

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



Disusun Oleh:
Muhammad Ralfi - 2211104054
SE-07-2

Dosen :
Wahyu andi

PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY
PURWOKERTO
2024

1. Tujuan

- a. Mengetahui apa itu ADT (abstrak data type).
- b. Memahami konsep Abstract Data Type (ADT) dan penggunaannya dalam pemrograman.
- c. Bisa mengimplementasikan ADT dalam pemrograman.

2. Landasan Teori

Tipe Data Abstrak (ADT) adalah tipe (atau kelas) untuk objek yang perilakunya ditentukan oleh serangkaian nilai dan serangkaian operasi. Definisi ADT hanya menyebutkan operasi apa yang akan dilakukan tetapi tidak menyebutkan bagaimana operasi ini akan diimplementasikan. Definisi ini tidak menentukan bagaimana data akan diatur dalam memori dan algoritma apa yang akan digunakan untuk mengimplementasikan operasi. Disebut "abstrak" karena memberikan tampilan yang tidak bergantung pada implementasi.

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

- a. ADT waktu yang terdiri dari ADT JAM dan ADT DATE
- b. Garis terdiri dari dua buah ADT POINT

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 dikelompokkan menjadi:

- a. Konstruktor/Kreator, pembentuk nilai type. Semua objek (variabel) bertipe tersebut harus melalui konstruktor. Biasanya namanya diawali Make.
- b. Selector, untuk mengakses tipe komponen (biasanya namanya diawali Get).
- c. Prosedur pengubah nilai komponen (biasanya namanya diawali Set).
- d. Tipe validator komponen, yang dipakai untuk mentest apakah dapat membentuk tipe sesuai dengan batasan.
- e. Destruktor/Dealokator yaitu untuk "menghancurkan" nilai objek/variabel (sekali-gus memori penyimpanannya).
- f. Baca/Tulis, untuk interface dengan input/output device.
- g. Operator relasional, terhadap tipe tersebut untuk mendefinisikan lebih besar, lebih kecil, sama dengan dan sebagainya.
- h. Aritmatika terhadap tipe tersebut, karena biasanya aritmatika dalam bahasa C hanya terdefinisi untuk bilangan numerik.
- i. Konversi dari tipe tersebut ke tipe dasar dan sebaliknya.

3. Guided

Belajar cara membuat sebuah tipe data bentukan atau struct dan memanggilnya dengan prosedur.

```
#include <iostream>

using namespace std;

// structure di tulis sebelum int main
struct mahasiswa{
    char nim[10];
    int nilail, nilai2;
};

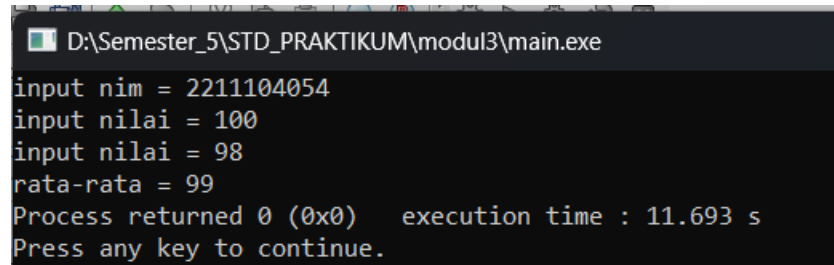
void inputMhs(mahasiswa &m);
float rata2(mahasiswa m);

int main(){
    // panggil struct
    mahasiswa mhs;
    inputMhs(mhs);
    cout << "rata-rata = " << rata2(mhs);
    return 0;
}

// prosedur
void inputMhs(mahasiswa &m){
    cout << "input nim = ";
    cin >> (m).nim; // memanggil structure
    cout << "input nilai = ";
    cin >> (m).nilail;
    cout << "input nilai = ";
    cin >> (m).nilai2;
}

float rata2(mahasiswa m){
    return (m.nilail+m.nilai2)/2;
}
```

Output:



```
D:\Semester_5\STD_PRAKTIKUM\modul3\main.exe
input nim = 2211104054
input nilai = 100
input nilai = 98
rata-rata = 99
Process returned 0 (0x0) execution time : 11.693 s
Press any key to continue.
```

Program meminta user menginputkan nim dan nilai, jumlah inputan nilai dapat disesuaikan didalam code program.

