BAB I
PENGEMBANGAN BERORIENTASI OBYEK

A. KOMPETENSI DASAR

3.1 Menganalisis pengembangan berorientasi obyek
4.1 Menyajikan hasil pengembangan berorientasi obyek

B. TUJUAN PEMBELAJARAN

Setelah mempelajari bab ini, siswa diharapkan mampu:

1. Siswa dapat memahami metedologi berorientasi obyek
2. Siswa dapat memahami strutur obyek
3. Siswa dapat memahami obyek oriented analysis and design
4. Siswa dapat memahami class diagram dan sequence diagram

MATERI PEMBELAJARAN

1. PENGERTIAN OOP

Pemrograman Berorientasi Object (OOP) adalah model pemrograman yang paling banyak dipakai saat ini.

OOP telah menggantikan teknik pemrograman prosedural yang telah dipakai sejak tahun 1970-an

Java adalah bahasa yang berorientasi objek, karena itu Anda harus paham OOP agar dapat produktif menggunakan Java.

Program yang Berorientasi Objek akan terdiri dari objek-objek.

Objek-objek ini seringkali merepresentasikan apa yang ada di dunia nyata.

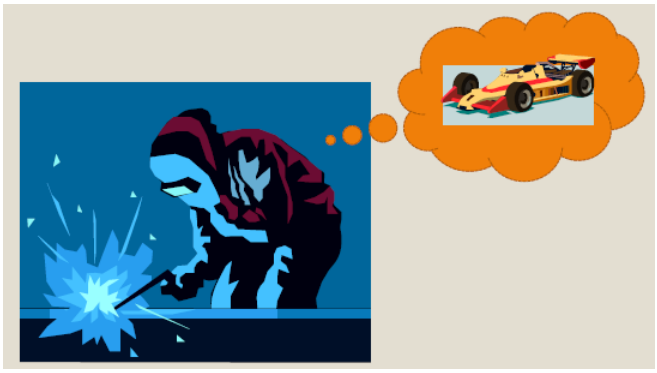
2. ISTILAH-ISTILAH DALAM OOP

- Class
- Object
- Relationship (antar class)
  a. Dependence
  b. Aggregation
  c. Inheritance
- Encapsulation
- Instantiation & Instance
- Instance Variable
- Method (mutator & accessor method)

3. CONTOH DALAM KEHIDUPAN SEHARI-HARI

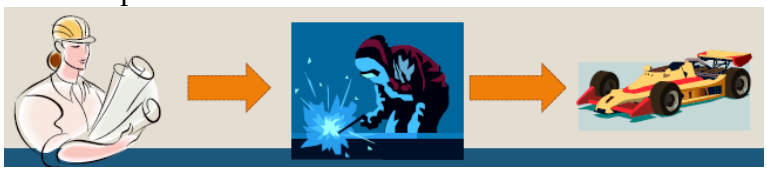
Pabrik Mobil

Sebelum seseorang mengendarai mobil, tentu Anda harus **membuat mobil** tersebut terlebih dahulu



Bagaimana cara membuat mobil ?

- **Langkah 1 :** Menggambar desain mobil.
- **Langkah 2 :** Menambahkan detil mobil tersebut, misal :
  - Desain pedal gas untuk maju
  - Desain rem untuk berhenti
  - Desain setir untuk belok kiri/kanan, dll.
- **Langkah 3 :** Proses pembuatan mobil dimulai.

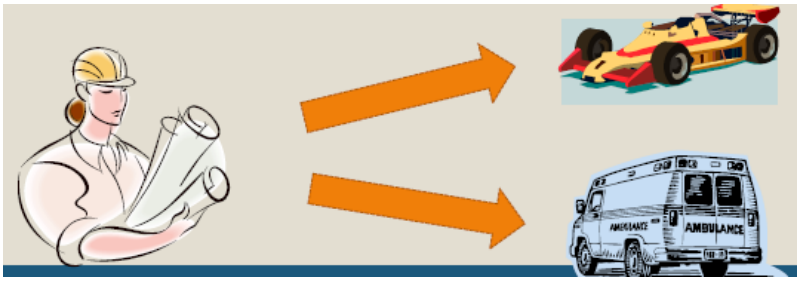


Bagaimana pedal gas bekerja ?

