

**LAPORAN PRAKTIKUM**  
**MODUL 3**  
**ABSTRACT DATA TYPE (ADT)**



**Disusun Oleh:**  
**Rizaldy Aulia Rachman (2311104051)**  
**S1SE-07-02**

**Dosen :**  
**Wahyu Andi Saputra, S.Pd., M.Eng**

**PROGRAM STUDI S1 SOFTWARE ENGINEERING**  
**FAKULTAS INFORMATIKA**  
**TELKOM UNIVERSITY PURWOKERTO**  
**2024**

## I. TUJUAN

- Menyembunyikan implementasi internal dari struktur data yang kompleks.
- Memfokuskan pada perilaku dan interface dari data, bukan pada bagaimana data tersebut disimpan di memori.
- Memungkinkan perubahan implementasi internal tanpa mempengaruhi bagian lain dari program yang menggunakan ADT tersebut.

## II. LANDASAN TEORI

### 2.1 Abstract Data Type (ADT)

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 biasanya diimplementasikan menjadi dua buah modul utama dan 1 modul *interface* program utama (*driver*). Dua modul tersebut adalah sebagai berikut:

1. Definisi/Spesifikasi *Type* dan Primitif/*Header* fungsi (.h)
  - Spesifikasi *type* sesuai dengan kaidah bahasa yang dipakai
  - Spesifikasi dari primitif sesuai dengan kaidah dalam konteks prosedural, yaitu:
    - Fungsi : nama, domain, *range*, dan prekondisi jika ada
    - Prosedur : *Initial state*, *Final state*, dan proses yang dilakukan
2. *Body*/realisasi dari primitif (.c)

Berupa kode program dalam bahasa yang bersangkutan (dalam praktikum ini berarti dengan bahasa C++). Realisasi fungsi dan prosedur harus sedapat mungkin memanfaatkan *selector* dan konstruktor.

### III. GUIDED

#### 1. Hasil Codingan Dari Praktikum Minggu Ke-3

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 int main() {
14     mahasiswa mhs;
15     inputMhs(mhs);
16     cout << "Rata-rata = " << rata2(mhs) << endl;
17     return 0;
18 }
19
20 void inputMhs(mahasiswa &m) {
21     cout << "Input NIM = ";
22     cin >> m.nim;
23     cout << "Input nilai 1 = ";
24     cin >> m.nilai1;
25     cout << "Input nilai 2 = ";
26     cin >> m.nilai2;
27 }
28
29 float rata2(mahasiswa m) {
30     return (m.nilai1 + m.nilai2) / 2.0f;
31 }
```

Output:

```
Input NIM = 2311104051
Input nilai 1 = 98
Input nilai 2 = 95
Rata-rata = 96.5
```

#### IV. 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  $0.3*uts+0.4*uas+0.3*tugas$ .

Jawaban:

Code:

```

1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  struct Mahasiswa {
7      string nama;
8      string nim;
9      float uts, uas, tugas, nilaiAkhir;
10 };
11
12 // Fungsi untuk menghitung nilai akhir
13 float hitungNilaiAkhir(float uts, float uas, float tugas) {
14     return (0.3 * uts) + (0.4 * uas) + (0.3 * tugas);
15 }
16
17 // Fungsi untuk memasukkan data mahasiswa
18 void inputMahasiswa(Mahasiswa &m) {
19     cout << "Input Nama: ";
20     cin.ignore(); // Membersihkan buffer agar getline bisa berfungsi
21     getline(cin, m.nama);
22
23     cout << "Input NIM: ";
24     cin >> m.nim;
25
26     cout << "Input nilai UTS: ";
27     cin >> m.uts;
28
29     cout << "Input nilai UAS: ";
30     cin >> m.uas;
31
32     cout << "Input nilai Tugas: ";
33     cin >> m.tugas;
34
35     m.nilaiAkhir = hitungNilaiAkhir(m.uts, m.uas, m.tugas);
36 }
37
38 // Fungsi untuk menampilkan data mahasiswa
39 void tampilMahasiswa(Mahasiswa m) {
40     cout << "\nNama: " << m.nama;
41     cout << "\nNIM: " << m.nim;
42     cout << "\nNilai UTS: " << m.uts;
43     cout << "\nNilai UAS: " << m.uas;
44     cout << "\nNilai Tugas: " << m.tugas;
45     cout << "\nNilai Akhir: " << m.nilaiAkhir << endl;
46 }
47
48 int main() {
49     const int MAX_MHS = 10; // Maksimal 10 mahasiswa
50     Mahasiswa mahasiswa[MAX_MHS];
51     int jumlahMahasiswa;
52
53     cout << "Masukkan jumlah mahasiswa (maks 10): ";
54     cin >> jumlahMahasiswa;
55
56     if(jumlahMahasiswa > MAX_MHS) {
57         cout << "Jumlah mahasiswa melebihi batas maksimal!" << endl;
58         return 1;
59     }
60
61     // Input data mahasiswa
62     for (int i = 0; i < jumlahMahasiswa; i++) {
63         cout << "\nData Mahasiswa ke-" << i + 1 << ": " << endl;
64         inputMahasiswa(mahasiswa[i]);
65     }
66
67     // Tampilkan data mahasiswa
68     cout << "\nDaftar Mahasiswa:\n";
69     for (int i = 0; i < jumlahMahasiswa; i++) {
70         cout << "\nMahasiswa ke-" << i + 1 << ": " << endl;
71         tampilMahasiswa(mahasiswa[i]);
72     }
73
74     return 0;
75 }

```

Output:

```
Daftar Mahasiswa:

Mahasiswa ke-1:

Nama: Rizaldy Aulia Rachman
NIM: 2311104051
Nilai UTS: 99
Nilai UAS: 100
Nilai Tugas: 98
Nilai Akhir: 99.1

Mahasiswa ke-2:

Nama: Cristiano Ronaldo
NIM: 234453455
Nilai UTS: 90
Nilai UAS: 95
Nilai Tugas: 98
Nilai Akhir: 94.4
```

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

```
tipe 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;
}
```

Jawaban:

Code:

```

1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  // Struktur data untuk merepresentasikan sebuah pelajaran
7  struct Pelajaran {
8      string namaMapel;
9      string kodeMapel;
10 };
11
12 // Fungsi untuk membuat objek pelajaran baru
13 Pelajaran create_pelajaran(string nama, string kode) {
14     Pelajaran pelajaran;
15     pelajaran.namaMapel = nama;
16     pelajaran.kodeMapel = kode;
17     return pelajaran;
18 }
19
20 // Prosedur untuk menampilkan informasi tentang sebuah pelajaran
21 void tampil_pelajaran(Pelajaran pelajaran) {
22     cout << "Nama Mata Pelajaran: " << pelajaran.namaMapel << endl;
23     cout << "Kode Mata Pelajaran: " << pelajaran.kodeMapel << endl;
24 }
25
26 int main() {
27     // Membuat objek pelajaran
28     Pelajaran pelajaran1 = create_pelajaran("Struktur Data", "STD");
29
30     // Menampilkan informasi pelajaran
31     tampil_pelajaran(pelajaran1);
32
33     return 0;
34 }

```

Output:

```

Nama Mata Pelajaran: Struktur Data
Kode Mata Pelajaran: 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
- fungsi/prosedur yang akan menukarkan isi dari variabel yang ditunjuk oleh 2 buah *pointer*

Jawaban:

Code:

```

1  #include <iostream>
2
3  using namespace std;
4
5  // Fungsi untuk menampilkan isi array 2D
6  void tampilArray(int arr[3][3]) {
7      for (int i = 0; i < 3; i++) {
8          for (int j = 0; j < 3; j++) {
9              cout << arr[i][j] << " ";
10             }
11             cout << endl;
12         }
13     }
14
15     // Fungsi untuk menukarkan dua nilai array pada posisi tertentu
16     void tukarArray(int arr1[3][3], int arr2[3][3], int baris, int kolom) {
17         int temp = arr1[baris][kolom];
18         arr1[baris][kolom] = arr2[baris][kolom];
19         arr2[baris][kolom] = temp;
20     }
21
22     // Fungsi untuk menukarkan isi variabel yang ditunjuk oleh dua pointer
23     void tukarPointer(int *ptr1, int *ptr2) {
24         int temp = *ptr1;
25         *ptr1 = *ptr2;
26         *ptr2 = temp;
27     }
28
29     int main() {
30         // Deklarasi 2 buah array 2D berukuran 3x3
31         int arr1[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
32         int arr2[3][3] = {{9, 8, 7}, {6, 5, 4}, {3, 2, 1}};
33
34         // Deklarasi 2 pointer integer
35         int a = 10, b = 20;
36         int *ptr1 = &a;
37         int *ptr2 = &b;
38
39         // Menampilkan isi array sebelum ditukar
40         cout << "Array 1 sebelum ditukar:" << endl;
41         tampilArray(arr1);
42         cout << "Array 2 sebelum ditukar:" << endl;
43         tampilArray(arr2);
44
45         // Menukarkan isi array di posisi tertentu
46         tukarArray(arr1, arr2, 1, 1); // Tukar elemen di baris 1, kolom 1
47
48         // Menampilkan isi array setelah ditukar
49         cout << "\nArray 1 setelah ditukar:" << endl;
50         tampilArray(arr1);
51         cout << "Array 2 setelah ditukar:" << endl;
52         tampilArray(arr2);
53
54         // Menampilkan nilai pointer sebelum ditukar
55         cout << "\nNilai sebelum pointer ditukar:" << endl;
56         cout << "Nilai ptr1 (a): " << *ptr1 << endl;
57         cout << "Nilai ptr2 (b): " << *ptr2 << endl;
58
59         // Menukarkan isi dari variabel yang ditunjuk oleh pointer
60         tukarPointer(ptr1, ptr2);
61
62         // Menampilkan nilai pointer setelah ditukar
63         cout << "\nNilai setelah pointer ditukar:" << endl;
64         cout << "Nilai ptr1 (a): " << *ptr1 << endl;
65         cout << "Nilai ptr2 (b): " << *ptr2 << endl;
66
67         return 0;
68     }

```

Output:



Array 1 sebelum ditukar:

1 2 3

4 5 6

7 8 9

Array 2 sebelum ditukar:

9 8 7

6 5 4

3 2 1

Array 1 setelah ditukar:

1 2 3

4 5 6

7 8 9

Array 2 setelah ditukar:

9 8 7

6 5 4

3 2 1

Nilai sebelum pointer ditukar:

Nilai ptr1 (a): 10

Nilai ptr2 (b): 20

Nilai setelah pointer ditukar:

Nilai ptr1 (a): 20

Nilai ptr2 (b): 10