

TUGAS PENDAHULUAN
MODUL 2
“Pengenalan C++: Subprogram & Array”



Disusun Oleh:
Isabelle Putri Ardini - 2311104030
SE-07-01

Dosen:
Yudha Islami Sulistya

PROGRAM STUDI SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

Soal Tugas Pendahuluan

1. Jelaskan apa yang dimaksud dengan pointer!

Jawaban:

Pointer adalah suatu variabel yang dapat menyimpan alamat memori dari variabel lain.

2. Bagaimana cara menampilkan alamat memori dari suatu variabel dalam program C++? Berikan contoh!

Jawaban:

Alamat memori ditampilkan menggunakan operator alamat (&), di mana operator ini dapat menunjukkan alamat memori tempat variabel disimpan.

Contoh program:

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

int main() {
    int a = 42; // Deklarasi variabel 'a' dengan nilai 42
    double b = 3.14; // Deklarasi variabel 'b' dengan nilai 3.14

    // Menampilkan alamat memori dari variabel a dan b
    cout << "Alamat memori variabel a: " << &a << endl;
    cout << "Alamat memori variabel b: " << &b << endl;

    return 0;
}
```

3. Bagaimana cara menggunakan pointer dalam program C++? Berikan contoh cara menampilkan nilai yang tersimpan pada suatu alamat melalui pointer!

Jawaban:

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

int main() {
    int a = 42; // Deklarasi variabel a dengan nilai 42
    int *ptr = &a; // Deklarasi pointer ptr yang menunjuk ke alamat memori variabel a

    // Menampilkan nilai a secara langsung
    cout << "Nilai variabel a: " << a << endl;

    // Menampilkan alamat memori variabel a melalui pointer
    cout << "Alamat memori variabel a (melalui pointer): " << ptr << endl;
}
```

```

// Menampilkan nilai variabel a melalui dereferensi pointer
cout << "Nilai yang ditunjuk oleh pointer ptr: " << *ptr << endl;

// Mengubah nilai variabel a melalui pointer
*ptr = 100;
cout << "Nilai variabel a setelah diubah melalui pointer: " << a << endl;

return 0;
}

```

Penjelasan:

- Deklarasi variabel a : `int a = 42;` menyimpan nilai 42 di variabel a.
- Deklarasi pointer ptr : `int *ptr = &a;` membuat pointer ptr yang menyimpan alamat memori dari variabel a.
- Menampilkan alamat a : Dengan mencetak ptr, kita bisa melihat alamat memori yang ditunjuk pointer tersebut.
- Dereferensi pointer (*) : Dengan menggunakan *ptr, kita bisa mengakses nilai yang disimpan di alamat memori yang ditunjuk oleh pointer (yaitu nilai a).
- Mengubah nilai melalui pointer : Menggunakan *ptr = 100; ita mengubah nilai a melalui pointer.

- Jelaskan apa yang dimaksud dengan Abstract Data Type (ADT)?

Jawaban:

ADT adalah TYPE dan sekumpulan PRIMITIF (operasi dasar) terhadap TYPE tersebut. TYPE adalah kumpulan nilai-nilai data dan operasi-operasi dasar yang berlaku pada data tersebut, dan berfokus pada apa yang dilakukan oleh data, bukan bagaimana data tersebut diimplementasikan, sehingga memungkinkan fleksibilitas dalam pemilihan cara implementasi di belakang layar.

- Berikan contoh ilustrasi sederhana di dalam dunia nyata, tetapi di luar konteks pemrograman!

Jawaban:

Contohnya remote TV. Pengguna hanya perlu mengetahui fungsi-fungsi dari remote tersebut, seperti mengubah channel TV, mengatur volume, menyalakan/mematikan TV. Pengguna tidak perlu mengetahui komponen-komponen pada bagian dalam remote untuk mengoperasikan remote tersebut. Sama seperti ADT yang menyembunyikan detail implementasi dan hanya menampilkan operasi yang dilakukan.

- Tuliskan ADT dari bangun ruang kerucut dalam bahasa C++!

Jawaban:

Struktur ADT Kerucut:

- Atribut (Properties)
 - Jari-jari alas (radius)

- Tinggi (height)
 - Garis pelukis (slant height)
- b. Operasi (Methods)
- Menghitung operasi kerucut
 - Menghitung luas permukaan kerucut

Implementasi dalam C++:

```
#include <iostream>
#include <cmath> // Untuk operasi matematika seperti sqrt dan M_PI

using namespace std;

// Definisi ADT Kerucut
class Kerucut {
private:
    double radius;
    double height;
public:
    Kerucut(double r, double h) {
        radius = r;
        height = h;
    }

    double hitungGarisPelukis() const {
        return sqrt((radius * radius) + (height * height));
    }

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

    double hitungLuasPermukaan() const {
        double slantHeight = hitungGarisPelukis();
        return (M_PI * radius * (radius + slantHeight));
    }

    void tampilkanAtribut() const {
        cout << "Jari-jari (radius): " << radius << endl;
        cout << "Tinggi (height): " << height << endl;
        cout << "Garis pelukis (slant height): " << hitungGarisPelukis() << endl;
    }
};
```

```
int main() {
    Kerucut kerucut(5, 12);

    kerucut.tampilkanAtribut();

    cout << "Volume kerucut: " << kerucut.hitungVolume() << " satuan kubik" << endl;

    cout << "Luas permukaan kerucut: " << kerucut.hitungLuasPermukaan() << " satuan persegi" << endl;

    return 0;
}
```

Output:

```
Jari-jari (radius): 5  
Tinggi (height): 12  
Garis pelukis (slant height): 13  
Volume kerucut: 314.159 satuan kubik  
Luas permukaan kerucut: 282.743 satuan persegi
```