

**LAPORAN PRAKTIKUM STRUKTUR DATA**

**PERTEMUAN 3**

**ABSTRACT DATA TYPE**



**Nama :**

Reyner Atira Prasetyo (2311104057)

**Dosen :**

Wahyu Andi Saputra S.pd, M.Eng

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

## I. TUJUAN

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

## II. TOOL

1. VSCode
2. GCC/G++/GDB

## III. DASAR TEORI

ADT (Abstract Data Type) adalah tipe data beserta sekumpulan operasi dasar (primitif) yang mendefinisikan manipulasi data tersebut. Setiap ADT memiliki invarian tipe dan aksioma yang mengatur hubungan antar komponennya. Definisi ADT bersifat statis, namun implementasinya bisa memuat ADT lain (seperti ADT `Waktu` yang terdiri dari ADT `Jam` dan `Tanggal`, atau ADT `Segi4` yang terdiri dari dua ADT `Point`).

Primitif dalam ADT dikelompokkan menjadi beberapa jenis:

1. **Konstruktor/Kreator:** Membentuk nilai awal dari tipe (dengan awalan `Make`).
2. **Selector:** Mengakses komponen tipe (dengan awalan `Get`).
3. **Modifier:** Mengubah nilai komponen (juga menggunakan `Get` dalam penamaan).
4. **Validator:** Memeriksa validitas nilai tipe sesuai batasan.
5. **Destruktor:** Menghapus nilai objek dan memori terkait.
6. **Baca/Tulis:** Berfungsi untuk input/output.
7. **Operator Relasional:** Mendefinisikan perbandingan antar objek ADT.
8. **Aritmatika:** Operasi matematika terhadap tipe yang bersangkutan.
9. **Konversi Tipe:** Mengonversi ADT ke tipe dasar atau sebaliknya.

ADT diimplementasikan dalam dua modul:

1. **Header (Spesifikasi Tipe dan Primitif):** Mendefinisikan tipe data dan primitif (fungsi, prosedur) dengan domain, range, serta kondisi awal/akhir.
2. **Body (Realisasi Primitif):** Implementasi kode yang memanfaatkan selector dan konstruktor untuk merealisasikan fungsi dan prosedur yang telah didefinisikan.

Driver program utama akan memanfaatkan kedua modul tersebut untuk menjalankan operasi ADT.

## IV. GUIDED

1. `guided.cpp`

```

1  #include<iostream>
2  #include<conio.h>
3  #include<stdlib.h>
4
5  using namespace std;
6
7  struct mahasiswa
8  {
9      char nim[10];
10     int nilai1, nilai2;
11 };
12
13 void inputMhs(mahasiswa &m);
14 float rata2(mahasiswa m);
15
16 int main(){
17     mahasiswa mhs;
18
19     inputMhs(mhs);
20     cout << "rata-rata = " << rata2(mhs);
21     return 0;
22 }
23
24 void inputMhs(mahasiswa &m){
25     cout << "Input nim = ";
26     cin >> (m).nim;
27     cout << "Input nilai = ";
28     cin >> (m).nilai1;
29     cout << "Input nilai = ";
30     cin >> (m).nilai2;
31 }
32
33 float rata2(mahasiswa m){
34     return(m.nilai1+m.nilai2/2);
35 }

```

Output :

```

--pid=Microsoft-MIEngine-Pid-hs014xze.pr4' --dbg
Input nim = 2311104000
Input nilai = 80
Input nilai = 79
rata-rata = 119
PS D:\PRAKTIKUM\Struktur Data\pertemuan3\guided>

```

2. mahasiswa.h

```

1  #ifndef MAHASISWA_H_INCLUDED
2  #define MAHASISWA_H_INCLUDED
3  struct mahasiswa{
4      char nim[10];
5      int nilai1, nilai2;
6  };
7
8  void inputMhs(mahasiswa &m);
9  float rata2(mahasiswa m);
10 #endif // MAHASISWA_H_INCLUDED

```

3. mahasiswa.cpp

```

1  #include<iostream>
2  #include "mahasiswa.h"
3
4  using namespace std;
5
6  void inputMhs(mahasiswa &m){
7      cout << "input nama =";
8      cin >> (m).nim;
9      cout << "input nilai =";
10     cin >> (m).nilai1;
11     cout << "input nilai2 =";
12     cin >> (m).nilai2;
13 }
14
15 float rata2(mahasiswa m){
16     return (m.nilai1+m.nilai2)/2;
17 }

```

#### 4. main.cpp

```

1  #include <iostream>
2  #include <conio.h>
3  #include <stdlib.h>
4  #include "mahasiswa.cpp"
5
6  using namespace std;
7
8  int main()
9  {
10     mahasiswa mhs;
11     inputMhs(mhs);
12     cout << "rata-rata = " << rata2(mhs); return 0;
13 }

