

# **LAPORAN PRAKTIKUM**

## **PERTEMUAN 2**

### **Pengenalan C++ : Abstract Data Type**



**Nama :**

Ahmad Ufii Lestari Ma'ruf (2311104015)

**Dosen :**

YUDHA ISLAMI SULISTYA, S.Kom., M.Kom.

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK**  
**FAKULTAS INFORMATIKA**  
**TELKOM UNIVERSITY PURWOKERTO**

**2024**

## 1. Praktek Bersama

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.

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

Maka output yang akan muncul akan seperti ini :

```
input nim = 2311104015
input nilai 1 = 91
input nilai 2 = 97
Rata-rata = 94
PS E:\ITTP\Modul ITTP\semester3\stru
```

## 2. Tugas Pendahuluan

- a. Pointer adalah variabel khusus dalam pemrograman yang menyimpan alamat memori dari suatu variabel lain. Dengan pointer, kita bisa mengakses atau memanipulasi data yang disimpan di lokasi memori yang ditunjukkannya. Pointer sangat berguna dalam pengelolaan memori yang efisien, dan sering digunakan

dalam pemrograman tingkat rendah, seperti manajemen array, string, atau alokasi dinamis.

- b. kita dapat menampilkan alamat memori suatu variabel menggunakan operator &.

Contoh code nya

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

int main() {
    int x = 10;
    cout << "Alamat memori variabel x: " << &x << endl;
    return 0;
}
```

- c. Di sini, `ptr` adalah pointer yang menyimpan alamat dari variabel `x`. Nilai yang tersimpan pada alamat tersebut bisa diakses dengan menggunakan operator dereferensi `*`. berikut contoh penggunaannya

```
include <iostream>
using namespace std;

int main() {
    int x = 20;
    int* ptr = &x; // Pointer ptr menyimpan alamat dari x

    cout << "Nilai x: " << x << endl;
    cout << "Alamat memori x: " << ptr << endl;
    cout << "Nilai yang disimpan di alamat yang ditunjuk ptr: " << *ptr << endl;

    return 0;
}
```

- d. Abstract Data Type (ADT) adalah model matematis dari tipe data yang mendefinisikan operasi yang dapat dilakukan pada data tanpa mempedulikan bagaimana data tersebut diimplementasikan. ADT hanya menekankan pada operasi yang bisa dilakukan dan bagaimana operasinya (dari sisi pengguna), bukan bagaimana data disimpan atau algoritma yang digunakan untuk implementasi operasi tersebut. Contoh ADT adalah stack, queue, list, dan set.

- e. Contoh ADT dalam dunia nyata adalah “Daftar Tugas Harian”. Pada ADT daftar tugas:

- Operasi yang bisa dilakukan adalah: menambah tugas, menghapus tugas, melihat tugas yang paling pertama atau terakhir, dan lain sebagainya.
- Implementasi sebenarnya dari daftar tugas (misalnya menggunakan kertas atau aplikasi) tidak relevan, selama operasi-operasi tersebut bisa dilakukan.

f. ADT dari bangun ruang kerucut

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

class Kerucut {
private :
    double radius;
    double tinggi;
public :
    // Constructor
    Kerucut(double r, double t){
        radius = r;
        tinggi = t;
    };

    double hitungVolume(){
        return (M_PI * radius * radius * tinggi) / 3;
    };

    double luasPermukaanKerucut(){
        double s = sqrt((radius * radius) + (tinggi * tinggi));
        return (M_PI * radius * (s + radius));
    };

    double tampilkanHasil(){
        cout << "jari - jari : " << radius << endl;
        cout << "tinggi : " << tinggi << endl;
        cout << "Volume : " << hitungVolume() << endl;
        cout << "Luas Permukaan : " << luasPermukaanKerucut() << endl;
        return 0;
    }
};

int main(){
    double r, t;
    cout << "masukan jari jari als kerucut : ";
    cin >> r;
    cout << "masukan tinggi kerucut : ";
    cin >> t;

    Kerucut kerucut(r, t);
    kerucut.tampilkanHasil();
}
```

