

**LAPORAN PRAKTIKUM**  
**Modul 3**  
**“ABSTRACT DATA TYPE (ADT)”**



**Disusun Oleh:**  
**Fahmi hasan asagaf -2311104074**  
**SE 07 02**

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

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

## 1. Tujuan Praktikum

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

## 2. Landasan Teori

### 3.1 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.

Definisi type dari sebuah ADT dapat mengandung sebuah definisi ADT lain. Misalnya :

3. ADT waktu yang terdiri dari ADT JAM dan ADT DATE
4. Garis terdiri dari duah buah ADT POINT

SEGI4 yang terdiri dari pasangan dua buah POINT (*Top,Left*) dan (*Bottom,Right*)

TYPE diterjemahkan menjadi *type* terdefinisi dalam bahasa yang bersangkutan. Jika dalam bahasa C menggunakan struct PRIMITIF, dalam konteks prosedural, diterjemahkan menjadi fungsi atau prosedur. PRIMITIF dikelompokan menjadi:

1. Konstruktor/Kreator, pemebentuk nilai *type*. Semua objek (variabel) bertipe tersebut harus melalui konstruktor. Biasanya namanya diawali Make.
2. *Selector*, untuk mengakses tipe komponen(biasanya namanya diawali Get).
3. Prosedur pengubah nilai komponen (biasanya namanya diawali Set).
4. Tipe validator komponen, yang dipakai untuk mentest apakah dapat membentuk tipe sesuai dengan batasan.
5. Destruktor/Dealokator yaitu untuk “menghancurkan” nilai objek/variabel (sekalgus memori penyimpanannya).
6. Baca/Tulis, untuk interface dengan *input/output* device.
7. Operator relasional, terhadap tipe tersebut untuk mendefinisikan lebih besar, lebih kecil, sama dengan dan sebagainya.
8. Aritmatika terhadap tipe tersebut, karena biasanya aritmatika dalam bahasa C hanya terdefinisi untuk bilangan numerik.
9. Konversi dari tipe tersebut ke tipe dasar dan sebaliknya.

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. Untuk memahami lebih jelas mengenai konsep ADT, perhatikan ilustrasi berikut.

Algoritma	C++
Program coba_ADT  Type mahasiswa < nim : char[10] nilai1, nilai2 : integer  Kamus mhs : mahasiswa  procedure inputMhs(input/output m : mahasiswa ) function rata2(input: m : mahasiswa) : real  Algoritma inputMhs(mhs) output( rata2(mhs) )  procedure inputMhs(input/output m : mahasiswa ) kamus algoritma input(m.nim, m.nilai1, m.nilai2)  function rata2(input: m : mahasiswa) : real kamus algoritma → (m.nilai1 + m.nilai1) / 2	<pre> #include &lt;iostream&gt; #include &lt;conio.h&gt; #include &lt;stdlib.h&gt;  using namespace std;  struct mahasiswa{     char nim[10];     int nilai1, nilai2; };  void inputMhs(mahasiswa &amp;m); float rata2(mahasiswa m);  int main() {     mahasiswa mhs;     inputMhs(mhs);     cout &lt;&lt; "rata-rata = " &lt;&lt; rata2(mhs);     return 0; }  void inputMhs(mahasiswa &amp;m){     cout &lt;&lt; "input nama = ";     cin &gt;&gt; (m).nim;     cout &lt;&lt; "input nilai = ";     cin &gt;&gt; (m).nilai1;     cout &lt;&lt; "input nilai2 = ";     cin &gt;&gt; (m).nilai2; }  float rata2(mahasiswa m){     return (m.nilai1+m.nilai2)/2; } </pre> <div style="position: absolute; top: 220px; right: 100px; border: 1px solid black; padding: 5px;">       Definisi/        Spesifikasi Type        dan Primitif /        Header fungsi        (&amp;.h)     </div> <div style="position: absolute; top: 440px; right: 100px; border: 1px solid black; padding: 5px;">       Body/ relisasi        dari primitif        (&amp;.c)     </div>

Untuk menerapkan konsep ADT, kita harus memisah deklarasi tipe, variabel, dan fungsi dari program ke dalam sebuah file.h dan memisah definisi fungsi dari program ke sebuah file.cpp. Sehingga jika kita menerapkan konsep ADT berdasarkan contoh program di atas, bentuk code program akan dipisah menjadi seperti berikut.

