

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



Disusun Oleh:
Tiurma Grace Angelina 2311104042
SE-07-02

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 nilail,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).nilail;
25     cout << "input nilai = ";
26     cin >> (m).nilai2;
27 }
```

```

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 }
32

```

OUTPUT:

```

C:\Users\USER\Documents\fgh\guided\bin\Debug\guided.exe
Input nim= 2311104042
input nilai =99
input nilai =97
rata-rata = 98
Process returned 0 (0x0) execution time : 18.766 s
Press any key to continue.

```

PENJELASAN:

1. Struct `mahasiswa`:

- Digunakan untuk menyimpan data mahasiswa, yaitu NIM, nilai pertama (`nilai1`), dan nilai kedua (`nilai2`).

2. Fungsi `inputMhs(mahasiswa &m)`:

- Mengambil input dari pengguna berupa NIM, nilai pertama, dan nilai kedua untuk disimpan di struct `mahasiswa`.

3. Fungsi `rata2(mahasiswa m)`:

- Menghitung rata-rata dari dua nilai mahasiswa yang diberikan, kemudian mengembalikannya sebagai nilai float.

4. Fungsi `main()`:

- Memanggil fungsi `inputMhs()` untuk mengisi data mahasiswa, kemudian memanggil fungsi `rata2()` untuk menghitung rata-rata dan menampilkannya.

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

CODE :

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  struct Mahasiswa {
6      string nama;
7      string nim;
8      float nilaiUTS;
9      float nilaiUAS;
10     float nilaiTugas;
11     float nilaiAkhir;
12 };
13
14 float hitungNilaiAkhir(float uts, float uas, float tugas) {
15     float nilaiAkhir = (0.3 * uts) + (0.4 * uas) + (0.3 * tugas);
16     return nilaiAkhir;
17 }
18
19 int main() {
20     const int maxMahasiswa = 10;
21     Mahasiswa dataMahasiswa[maxMahasiswa];
22     int jumlah;
23
24     cout << "Berapa jumlah mahasiswa yang akan diinput (maksimal 10): ";
25     cin >> jumlah;
26
27     if (jumlah > maxMahasiswa) {
```

```
21     Mahasiswa dataMahasiswa[maxMahasiswa];
22     int jumlah;
23
24     cout << "Berapa jumlah mahasiswa yang akan diinput (maksimal 10): ";
25     cin >> jumlah;
26
27     if (jumlah > maxMahasiswa) {
28         cout << "Jumlah tidak boleh lebih dari " << maxMahasiswa << endl;
29         return 1;
30     }
31
32     for (int i = 0; i < jumlah; i++) {
33         cout << "\nMahasiswa ke-" << (i + 1) << endl;
34         cout << "Nama: ";
35         cin.ignore();
36         getline(cin, dataMahasiswa[i].nama);
37         cout << "NIM: ";
38         cin >> dataMahasiswa[i].nim;
39         cout << "Nilai UTS: ";
40         cin >> dataMahasiswa[i].nilaiUTS;
41         cout << "Nilai UAS: ";
42         cin >> dataMahasiswa[i].nilaiUAS;
43         cout << "Nilai Tugas: ";
44         cin >> dataMahasiswa[i].nilaiTugas;
45
46         dataMahasiswa[i].nilaiAkhir = hitungNilaiAkhir(
47             dataMahasiswa[i].nilaiUTS,
```

```

45
46         dataMahasiswa[i].nilaiAkhir = hitungNilaiAkhir(
47             dataMahasiswa[i].nilaiUTS,
48             dataMahasiswa[i].nilaiUAS,
49             dataMahasiswa[i].nilaiTugas
50         );
51     }
52
53
54     cout << "\nData Mahasiswa dan Nilai Akhir:\n";
55     for (int i = 0; i < jumlah; i++) {
56         cout << "\nMahasiswa " << (i + 1) << endl;
57         cout << "Nama: " << dataMahasiswa[i].nama << endl;
58         cout << "NIM: " << dataMahasiswa[i].nim << endl;
59         cout << "Nilai UTS: " << dataMahasiswa[i].nilaiUTS << endl;
60         cout << "Nilai UAS: " << dataMahasiswa[i].nilaiUAS << endl;
61         cout << "Nilai Tugas: " << dataMahasiswa[i].nilaiTugas << endl;
62         cout << "Nilai Akhir: " << dataMahasiswa[i].nilaiAkhir << endl;
63         cout << "-----" << endl;
64     }
65
66     return 0;
67 }
68
69
70

```

OUTPUT:

```

C:\Users\USER\Documents\fgjh\unguided1\bin\Debug\unguided1.exe
Berapa jumlah mahasiswa yang akan diinput (maksimal 10): 2

Mahasiswa ke-1
Nama: TIUR
NIM: 2311104042
Nilai UTS: 99
Nilai UAS: 99
Nilai Tugas: 99

Mahasiswa ke-2
Nama: RENGGANIS
NIM: 23111040465
Nilai UTS: 99
Nilai UAS: 99
Nilai Tugas: 99

Data Mahasiswa dan Nilai Akhir:

Mahasiswa 1
Nama: TIUR
NIM: 2311104042
Nilai UTS: 99
Nilai UAS: 99
Nilai Tugas: 99
Nilai Akhir: 99
-----

Mahasiswa 2
Nama: RENGGANIS
NIM: 23111040465

