

Pewarisan/Inheritance

6.1 Pengertian Inheritance

Inheritance/Pewarisan adalah suatu cara pembuatan class baru dengan menggunakan kembali class yang sudah didefinisikan sebelumnya dengan menambahkan attribute dan method baru. Sehingga demikian class baru yang dibuat tetap memiliki attribute dan method yang dimiliki oleh class induknya. Pada konsep pewarisan/ inheritance terdapat beberapa istilah yang perlu diketahui yaitu:

- Sub Class, digunakan untuk menunjukkan class anak atau turunan secara hirarkis dari super class
- Super Class, digunakan untuk menunjukkan class induk secara hirarkis dari sub class (class anak)

6.1.1 Studi Kasus

Buatlah sebuah project yang berisi tiga class berikut ini:

| Virus |
|---|
| + nama: String +size : int +kemampuan : String +banyak : int |
| +Menyerang() : void +MemperbanyakDiri() : void |

| Trojan |
|--|
| +nama : String +size : int +kemampuan : String |
| +Menyerang () : void +MenyembunyikanFile() : void +MemblokirCMD() : void |

| Worm |
|---|
| +nama : String +size : int +kemampuan : String |
| +Menyerang() : void +MenginfeksiRegistry() : void +Menghapus NTLDR() : void |

Berikut ini adalah kode program dari bagan diatas:

- **Virus.cs**

```
class Virus
{
    public String nama;
    public int size;
    public String kemampuan;
    public int banyak;

    public void menyerang()
    {
        Console.WriteLine(" {0} menyerang dengan {1} ", this.nama, this.kemampuan);
    }

    public void MemperbanyakDiri()
    {
        Console.WriteLine(" {0} memperbanyak file hingga {1} kali", this.nama, this.banyak*5);
    }
}
```

- **Trojan.cs**

```
class Trojan
{
    public String nama;
    public int size;
    public String kemampuan;

    public void Menyerang()
    {
        Console.WriteLine(" {0} menyerang dengan {1} ", this.nama, this.kemampuan);
    }

    public void MenyembunyikanFile()
    {
        Console.WriteLine(" {0} Sembunyikan File yang ada di C:/Program File/ Microsoft Office", this.nama);
    }
    public void MemblokirCMD()
    {
        Console.WriteLine(" {0} Blokir setiap kegiatan melalui CommandPrompt", this.nama);
    }
}
```

- **Worm.cs**

```
class Worm
{
    public String nama;
    public int size;
    public String kemampuan;

    public void menyerang()
    {
        Console.WriteLine(" {0} menyerang dengan {1} ", this.nama, this.kemampuan);
    }

    public void MenginfeksiRegistry()
    {
        Console.WriteLine(" {0} Menginfeksi Registry ", this.nama);
    }

    public void MenghapusNTLDR()
    {
        Console.WriteLine(" {0} Menghapus NTLDR ", this.nama);
    }
}
```