Algoritma	C++
Program coba_ADT  Type mahasiswa < nim : char[10] nilai1, nilai2 : integer  Kamus mhs : mahasiswa  procedure inputMhs(i/o m : mahasiswa ) function rata2(input: m : mahasiswa) : real  Algoritma inputMhs(mhs) output( rata2(mhs) ) procedure inputMhs(input/output m : mahasiswa ) kamus algoritma input(m.nim, m.nilai1, m.nilai2)	mahasiswa.h
	<pre> #ifndef MAHASISWA_H_INCLUDED #define MAHASISWA_H_INCLUDED struct mahasiswa{     char nim[10];     int nilai1, nilai2; };  void inputMhs(mahasiswa &amp;m); float rata2(mahasiswa m); #endif // MAHASISWA_H_INCLUDED </pre>
	mahasiswa.cpp
	<pre> void inputMhs(mahasiswa &amp;m){     cout &lt;&lt; "input nama = ";     cin &gt;&gt; (m).nim;     cout &lt;&lt; "input nilai = ";     cin &gt;&gt; (m).nilai1;     cout &lt;&lt; "input nilai2 = ";     cin &gt;&gt; (m).nilai2; } </pre>

<pre>function rata2(input:   m : mahasiswa) : real kamus   algoritma   → (m.nilai1 + m.nilai2) / 2</pre>	<pre>float rata2(mahasiswa m){   return (m.nilai1+m.nilai2)/2; }</pre> <p style="text-align: center;"><b>main.cpp</b></p> <pre>#include &lt;iostream&gt; #include &lt;conio.h&gt; #include &lt;stdlib.h&gt; #include "mahasiswa.cpp"  using namespace std;  int main() {   mahasiswa mhs;   inputMhs(mhs);   cout &lt;&lt; "rata-rata = " &lt;&lt; rata2(mhs);   return 0; }</pre>
--	--

### 3. Guided

```
main.cpp X
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
15     mahasiswa mhs;
16     inputMhs(mhs);
17     cout << "rata-rata = " << rata2(mhs);
18     return 0;
19 }
20
21
22 void inputMhs(mahasiswa &m){
23     cout << "input nim = ";
24     cin >> (m).nim;
25     cout << "input nilai = ";
26     cin >> (m).nilai1;
27     cout << "input nilai = ";
28     cin >> (m).nilai2;
29 }
30
31 float rata2(mahasiswa m){
32     return (m.nilai1+m.nilai2)/2;
33 }
34
```

## Output

```
"C:\codeblocks file\pertemuan4\bin\Debug\pertemuan4.exe"  
input nim = 2311104074  
input nilai = 95  
input nilai = 98  
rata-rata = 96  
Process returned 0 (0x0)   execution time : 16.210 s  
Press any key to continue.
```

### Penjelasan Program

#### **Struct mahasiswa:**

- Struct ini adalah tipe data yang menyimpan beberapa informasi terkait mahasiswa, yaitu NIM (nomor induk mahasiswa) dan dua nilai (nilai1 dan nilai2).
- char nim[10]: menyimpan NIM sebagai array dari karakter.
- int nilai1, nilai2: menyimpan dua nilai angka dari mahasiswa.

#### **Fungsi inputMhs(mahasiswa &m):**

- Fungsi ini digunakan untuk menerima input dari user berupa NIM dan dua nilai.
- Simbol & menandakan bahwa argumen yang diterima oleh fungsi ini adalah referensi dari variabel mahasiswa, sehingga perubahan yang dilakukan dalam fungsi ini juga memengaruhi variabel aslinya.
- Fungsi ini meminta pengguna untuk memasukkan NIM, nilai pertama, dan nilai kedua.