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

Code:

```

struct mahasiswa {
    string nama;
    char nim[10];
    int nilaiUts, nilaiUas, nilaiTugas;
};

void inputMhs(mahasiswa &m);
float nilaiAkhir(mahasiswa m);

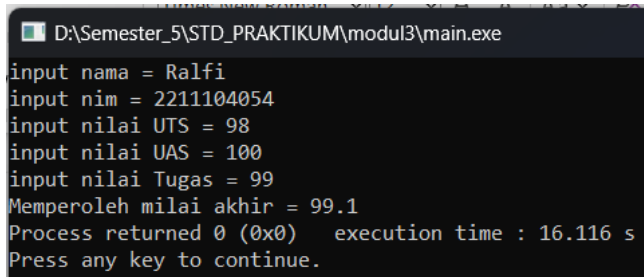
int main(){
    mahasiswa mhs;
    inputMhs(mhs);
    cout << "Memperoleh milai akhir = " << nilaiAkhir(mhs);
    return 0;
}

void inputMhs(mahasiswa &m){
    cout << "input nama = ";
    cin >> (m).nama;
    cout << "input nim = ";
    cin >> (m).nim;
    cout << "input nilai UTS = ";
    cin >> (m).nilaiUts;
    cout << "input nilai UAS = ";
    cin >> (m).nilaiUas;
    cout << "input nilai Tugas = ";
    cin >> (m).nilaiTugas;
}

float nilaiAkhir(mahasiswa m){
    return(0.3*m.nilaiUts+0.4*m.nilaiUas+0.3*m.nilaiTugas);
}

```

Output:



```

D:\Semester_5\STD_PRAKTIKUM\modul3\main.exe
input nama = Ralfi
input nim = 2211104054
input nilai UTS = 98
input nilai UAS = 100
input nilai Tugas = 99
Memperoleh milai akhir = 99.1
Process returned 0 (0x0) execution time : 16.116 s
Press any key to continue.

```

2. Buatlah ADT pelajaran sebagai berikut di dalam file “pelajaran.h”, Buatlah implementasi ADT pelajaran pada file “pelajaran.cpp” Cobalah hasil implementasi ADT pada file “main.cpp”

Pelajaran.h

```

// pelajaran.h
#ifndef PELAJARAN_H
#define PELAJARAN_H

#include <string>
using namespace std;

struct pelajaran {
    string namaMapel;
    string kodeMapel;
};

pelajaran create_pelajaran(string namapel, string kodepel);

void tampil_pelajaran(pelajaran pel);

#endif

```

Pelajaran.cpp

```
// pelajaran.cpp
#include "pelajaran.h"
#include <iostream>

pelajaran create_pelajaran(string namapel, string kodepel) {
    pelajaran pel;
    pel.namaMapel = namapel;
    pel.kodeMapel = kodepel;
    return pel;
}

void tampil_pelajaran(pelajaran pel) {
    cout << "nama pelajaran : " << pel.namaMapel << endl;
    cout << "nilai : " << pel.kodeMapel << endl;
}
}
```

Main.cpp

```
// main.cpp
#include <iostream>
#include "pelajaran.h"

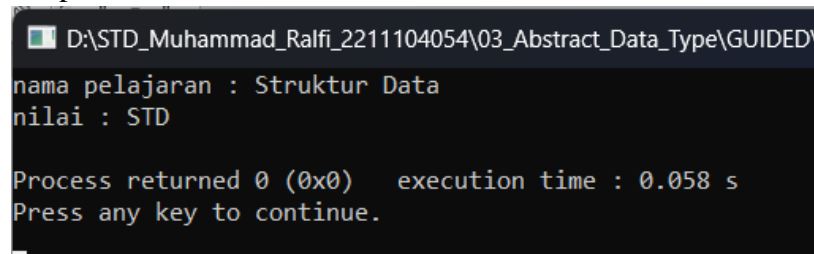
using namespace std;

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

    pelajaran pel = create_pelajaran(namapel, kodepel);
    tampil_pelajaran(pel);

    return 0;
}
```

Output:



```
D:\STD_Muhammad_Ralfi_2211104054\03_Abstract_Data_Type\GUIDED
nama pelajaran : Struktur Data
nilai : STD

Process returned 0 (0x0)   execution time : 0.058 s
Press any key to continue.
```

3. Buatlah program dengan ketentuan :
 - a. 2 buah array 2D integer berukuran 3x3 dan 2 buah pointer integer
 - b. Fungsi/prosedur yang menampilkan isi sebuah array integer 2D
 - c. Fungsi/prosedur yang akan menukarkan isi dari 2 array integer 2D pada posisi tertentu.
 - d. Fungsi/prosedur yang akan menukarkan isi dari variabel yang ditunjuk oleh 2 buah pointer

