

Inheritance (Pewarisan)

Purwo Agus Sucipto, S.Kom, M.Kom

Pewarisan

Pewarisan memungkinkan satu kelas untuk menggunakan kembali atribut dan metode dari kelas lain.

Ini membantu menulis kode yang lebih bersih dan efisien dengan menghindari duplikasi.

Pengelompokkan "konsep warisan" menjadi dua kategori:

- **kelas turunan** (anak) - kelas yang mewarisi dari kelas lain
- **kelas dasar** (induk) - kelas yang diwarisi

Untuk mewarisi dari suatu kelas, gunakan symbol : .

Dalam contoh di bawah, Carkelas (anak) mewarisi atribut dan metode dari Vehiclekelas (induk).

Contoh

```
#include <iostream>
```

```
Using namespace std;
```

```
// Base class
```

```
class Kendaraan {  
public:  
    string merk = "Ford";  
    void klakson() {  
        cout << "Tuut, tuut! \n" ;  
    }  
};
```

```
// Derived class
```

```
class Mobil: public Kendaraan {  
public:  
    string model = "Mustang";  
};
```

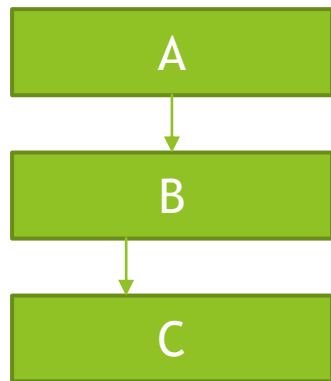
```
int main() {  
    Car myCar;  
    myCar.klakson();  
    cout << myCar.merk + " " + myCar.model;  
    return 0;  
}
```

Pewarisan Multilevel (Bertingkat)

Pewarisan Bertingkat

Suatu kelas juga dapat diturunkan dari satu kelas, yang sudah diturunkan dari kelas lain.

Dalam contoh berikut, Cucu diturunkan dari kelas Anak (yang diturunkan dari MyClass).



Contoh

```
#include <iostream>

Using namespace std;

class MyClass {
public:
    void myFunction() {
        cout << "Some content in parent class." ;
    }
};

// Derived class (child)
class MyChild: public MyClass {
};

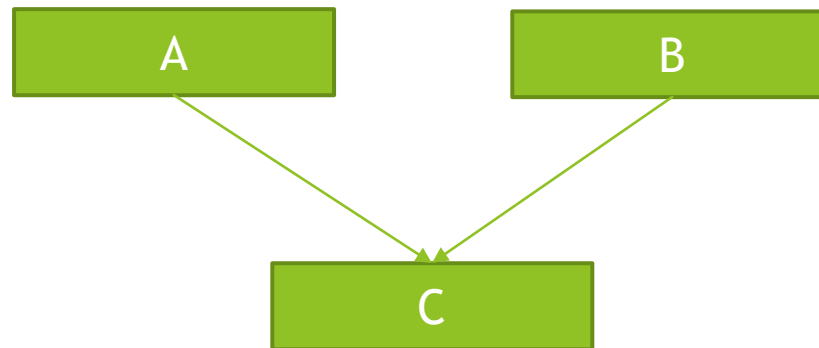
// Derived class (grandchild)
class MyGrandChild: public MyChild {
};

int main() {
    MyGrandChild myObj;
    myObj.myFunction();
    return 0;
}
```

Pewarisan Berganda

Suatu kelas juga dapat diturunkan dari lebih dari satu kelas dasar, menggunakan daftar yang dipisahkan dengan koma.

Class Anak: Public Ortu, public kakek{};



Contoh

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

class MyClass {
public:
    void myFunction() {
        cout << "Some content in parent class." ;
    }
};

// Another base class
class MyOtherClass {
public:
    void myOtherFunction() {
        cout << "Some content in another class." ;
    }
};

// Derived class
class MyChildClass: public MyClass, public MyOtherClass {
};

int main() {
    MyChildClass myObj;
    myObj.myFunction();
    myObj.myOtherFunction();
    return 0;
}
```