#### **Fungsi rata2(mahasiswa m):**

- Fungsi ini digunakan untuk menghitung rata-rata dari dua nilai yang dimiliki oleh mahasiswa.
- Mengembalikan hasil perhitungan rata-rata dari nilai1 dan nilai2.

#### **Fungsi main():**

- Fungsi utama program.
- Membuat sebuah variabel mhs bertipe mahasiswa.
- Memanggil fungsi inputMhs() untuk meminta input data mahasiswa.
- Memanggil fungsi rata2() untuk menghitung dan menampilkan rata-rata dua nilai mahasiswa.

## 4. Unguided

### 1.

```

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

## Output

"C:\codeblocks file\UNGUIDEDMODUL3\bin\Debug\UNGUIDEDMODUL3.exe"

Masukkan jumlah mahasiswa (maksimal 10): 2

Data Mahasiswa ke-1

Masukkan Nama: alvin

Masukkan NIM: 2311104070

Masukkan Nilai UTS: 10

Masukkan Nilai UAS: 5

Masukkan Nilai Tugas: 7

Data Mahasiswa ke-2

Masukkan Nama: Ramdan

Masukkan NIM: 2311104066

Masukkan Nilai UTS: 6

Masukkan Nilai UAS: 20

Masukkan Nilai Tugas: 6

Data Mahasiswa yang telah dimasukkan:

Mahasiswa ke-1

Nama: alvin

NIM: 2311104070

Nilai UTS: 10

Nilai UAS: 5

Nilai Tugas: 7

Nilai Akhir: 7.1

Mahasiswa ke-2

Nama: Ramdan

NIM: 2311104066

Nilai UTS: 6

Nilai UAS: 20

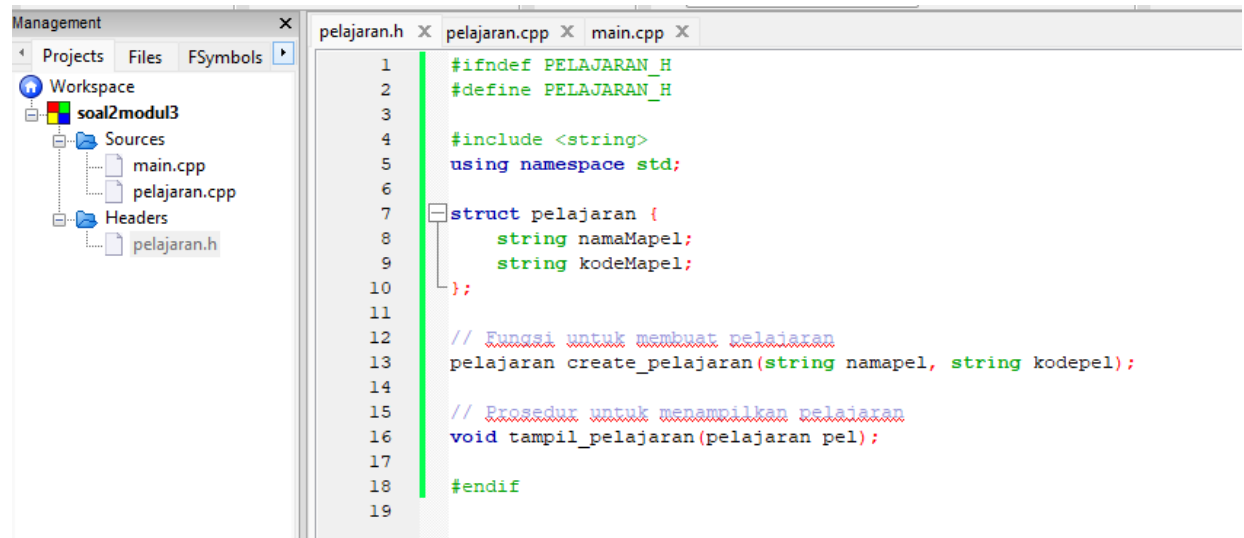
Nilai Tugas: 6

Nilai Akhir: 11.6

Process returned 0 (0x0) execution time : 37.613 s

Press any key to continue.

## 2. pelajaran.h

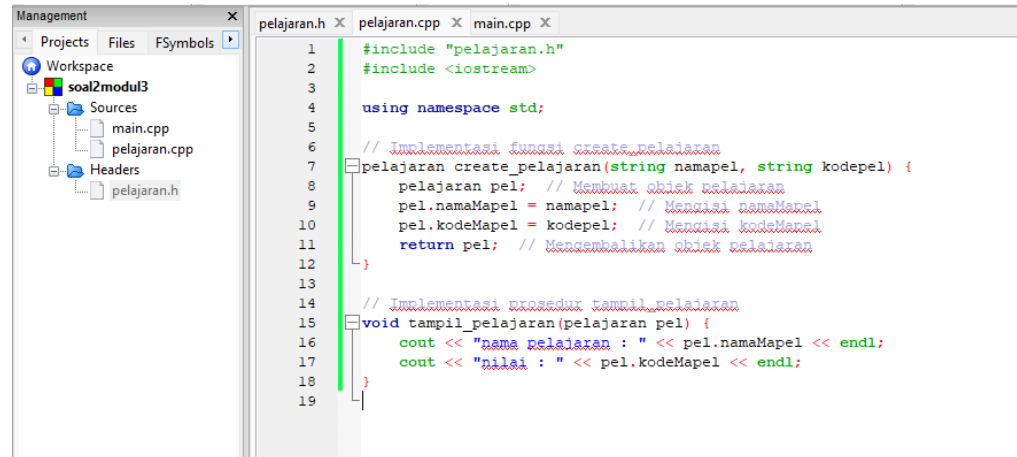


