

LAPORAN PRAKTIKUM
MODUL 3
Abstract Data Type (ADT)



Nama :

Candra Dinata (2311104061)

Kelas :

S1SE 07 02

Dosen :

Wahyu Andi Saputra

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

I. TUJUAN

Tujuan dari pengenalan C++ dalam laporan praktikum adalah untuk membantu mahasiswa memahami dasar-dasar pemrograman, seperti sintaks, tipe data, operator, dan struktur kontrol, serta menguasai penggunaan fungsi untuk modularitas program. Selain itu, mahasiswa juga diperkenalkan dengan konsep Object-Oriented Programming (OOP) seperti kelas, objek, pewarisan, dan enkapsulasi. Melalui latihan ini, mahasiswa diharapkan mampu menerapkan konsep pemrograman untuk menyelesaikan masalah dan memahami cara kerja kompilator C++ dalam mengubah kode sumber menjadi program yang dapat dijalankan.

II. LANDASAN TEORI

C++ adalah bahasa pemrograman yang dikembangkan sebagai pengembangan dari bahasa C dengan menambahkan fitur-fitur pemrograman berorientasi objek (Object-Oriented Programming, OOP). Dalam C++, konsep OOP seperti enkapsulasi, pewarisan, dan polimorfisme menjadi dasar untuk membuat program yang lebih modular, terstruktur, dan dapat digunakan kembali. Bahasa ini mendukung penggunaan fungsi, manipulasi memori secara langsung melalui pointer, serta memiliki kemampuan untuk menangani berbagai tipe data dasar dan struktural. C++ juga mendukung konsep pemrograman prosedural dan modular, yang memudahkan pengembangan program yang kompleks. Kompilasi program C++ dilakukan melalui kompilator yang menerjemahkan kode sumber menjadi file eksekusi yang bisa dijalankan oleh komputer.

III. GUIDED

1.

```
mod3 > G- guided.cpp > ...
1  #include<iostream>
2
3  using namespace std;
4
5  struct mahasiswa{
6      char nim[10];
7      int nilai1, nilai2;
8  };
9
10 int main(){
11
12 }
13
14
```

Kode tersebut mendefinisikan sebuah struktur bernama `mahasiswa` yang berisi atribut `nim` (Nomor Induk Mahasiswa) berupa array karakter dan dua nilai bertipe `int`. Struktur ini dapat digunakan untuk menyimpan data mahasiswa. Selain itu, terdapat fungsi `main()` sebagai titik awal program, namun belum ada logika yang diimplementasikan dalam fungsi tersebut.

2.

```
mod3 > guided.cpp > ...
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 int main(){
14     mahasiswa mhs;
15     inputMhs(mhs);
16     cout<<"rata-rata = " << rata2(mhs);
17     return;
18 }
19
20
21 |
```

Kode tersebut mendefinisikan struktur `mahasiswa` dengan atribut `nim`, `nilai1`, dan `nilai2`. Ada dua fungsi: `inputMhs(mahasiswa &m)` yang menerima input data mahasiswa (belum diimplementasikan dalam kode) dan `rata2(mahasiswa m)` yang menghitung rata-rata dari dua nilai mahasiswa (juga belum diimplementasikan). Di dalam fungsi `main()`, sebuah variabel `mhs` bertipe `mahasiswa` dibuat, kemudian data dimasukkan melalui fungsi `inputMhs`, dan hasil rata-rata nilainya ditampilkan dengan fungsi `rata2`.

3.

```
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 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 float rata2(mahasiswa m){
29     return(m.nilai1+m.nilai2)/2;
30 }
31 }
```

Kode tersebut mendefinisikan struktur `mahasiswa` dengan atribut `nim`, `nilai1`, dan `nilai2`, serta dua fungsi: `inputMhs` dan `rata2`. Fungsi `inputMhs(mahasiswa &m)` menerima input data mahasiswa, yaitu NIM, nilai pertama, dan nilai kedua. Fungsi `rata2(mahasiswa m)` menghitung rata-rata dari dua nilai mahasiswa. Di fungsi `main()`, sebuah variabel `mhs` bertipe `mahasiswa` dibuat, kemudian data dimasukkan melalui `inputMhs`, dan hasil rata-rata nilai ditampilkan menggunakan fungsi `rata2`.

