“PENJELASAN TENTANG ENCAPSULATION DAN INHERITANCE DALAM PEMROGRAMAN C++ BESERTA CONTOH PROGRAMN”

Tugas Pemograman Berorientasi Objek

Disusun untuk Memenuhi Tugas Mata Kuliah Pemograman Berorientasi Objek, Semester 2 (Rombel A)

Disusun Oleh:

Fachrezi Bachri

2401020010

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK DAN TEKNOLOGI KEMARITIMAN

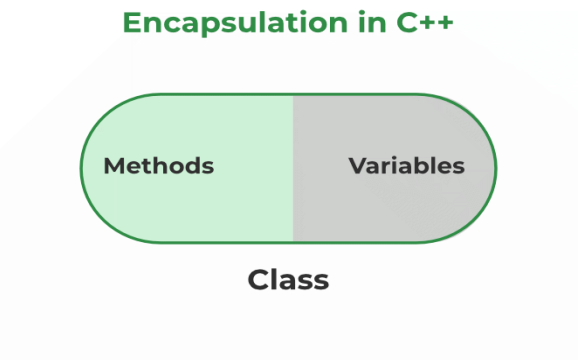
UNIVERSITAS MARITIM RAJA ALI HAJI

TAHUN PELAJARAN 2025/2026

1. Encapsulation

Pengertian Encapsulation :
Secara sederhana, enkapsulasi didefinisikan sebagai pembungkusan data dan informasi di bawah satu unit. Dalam Pemrograman Berorientasi Objek, enkapsulasi didefinisikan sebagai pengikatan data dan fungsi yang memanipulasinya bersama-sama dalam suatu kelas.

Contoh :



Contoh Program :

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

class Employee {
private:
    int salary;

public:
    void setSalary(int s) {
        salary = s;
    }

    int getSalary() {
        return salary;
    }
};

int main() {
    Employee myObj;
    myObj.setSalary(50000);
    cout << myObj.getSalary();
    return 0;
}
```

Penjelasan Dari Contoh :

- Atribut salary bersifat private, yang artinya memiliki akses terbatas.
- Metode public setSalary() mengambil parameter (s) dan menetapkannya ke atribut salary (salary = s).
- Metode public getSalary() mengembalikan nilai atribut private salary.
- Di dalam main(), kita membuat objek kelas Employee. Sekarang kita dapat menggunakan metode setSalary() untuk menetapkan nilai atribut privat menjadi 50000. Kemudian kita memanggil metode getSalary() pada objek untuk mengembalikan nilai.

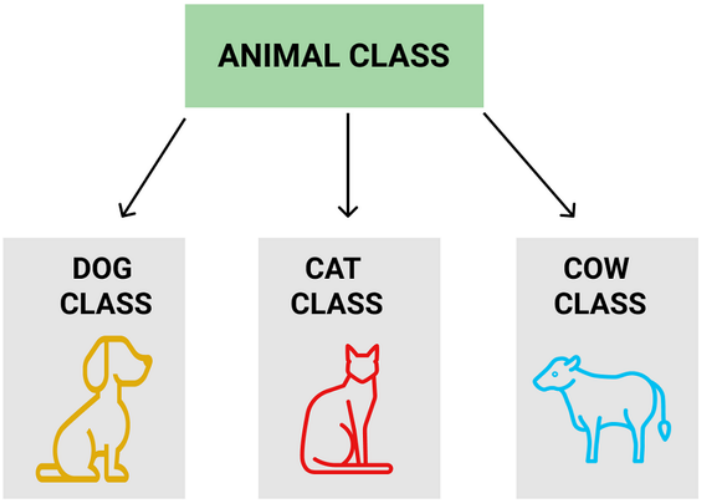
2. Inheritance

Pengertian Inheritance :
Kemampuan suatu kelas untuk memperoleh properti dan karakteristik dari kelas lain disebut Pewarisan. Pewarisan merupakan salah satu fitur terpenting dalam Pemrograman Berorientasi Objek.

- Sub Kelas: Kelas yang mewarisi properti dari kelas lain disebut Sub kelas atau Kelas Turunan.

- Kelas Super: Kelas yang propertinya diwarisi oleh subkelas disebut Kelas Dasar atau Kelas Super

Contoh :

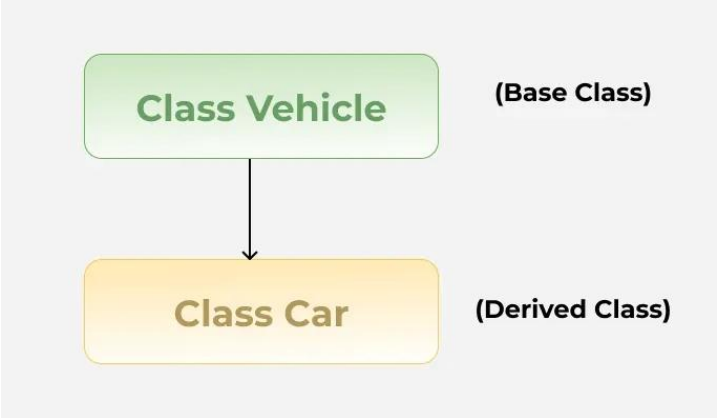


Dog, Cat, Cow dapat merupakan Kelas Turunan dari Kelas Animal.

Jenis Inheritance :

1. Single Inheritance :
   Dalam Single Inheritance, sebuah kelas hanya diperbolehkan mewarisi dari satu kelas saja. Dengan kata lain, satu kelas dasar diwarisi oleh satu kelas turunan saja.

Contoh :



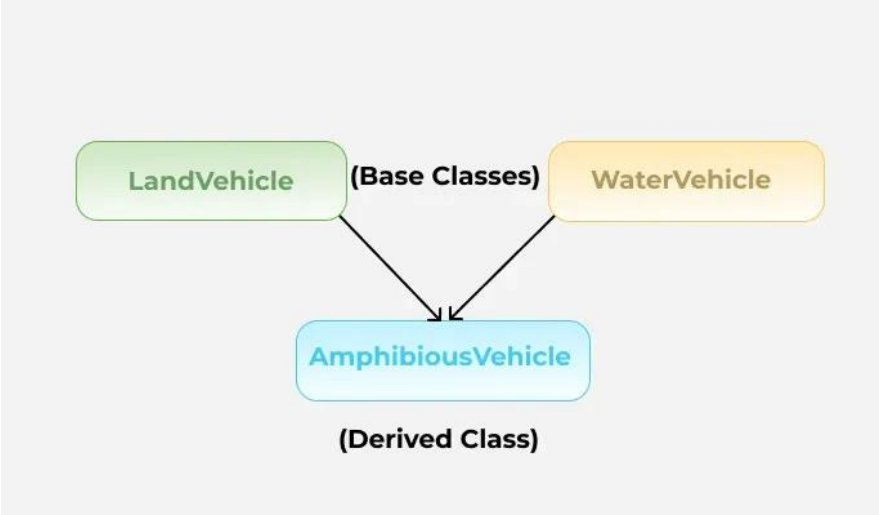
Contoh Program :

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

class Vehicle {
public:
    Vehicle() {
        cout << "This is a Vehicle"<< endl;
    }
};

class Car : public Vehicle {
public:
    Car() {
        cout << "This Vehicle is Car"<< endl;
    }
};

int main() {
    Car obj;
    return 0;
}
```