```

1  #ifndef PELAJARAN_H
2  #define PELAJARAN_H
3
4  #include <string>
5  using namespace std;
6
7  struct pelajaran {
8      string namaMapel;
9      string kodeMapel;
10 };
11
12 // Fungsi untuk membuat pelajaran
13 pelajaran create_pelajaran(string namapel, string kodepel);
14
15 // Prosedur untuk menampilkan pelajaran
16 void tampil_pelajaran(pelajaran pel);
17
18 #endif
19

```

## Pelajaran.cpp

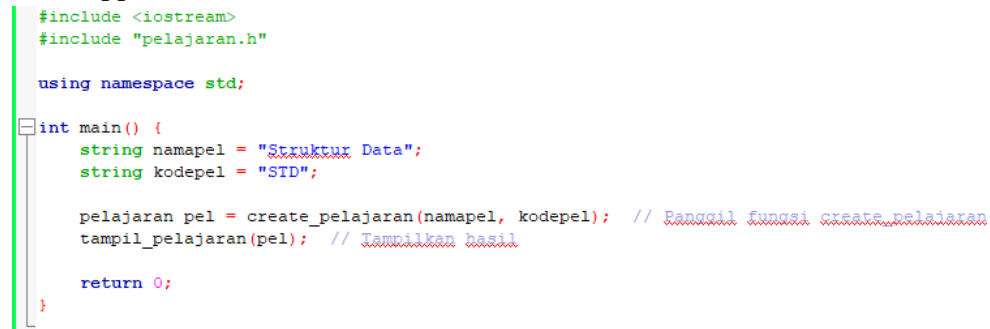


```

1  #include "pelajaran.h"
2  #include <iostream>
3
4  using namespace std;
5
6  // Implementasi fungsi create_pelajaran
7  pelajaran create_pelajaran(string namapel, string kodepel) {
8      pelajaran pel; // Membuat objek pelajaran
9      pel.namaMapel = namapel; // Mengisi namaMapel
10     pel.kodeMapel = kodepel; // Mengisi kodeMapel
11     return pel; // Mengembalikan objek pelajaran
12 }
13
14 // Implementasi prosedur tampil_pelajaran
15 void tampil_pelajaran(pelajaran pel) {
16     cout << "nama pelajaran : " << pel.namaMapel << endl;
17     cout << "nilai : " << pel.kodeMapel << endl;
18 }
19

```

## Main.cpp