- Apa yang sebenarnya terjadi di dalam mesin pada saat pedal gas diinjak?
- Apakah Anda sebagai pengendara perlu tahu, bagaimana cara kerja detil dari pedal gas?
- Tentu tidak. Yang perlu Anda ketahui hanyalah bahwa kalau pedal gas diinjak akan mempercepat laju mobil!
- Artinya, detil dan cara kerja pedal disembunyikan dari penggunanya.
- Anda cukup tahu aturan dan fungsinya saja:
  Injak pedal gas sebelah tengah maka mobil akan melaju

Desain mobil ⟶ Obyek mobil

- Jadi kesimpulannya, sebelum Anda membuat objek mobil, seseorang harus merancang terlebih dahulu.
- Di dalam rancangan tersebut, semua detil cara kerja mobil disembunyikan dari para penggunanya kelak.
- Setelah desain mobil dibuat, maka kita dapat mulai membuat berbagai objek mobil yang diinginkan.



Apa hubungannya dengan OOP?

- Sama halnya dengan penyembunyian detil PEDAL pada sebuah MOBIL, di OOP kita menyembunyikan detil pemrograman di dalam sebuah **METHOD** (FUNGSI)
- METHOD-METHOD ini dibuat dan disimpan bersama-sama di dalam suatu desain yang disebut dengan **CLASS**.
    - METHOD → injak gas, injak rem, belok kiri, belok kanan, dll
    - CLASS → rancangan mobil

4. CLASS DAN OBYEK

CLASS adalah suatu template/blueprint/rancangan dari object yang akan dibuat.
OBJECT adalah sesuatu yang diciptakan dari Class.
Analogi lain :

a. Class = cetakan kue
b. Object = kue-nya

5. MERANCANG APLIKASI PROGRAM BERBASIS OOP

Sekarang, Anda diminta merancang suatu program untuk sebuah BANK:

a. Di aplikasi BANK, pasti ada yang namanya REKENING
b. Dengan menggunakan konsep yang sama: Anda dapat mendesain sebuah CLASS yang merepresentasikan REKENING.
c. REKENING tersebut didesain sedemikian rupa sehingga memiliki 3 METHOD, yaitu:
    - METHOD untuk menyimpan uang (tabung)
    - METHOD untuk mengambil uang (tarik)
    - METHOD untuk memeriksa saldo (cek)
    -

Analogi pabrik mobil danprogram BANK

PABRIK MOBIL	PROGRAM BANK
Desain MOBIL	Desain CLASS REKENING
Detil cara kerja MOBIL: -Injak pedal -Injak rem -Stir kiri/kanan	Detil REKENING: -Method menabung -Method menarik uang -Method memeriksa saldo
MOBIL memiliki atribut: -Beberapa lampu -Beberapa ban -Beberapa kursi -Satu setir, dll	REKENING memiliki atribut: -Nomor -Nama Pemilik -Jumlah saldo
Buat objek MOBIL untuk Anda kendarai	Buat objek REKENING untuk Anda miliki

TABEL 1: MOBIL vs REKENING

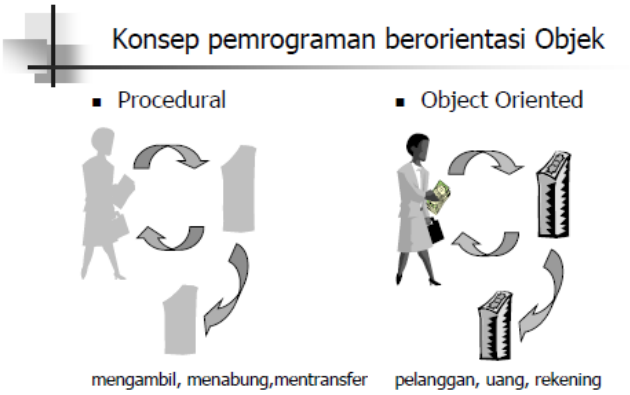
6. MEMBUAT CLASS DI JAVA REKENING.JAVA

```
public class Rekening {
}//end class Rekening
Note:class selalu diawali dengan huruf KAPITAL
).
```

Class rekening + method

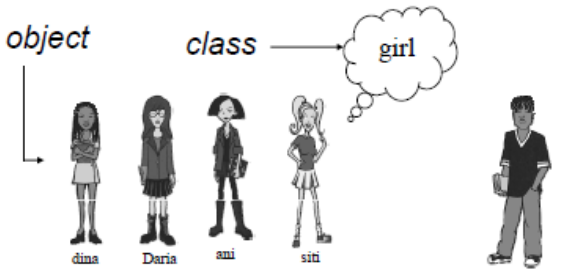
```
public class Rekening{
        // Method untuk mengecek saldo
        public void cekSaldo() {
                System.out.println("Selamat Datang di Rekening Anda!" );
                System.out.println("Saldo Anda adalah" + 150000);
        } // end method cekSaldo
} // end class Rekening
```