```

Output :

```

--pid=Microsoft-MIEngine-Pid-c3cmdcv.kx1' '--dbgE
input nama =Rey
input nilai =88
input nilai2 =73
rata-rata = 80
PS D:\PRAKTIKUM\Struktur Data\pertemuan3\guided>

```

## V. UNGUIDED

1. Program menyimpan mahasiswa
  - a. mahasiswa.h

```

1  #ifndef MAHASISWA_H
2  #define MAHASISWA_H
3
4  #include <string>
5  using namespace std;
6
7  // Struktur data mahasiswa
8  struct Mahasiswa {
9      string nama;
10     string nim;
11     float uts;
12     float uas;
13     float tugas;
14     float nilaiAkhir;
15 };
16
17 // Deklarasi fungsi untuk menghitung nilai akhir
18 float hitungNilaiAkhir(float uts, float uas, float tugas);
19
20 #endif

```

## b. mahasiswa.cpp

```

1  #include <iostream>
2  #include "mahasiswa.h"
3
4  using namespace std;
5
6  float hitungNilaiAkhir(float uts, float uas, float tugas) {
7      return (0.3 * uts) + (0.4 * uas) + (0.3 * tugas);
8  }
9
10 int main() {
11     Mahasiswa mahasiswa[10];
12     int jumlahMahasiswa;
13
14     cout << "Masukkan jumlah mahasiswa (maksimal 10): ";
15     cin >> jumlahMahasiswa;
16
17     for (int i = 0; i < jumlahMahasiswa; i++) {
18         cout << "\nMasukkan data mahasiswa ke-" << i+1 << endl;
19         cout << "Nama: ";
20         cin.ignore();
21         getline(cin, mahasiswa[i].nama);
22         cout << "NIM: ";
23         getline(cin, mahasiswa[i].nim);
24         cout << "Nilai UTS: ";
25         cin >> mahasiswa[i].uts;
26         cout << "Nilai UAS: ";
27         cin >> mahasiswa[i].uas;
28         cout << "Nilai tugas: ";
29         cin >> mahasiswa[i].tugas;
30
31         mahasiswa[i].nilaiAkhir = hitungNilaiAkhir(mahasiswa[i].uts, mahasiswa[i].uas, mahasiswa[i].tugas);
32     }
33
34     cout << "\nData Mahasiswa:" << endl;
35     for (int i = 0; i < jumlahMahasiswa; i++) {
36         cout << "\nMahasiswa ke-" << i+1 << endl;
37         cout << "Nama: " << mahasiswa[i].nama << endl;
38         cout << "NIM: " << mahasiswa[i].nim << endl;
39         cout << "Nilai UTS: " << mahasiswa[i].uts << endl;
40         cout << "Nilai UAS: " << mahasiswa[i].uas << endl;
41         cout << "Nilai tugas: " << mahasiswa[i].tugas << endl;
42         cout << "Nilai Akhir: " << mahasiswa[i].nilaiAkhir << endl;
43     }
44
45     return 0;
46 }
47

```

Output :

Masukkan jumlah mahasiswa (maksimal 10): 4

Masukkan data mahasiswa ke-1

Nama: Rusdi

NIM: Ngawi

Nilai UTS: 90

Nilai UAS: 87

Nilai Tugas: 78

Masukkan data mahasiswa ke-2

Nama: Fuad

NIM: Menggoda

Nilai UTS: 78

Nilai UAS: 84

Nilai Tugas: 95

Masukkan data mahasiswa ke-3

Nama: Amba

NIM: Tukam

Nilai UTS: 90

Nilai UAS: 95

Nilai Tugas: 98

Masukkan data mahasiswa ke-4

Nama: Sigit

NIM: Rendang

Nilai UTS: 67

Nilai UAS: 71

Nilai Tugas: 88

```
Data Mahasiswa:

Mahasiswa ke-1
Nama: Rusdi
NIM: Ngawi
Nilai UTS: 90
Nilai UAS: 87
Nilai Tugas: 78
Nilai Akhir: 85.2

Mahasiswa ke-2
Nama: Fuad
NIM: Menggoda
Nilai UTS: 78
Nilai UAS: 84
Nilai Tugas: 95
Nilai Akhir: 85.5

Mahasiswa ke-3
Nama: Amba
NIM: Tukam
Nilai UTS: 90
Nilai UAS: 95
Nilai Tugas: 98
Nilai Akhir: 94.4

Mahasiswa ke-4
Nama: Sigit
NIM: Rendang
Nilai UTS: 67
Nilai UAS: 71
Nilai Tugas: 88
Nilai Akhir: 74.9
PS D:\PRAKTIKUM\Struktur Data\pertemuan3\unguided>
```

## 2. Program Pelajaran

### a. 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 pelajaran create_pelajaran(string namapel, string kodepel);
13
14 void tampil_pelajaran(pelajaran pel);
15
16 #endif
17
```

b. pelajaran.cpp

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

c. main.cpp