IV. UNGUIDED

1.

```
mod3 > unguided.cpp > tampilkanMahasiswa(const Mahasiswa &);
1  #include <iostream>
2  using namespace std;
3
4  struct Mahasiswa {
5      string nama;
6      string nim;
7      float uts;
8      float uas;
9      float tugas;
10     float nilai_akhir;
11 };
12
13 void hitungNilaiAkhir(Mahasiswa &mhs) {
14     mhs.nilai_akhir = 0.3 * mhs.uts + 0.4 * mhs.uas + 0.3 * mhs.tugas;
15 }
16
17 void tampilkanMahasiswa(const Mahasiswa &mhs) {
18     cout << "Nama: " << mhs.nama << endl;
19     cout << "NIM: " << mhs.nim << endl;
20     cout << "UTS: " << mhs.uts << endl;
21     cout << "UAS: " << mhs.uas << endl;
22     cout << "Tugas: " << mhs.tugas << endl;
23     cout << "Nilai Akhir: " << mhs.nilai_akhir << endl;
24 }
25
26 int main() {
27     Mahasiswa mahasiswa[10];
28     int jumlahMahasiswa;
29
30     cout << "Masukkan jumlah mahasiswa (maks 10): ";
31     cin >> jumlahMahasiswa;
```

```
Nama: candra
NIM: 2311104061
UTS: 85
UAS: 90
Tugas: 95
Nilai Akhir: 90

Nama: fajar
NIM: 2311104098
UTS: 80
UAS: 95
Tugas: 95
Nilai Akhir: 90.5

Nama: jamal
NIM: 2311104099
UTS: 90
UAS: 90
Tugas: 90
Nilai Akhir: 90

PS C:\Users\Candra Dinata\pertemuan1\mod3\output>
```

Kode tersebut mendefinisikan struktur `Mahasiswa` dengan atribut nama, NIM, nilai UTS, UAS, tugas, dan nilai akhir. Fungsi `hitungNilaiAkhir` menghitung nilai akhir mahasiswa berdasarkan bobot 30% untuk UTS, 40% untuk UAS, dan 30% untuk tugas. Fungsi `tampilkanMahasiswa` menampilkan data lengkap mahasiswa. Dalam fungsi `main`, program meminta input jumlah mahasiswa (maksimal 10) lalu untuk setiap mahasiswa, diminta nama, NIM, nilai UTS, UAS, dan tugas. Setelah itu, nilai akhir dihitung dan data setiap mahasiswa ditampilkan secara terpisah.

2.

```
5 // Definisi ADT Pelajaran
6 struct pelajaran {
7     string namaMapel;
8     string kodeMapel;
9 };
10
11 // Fungsi untuk membuat pelajaran
12 pelajaran create_pelajaran(string namapel, string kodepel) {
13     pelajaran pel;
14     pel.namaMapel = namapel;
15     pel.kodeMapel = kodepel;
16     return pel;
17 }
18
19 // Fungsi untuk menampilkan pelajaran
20 void tampil_pelajaran(pelajaran pel) {
21     cout << "Nama Pelajaran: " << pel.namaMapel << endl;
22     cout << "Kode Pelajaran: " << pel.kodeMapel << endl;
23 }
24
25 int main() {
26     // Inisialisasi nama pelajaran dan kode pelajaran
27     string namapel = "Struktur Data";
28     string kodepel = "STD";
29
30     // Membuat pelajaran dan menampilkannya
31     pelajaran pel = create_pelajaran(namapel, kodepel);
32     tampil_pelajaran(pel);
33
34     return 0;
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Candra Dinata\pertemuan1\mod3\output> cd 'c:\Users\Candra Dinata\pertemuan1\mod3\output'
PS C:\Users\Candra Dinata\pertemuan1\mod3\output> & .\u2.exe
Nama Pelajaran: Struktur Data
Kode Pelajaran: STD
PS C:\Users\Candra Dinata\pertemuan1\mod3\output>
```