Code:

```
#include <iostream>
using namespace std;

void displayArray(int arr[3][3]) {
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            cout << arr[i][j] << " ";
        }
        cout << endl;
    }
}
```

```
    }  
    cout << endl;  
}  
  
void swapArrayElements(int arr1[3][3], int arr2[3][3], int row, int  
col) {  
    int temp = arr1[row][col];  
    arr1[row][col] = arr2[row][col];  
    arr2[row][col] = temp;  
}  
  
void swapPointerValues(int *ptr1, int *ptr2) {  
    int temp = *ptr1;  
    *ptr1 = *ptr2;  
    *ptr2 = temp;  
}  
  
int main() {  
  
    int array1[3][3] = {  
        {1, 2, 3},  
        {4, 5, 6},  
        {7, 8, 9}  
    };  
  
    int array2[3][3] = {  
        {9, 8, 7},  
        {6, 5, 4},  
        {3, 2, 1}  
    };  
  
    // Dua buah pointer integer  
    int a = 10, b = 20;  
    int *ptr1 = &a;  
    int *ptr2 = &b;  
  
    // Menampilkan array sebelum penukaran  
    cout << "Array 1 sebelum penukaran:\n";  
    displayArray(array1);  
    cout << "Array 2 sebelum penukaran:\n";  
    displayArray(array2);  
  
    // Menukar elemen pada posisi tertentu  
    swapArrayElements(array1, array2, 1, 1);  
  
    cout << "Array 1 setelah penukaran:\n";  
    displayArray(array1);  
    cout << "Array 2 setelah penukaran:\n";  
    displayArray(array2);  
}
```

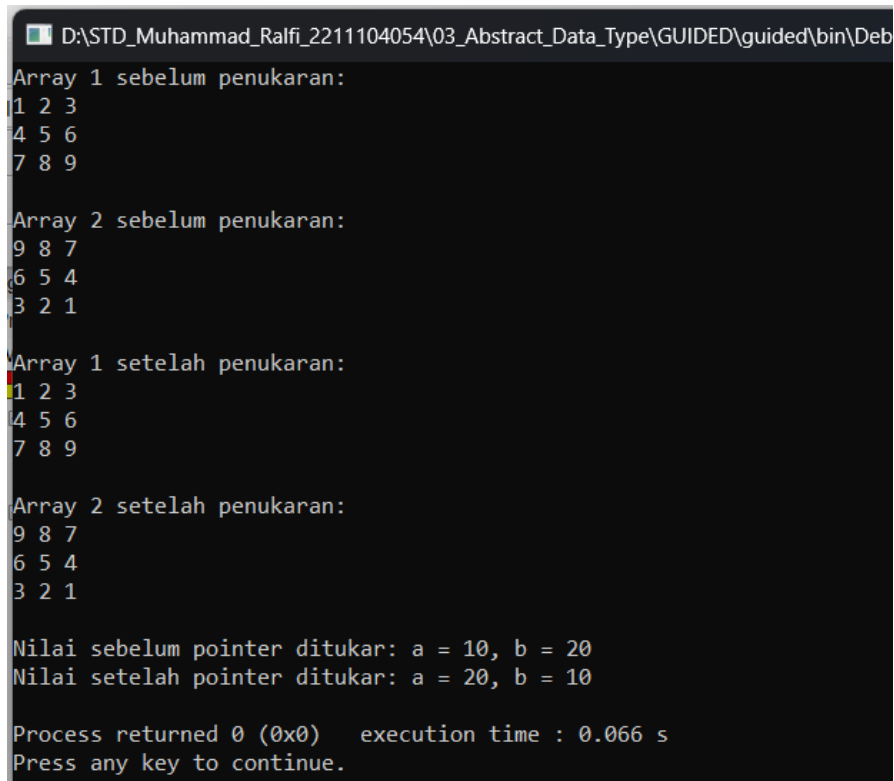
```
    cout << "Nilai sebelum pointer ditukar: a = " << *ptr1 << ", b = "
    << *ptr2 << endl;

    swapPointerValues(ptr1, ptr2);

    cout << "Nilai setelah pointer ditukar: a = " << *ptr1 << ", b = "
    << *ptr2 << endl;

    return 0;
}
```

Output:



```
D:\STD_Muhammad_Ralfi_2211104054\03_Abstract_Data_Type\GUIDED\guided\bin\Deb
Array 1 sebelum penukaran:
1 2 3
4 5 6
7 8 9

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

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

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

Nilai sebelum pointer ditukar: a = 10, b = 20
Nilai setelah pointer ditukar: a = 20, b = 10

Process returned 0 (0x0)   execution time : 0.066 s
Press any key to continue.
```

5. Kesimpulan

ADT merupakan sebuah konsep yang mendefinisikan objek berdasarkan perilakunya tanpa memperhatikan detail implementasi. ADT hanya mendeskripsikan nilai-nilai dan operasi yang dapat dilakukan pada objek, tanpa menyebutkan bagaimana data disimpan atau bagaimana operasi diimplementasikan secara spesifik. ADT bersifat "abstrak" karena menawarkan tampilan yang tidak bergantung pada detail implementasi, sehingga memungkinkan pengembangan program yang lebih modular dan fleksibel.

Dalam implementasinya, tipe dari sebuah ADT bisa terdiri dari ADT lainnya, seperti ADT waktu yang terdiri dari ADT JAM dan ADT DATE. Pada bahasa pemrograman seperti C++, ADT diterjemahkan dalam bentuk struktur (struct), dan operasi pada ADT dikelompokkan menjadi beberapa kategori seperti konstruktor, selector, prosedur pengubah nilai, validasi, destruktur, input/output, operator relasional, aritmatika, serta konversi tipe.