```

```
C:\Users\USER\Documents\fgjh\unguided1\bin\Debug\unguided1.exe

Mahasiswa 1
Nama: TIUR
NIM: 2311104042
Nilai UTS: 99
Nilai UAS: 99
Nilai Tugas: 99
Nilai Akhir: 99
-----

Mahasiswa 2
Nama: RENGGANIS
NIM: 23111040465
Nilai UTS: 99
Nilai UAS: 99
Nilai Tugas: 99
Nilai Akhir: 99
-----

Process returned 0 (0x0)   execution time : 38.363 s
Press any key to continue.
```

Penjelasan :

1. **Struct Mahasiswa:** Menyimpan data mahasiswa dengan atribut nama, nim, uts, uas, tugas, dan nilaiAkhir.
 2. **Fungsi hitungNilaiAkhir:** Menghitung nilai akhir mahasiswa berdasarkan rumus yang diberikan: $0.3 \cdot \text{uts} + 0.4 \cdot \text{uas} + 0.3 \cdot \text{tugas}$.
 3. **Array mahasiswa[10]:** Menyimpan data maksimal 10 mahasiswa.
 4. **Input dan Output:** Program meminta input data dari pengguna dan menampilkan data serta nilai akhir masing-masing mahasiswa. Program ini terbatas pada 10 mahasiswa, tetapi dapat dimodifikasi sesuai kebutuhan.
2. Buatlah ADT pelajaran sebagai berikut di dalam *file* “pelajaran.h”:

```
type pelajaran <
    namaMapel : string
    kodeMapel : string
>
fungsi create_pelajaran( namaPel : string, kodePel : string ) →
    pelajaran
prosedur tampil_pelajaran( pel : pelajaran )
```

Buatlah implementasi ADT pelajaran pada *file* “pelajaran.cpp”

Cobalah hasil implementasi ADT pada *file* “main.cpp”

```

using namespace std;
int main() {
    string namapel = "Struktur Data";
    string kodepel = "STD";
    pelajaran pel = create_pelajaran(namapel, kodepel);
    tampil_pelajaran(pel);

    return 0;
}

```

Gambar 3-1 Main.cpp pelajaran

CODE:

```

1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  struct pelajaran {
7      string namaMapel;
8      string kodeMapel;
9  };
10
11  pelajaran create_pelajaran(string namaMapel, string kodepel) {
12      pelajaran pel;
13      pel.namaMapel = namaMapel;
14      pel.kodeMapel = kodepel;
15      return pel;
16  }
17
18  void tampil_pelajaran(pelajaran pel) {
19      cout << "nama pelajaran : " << pel.namaMapel << endl;
20      cout << "nilai : " << pel.kodeMapel << endl;
21  }
22
23  int main() {
24
25      string namapel = "Struktur Data";
26      string kodepel = "STD";
27

```

```

9  };
10
11  pelajaran create_pelajaran(string namaMapel, string kodepel) {
12      pelajaran pel;
13      pel.namaMapel = namaMapel;
14      pel.kodeMapel = kodepel;
15      return pel;
16  }
17
18  void tampil_pelajaran(pelajaran pel) {
19      cout << "nama pelajaran : " << pel.namaMapel << endl;
20      cout << "nilai : " << pel.kodeMapel << endl;
21  }
22
23  int main() {
24
25      string namapel = "Struktur Data";
26      string kodepel = "STD";
27
28      pelajaran pel = create_pelajaran(namapel, kodepel);
29
30      tampil_pelajaran(pel);
31
32      return 0;
33  }
34

```

OUTPUT:

```
C:\Users\USER\Documents\fgjh\unguided2\bin\Debug\unguided2.exe
nama pelajaran : Struktur Data
nilai : STD

Process returned 0 (0x0)   execution time : 0.096 s
Press any key to continue.
```

PENJELASAN:

- **#include <iostream> dan #include <string>**: Digunakan untuk mengakses fungsi input-output (cout, cin) dan tipe data string.
- **Struct pelajaran**: Mendefinisikan tipe data pelajaran yang menyimpan namaMapel dan kodeMapel.
- **Fungsi create_pelajaran**: Mengembalikan objek pelajaran yang berisi nama dan kode pelajaran.
- **Fungsi tampil_pelajaran**: Mencetak nama dan kode pelajaran ke layar.
- **Fungsi main()**: Kode utama yang menjalankan program. Program membuat objek pelajaran, kemudian menampilkan informasinya.