• Program.cs

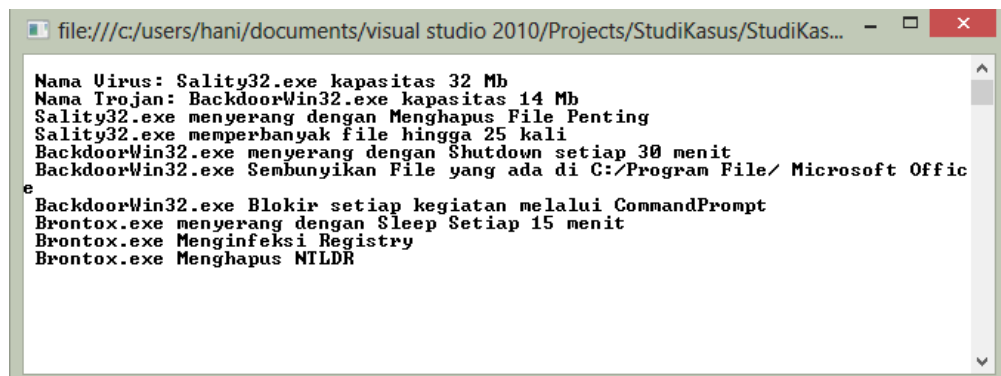
```
static void Main(string[] args)
{
    Virus Virus1 = new Virus();
    Trojan Trojan1 = new Trojan();
    Worm Worm1 = new Worm();

    //pengaksesan atribut
    virus1.nama = "Sality32.exe";
    virus1.size = 32;
    virus1.kemampuan = "Menghapus File Penting";
    virus1.banyak = 5;
    Trojan1.nama = "BackdoorWin32.exe";
    Trojan1.size = 14;
    Trojan1.kemampuan = "Shutdown setiap 30 menit";
    Worm1.nama = "Brontox.exe";
    Worm1.size = 23;
    Worm1.kemampuan = "Sleep Setiap 15 menit";

    //display
    Console.WriteLine("\n Nama Virus: {0} kapasitas {1} Mb", virus1.nama, virus1.size);
    Console.WriteLine(" Nama Trojan: {0} kapasitas {1} Mb", Trojan1.nama, Trojan1.size);

    //pengaksesan method
    virus1.menyerang();
    virus1.MemperbanyakDiri();
    Trojan1.Menyerang();
    Trojan1.MenyembunyikanFile();
    Trojan1.MemblokirCMD();
    Worm1.menyerang();
    Worm1.MenginfeksiRegistry();
    Worm1.MenghapusNTLDR();

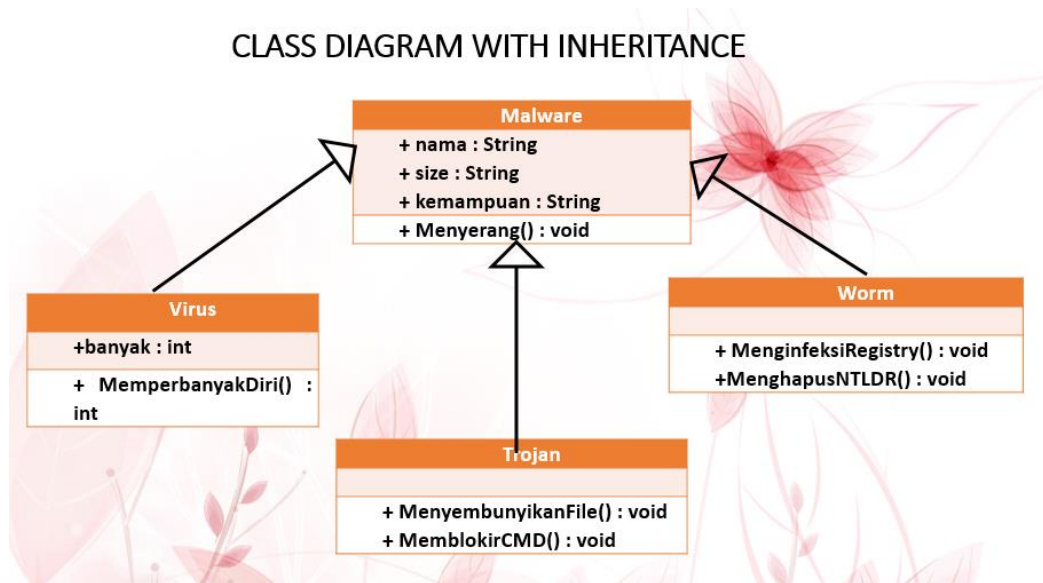
    Console.Read();
}
```



6.1.2 Latihan Inheritance

Setelah berlatih dengan program yang diatas, bagaimana menurut anda? Menulis berulang kali method dan atribut yang sama sangatlah melelahkan dan membosankan bukan?? Nah disinilah peranan inheritance diperlukan dengan inheritace/ pewarisan programmer hanya perlu membuat satu parent class (Super Class) yang mana atribut dan methodnya akan diturunkan kepada anak-anak kelasnya (Sub Class) nya. Kemudian dari sub class jika ingin menambahkan method baru , programmer tinggal menambahkan method yang diperlukan pada class tersebut, tanpa mengubah atau menghapus method yang

diturunkan, oleh karena itu struktur diagramnya akan berubah menjadi seperti ini:



