



# Back End Development 1

## sesi 5

---



# **Object Oriented Programming**

## Object Oriented Programming - Sesi 05

### Konsep

Paradigma Pemrograman dengan memecah alur program menjadi modul-modul sederhana yang disebut dengan objek.

Pendekatan OOP berbeda dengan Prosedural Programming seperti Cobol, Basic, Pascal dll yang memecah program menjadi function.

Sudah di mention di sesi awal jika C# menggunakan konsep OOP.

Untuk memudahkan kita dalam mengenal OOP, Perlu diperhatikan istilah dibawah ini akan selalu kita gunakan, yaitu :

1. Class, merupakan pengelompokkan. Contoh : Class Smartphone dan Laptop
2. Objek, merupakan isi dari suatu class dan tentunya mempunyai ciri unik, contoh :  
Class Smartphone : Samsung, iPhone, XiaoMi  
Class Laptop : Asus, Lenovo, Acer
3. Property, perlengkapan yang dimiliki oleh sebuah objek, contoh : Samsung memiliki properti : Warna, Berat, Tinggi, dll. Untuk mendetailkan objek secara jelas.
4. Method, suatu aksi yang bisa dilakukan oleh suatu objek, contoh : chatting, sosial media, online shopping dll..



```

Sesi4 > C: Laptop.cs > ...
1  using System;
2  //
2 references
3  public class Laptop
4  {
5      0 references
6      ....class Laptop
7      ....{
8      ....//Attribute/Variable
9      3 references
10     ....public string merk;
11     0 references
12     ....public int ram;
13     0 references
14     ....public int memory;
15     //Method
16     0 references
17     ....public void Chatting()
18     {
19     ....Console.WriteLine("\n{0} sedang chatting", this.merk);
20     }
21     0 references
22     ....public void Sosmed()
23     {
24     ....Console.WriteLine("{0} sedang sosmed", this.merk);
25     }
26     0 references
27     ....public void OnlineShop()
28     {
29     ....Console.WriteLine("{0} sedang online shop", this.merk);
30     }
31 }

```

laptop.cs

```

C: Program.cs x
Sesi4 > C: Program.cs > Program
1  using System;
2  //
0 references
3  class Program
4  {
5      0 references
6      public static void Main(string[] args)
7      {
8          //instant objek
9          Laptop laptop1 = new Laptop();
10
11         //akses atribut
12         laptop1.merk="lenovo";
13         laptop1.ram=4;
14         laptop1.memory=128;
15
16         //Tampilkan
17         Console.WriteLine("\n Merk Laptop adalah {0}, laptop1.merk");
18         Console.WriteLine("Kapasitas Ram ada {0}", laptop1.ram);
19         Console.WriteLine("Kapasitas Memory ada {0}", laptop1.memory);
20
21         //Akses Method
22         laptop1.Chatting();
23         laptop1.Sosmed();
24         laptop1.OnlineShop();
25
26         Console.Read();
27     }
}

```

program.cs



HACKTIV8

## Object Oriented Programming - Sesi 05

# Encapsulation

Enkapsulasi sebenarnya merupakan sebuah metode untuk menyembunyikan elemen tertentu dari sebuah class sehingga tidak dapat diakses penuh atau full control oleh kelas lain, Hal ini memungkinkan pengguna lain hanya sebatas dapat membacanya saja atau read saja.

Sebelum memberikan contoh mengenai Encapsulation ada baiknya kita mengenal yang namanya access identifier karena dalam bahasa pemrograman tidaklah lepas dari namanya access identifier yang memungkinkan atau tidaknya sebuah pengguna dapat mengakses properti yang akan digunakan.

Ada beberapa access identifier yang sering digunakan diantaranya adalah :

- a. Public , berarti bahwa anggota class tersebut (attribut/method/property) dapat diakses oleh siapa saja atau class yang lain.
- b. Private, berarti bahwa anggota class tersebut hanya boleh diakses oleh class itu sendiri
- c. Protected, berarti bahwa anggota class tersebut hanya boleh diakses oleh dirinya sendiri dan Class – Class turunanya.



