament"

#inches electrone

Let help()

#id: string namepel = "Miratras Data";

#id: string namepel = "Miratras Data";

#id: string namepel = "PROF";

pelajamen pel = Urmito pelajamenmangel, Nomepel):

tampil pelajamen peli)

petant ()
```

Outputnya

```
nama pelajaran : Struktur Data
nilai : STD
```

3. 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 STRUKTUR DATA 46
- fungsi/prosedur yang akan menukarkan isi dari variabel yang ditunjuk oleh 2
 buah pointer

Jawab:

Kode

```
The control of the co
```

Outputnya

```
Of Tombhouseman 2 | Part |
Description of the part |
Description of th
```

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.