

Modul 2

1. Capaian Pembelajaran Matakuliah (CPMK)

CPMK142:

Mampu menerapkan teknik pemrograman terstruktur, modular, dan berorientasi objek untuk menyelesaikan permasalahan komputasi sesuai kebutuhan aplikasi sistem komputer.

Indikator Ketercapaian:

Mahasiswa mampu membuat class dan object sederhana.

2. Dasar Teori

KONSEP CLASS, OBJECT, ATTRIBUTE, DAN METHOD

Class adalah struktur data buatan pengguna (*user-defined data type*) yang menjadi cetak biru (*blueprint*) untuk membuat objek. Class mengelompokkan variabel (penyimpan data) dan fungsi (pemroses data) ke dalam satu wadah.

Dalam C++, deklarasi *class* menggunakan kata kunci *class* dan diawali dengan penentu akses (*access modifier*) seperti *public*, *private*, atau *protected*.

```
class NamaClass {
public:
    // Attribute
    std::string namaAttribute;

    // Method
    void namaMethod() {
        // Anggota aksi
    }
}; // Catatan: Sintaks class diakhiri dengan titik koma (;)
```

Object (Instansi Class)

Object adalah bentuk nyata (*instance*) dari suatu *class*. Ketika *class* didedikasikan, belum ada memori yang dialokasikan. Memori baru dialokasikan saat *object* dibuat di dalam fungsi *main()* atau fungsi lainnya

```
NamaClass namaObject; // Proses instansiasi objek
```

Attribute (Properti/Data)

Attribute adalah variabel yang dideklarasikan di dalam kelas. Variabel ini merepresentasikan karakteristik, keadaan (*state*), atau data yang dimiliki oleh suatu objek.

Method (Fungsi Anggota)

Method adalah fungsi yang dideklarasikan di dalam atau di luar definisi *class*. *Method* menentukan perilaku (*behavior*) atau tindakan yang dapat dilakukan oleh objek.

Akses Anggota Class

Untuk mengakses *attribute* atau memanggil *method* dari suatu objek, digunakan operator titik (.).

Kelas adalah tipe data yang didefinisikan pengguna yang dapat kita gunakan dalam program kita, dan berfungsi sebagai konstruktor objek, atau "cetak biru" untuk membuat objek.

3. Alat dan Bahan

Hardware :

1. Laptop/PC
2. LCD/proyektor
3. Jaringan internet

Software :

1. Sistem operasi Windows/Linux/macOS
2. Java Development Kit (JDK)
3. IDE pemrograman: NetBeans / IntelliJ IDEA / Visual Studio Code
4. LMS untuk materi dan aktivitas pembelajaran

4. Langkah Kerja / Prosedur Praktikum

Create a Class

To create a class, use the **class** keyword:

Example:

Create a class called "**MyClass**":

```
class MyClass {           // The class
    public:                // Access specifier
        int myNum;         // Attribute (int variable)
        string myString;   // Attribute (string variable)
};
```

Contoh yang dijelaskan

- Kata kunci tersebut `class` digunakan untuk membuat kelas bernama `MyClass`.
- Kata kunci tersebut `public` merupakan **penentu akses**, yang menentukan bahwa anggota (atribut dan metode) dari kelas tersebut dapat diakses dari luar kelas. Anda akan mempelajari lebih lanjut tentang penentu akses nanti.
- Di dalam kelas, terdapat variabel integer `myNum` dan variabel string `myString`. Ketika variabel dideklarasikan di dalam kelas, variabel tersebut disebut **atribut**.
- Terakhir, akhiri definisi kelas dengan titik koma `;`.

Create an Object

Dalam C++, sebuah objek dibuat dari sebuah kelas. Kita telah membuat kelas bernama `MyClass`, jadi sekarang kita dapat menggunakannya untuk membuat objek. Untuk membuat objek dari `MyClass`, tentukan nama kelas, diikuti dengan nama objek.

