

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



**Disusun Oleh:**  
**Berlian Seva Astryana -2311104067**  
**S1SE-07-B**

**Dosen :**  
**Arief Rais Bahtiar, S.Kom., M.Kom**

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

## 1. Tujuan

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

## 2. Landasan Teori

### 3.1 Pengertian

Abstract Data Type (ADT) adalah tipe data yang mendefinisikan oleh sekumpulan operasi dasar (primitif) yang dapat dilakukan pada tipe tersebut. ADT menyajikan cara untuk mengabstraksi data dan operasi yang dapat dilakukan, memisahkan antara implementasi dan penggunaan. Dalam ADT terdapat definisi invariant dari tipe dan aksioma yang berlaku, menjadikannya sebagai definisi statis. Contoh ADT yang umum meliputi struktur data seperti tumpukan (stack), antrian (queue) dan graf.

### 3.2 Struktur ADT

- a. Tipe dari sebuah ADT mencakup definisi ADT lain. Misalnya, ADT waktu terdiri dari ADT jam dan tanggal.
- b. Pimitif dalam ADT dibagi menjadi beberapa kategori:
  1. Konstruktor
  2. Selector
  3. Prosedur mengubah
  4. Validator
  5. Baca/Tulis
  6. Destruktor/dealokator
  7. Operasional relasional
  8. Aritmatika
  9. Konversi

## 3. Guided

```
main.cpp x
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 }
28
29 float rata2(mahasiswa m){
30     return(m.nilai1+m.nilai2)/2;
31 }
32

```

Program C++ di atas mendefinisikan sebuah struktur data bernama “mahasiswa” yang digunakan untuk menyimpan informasi tentang seorang mahasiswa, termasuk NIM dan dua nilai akademik. Program dimulai dengan mengimpor Pustaka input/output standar melalui “#include <iostream>” dan menggunakan namespace “std” untuk memudahkan akses ke elemen-elemen standar. Di dalam fungsi “inputMhs” dipanggil untuk meminta pengguna memasukkan NIM serta dua nilai mahasiswa, yang kemudian disimpan dalam variable “mhs”. Fungsi “inputMhs” menerima parameter referensi dari tipe “mahasiswa”, sehingga setiap perubahan yang dilakukan pada parameter tersebut akan mempengaruhi objek asli. Setelah data diinput, program menghitung rata rata dari kedua nilai dengan memanggil fungsi “rata2” yang menerima objek “mahasiswa” sebagai parameter dan mengembalikan hasil perhitungan rata rata nilai. Hasil rata-rata ini kemudian ditampilkan ke layar. Program ini menunjukkan cara sederhana untuk menggunakan struktur data dan fungsi dalam C++ untuk mengelola data mahasiswa dan melakukan perhitungan dasar. Output yang dihasilkan dari program tersebut:

```

input nim = 2311104067
input nilai = 98
input nilai = 95
rata rata = 96

```

#### 4. Unguided

- 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  $0.3 \cdot uts + 0.4 \cdot uas + 0.3 \cdot tugas$ .
- 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;
}
```

Pelajaran.h :

```
1  #ifndef PELAJARAN_H_INCLUDED
2  #define PELAJARAN_H_INCLUDED
3
4  #include <string>
5  #include <iostream>
6
7  using namespace std;
8
9  // Definisi tipe pelajaran
10 struct pelajaran {
11     string namaMapel;
12     string kodeMapel;
13 };
14
15 // Fungsi untuk membuat objek pelajaran
16 pelajaran create_pelajaran(string namapel, string kodepel);
17
18 // Prosedur untuk menampilkan informasi pelajaran
19 void tampil_pelajaran(pelajaran pel);
20
21 #endif // PELAJARAN_H_INCLUDED
```

pelajaran.cpp:

