

Tugas Pendahuluan Modul 3  
STRUKTUR DATA - Ganjil 2024/2025  
"Abstract Data Type"

A. Ketentuan Tugas Pendahuluan

1. Tugas Pendahuluan dikerjakan secara **Individu**.
2. TP ini bersifat **WAJIB**, tidak mengerjakan = **PENGURANGAN POIN JURNAL / TES ASESMEN**.
3. Hanya **MENGUMPULKAN** tetapi **TIDAK MENGERJAKAN** = **PENGURANGAN POIN JURNAL / TES ASESMEN**.
4. Deadline pengumpulan TP Modul 2 adalah Senin, 23 September 2024 pukul 06.00 WIB.
5. **TIDAK ADA TOLERANSI KETERLAMBATAN, TERLAMBAT ATAU TIDAK MENGUMPULKAN TP MAKA DIANGGAP TIDAK MENGERJAKAN**.
6. **DILARANG PLAGIAT (PLAGIAT = E)**.
7. Kerjakan TP dengan jelas agar dapat dimengerti.
8. File diupload di LMS menggunakan format **PDF** dengan ketentuan: **TP\_MOD\_[XX]\_NIM\_NAMA.pdf**

**CP (WA):**

- Andini (082243700965)
- Imelda (082135374187)

**SELAMAT MENGERJAKAN^^**

# **LAPORAN PRAKTIKUM**

## **PERTEMUAN 3**

### **Abstract Data Type**



**Nama :**

Maulana Kaka Halin Widyadhana (2311104034)

**Dosen :**

Yudha Islami Sulistya

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

## B. Soal Tugas Pendahuluan

1. Jelaskan apa yang dimaksud dengan pointer!
2. Bagaimana cara menampilkan alamat memori dari suatu variabel dalam program C++? Berikan contoh!
3. Bagaimana cara menggunakan pointer dalam program C++? Berikan contoh cara menampilkan nilai yang tersimpan pada suatu alamat melalui pointer!
4. Jelaskan apa yang dimaksud dengan Abstract Data Type (ADT)!
5. Berikan contoh ilustrasi sederhana di dalam dunia nyata, tetapi diluar konteks pemrograman!
6. Tuliskan ADT dari bangun ruang kerucut dalam bahasa C++!

**Semoga Selalu diberi kemudahan^^**

Jawab:

1. Pointer adalah sebuah variabel yang berfungsi untuk menyimpan alamat memori objek. Pointer digunakan secara ekstensif pada bahasa pemrograman C dan C++ untuk 3 tujuan utama, yaitu:

- Untuk mengalokasikan objek baru pada timbunan,
- Untuk meneruskan fungsi ke fungsi lain.
- Untuk melakukan perulangan atau iterasi atas elemen dalam array atau struktur data lainnya.

2. Untuk menampilkan alamat memori dari suatu variabel pada bahasa C++, gunakan operator "&"

Contoh:

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

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

3. Cara menggunakannya yaitu dengan mendeklarasikan pointer dengan tipe data yang sama dengan variabel yang ingin ditunjuk.

Contoh:

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

int main() {
    int a = 20;
    int *ptr = &a; // Pointer menunjuk ke alamat memori variabel a
    cout << "Nilai a: " << a << endl; // Menampilkan nilai a
    cout << "Nilai a melalui pointer: " << *ptr << endl; // Mengakses nilai a melalui pointer
    cout << "Alamat memori pointer ptr: " << ptr << endl; // Menampilkan alamat memori yang ditunjuk oleh pointer
    return 0;
}
```

4. Abstract Data Type (ADT) adalah suatu model data yang hanya memaparkan cara interaksi dengan data tersebut tanpa menunjukkan detail implementasi. ADT berfungsi untuk mendefinisikan operasi-operasi yang dapat dilakukan pada data tetapi tidak memperlihatkan bagaimana operasi tersebut diimplementasikan.

5. Contoh ilustrasi ADT pada dunia nyata diluar konteks pemrograman yaitu dapat diambil dari ATM. Pengguna berinteraksi dengan ATM melalui serangkaian operasi seperti menarik uang, memasukkan uang, memeriksa saldo dan sebagainya. Pengguna tidak perlu tau bagaimana cara mesin tersebut memproses data didalamnya karena mereka hanya perlu memahami fungsi dan operasi yang disediakan.

6. Dalam ADT kerucut, definisikan beberapa atribut seperti jari-jari dan tinggi serta operasi untuk menghitung luas permukaan kerucut dan volume pada kerucut.

```
#include <iostream>
#include <cmath> // Fungsi matematika

using namespace std;

class Kerucut {
private:
    double JariJari;
    double Tinggi;
public:
    // Konstruktor
    Kerucut(double r, double t) : JariJari(r), Tinggi(t) {}

    // Menghitung volume kerucut
    double hitungVolume() {
        return (1.0/3.0) * M_PI * JariJari * JariJari * Tinggi;
    }

    // Menghitung luas permukaan kerucut
    double hitungLuasPermukaan() {
        double s = sqrt((JariJari * JariJari) + (Tinggi * Tinggi)); // s = sisi miring
        return M_PI * JariJari * (JariJari + s);
    }
};

int main() {
    // Membuat objek kerucut dengan jari-jari 7 dan tinggi 12
    Kerucut kerucut(7.0, 12.0);

    // Menunjukkan volume dan Luas permukaan kerucut
    cout << "Volume Kerucut: " << kerucut.hitungVolume() << endl;
    cout << "Luas Permukaan Kerucut: " << kerucut.hitungLuasPermukaan() << endl;

    return 0;
}
```