Dari bagan/ diagram diatas programmer perlu membuat empat kelas baru yaitu kelas malware, virus, trojan, worm. Mengapa lebih mudah menggunakan inheritance? Karena setiap malware (virus/trojan/worm) sama sama memiliki attribut **nama**, **size**, dan **kemampuan**, juga memiliki method **Menyerang()**. Agar lebih jelasnya mari kita buat kode programnya:

- **Malware.cs**

```
class Malware
{
    public string nama;
    public int size;
    public string kemampuan;

    public void Menyerang()
    {
        Console.WriteLine("\n {0} menyerang dengan cara {1}", this.nama, this.kemampuan);
    }
}
```

- **Virus.cs** → Menandakan class virus turunan dari class malware

```
class Virus: Malware
{
    public int banyak;

    public void MemperbanyakDiri()
    {
        Console.WriteLine(" {0} memperbanyak diri sebanyak {1} kali", this.nama, this.banyak*5);
    }
}
```

- Trojan.cs

```
class Trojan : Malware
{
    public void menyembunyikanFile()
    {
        Console.WriteLine(" Sembunyikan file yang ada di C:/Program File/Microsoft Office");
    }
    public void memblokirCMD()
    {
        Console.WriteLine(" Blokir semua aktifitas yang menggunakan Command Prompt");
    }
}
```

- Worm.cs

```
class Worm : Malware
{
    public void menginfeksiRegistry()
    {
        Console.WriteLine(" {0} menginfeksi registry", this.nama);
    }

    public void menghapusNTLDR()
    {
        Console.WriteLine(" {0} menghapus NT Loader", this.nama);
    }
}
```

- Program.cs

```
class Program
{
    static void Main(string[] args)
    {
        Virus virus1 = new Virus();
        Trojan trojan1 = new Trojan();
        Worm worm1 = new Worm();

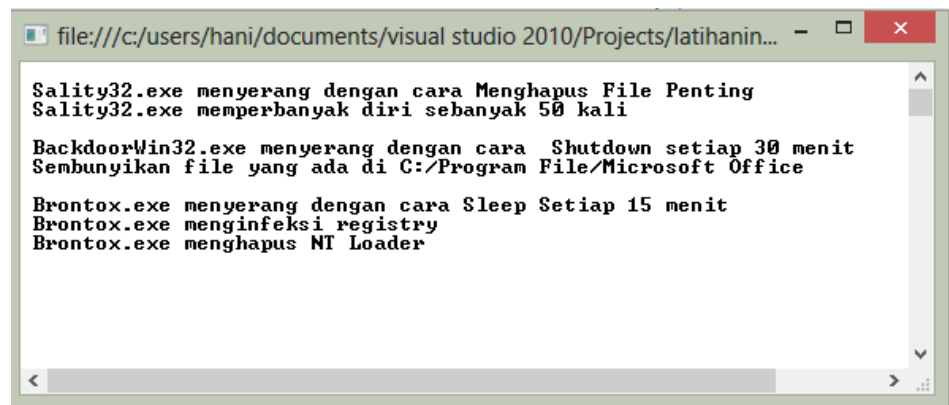
        virus1.nama = "Sality32.exe";
        virus1.size = 32;
        virus1.banyak = 10;
        virus1.kemampuan = "Menghapus File Penting";
        virus1.Menyerang();
        virus1.MemperbanyakDiri();

        trojan1.nama = "BackdoorWin32.exe";
        trojan1.kemampuan = " Shutdown setiap 30 menit";
        trojan1.Menyerang();
        trojan1.menyembunyikanFile();

        worm1.nama = "Brontox.exe";
        worm1.size = 14;
        worm1.kemampuan = "Sleep Setiap 15 menit";
        worm1.Menyerang();
        worm1.menginfeksiRegistry();
        worm1.menghapusNTLDR();

        Console.Read();
    }
}
```