### 3. Unguided

#### a. Soal satu

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

struct Mahasiswa{
    string nama;
    string nim;
    float uts;
    float uas;
    float tugas;
    float nilaiAakhir;
};

float nilaiAakhir(float uts, float uas, float tugas){
    float nilai = (uts * 0.3) + (uas * 0.4) + (tugas * 0.3);
    return nilai;
}

void tampilkanNilai(Mahasiswa mahasiswa[], int jumlah){
    cout << "\n Data mahasiswa \n";
    cout << "Nama\tNIM\tUTS\tUAS\tTugas\tNilai Akhir\n";

    for(int i = 0; i < jumlah; i++){
        cout << mahasiswa[i].nama << "\t"
              << mahasiswa[i].nim << "\t"
              << mahasiswa[i].uts << "\t"
              << mahasiswa[i].uas << "\t"
              << mahasiswa[i].tugas << "\t"
              << mahasiswa[i].nilaiAakhir << endl;
    }
};

int main() {
    const int maxMahasiswa = 10;
    Mahasiswa mhs[maxMahasiswa];

    int jumlahMahasiswa = 0;
    char pilihan;

    for(int i = 0 ; i < maxMahasiswa ; i++){
        cout << "Data Mahasiswa ke-" << i+1 << endl;
        cout << "Masukan Nama : ";
        cin >> mhs[i].nama;
        cout << "Masukan NIM : ";
        cin >> mhs[i].nim;
        cout << "Masukan Nilai UTS : ";
        cin >> mhs[i].uts;
        cout << "Masukan Nilai UAS : ";
        cin >> mhs[i].uas;
        cout << "Masukan Nilai Tugas : ";
        cin >> mhs[i].tugas;

        mhs[i].nilaiAakhir = nilaiAakhir(mhs[i].uts, mhs[i].uas, mhs[i].tugas);

        jumlahMahasiswa++;

        cout << "Apakah ingin memasukan data mahasiswa lagi Y / N";
        cin >> pilihan;

        if(pilihan == 'n' || pilihan == 'N'){
            break;
        }
    }

    tampilkanNilai(mhs, jumlahMahasiswa);

    return 0;
}
```

**Output yang dihasilkan :**

```

C:\Users\user> python34\python.exe -i interpreter.py
Data Mahasiswa ke-1
Masukan Nama : ahmad
Masukan NIM : 23111
Masukan Nilai UTS : 98
Masukan Nilai UAS : 89
Masukan Nilai Tugas : 91
Apakah ingin memasukan data mahasiswa lagi Y / Ny
Data Mahasiswa ke-2
Masukan Nama : uffi
Masukan NIM : 23114
Masukan Nilai UTS : 99
Masukan Nilai UAS : 85
Masukan Nilai Tugas : 89
Apakah ingin memasukan data mahasiswa lagi Y / Ny
Data Mahasiswa ke-3
Masukan Nama : intan azizah
Masukan NIM : Masukan Nilai UTS : 91
Masukan Nilai UAS : 89
Masukan Nilai Tugas : 96
Apakah ingin memasukan data mahasiswa lagi Y / N
n

  Data mahasiswa
Nama      NIM      UTS      UAS      Tugas      Nilai Akhir
ahmad     23111    98      89      91      92.3
uffi      23114    99      85      89      90.4
intan     azizah   91      89      96      91.7

```

**b. Soal sua**

File 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
```

File 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;
}

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

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

    return 0;
}
```

Output yang dihasilkan:

```
Nama pelajaran : Struktur Data
Nilai : STD
```

### c. Soal Tiga

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

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

void pindahElement(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 pindahPointer(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}};

    int a = 10, b = 20;
    int *ptrA = &a, *ptrB = &b;

    cout << "Isi Array 1:" << endl;
    tampilkanArray(array1);
    cout << "Isi Array 2:" << endl;
    tampilkanArray(array2);

    pindahElement(array1, array2, 1, 1);

    cout << "Setelah menukar elemen di posisi [1][1]: " << endl;
    cout << "Isi Array 1:" << endl;
    tampilkanArray(array1);
    cout << "Isi Array 2:" << endl;
    tampilkanArray(array2);

    cout << "Sebelum menukar nilai menggunakan pointer:" << endl;
    cout << "Nilai a = " << a << ", Nilai b = " << b << endl;

    pindahPointer(ptrA, ptrB);

    cout << "Setelah menukar nilai menggunakan pointer:" << endl;
    cout << "Nilai a = " << a << ", Nilai b = " << b << endl;

    return 0;
}
```

### Output yang keluar

```
Isi Array 1:
1 2 3
4 5 6
7 8 9
Isi Array 2:
9 8 7
6 5 4
3 2 1
Setelah menukar elemen di posisi [1][1]:
Isi Array 1:
1 2 3
4 5 6
7 8 9
Isi Array 2:
9 8 7
6 5 4
3 2 1
Sebelum menukar nilai menggunakan pointer:
Nilai a = 10, Nilai b = 20
Setelah menukar nilai menggunakan pointer:
Nilai a = 20, Nilai b = 10
```