

**IF34348 - PEMROGRAMAN LANJUT**

# **OBJECT ORIENTED PROGRAMMING (OOP)**

**06**

Oleh : Andri Heryandi, M.T.

# DEFINISI

IF34348 - Pemrograman Lanjut

- **Object Oriented Programming (OOP) / Pemrograman Berbasis Objek (PBO)** adalah konsep pemrograman yang menerapkan konsep objek, dimana objek terdiri dari atribut (informasi-informasi mengenai objek) dan method (prosedur/proses) yang bisa dilakukan oleh objek tersebut.
- **Software/Perangkat lunak** terdiri dari objek-objek yang saling berinteraksi.



Oleh : Andri Heryandi, M.T.

# CONTOH

IF34348 - Pemrograman Lanjut

## ■ Contoh :

### ■ Object : Manusia

- **Attribut** : Nama, tinggi, umur, berat badan dll
- **Method** : Makan, Minum, Berjalan, Bekerja

### ■ Object : Windows

- **Attribut** : Left, Top, Width, Height, BackgroundColor,
- **Method** : OnClick, OnClose, OnDoubleClick

### ■ Object : Keluarga

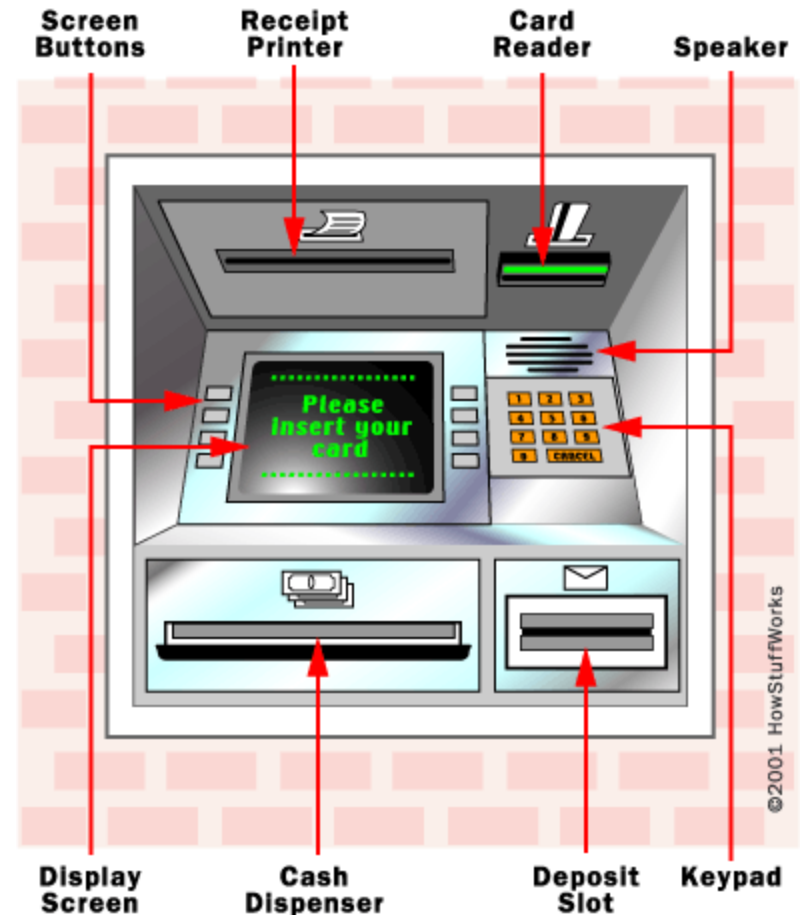
- **Attribut** : ayah, Ibu, Anak[]
- **Method** : TambahAnak,



