



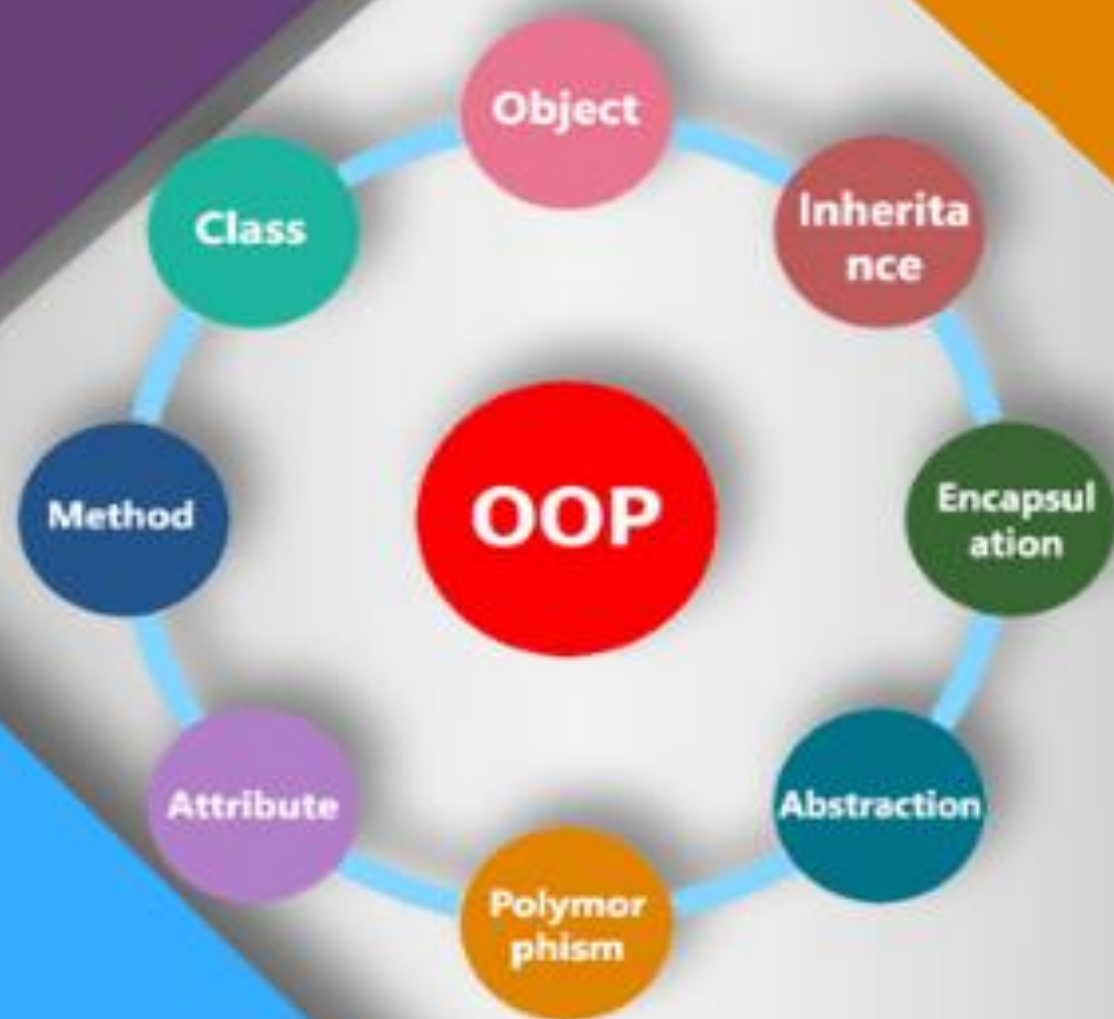
BBPPMPV  
BMTI

# KONSEP OOP

PEMROGRAMAN

BERORIENTASI

OBJEK



## KONSEP OOP

- Pemrograman yang berorientasi obyek (object oriented programming = OOP) adalah suatu bentuk perancangan modular yang berdasarkan pemikiran dalam bentuk obyek-obyek, operasi-operasi, metoda-metoda dan pesan-pesan yang melalui obyek terkait.
- Semua data dan fungsi di dalam paradigma ini dibungkus dalam Class-Class atau objek-objek. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya.

## KARAKTERISTIK OOP

- Objek  
membungkus data dan fungsi bersama menjadi suatu unit dalam sebuah program komputer; objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek.