```
1  #include "pelajaran.h"
2
3  // Implementasi fungsi create pelajaran
4  pelajaran create_pelajaran(string namapel, string kodepel) {
5      pelajaran pel;
6      pel.namaMapel = namapel;
7      pel.kodeMapel = kodepel;
8      return pel;
9  }
10
11 // Implementasi prosedur tampil pelajaran
12 void tampil_pelajaran(pelajaran pel) {
13     cout << "Nama Mata Pelajaran: " << pel.namaMapel << endl;
14     cout << "Kode Mata Pelajaran: " << pel.kodeMapel << endl;
15 }
16
```

main.cpp:

```
#include <iostream>
#include "pelajaran.h"
#include <string>

using namespace std;

int main() {
    string namapel = "Struktur Data";
    string kodepel = "STD";

    // Membuat objek pelajaran menggunakan create_pelajaran
    pelajaran pel = create_pelajaran(namapel, kodepel);

    // Menampilkan informasi pelajaran
    tampil_pelajaran(pel);

    return 0;
}
```

c. Buatlah program dengan ketentuan :

1. 2 buah array 2D integer berukuran 3x3 dan 2 buah pointer integer
2. fungsi/prosedur yang menampilkan isi sebuah array integer 2D
3. fungsi/prosedur yang akan menukarkan isi dari 2 array integer 2D pada posisi tertentu
4. fungsi/prosedur yang akan menukarkan isi dari variabel yang ditunjuk oleh 2 buah pointer

```
1  #include <iostream>
2
3  using namespace std;
4
5  void tampilkanArray(int arr[3][3]) {
6      for (int i = 0; i < 3; i++) {
7          for (int j = 0; j < 3; j++) {
8              cout << arr[i][j] << " ";
9          }
10         cout << endl;
11     }
12 }
13
14 void tukarArray(int arr1[3][3], int arr2[3][3], int posisi[2]) {
15     int temp = arr1[posisi[0]][posisi[1]];
16     arr1[posisi[0]][posisi[1]] = arr2[posisi[0]][posisi[1]];
17     arr2[posisi[0]][posisi[1]] = temp;
18 }
19
20 void tukarPointer(int *ptr1, int *ptr2) {
21     int temp = *ptr1;
22     *ptr1 = *ptr2;
23     *ptr2 = temp;
24 }
```

```

26 int main() {
27
28     int arr1[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
29     int arr2[3][3] = {{10, 11, 12}, {13, 14, 15}, {16, 17, 18}};
30
31     int var1 = 20;
32     int var2 = 30;
33     int *ptr1 = &var1;
34     int *ptr2 = &var2;
35
36     cout << "Isi array 1:" << endl;
37     tampilkanArray(arr1);
38     cout << "Isi array 2:" << endl;
39     tampilkanArray(arr2);
40
41     int posisi[2] = {1, 1};
42     tukarArray(arr1, arr2, posisi);
43
44     cout << "Isi array 1 setelah ditukar:" << endl;
45     tampilkanArray(arr1);
46     cout << "Isi array 2 setelah ditukar:" << endl;
47     tampilkanArray(arr2);
48
49     tukarPointer(ptr1, ptr2);
50
51     cout << "Isi var1 setelah ditukar: " << *ptr1 << endl;
52     cout << "Isi var2 setelah ditukar: " << *ptr2 << endl;
53
54     return 0;
55 }
56

```

Output:

```

Isi array 1:
1 2 3
4 5 6
7 8 9
Isi array 2:
10 11 12
13 14 15
16 17 18
Isi array 1 setelah ditukar:
1 2 3
4 14 6
7 8 9
Isi array 2 setelah ditukar:
10 11 12
13 5 15
16 17 18
Isi var1 setelah ditukar: 30
Isi var2 setelah ditukar: 20

```

## 5. Kesimpulan

ADT merupakan alat penting dalam pemrograman yang memungkinkan pengembangan untuk membuat tipe data baru dengan operasi yang terdefinisi dengan baik. Dengan memisahkan antara antarmuka dan implementasi, ADT memberikan fleksibilitas dan modularitas dalam pengembangan perangkat lunak.