- **Running Program**



```
file:///c:/users/hani/documents/visual studio 2010/Projects/latihanin... - □ ×  
  
Sality32.exe menyerang dengan cara Menghapus File Penting  
Sality32.exe memperbanyak diri sebanyak 50 kali  
  
BackdoorWin32.exe menyerang dengan cara Shutdown setiap 30 menit  
Sembunyikan file yang ada di C:/Program File/Microsoft Office  
  
Brontox.exe menyerang dengan cara Sleep Setiap 15 menit  
Brontox.exe menginfeksi registry  
Brontox.exe menghapus NT Loader
```

6.2 Virtual dan Override Methode

Dengan menggunakan konsep inheritance programmer dapat mempermudah pengerjaan program menjadi lebih simple dan efektif dengan dapat menggunakan method yang sama dengan yang ada pada parent class (super class), akan tetapi ada kalannya method yang diwarisi oleh super class pada sub class tidak sesuai dengan tempatnya. misalkan saja programmer membuat class dengan nama **“Hewan.cs”** sebagai parent class, kemudian terdapat sub class dengan nama **“Buaya.cs”**, dan **“Banteng.cs”**, lalu dalam class **“Hewan.cs”** terdapat method **menyerang()**, maka secara otomatis buaya dan banteng memiliki method menyerang, akan tetapi pada realnya (pada kehidupan nyata) buaya dan banteng memiliki cara menyerang yang berbeda, lalu bagaimana cara inheritance dapat menurunkan method untuk sesuatu yang lebih spesifik lagi? Yaitu dengan cara membuat override method, override method yaitu menimpa method yang ada pada parent class untuk diterapkan pada sub class dengan cara member awalan keyword **“Override”** pada nama method yang sama yang ada pada parent/super class.

Sedangkan virtual adalah keyword yang digunakan pada method yang ada pada parent class yang nantinya akan di override melalui method yang ada di sub class. Jadi jika pada class turunan/ sub class ditambahkan keyword override, maka pada super class method yang nantinya akan di override/ ditimpa menggunakan keyword **“Virtual”**. Berikut adalah contoh penerapan virtual dan override method.

- **Malware.cs**

Menandakan Menyerang dapat di override

```
class Malware
{
    public string nama;
    public int size;
    public string kemampuan;

    public virtual void Menyerang()
    {
        Console.WriteLine("\n {0} menyerang dengan cara {1}", this.nama, this.kemampuan);
    }
}
```

- **Virus.cs**

class

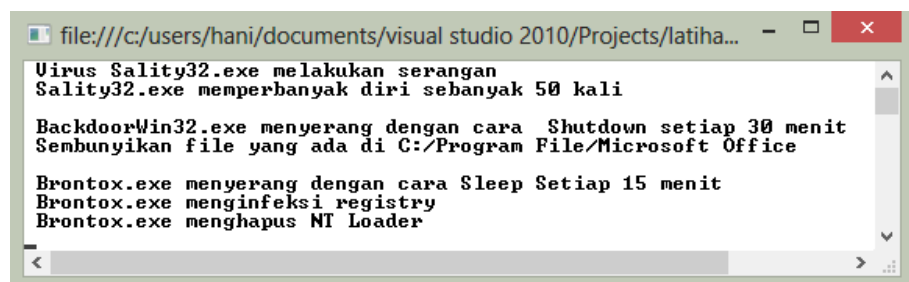
Menimpa void menyerang() yg ada pada parent

```
class Virus: Malware
{
    public int banyak;

    public void MemperbanyakDiri()
    {
        Console.WriteLine(" {0} memperbanyak diri sebanyak {1} kali", this.nama, this.banyak*5);
    }

    public override void Menyerang()
    {
        Console.WriteLine(" Virus {0} melakukan serangan ", this.nama);
    }
}
```

- **Running Program**



The screenshot shows a console window titled "file:///c:/users/hani/documents/visual studio 2010/Projects/latiha...". The output text is as follows:

```
Virus Sality32.exe melakukan serangan
Sality32.exe memperbanyak diri sebanyak 50 kali

BackdoorWin32.exe menyerang dengan cara Shutdown setiap 30 menit
Sembunyikan file yang ada di C:/Program File/Microsoft Office

Brontox.exe menyerang dengan cara Sleep Setiap 15 menit
Brontox.exe menginfeksi registry
Brontox.exe menghapus NT Loader
```