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

```
1  #include <iostream>
2  using namespace std;
3
4  void cetak_array(int matriks[3][3]) {
5      for (int i = 0; i < 3; i++) {
6          for (int j = 0; j < 3; j++) {
7              cout << matriks[i][j] << " ";
8          }
9          cout << endl;
10     }
11 }
12
13 void tukar_elemen_array(int matriks1[3][3], int matriks2[3][3], int baris, int kolom) {
14     int sementara = matriks1[baris][kolom];
15     matriks1[baris][kolom] = matriks2[baris][kolom];
16     matriks2[baris][kolom] = sementara;
17 }
18
19 void tukar_nilai_pointer(int* penunjuk1, int* penunjuk2) {
20     int sementara = *penunjuk1;
21     *penunjuk1 = *penunjuk2;
22     *penunjuk2 = sementara;
23 }
24
25 int main() {
26     int matriks1[3][3] = { {1, 2, 3}, {4, 5, 6}, {7, 8, 9} };
27     int matriks2[3][3] = { {9, 8, 7}, {6, 5, 4}, {3, 2, 1} };
```



```

25 int main() {
26     int matriks1[3][3] = { {1, 2, 3}, {4, 5, 6}, {7, 8, 9} };
27     int matriks2[3][3] = { {9, 8, 7}, {6, 5, 4}, {3, 2, 1} };
28
29     int angka1 = 15, angka2 = 25;
30     int* penunjuk1 = &angka1;
31     int* penunjuk2 = &angka2;
32
33     cout << "Isi matriks1 sebelum pertukaran:" << endl;
34     cetak_array(matriks1);
35
36     cout << "Isi matriks2 sebelum pertukaran:" << endl;
37     cetak_array(matriks2);
38
39     tukar_elemen_array(matriks1, matriks2, 1, 1);
40
41     cout << "\nIsi matriks1 setelah pertukaran pada posisi [1][1]:" << endl;
42     cetak_array(matriks1);
43
44     cout << "Isi matriks2 setelah pertukaran pada posisi [1][1]:" << endl;
45     cetak_array(matriks2);
46
47     cout << "\nNilai sebelum pertukaran pointer:" << endl;
48     cout << "angka1 = " << angka1 << ", angka2 = " << angka2 << endl;
49
50     tukar_nilai_pointer(penunjuk1, penunjuk2);
51
52     cout << "\nNilai setelah pertukaran pointer:" << endl;
53     cout << "angka1 = " << angka1 << ", angka2 = " << angka2 << endl;
54
55     return 0;
56 }
57
58
59

```

OUTPUT:

```

C:\Users\USER\Documents\fgjh\unguided3\bin\Debug\unguided3.exe
Isi matriks1 sebelum pertukaran:
1 2 3
4 5 6
7 8 9
Isi matriks2 sebelum pertukaran:
9 8 7
6 5 4
3 2 1

Isi matriks1 setelah pertukaran pada posisi [1][1]:
1 2 3
4 5 6
7 8 9
Isi matriks2 setelah pertukaran pada posisi [1][1]:
9 8 7
6 5 4
3 2 1

Nilai sebelum pertukaran pointer:
angka1 = 15, angka2 = 25
Nilai setelah pertukaran pointer:
angka1 = 25, angka2 = 15

Process returned 0 (0x0)   execution time : 0.115 s
Press any key to continue.

```

PENJELASAN :

Nama variabel:

- array1 dan array2 diganti menjadi matriks1 dan matriks2 untuk memberikan nama yang lebih spesifik.
- ptr1 dan ptr2 diganti menjadi penunjuk1 dan penunjuk2 agar lebih jelas sebagai

pointer dalam konteks bahasa Indonesia.

- Variabel a dan b diganti menjadi angka1 dan angka2.

Nama fungsi:

- tampil_array diubah menjadi cetak_array untuk memberi nuansa nama fungsi yang lebih sesuai.
- tukar_array diubah menjadi tukar_elemen_array untuk lebih menjelaskan bahwa fungsi ini menukar elemen di posisi tertentu dari dua array.
- tukar_pointer diubah menjadi tukar_nilai_pointer agar lebih jelas bahwa fungsi ini menukar nilai yang ditunjuk oleh pointer.

5. KESIMPULAN

1. Pemahaman tentang Array 2D dan Pointer:

- Program ini menunjukkan cara kerja array 2D dan penggunaan pointer dalam menukar nilai-nilai variabel. Dua array 2D digunakan untuk menyimpan angka-angka dan ada beberapa fungsi yang menangani pertukaran elemen-elemen di posisi tertentu.

2. Penggunaan Fungsi/Prosedur:

- Program ini mengimplementasikan fungsi untuk menampilkan isi array 2D, menukar elemen-elemen di dalam array pada posisi yang spesifik, dan menukar nilai yang ditunjuk oleh dua pointer.
- Fungsi `cetak_array()` berfungsi untuk mencetak isi array 2D ke layar, sedangkan `tukar_elemen_array()` bertanggung jawab untuk menukar elemen antara dua array 2D di indeks yang sama.

3. Manfaat Pointer dalam Pertukaran Nilai:

- Dengan pointer, program ini dapat menukar nilai-nilai antara dua variabel secara langsung di dalam memori, yang memperlihatkan pentingnya pemahaman tentang pointer dalam pemrograman C++.

4. Abstraksi yang Lebih Modular:

- Program ini mengimplementasikan abstraksi sederhana yang membuat kode menjadi lebih modular. Setiap fungsi memiliki tanggung jawab yang jelas, sehingga program lebih mudah dikelola dan dikembangkan lebih lanjut.