



MATERI AJAR

Pemrograman Berorientasi Objek

Materi Inheritance



1. Pengertian Inheritance

Inheritance (pewarisan) adalah konsep dalam Pemrograman Berorientasi Objek (OOP) yang memungkinkan sebuah class untuk mewarisi atribut dan metode dari class lain. Class yang diwarisi disebut parent class (superclass), sedangkan class yang mewarisi disebut child class (subclass).

Menurut (Booch, 2007), inheritance memungkinkan reuse (penggunaan ulang) kode dengan cara mendefinisikan class yang lebih umum sebagai dasar, sehingga subclass dapat menambahkan atau mengubah perilaku sesuai kebutuhan tanpa menulis ulang seluruh kode.

Contoh Konsep Inheritance:

Jika kita memiliki class Hewan, maka class Kucing dan Anjing bisa mewarisi atribut dan metode dari class Hewan, seperti nama, umur, dan suara().

2. Tujuan Inheritance

- Mengurangi duplikasi kode – Subclass dapat menggunakan kembali kode dari superclass.
- Mempermudah pemeliharaan – Perubahan di superclass akan berpengaruh pada semua subclass.
- Meningkatkan modularitas – Membantu dalam pembuatan struktur kode yang lebih rapi dan terorganisir.
- Mendukung konsep polymorphism – Subclass dapat mengubah metode superclass sesuai kebutuhan.

3. Konsep Dasar Inheritance

Inheritance memungkinkan hubungan "is-a" (adalah suatu) antara superclass dan subclass.

Contoh Hubungan "is-a" dalam Inheritance

- Burung is-a Hewan
- Mobil is-a Kendaraan
- Dosen is-a Orang

4. Manfaat Inheritance

- Efisiensi Kode → Kode tidak perlu ditulis ulang karena subclass dapat menggunakan kode dari superclass.
- Pengelompokan yang Logis → Class dapat dikategorikan berdasarkan fungsinya.
- Kemudahan Pengembangan → Jika ada perubahan pada superclass, maka subclass otomatis mendapat perubahannya.

- Mendukung Polymorphism → Metode dalam superclass bisa diubah dalam subclass sesuai kebutuhan.

5. Contoh Implementasi

```
using System;

class Hewan // Superclass (Parent)
{
    public string Nama;
    public int Umur;

    public void Info()
    {
        Console.WriteLine($"Nama: {Nama}, Umur: {Umur} tahun");
    }
}

class Kucing : Hewan // Subclass (Child)
{
    public void Meong()
    {
        Console.WriteLine("Meow! Meow!");
    }
}

class Program
{
    static void Main()
    {
        Kucing kucingA = new Kucing();
        kucingA.Nama = "Milo";
        kucingA.Umur = 2;
        kucingA.Info(); // Mewarisi metode dari superclass
        kucingA.Meong(); // Metode khusus milik class Kucing
    }
}
```

Output

```
Nama: Milo, Umur: 2 tahun
Meow! Meow!
```

Referensi

Booch, G. (2007). Object-Oriented Analysis and Design with Applications. Addison-Wesley.

Wegner, P. (1990). Concepts and Paradigms of Object-Oriented Programming. ACM SIGPLAN OOPS Messenger.



LKPD

Pemrograman Berorientasi Objek

Materi Inheritance



Mata Pelajaran	Dasar-dasar Pemrograman
Materi	Inheritance
Kelas/Semester	X/2
Alokasi Waktu	45 Menit
Hari/Tanggal	
Nama	
Kelas	

Soal Essay

1. Jelaskan konsep inheritance dalam OOP dengan bahasa sendiri dan berikan contohnya!

2. Apa perbedaan antara superclass dan subclass dalam inheritance?

3. Diberikan class berikut, tambahkan satu subclass bernama Mobil yang mewarisi Kendaraan dan tambahkan atribut baru jumlahPintu:

```
class Kendaraan
{
    public string Merk;
    public int Tahun;
}
```

4. Apa manfaat menggunakan inheritance dalam pemrograman? Jelaskan minimal tiga manfaat!

5. Berikut adalah class Hewan:

```
class Hewan
{
    public void Suara()
    {
        Console.WriteLine("Hewan mengeluarkan suara");
    }
}
```

Buat subclass Kucing yang mengubah metode Suara() agar menghasilkan output "Meow! Meow!"