Kode tersebut mendefinisikan struktur `pelajaran` dengan dua atribut, yaitu `namaMapel` (nama mata pelajaran) dan `kodeMapel` (kode mata pelajaran). Fungsi `create_pelajaran` digunakan untuk membuat objek `pelajaran` dengan mengisi nama dan kode pelajaran, sementara fungsi `tampil_pelajaran` menampilkan informasi pelajaran tersebut. Dalam fungsi `main`, sebuah pelajaran dengan nama "Struktur Data" dan kode "STD" dibuat menggunakan `create_pelajaran`, kemudian informasi pelajaran tersebut ditampilkan menggunakan `tampil_pelajaran`.

3.

```
mod3 > u3.cpp > main()
4 void tampilkanArray2D(int arr[3][3]) {
5     for (int i = 0; i < 3; i++) {
6         for (int j = 0; j < 3; j++) {
7             cout << arr[i][j] << " ";
8         }
9         cout << endl;
10    }
11 }
12
13 void tukarNilai(int* ptr1, int* ptr2) {
14     int temp = *ptr1;
15     *ptr1 = *ptr2;
16     *ptr2 = temp;
17 }
18
19 int main() {
20     int array2D[3][3] = {
21         {1, 2, 3},
22         {4, 5, 6},
23         {7, 8, 9}
24     };
25
26     int* ptr1 = &array2D[0][0]; // pointer ke elemen pertama
27     int* ptr2 = &array2D[2][2]; // pointer ke elemen terakhir
28
29     cout << "Array sebelum penukaran:\n";
30     tampilkanArray2D(array2D);
31
32     tukarNilai(ptr1, ptr2);
33
34     cout << "\nArray setelah penukaran:\n";
35     tampilkanArray2D(array2D);
36 }
PS C:\Users\Candra Dinata\pertemuan1\mod3\output> cd 'c:\Users\Candra Dinata\pertemuan1\mod3\output' & .\u3.exe
Array sebelum penukaran:
1 2 3
4 5 6
7 8 9

Array setelah penukaran:
9 2 3
4 5 6
7 8 1
PS C:\Users\Candra Dinata\pertemuan1\mod3\output>
```

Kode tersebut mendefinisikan fungsi `tampilkanArray2D` untuk menampilkan elemen-elemen dari sebuah array 2D berukuran 3x3 dan fungsi `tukarNilai` untuk menukar nilai antara dua elemen menggunakan pointer. Di dalam fungsi `main`, sebuah array 2D dengan nilai awal 1 hingga 9 dideklarasikan. Pointer `ptr1` menunjuk ke elemen pertama (nilai 1) dan `ptr2` menunjuk ke elemen terakhir (nilai 9). Sebelum penukaran, array ditampilkan, kemudian nilai elemen pertama dan terakhir ditukar menggunakan `tukarNilai`. Setelah penukaran, array ditampilkan lagi untuk menunjukkan perubahan.

V. KESIMPULAN

Abstract Data Type (ADT) merupakan konsep fundamental dalam pemrograman yang memungkinkan pengembang untuk mendefinisikan tipe data berdasarkan perilaku dan operasi yang dapat dilakukan, tanpa memperhatikan implementasi detailnya. ADT memberikan cara untuk mengorganisir data secara logis dan mengabstraksi operasi-operasi yang berkaitan dengan data tersebut, seperti pada struktur data seperti tumpukan (stack), antrian (queue), atau daftar (list). Penggunaan ADT mempermudah proses desain program dengan memungkinkan pemisahan antara antarmuka (interface) dan implementasi, sehingga mendukung prinsip modularitas dan enkapsulasi. Melalui praktik penggunaan ADT, programmer dapat meningkatkan efisiensi dan keterbacaan kode, serta meminimalisir kesalahan dengan memfokuskan perhatian pada fungsi yang diinginkan daripada detail teknis implementasi.