## Object Oriented Programming - Sesi 05

### Identifier

Tabel Hubungan antara Class ,Atribut dan Method

<b>CLASS</b>	<b>laptop</b>
<b>ATRIBUT</b> - => <b>private</b> + => <b>public</b>	+ Merk = String - Ram = int - Memory = int
<b>METHOD</b>	+Chatting() +Sosmed() +OnlineShop()

Untuk membungkus suatu objek pada C# kita dapat menggunakan metode getter dan setter , untuk lebih memahaminya mari langsung saja kita praktikan.

```
1  using System;
2
3  0 references
4  public class Pesawat
5  {
6      0 references
7      class class_pesawat
8      {
9          1 reference
10         public string nama;
11         2 references
12         private string ketinggian;
13         2 references
14         public string ketinggian
15         {
16             get { return ketinggian; }
17             set { ketinggian = value; }
18         }
19         0 references
20         public void terbang()
21         {
22             Console.WriteLine("Pesawat dengan nama {0}, sedang take off", this.nama);
23         }
24         0 references
25         public void sudahterbang()
26         {
27             Console.WriteLine("Pesawat sekarang berada pada ketinggian {0}, this.Ketinggian");
28         }
29     }
30 }
```

```
1  using System;
2
3  0 references
4  class Program
5  {
6      0 references
7      public static void Main(string[] args)
8      {
9          class_pesawat pesawat = new class_pesawat();
10         pesawat.nama = "GARUDAKU";
11         pesawat.ketinggian = "2500 kaki";
12
13         pesawat.terbang();
14         pesawat.sudahterbang();
15         Console.Read();
16     }
17 }
```

program.cs



HACKTIV8



## Object Oriented Programming - Sesi 05

# Inheritance

Dikenal dengan Pewarisan sifat yang artinya cara untuk membentuk kelas baru menggunakan kelas yang telah ditetapkan.

Ilustrasinya yaitu jika sebuah class pesawat mempunyai atribut

(+ Nama : String,  
-ketinggian : String,  
-JumlahPenumpang : int,  
-JumlahRoda : int) ,

maka atribut dari class pesawat tersebut dapat diwariskan ke kelas lainnya contoh Class Pesawat\_tempur, dan atribut pada class pesawat juga biasa digunakan pada method di class pesawat\_tempur.

Contoh :

- Buatlah projek baru dengan nama Inheritance, setelah itu tambahkan juga class baru dengan nama pesawat.cs lalu ketikan listing program dibawah.

```

1 using System;
2
3 1 reference
4 public class pesawat
5 {
6     0 references
7     class pesawat_tempur:pesawat // ini merupakan inheritance dari class pesawat
8     {
9         0 references
10        public void terbang()
11        {
12            Console.WriteLine("Pesawat Tempur dengan nama {0}, yang mempunyai jumlah roda {1},sedang berada pada ketinggian{2} dengan membawa jumlah penumpang sebanyak{3}
13            Akan meledakan senjata", this>Nama, this.JumlahRoda, this.Ketinggian, this.JumlahPenumpang);
14        }
15    }
16 }

```



```

1  using System;
2
3  0 references
4  class Program
5  {
6      0 references
7      public static void Main(string[] args)
8      {
9          pesawat pesawat = new pesawat();// instantiasi objek
10         pesawat_tempur pswttempur = new pesawat_tempur();//instantiasi objek
11
12         // seting atribut/variabel
13         pesawat>Nama = "Helly";
14         pesawat.JumlahRoda = 3;
15         pesawat.Ketinggian = "1000 kaki";
16         pesawat.JumlahPenumpang = "3 penumpang";
17         pswttempur>Nama = "AZ500TPU";
18         pswttempur.JumlahRoda = 5;
19         pswttempur.Ketinggian = "200 kaki";
20         pswttempur.JumlahPenumpang = "2 Penumpang plus kopilot dan asistance";
21
22         //memanggil method dari suatu kelas
23         pesawat.terbang();
24         pswttempur.terbangtinggi();
25         Console.Read();
26     }
27 }

```



## Object Oriented Programming - Sesi 05

# Polimorfisme

Polimorfisme merupakan kemampuan suatu objek untuk mengungkapkan banyak hal melalui satu cara yang sama.