### Contoh Objek:

- suatu faktur,
- sebuah organisasi,
- suatu bentuk ketajaman dalam program untuk gambar,

## CLASS

- Class adalah suatu implementasi dari sebuah tipe obyek. Class memiliki suatu struktur data dan metoda-metoda yang memperinci operasi-operasi yang dapat digunakan dalam struktur data yang bersangkutan. Implementasi Class terdiri dari struktur data untuk masing-masing obyek.
- Contoh : Class Karyawan dapat terdiri dari data tentang jabatan (kedudukan), gaji, ekstensi pesawat telepon dan lain-lain.



## OBJEK DAN CLASS

- Dalam konsep OOP, objek adalah instant dari class. Objek dalam konsep OOP terdiri dari 2 elemen prnyusun yaitu atribut dan method. Atribut adalah ciri-ciri dan keadaan dari objek, sedangkan method adalah perilaku objek (fungsi yang digunakan untuk memanipulasi atribut)

## CONTOH CLASS

```
package siswa;
```

```
// @author cianni
```

```
class Siswa {
```

```
String nama= "Zahara Ika ";
```

```
public static void main (String[] args)
```

```
{
```

```
    Siswa nm = new Siswa();
```

```
    System.out.print("nama siswa adalah:" +nm.nama);
```

```
}
```

```
}
```

Nama kelas

Variable instance

Mendefinisikan objek dalam kelas (instansiasi)

## ENKAPSULASI

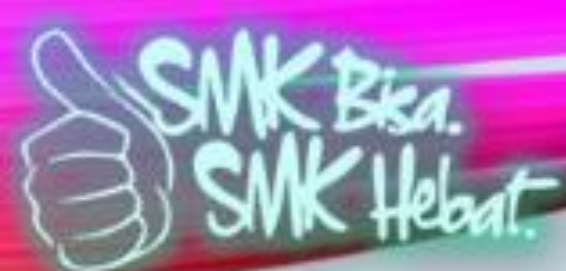
- Pembungkusan (encapsulation) merupakan suatu karakteristik OOP di mana program terbungkus (menjadi satu) data (property/field) dan perilaku(fungsi/method), artinya lebih memperhatikan aspek internal daripada aspek eksternal.



## ENKAPSULASI

- Encapsulation merupakan hal yang penting, karena hal ini memisahkan antara bagaimana tingkah laku dari obyek dan bagaimana obyek tersebut diimplementasikan. Hal ini mengizinkan implementasi obyek dimodifikasi tanpa harus memodifikasi aplikasinya.





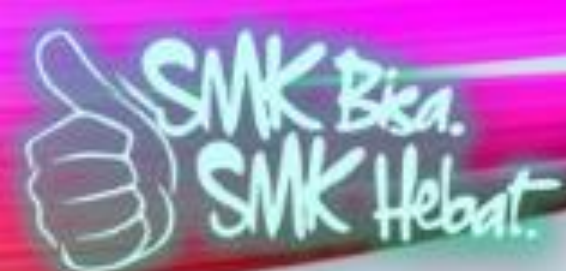
## TURUNAN (INHERITANCE)

Turunan merupakan suatu karakteristik OOP di mana suatu Class (parent class/baseclass) dapat diturunkan ke Class yang lain (child class/derived class), sehingga Class anak dapat memiliki data atau perilaku Class orang tuanya.

Inheritance merupakan pewarisan atribut dan method pada sebuah class yang diperoleh dari class yang telah terdefinisi tersebut.

## POLIMORFISME

- Polimorfisme merupakan suatu karakteristik OOP di mana objek yang berbeda dapat memakai method dengan nama yang sama pada suatu Class (class) Tidak bergantung kepada pemanggilan subrutin, bahasa orientasi objek dapat mengirim pesan; metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu di mana pesan tersebut dikirim.



## KELEBIHAN OOP

- Dengan menggunakan OOP maka dalam melakukan pemecahan suatu masalah kita tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut (terstruktur) tetapi objek-objek apa yang dapat melakukan pemecahan masalah tersebut. Sebagai contoh anggap kita memiliki sebuah departemen yang memiliki manager, sekretaris, petugas administrasi data dan lainnya. Misal manager tersebut ingin memperoleh data dari bag administrasi maka manager tersebut tidak harus mengambilnya langsung tetapi dapat menyuruh petugas bag administrasi untuk mengambilnya. Pada kasus tersebut seorang manager tidak harus mengetahui bagaimana cara mengambil data tersebut tetapi manager bisa mendapatkan data tersebut melalui objek petugas administrasi. Jadi untuk menyelesaikan suatu masalah dengan kolaborasi antar objek-objek yang ada karena setiap objek memiliki deskripsi tugasnya sendiri.