# CONTOH

IF34348 - Pemrograman Lanjut

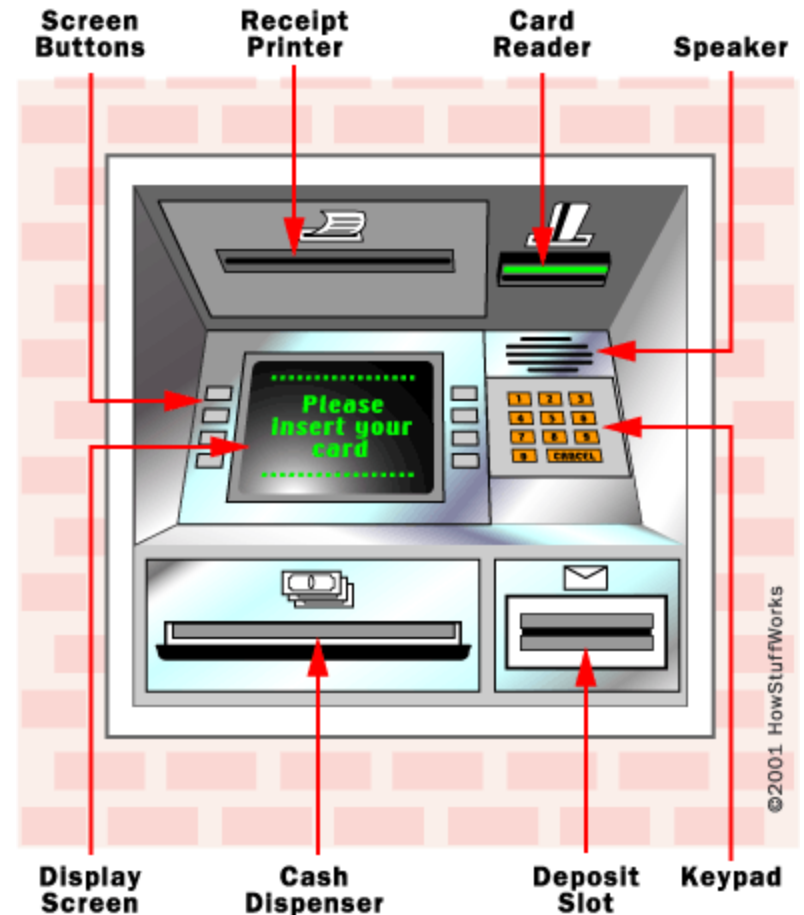
- Perhatikan mesin ATM
- Sebuah mesin ATM terdiri dari elemen-elemen berikut :
  - Display Screen (Layar)
  - Screen Button (Tombol)
  - Receipt Printer
  - Card Reader
  - Speaker
  - Cash Dispenser
  - Deposit Slot
  - Keypad
- Di dalam konsep OOP, setiap elemen tersebut berinteraksi dengan mengirimkan pesan (message) tertentu.



# CONTOH

IF34348 - Pemrograman Lanjut

- Interaksi ketika penekanan tombol Ambil Uang Rp. 200.000
  - Speaker mengeluarkan bunyi beep.
  - CashDispenser mengeluarkan uang 200.000
  - Jika uang diambil, maka Receipt Printer mencetak faktur,
  - Jika uang tidak diambil, Uang dimasukkan kembali ke Cash Dispenser,
  - Layar kembali ke menu Utama.



# FITUR-FITUR OOP

IF34348 - Pemrograman Lanjut

- **Encapsulation**

Penggabungan antara data (attribut) dengan prosedur (method) yang mengolahnya.

- **Inheritance**

Penurunan sifat (attribut dan method) dari Class Parent (SuperClass) ke Class Child (SubClass). Ini menandakan bahwa OOP mendukung konsep code reuse dimana data-data yang ada di class parent bisa di kenal di kelas child.

- **Polymorphism**

Sebuah kemampuan dari sebuah objek untuk bekerja dalam berbagai bentuk. Penggunaan umum polymorphism biasanya digunakan ketika sebuah reference dari class parent digunakan untuk mengacu ke class child.



# CLASS DAN OBJECT

IF34348 - Pemrograman Lanjut

## ■ Class

Class adalah cetak biru/prototipe/pendefinisian dari suatu benda. Didalam class-lah attribut dan method suatu object didefinisikan.

Contoh : Manusia, Window

## ■ Object

Object adalah bentuk instance/nyata/real/hidup dari sebuah class.

Contoh :

- Shelly:Manusia (Object Shelly mempunyai Class Manusia)
- Form1:Window (Object Form1 mempunyai class Window)



# CLASS DAN OBJECT

IF34348 - Pemrograman Lanjut

- Setiap object pasti memiliki class (sebagai templatanya)
- Setiap object harus diinstansiasi/dihidupkan terlebih dahulu sebelum digunakan. Instansiasi sebuah objek dapat dilakukan dengan keyword new. Contoh berikut :

```
NamaClass NamaObject;
```

```
NamaObject=new NamaClass(parameter_konstruktornya);
```

- Untuk mengakses atribut atau method suatu object, gunakan tanda titik setelah nama objeknya.

```
Kucing catty=new Kucing("Catty");
```

```
catty.warna="putih";
```

```
catty.jalan();
```

Mengakses  
atribut/method





# CONTOH PENGGUNAAN CLASS

IF34348 - Pemrograman Lanjut

- **String** di java sebenarnya adalah sebuah **Class**.
- **Method-method** yang ada di class **String**
  - **charAt(index)** : Mereturnkan huruf pada posisi ke-index. Index 0 menunjukkan huruf pertama.
  - **equals(string\_lain)** : Mereturnkan true jika isi string sama dengan isi string\_lain (case sensitive).
  - **equalsIgnoreCase(string\_lain)** : Mereturnkan true jika isi string sama dengan string lain dengan mode perbandingan case insensitive.
  - **length()** : Mereturnkan berapa banyak huruf dalam string.
  - **toUpperCase()** : Mereturnkan string yang berisi bentuk kapital dari stringnya.
  - **toLowerCase()** : Mereturnkan string yang berisi bentuk huruf kecil dari stringnya.