Polimorfisme mengizinkan kelas induk untuk mendefinisikan sebuah method general (bersifat umum) untuk semua kelas turunanya, dan selanjutnya kelas-kelas turunan dapat memperbaharui implementasi dari method tersebut secara lebih spesifik sesuai dengan karakteristiknya masing-masing.

Polimorfisme adalah proses menggunakan operator atau fungsi dalam cara yang berbeda untuk input data yang berbeda. Untuk mengimplementasikan polimorfisme dapat dilakukan menggunakan dua cara yaitu:

## A. Method overloading

Method yang membuat dua atau lebih method yang bernama sama, tetapi dengan jumlah atau jenis parameter yang berbeda. Contoh :

- Buatlah sebuah projek baru ,kemudian buat juga class dengan nama data.
- Masukan dua script berikut.
- Script untuk Data.cs

```
1  using System;
2
3  0 references
4  public class overloading
5  {
6      0 references
7      class data
8      {
9          2 references
10         public string nama;
11         2 references
12         public double nomor_telepon;
13         0 references
14         public void print(string nama)
15         {
16             this.nama = nama;
17             Console.WriteLine("Namanya adalah : {0}", this.nama);
18         }
19         0 references
20         public void print(double nomor_telepon)
21         {
22             this.nomor_telepon = nomor_telepon;
23             Console.WriteLine("Nomor HP : {0}", this.nomor_telepon);
24         }
25     }
26 }
```

overloading.cs



HACKTIV8

```

1  using System;
2
3  0 references
4  class Program
5  {
6      0 references
7      static void Main(string[] args)
8      {
9          data datanew = new data();
10         datanew.print("Foxy");
11         datanew.print(1641231231);
12         Console.Read();
13     }
14 }

```

program.cs

## B. Method overriding dengan pewarisan

Overriding digunakan oleh Subclass yang berusaha memodifikasi tingkah laku yang diwarisi dari superclass.

Tujuannya adalah supaya subclass memiliki tingkah laku yang lebih spesifik.

Ini dilakukan dengan cara mendeklarasikan kembali method milik parent class di subclass.

- Contoh Buatlah sebuah projek dengan nama overriding, dan ketikan script berikut.

```

1  using System;
2
3  3 references
4  class Program
5  {
6      9 references | 9 references
7      public int bilangan1, bilangan2;
8      2 references
9      public void bilangan(int angka1, int angka2)
10     {
11         bilangan1 = angka1;
12         bilangan2 = angka2;
13     }
14     2 references
15     public virtual void penjumlahanperkaliandll()
16     {
17         Console.WriteLine("Overiding 1.....----->>>>>>");
18         Console.WriteLine("Hasil penjumlahan angka {0} dan angka {1}= {2}",bilangan1,bilangan2,bilangan1+bilangan2);
19         Console.WriteLine("Hasil Perkalian angka {0} dan angka {1}={2} \n\n", bilangan1, bilangan2, bilangan1 * bilangan2);
20     }
21 }

```



```

18 class override:Program
19 {
    2 references
20 public override void penjumlahanperkaliandll()
21 {
22     Console.WriteLine("overiding 2....----->>>>>>>");
23     Console.WriteLine("Hasil Pembagian angka {0} dan angka {1} ={2}",bilangan1,bilangan2,bilangan1/bilangan2);
24     Console.WriteLine("Hasil Pengurangan angka {0} dan angka {1}= {2}", bilangan1, bilangan2, bilangan1 - bilangan2);
25 }
    0 references
26 static void Main(string[] args)
27 {
28     Program overiding = new Program();
29     overiding.bilangan(10, 5);
30     overiding.penjumlahanperkaliandll();
31     overiding = new override();
32     overiding.bilangan(20, 5);
33     overiding.penjumlahanperkaliandll();
34     Console.Read();
35 }
36 }

```