```
1 #include <iostream>
2 #include "pelajaran.h"
3 #include "pelajaran.cpp"
4
5 using namespace std;
6
7 int main() {
8     string namapel = "Struktur Data";
9     string kodepel = "STD";
10
11     pelajaran pel = create_pelajaran(namapel, kodepel);
12
13     tampil_pelajaran(pel);
14
15     return 0;
16 }
17
```

Output :

```
Microsoft Windows [Version 10.0.17134.0]
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Users\user> cd D:\PRAKTIKUM\Struktur Data\pertemuan3\unguided
C:\Users\user> g++ main.cpp pelajaran.cpp
C:\Users\user> .\main.exe
Nama Mata Pelajaran: Struktur Data
Kode Mata Pelajaran: STD
PS D:\PRAKTIKUM\Struktur Data\pertemuan3\unguided>
```

3. Program tukar isi array dan pointer

a. fungsi.h



```

1  #ifndef FUNGSI_H
2  #define FUNGSI_H
3
4  void tampilArray(int arr[3][3]);
5
6  void tukarIsiArray(int arr1[3][3], int arr2[3][3], int row, int col);
7
8  void tukarIsiPointer(int* p1, int* p2);
9
10 #endif
11

```

b. fungsi.cpp

```

1  #include <iostream>
2  #include "fungsi.h"
3
4  using namespace std;
5
6  void tampilArray(int arr[3][3]){
7      for (int i = 0; i < 3; i++)
8      {
9          for (int j = 0; j < 3; j++)
10         {
11             cout << arr [i][j] << " ";
12         }
13         cout << endl;
14     }
15     cout << endl;
16 }
17
18 void tukarIsiArray(int arr1[3][3], int arr2[3][3], int row, int col) {
19     int temp = arr1[row][col];
20     arr1[row][col] = arr2[row][col];
21     arr2[row][col] = temp;
22 }
23
24 void tukarIsiPointer(int* p1, int* p2){
25     int temp = *p1;
26     *p1 = *p2;
27     *p2 = temp;
28 }

```

c. main.cpp

```
1 #include <iostream>
2 #include "fungsi.h"
3 #include "fungsi.cpp"
4
5 using namespace std;
6
7 int main() {
8     int array1[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
9     int array2[3][3] = {{9, 8, 7}, {6, 5, 4}, {3, 2, 1}};
10
11     int a = 10, b = 20;
12     int* ptr1 = &a;
13     int* ptr2 = &b;
14
15     cout << "Array 1 sebelum tukar: " << endl;
16     tampilArray(array1);
17
18     cout << "Array 2 sebelum tukar: " << endl;
19     tampilArray(array2);
20
21     int row = 1, col = 1;
22     tukarIsiArray(array1, array2, row, col);
23
24     cout << "Array 1 setelah tukar posisi (" << row << ", " << col << "):" << endl;
25     tampilArray(array1);
26
27     cout << "Array 2 setelah tukar posisi (" << row << ", " << col << "):" << endl;
28     tampilArray(array2);
29
30     cout << "Nilai sebelum tukar pointer:" << endl;
31     cout << "Pointer 1 (nilai a): " << *ptr1 << endl;
32     cout << "Pointer 2 (nilai b): " << *ptr2 << endl;
33
34     tukarIsiPointer(ptr1, ptr2);
35
36     cout << "Nilai setelah tukar pointer:" << endl;
37     cout << "Pointer 1 (nilai a): " << *ptr1 << endl;
38     cout << "Pointer 2 (nilai b): " << *ptr2 << endl;
39
40     return 0;
41 }
```

Output :

```

Array 1 sebelum tukar:
1 2 3
4 5 6
7 8 9

Array 2 sebelum tukar:
9 8 7
6 5 4
3 2 1

Array 1 setelah tukar posisi (1,1):
1 2 3
4 5 6
7 8 9

Array 2 setelah tukar posisi (1,1):
9 8 7
6 5 4
3 2 1

Nilai sebelum tukar pointer:
Pointer 1 (nilai a): 10
Pointer 2 (nilai b): 20
Nilai setelah tukar pointer:
Pointer 1 (nilai a): 20
Pointer 2 (nilai b): 10
PS D:\PRAKTIKUM\Struktur Data\pertemuan3\unguided>

```

## VI. KESIMPULAN

Praktikum ini berhasil menunjukkan penerapan Abstract Data Type (ADT) dalam C++ untuk mengelola data secara modular dan terstruktur. Dengan memisahkan definisi tipe data di header (.h) dan implementasi fungsi di file .cpp, program menjadi lebih mudah dipahami dan dipelihara. Penggunaan primitif seperti konstruktor, selector, dan modifier mempermudah manipulasi objek ADT. Selain itu, pemanfaatan ADT seperti Point, Segi4, serta operasi penukaran pada array 2D dan pointer menegaskan pentingnya ADT dalam pengorganisasian data yang efisien.