# CONTOH PENGGUNAAN CLASS

IF34348 - Pemrograman Lanjut

```
public class TestString {  
    public static void main(String[] args) {  
        String nama="Universitas Komputer Indonesia";  
        System.out.println("ISI STRING : "+nama);  
        System.out.println("Panjang      : "+nama.length());  
        System.out.println("Upper Case  : "+nama.toUpperCase());  
        System.out.println("Lower Case : "+nama.toLowerCase());  
        System.out.println("=UNIKOM      : "+nama.equals("UNIKOM"));  
    }  
}
```

```
ISI STRING : Universitas Komputer Indonesia  
Panjang    : 30  
Upper Case : UNIVERSITAS KOMPUTER INDONESIA  
Lower Case : universitas komputer indonesia  
=UNIKOM    : false
```



# MEMBUAT CLASS SEDERHANA

IF34348 - Pemrograman Lanjut

## ■ Sintak pembuatan class sederhana

```
class NamaKelas{  
    tipe_data nama_atribut;  
    tipe_data nama_atribut;  
    nama_kelas nama_object;  
  
    NamaKelas(parameter){  
        ... // isi konstruktor  
    }  
    void nama_method(parameter){  
        ... // isi method berbentuk procedure  
    }  
    tipe_data nama_method_function(paramter){  
        ... // isi method berbentuk function  
        return ...;  
    }  
}
```

Daftar Atribut

Daftar Method  
(Procedure/Function  
/Constructor)



# CONTOH CLASS SEDERHANA

IF34348 - Pemrograman Lanjut

- **Buatlah sebuah class bernama Titik yang digunakan untuk menyimpan sebuah titik koordinat.**
  - Setiap titik mempunyai atribut posisi X dan atribut posisi Y.
  - Class ini harus dapat melakukan hal berikut :
    - Memberikan nilai default ( $X=0$ ,  $Y=0$ ) ketika X dan Y belum didefinisikan.
    - Mengisi Nilai X
    - Mengisi Nilai Y
    - Mengisi Nilai X dan Y (sekaligus)
    - Menampilkan nilai X dan Y
    - Pindah ke Koordinat Lain (berdasarkan jarak atau ke titik tertentu)
    - Menghitung Jarak Ke Titik Lain



# CONTOH CLASS SEDERHANA

IF34348 - Pemrograman Lanjut

```
class Titik {  
    double x;  
    double y;  
    Titik(){  
        x=0;  
        y=0;  
    }  
    Titik(double x1,double y1){  
        x=x1;  
        y=y1;  
    }  
    void tampil(){  
        System.out.println("("+x+", "+y+")");  
    }  
    void pindah(double x1, double y1){  
        x=x1;  
        y=y1;  
    }  
}
```



Oleh : Andri Heryandi, M.T.

# CONTOH CLASS SEDERHANA

IF34348 - Pemrograman Lanjut

```
void pindah(Titik t){
    x=t.x;
    y=t.y;
}
void isiX(double x1){
    x=x1;
}
void isiY(double y1){
    y=y1;
}
void isiXY(double x1,double y1){
    x=x1;
    y=y1;
}
```



Oleh : Andri Heryandi, M.T.

# CONTOH CLASS SEDERHANA

IF34348 - Pemrograman Lanjut

```
double jarakKe(double x1, double y1){  
    double jarak;  
    jarak=Math.pow(Math.pow(x-x1,2)+Math.pow(y-  
y1,2),0.5);  
    return jarak;  
}  
double jarakKe(Titik t2){  
    double jarak;  
    jarak=Math.pow(Math.pow(x-t2.x,2)+Math.pow(y-  
t2.y,2),0.5);  
    return jarak;  
}  
}
```


$$\text{Math.pow}(x,y) = x^y$$

# MEMBUAT KELAS TESTER

IF34348 - Pemrograman Lanjut

- Kelas tester digunakan sebagai class untuk melakukan uji coba terhadap class yang telah dibuat.
- Sebaiknya kelas tester dibuat terpisah dari file classnya.



Oleh : Andri Heryandi, M.T.