2. Multiple Inheritance :

   Multiple Inheritance merupakan fitur C++ yang memungkinkan satu kelas mewarisi lebih dari satu kelas. Dengan kata lain, satu subkelas diwarisi dari lebih dari satu kelas dasar.

Contoh :

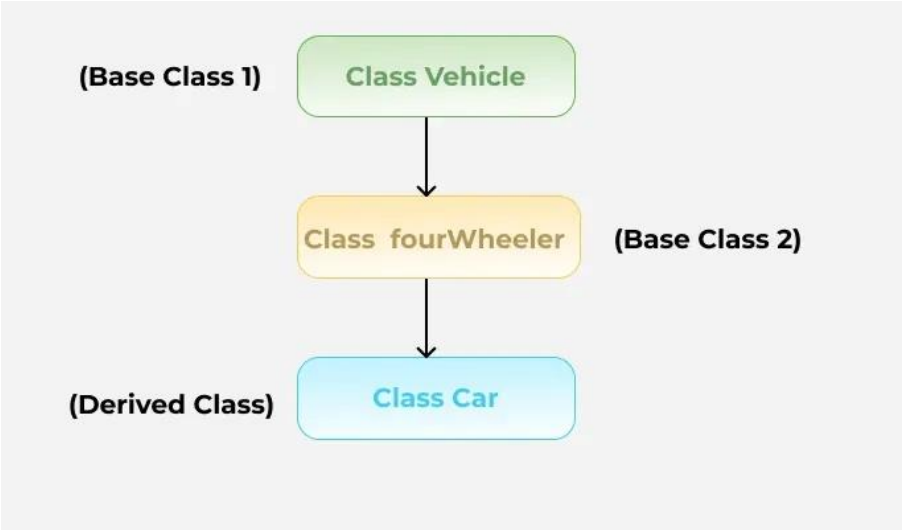


Contoh Program :