Untuk mengakses atribut kelas (`myNum` dan `myString`), gunakan sintaks titik (`.`) pada objek,

Contoh

Buat objek bernama "`myObj`" dan akses atributnya:

```
class MyClass {           // The class
public:                  // Access specifier
    int myNum;           // Attribute (int variable)
    string myString;     // Attribute (string variable)
};

int main() {
    MyClass myObj;       // Create an object of MyClass

    // Access attributes and set values
    myObj.myNum = 15;
    myObj.myString = "Some text";

    // Print attribute values
    cout << myObj.myNum << "\n";
    cout << myObj.myString;
    return 0;
}
```

Multiple Objects

Anda dapat membuat beberapa objek dari satu kelas:

Contoh

```
// Create a Car class with some attributes
class Car {
public:
    string brand;
    string model;
    int year;
};

int main() {
    // Create an object of Car
    Car carObj1;
    carObj1.brand = "BMW";
    carObj1.model = "X5";
    carObj1.year = 1999;

    // Create another object of Car
    Car carObj2;
    carObj2.brand = "Ford";
    carObj2.model = "Mustang";
    carObj2.year = 1969;

    // Print attribute values
    cout << carObj1.brand << " " << carObj1.model << " " <<
carObj1.year << "\n";
    cout << carObj2.brand << " " << carObj2.model << " " <<
carObj2.year << "\n";
    return 0;
}
```

5. Lembar Kerja

Latihan 2

Membuat Berkas Program:

Persiapan Workspace.

1. Buka aplikasi **Visual Studio Code**.
2. Buka folder kerja praktikum Anda (File > Open Folder).
3. Buat berkas baru bernama modul2_persegi.cpp

Menuliskan Class PersegiPanjang:

Langkah 1. Ketikkan kode berikut untuk mendefinisikan class, attribute, dan method perhitungan geometri sederhana:

```
C++#include <iostream>
// 1. Deklarasi Class
```

```

class PersegiPanjang {
public:
    // 2. Deklarasi Attribute
    double panjang;
    double lebar;

    // 3. Deklarasi Method
    double hitungLuas() {
        return panjang * lebar;
    }

    double hitungKeliling() {
        return 2 * (panjang + lebar);
    }

    void tampilkanInfo() {
        std::cout << "--- Detail Persegi Panjang ---" << std::endl;
        std::cout << "Panjang : " << panjang << std::endl;
        std::cout << "Lebar   : " << lebar << std::endl;
        std::cout << "Luas    : " << hitungLuas() << std::endl;
        std::cout << "Keliling : " << hitungKeliling() << std::endl;
    }
};

```

Membuat Object di Fungsi main():

Langkah 2.Lanjutkan dengan menambahkan fungsi main() di bawah deklarasi class pada berkas yang sama:

```

C++int main() {
    // Instansiasi Object Pertama
    PersegiPanjang kotak1;
    kotak1.panjang = 10.0;
    kotak1.lebar = 5.0;

    // Instansiasi Object Kedua
    PersegiPanjang kotak2;
    kotak2.panjang = 7.5;
    kotak2.lebar = 3.0;

    // Memanggil Method untuk kotak1
    std::cout << "OBJEK 1:" << std::endl;
}

```

```
    kotak1.tampilkanInfo();

    std::cout << std::endl;

    // Memanggil Method untuk kotak2
    std::cout << "OBJEK 2:" << std::endl;
    kotak2.tampilkanInfo();

    return 0;
}
```

Kompilasi dan Eksekusi Program:

Langkah 3.

Buka Terminal di VS Code (Ctrl + ~ atau Terminal > New Terminal)

Jalankan perintah kompilasi: `Bash g++ modul2_persegi.cpp -o modul2_persegi`

Jalankan program hasil kompilasi:

Windows: `.\modul2_persegi.exe`

Linux/macOS: `./modul2_persegi`

Amati output yang muncul di terminal.

HASIL PERCOBAAN:

Screenshot Hasil LATIHAN 2 dan Berikan Analisis.
(UPLOAD ke LMS → TUGAS PRAKTIKUM 2)

Tugas Mandiri 2

(Kerjakan Tugas Mandiri 2 Upload ke LMS → **TUGAS TEORI 2**)

1. Apakah yang dimaksud dengan CLASS, OBJECT, METHOD?
2. Sebutkan min. 5 atribut dan 3 method yang melekat pada diri anda ?
3. Mengapa metodologi berorientasi objek berkembang lebih pesat dibandingkan dengan metode-metode yang lain dalam bahasa pemrograman
4. Di dalam konsep OOP, apakah fungsi utama dari sebuah **method** di dalam **class**?
5. Mengapa pada contoh kode di atas kita menggunakan kata kunci public: di dalam definisi PersegiPanjang? Apa yang akan terjadi jika kata kunci tersebut dihilangkan atau diganti menjadi private: saat kita mengakses kotak1.panjang = 10.0; dari dalam fungsi main()?



Program Studi
SISTEM KOMPUTER