# MEMBUAT KELAS TESTER

IF34348 - Pemrograman Lanjut

```
public class TitikTester {  
    public static void main(String[] args) {  
        Titik t1,t2;  
        t1=new Titik();// Tanpa parameter x=0, y=0  
        t2=new Titik(9,4);// x=9, y=4  
        System.out.print("T1 : ");  
        t1.tampil();  
        System.out.print("T2 : ");  
        t2.tampil();  
        t1.pindah(5,1);  
        System.out.print("Setelah pindah T1 : ");  
        t1.tampil();  
        double jarak;  
        jarak=t1.jarakKe(t2);  
        System.out.println("Jarak dari T1 ke T2 : "+jarak);  
    }  
}
```

```
T1 : (0.0,0.0)  
T2 : (9.0,4.0)  
Setelah pindah T1 : (5.0,1.0)  
Jarak dari T1 ke T2 : 5.0
```



Oleh : Andri Heryandi, M.T.

# CLASS CALCULATOR

IF34348 - Pemrograman Lanjut

## ■ Buatlah class Calculator

### ■ **Attribut** :

- Operan1 bertipe double
- Operan2 bertipe double

### ■ **Method** :

- IsiOperan1(double x) : Mengisi atribut operan1 dengan nilai x
- IsiOperan2(double x) : Mengisi atribut operan2 dengan nilai x
- Tambah() : Mereturnkan nilai Operan1 + Operan2
- Kurang() : Mereturnkan nilai Operan1 - Operan2
- Kali() : Mereturnkan nilai Operan1 \* Operan2
- Bagi() : Mereturnkan nilai Operan1 / Operan2
- Pangkat() : Mereturnkan nilai Operan1^Operan2

## ■ Kelas tersebut harus bisa dijalankan dengan menggunakan Class CalculatorTester (di slide berikutnya)



# CLASS CALCULATORTESTER

IF34348 - Pemrograman Lanjut

```
public class CalculatorTester {  
    public static void main(String[] args) {  
        Calculator c=new Calculator();  
        c.isiOperan1(7);  
        c.isiOperan2(5);  
        System.out.println("Tambah      : "+c.tambah());  
        System.out.println("Kurang     : "+c.kurang());  
        System.out.println("Kali       : "+c.kali());  
        System.out.println("Bagi       : "+c.bagi());  
        System.out.println("Pangkat    : "+c.pangkat());  
    }  
}
```

```
Tambah      : 12.0  
Kurang     : 2.0  
Kali       : 35.0  
Bagi       : 1.4  
Pangkat    : 16807.0
```



Oleh : Andri Heryandi, M.T.

# CLASS NILAI

IF34348 - Pemrograman Lanjut

## ■ Buatlah class Nilai

- **Attribut :**
  - Quis bertipe double
  - UTS bertipe double
  - UAS bertipe double
- **Method :**
  - `setQuis(double x)` : Mengisi nilai quis
  - `setUTS(double x)` : Mengisi nilai UTS
  - `setUAS(double x)` : Mengisi nilai UAS
  - `getNA()` : Mereturnkan nilai akhir berupa double
  - `getIndex()` : Mereturnkan index berupa char
  - `getKeterangan()` : Mereturnkan keterangan berupa String

## ■ Rumus NilaiAkhir

$$NA = 20\% \text{ QUIZ} + 30\% * UTS + 50\% * UAS$$

## ■ Aturan Index

- NA 80..100 Index='A'
- NA 68..80 Index='B'
- NA 56..68 Index='C'
- NA 45..56 Index='D'
- NA 0..45 Index='E'

## ■ Aturan Keterangan

- Index='A' : Sangat Baik
- Index='B' : Baik
- Index='C' : Cukup
- Index='D' : Kurang
- Index='E' : Sangat Kurang



# CLASS NILAITESTER

IF34348 - Pemrograman Lanjut

```
public class NilaiTester {  
    public static void main(String[] args) {  
        Nilai n=new Nilai();  
        n.setQuis(60);  
        n.setUTS(80);  
        n.setUAS(75);  
        System.out.println("Quis           : "+n.Quis);  
        System.out.println("UTS           : "+n.UTS);  
        System.out.println("UAS           : "+n.UAS);  
        System.out.println("NA            : "+n.getNA());  
        System.out.println("Index         : "+n.getIndex());  
        System.out.println("Keterangan    : "+n.getKeterangan());  
    }  
}
```

```
Quis           : 60.0  
UTS            : 80.0  
UAS            : 75.0  
NA             : 73.5  
Index          : B  
Keterangan     : Baik
```



Oleh : Andri Heryandi, M.T.