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

class LandVehicle {
public:
    LandVehicle() {
        cout << "This is a LandVehicle"<< endl;
    }
};

class WaterVehicle {
public:
    WaterVehicle() {
        cout << "This is a WaterVehicle"<< endl;
    }
};

class AmphibiousVehicle : public WaterVehicle, public LandVehicle {
public:
    AmphibiousVehicle() {
        cout << "This is an AmphibiousVehicle"<< endl;
    }
};

int main() {
    AmphibiousVehicle obj;
    return 0;
}
```

3. Multilevel Inheritance :
   Dalam Multilevel Inheritance, kelas turunan dibuat dari kelas turunan lain dan kelas turunan tersebut dapat diturunkan dari kelas dasar atau kelas turunan lainnya. Tingkatannya bisa berapa saja.

Contoh :

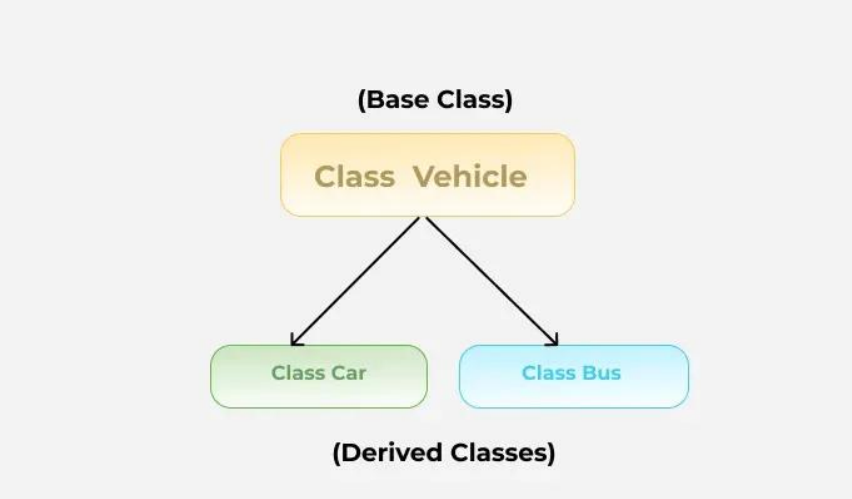


Contoh Program :

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

class Vehicle {
public:
    Vehicle() {
        cout << "This is a Vehicle"<< endl;
    }
};

class fourWheeler : public Vehicle {
public:
    fourWheeler() {
        cout << "4 Wheeler Vehicles"<< endl;
    }
};

class Car : public fourWheeler {
public:
    Car() {
        cout << "This 4 Wheeler Vehical is a Car";
    }
};

int main() {
    Car obj;
    return 0;
}
```

4. Hierarchical Inheritance :
   Dalam Hierarchical Inheritance, lebih dari satu subkelas diwarisi dari satu kelas dasar. Yaitu lebih dari satu kelas turunan dibuat dari satu kelas dasar.

Contoh :

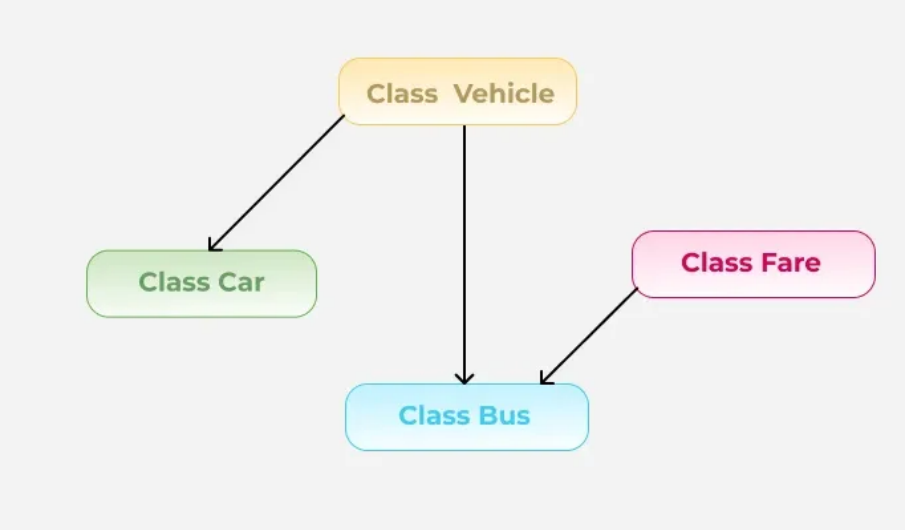


Contoh Program :

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

class Vehicle {
public:
    Vehicle() {
        cout << "This is a Vehicle"<< endl;
    }
};

class Car : public Vehicle {
public:
    Car() {
        cout << "This Vehicle is Car"<< endl;
    }
};

class Bus : public Vehicle {
public:
    Bus() {
        cout << "This Vehicle is Bus"<< endl;
    }
};

int main() {
    Car obj1;
    Bus obj2;
    return 0;
}
```

5. Hybrid Inheritance :
   Hybrid Inheritance di implementasikan dengan menggabungkan lebih dari satu jenis pewarisan.

Contoh :



Contoh Program :

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

class Vehicle {
public:
    Vehicle() {
        cout << "This is a Vehicle"<< endl;
    }
};

class Fare {
public:
    Fare() {
        cout << "Fare of Vehicle"<< endl;
    }
};

class Car : public Vehicle {
    public:
    Car() {
        cout << "This Vehical is a Car"<< endl;
    }
};

class Bus : public Vehicle, public Fare {
    public:
    Bus() {
        cout << "This Vehicle is a Bus with Fare";
    }
};

int main() {
    Bus obj2;
    return 0;
}
```