```

1  #include <iostream>
2  #include "pelajaran.h"
3
4  using namespace std;
5
6  int main() {
7      string namapel = "Struktur Data";
8      string kodepel = "STD";
9
10     pelajaran pel = create_pelajaran(namapel, kodepel); // Panggil fungsi create_pelajaran
11     tampil_pelajaran(pel); // Tampilkan hasil
12
13     return 0;
14 }

```

## Output

```

C:\codeblocks\file\soal2modul3>program.exe
nama pelajaran : Struktur Data
nilai : STD

```



### 3.

```

main.cpp x
1  #include <iostream>
2
3  using namespace std;
4
5
6  void tampilkanArray2D(int array[3][3]) {
7      for (int i = 0; i < 3; i++) {
8          for (int j = 0; j < 3; j++) {
9              cout << array[i][j] << " ";
10             }
11             cout << endl;
12         }
13     }
14
15
16 void tukarArray2D(int array1[3][3], int array2[3][3], int baris, int kolom) {
17
18     if (baris >= 0 && baris < 3 && kolom >= 0 && kolom < 3) {
19         int temp = array1[baris][kolom];
20         array1[baris][kolom] = array2[baris][kolom];
21         array2[baris][kolom] = temp;
22     } else {
23         cout << "Indeks di luar batas!" << endl;
24     }
25 }
26
27
28 void tukarPointer(int *ptr1, int *ptr2) {
29     int temp = *ptr1;
30     *ptr1 = *ptr2;
31     *ptr2 = temp;
32 }
33
34 int main() {
35
36     int array1[3][3] = {
37         {1, 2, 3},
38         {4, 5, 6},
39         {7, 8, 9}
40     };
41
42     int array2[3][3] = {
43         {9, 8, 7},
44         {6, 5, 4},
45         {3, 2, 1}
46     };
47
48
49     cout << "Isi Array 1:" << endl;
50     tampilkanArray2D(array1);
51
52     cout << "\nIsi Array 2:" << endl;
53     tampilkanArray2D(array2);
54
55
56     int baris = 1, kolom = 1;
57     cout << "\nMenukar posisi (" << baris << ", " << kolom << ") antara Array 1 dan Array 2." << endl;
58     tukarArray2D(array1, array2, baris, kolom);
59
60
61     cout << "\nIsi Array 1 setelah penukaran:" << endl;
62     tampilkanArray2D(array1);
63
64     cout << "\nIsi Array 2 setelah penukaran:" << endl;
65     tampilkanArray2D(array2);
66
67
68     int a = 10, b = 20;
69     int *ptr1 = &a;
70     int *ptr2 = &b;
71
72     cout << "\nSebelum penukaran: a = " << a << ", b = " << b << endl;
73     tukarPointer(ptr1, ptr2);
74     cout << "Setelah penukaran: a = " << a << ", b = " << b << endl;
75
76     return 0;
77 }
78
  
```

## Output

```
"C:\codeblocks file\soal3modul3\bin\Debug\soal3modul3.exe"
Isi Array 1:
1 2 3
4 5 6
7 8 9

Isi Array 2:
9 8 7
6 5 4
3 2 1

Menukar posisi (1,1) antara Array 1 dan Array 2.

Isi Array 1 setelah penukaran:
1 2 3
4 5 6
7 8 9

Isi Array 2 setelah penukaran:
9 8 7
6 5 4
3 2 1

Sebelum penukaran: a = 10, b = 20
Setelah penukaran: a = 20, b = 10

Process returned 0 (0x0)   execution time : 0.060 s
Press any key to continue.
```

## 5. Kesimpulan

Secara keseluruhan, praktikum ini bertujuan untuk memberikan pemahaman dan keterampilan dalam menggunakan ADT untuk membuat program yang lebih baik. Terima kasih asprak semoga kita semua sukses