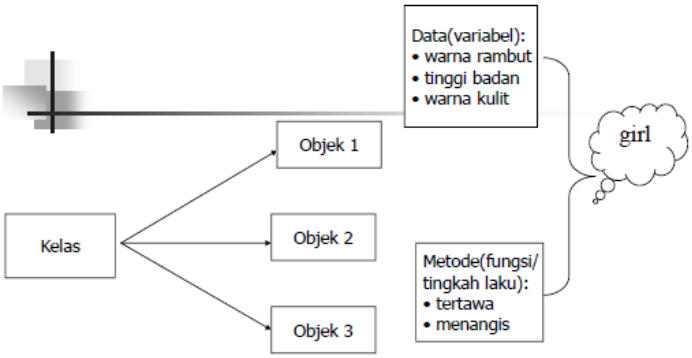
NOTE: class REKENING **hanya bisa di-compile**, tidak bisa di-run
Hal ini dikarenakan class Rekening bukanlah program aplikasi
(tidak memiliki main method)



Objek dan kelas

- Kelas adalah pemodelan dari objek yang berisi informasi tentang karakteristik(data) dan tingkah laku yang dimiliki oleh objek tersebut(metode), sedangkan objek merupakan perwujudan dari suatu kelas .

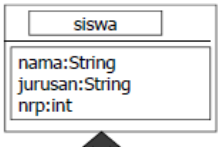




Menciptakan Kelas

- Kelas pada java didefinisikan dengan menggunakan kata kunci class
- Bentuk umum:

```
class namakelas
{
    class body
}
```



- Contoh:

```
class siswa
{
   String nama;
   String jurusan;
   int nrp;
}
```

UML class diagram

Membuat Objek dari Suatu Kelas

- Dibutuhkan operator new untuk membuat objek dari suatu kelas
- Bentuk umum:

```
namakelas variabelreferensiobjek=new namakelas();
```

- Contoh:

siswa a=new siswa();

Ungkapan diatas merupakan bentuk singkat dari proses berikut:

a. Mendeklarasikan variabel referensi objek
   siswa a;
b. Menciptakan objek dari kelas, dengan operator new
   new siswa();
a. Menugaskan(meng-assign) variabel kepada objek yang telah dibuat
   a=new siswa();

Mengakses Variabel dari suatu kelas

Program siswaku.java:

```
class siswa
{
   String nama;
   String jurusan;          variabel
   int nrp;
}
public class siswaku
{                                   variabelreferensiobjek.nama variabel
   siswa a=new siswa();
   a.nama="cita";
   a.jurusan="telkom";
   a.nrp=7206;
   System.out.println("nama" + a.nama + "jurusan"
   + a.jurusan+"nrp"+a.nrp);
}
```

Mengakses Variabel dan Metode dari suatu kelas

Contoh:siswalg.java

Variabelreferensiobjek.nama metode

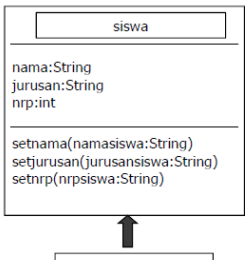
```
class siswa
{
   String nama;
   String jurusan;
   int nrp;                          metode

   void setnama(String namasiswa)
   {
        nama=namasiswa;
   }
   void setjurusan(String jurusansiswa)
   {
        jurusan=jurusansiswa;
   }
   void setnrp(String nrpsiswa)
   {
        nrp=nrpsiswa;
   }

}
public class siswalg
{

}
```

Apa yang harus ditambahkan
Agar muncul tampilan:
Cita adalah namaku telkom adalah
Jurusanku dan nrpku adalah 7206



UML class diagram

7. SEJARAH METEDOLOGI BERORIENTASI OBYEK

Metodologi berorientasi obyek mulai berkembang ketika grady Booch pada tahun 80 anmempublikasikan suatu paper bagaimana melakukan perancangan untuk bahasa ADA namun memberi judul paper tersebut dengan : Obyek Oriented Design. Selanjutnya ide tersebut terus ia kembangkan sampai tahun 90 an.

Peter Coad dan Yourdon

Pada tahun 1991 Peter Coad dan Yourdon memperkenalkan metode berorientasi obyek yang sederhana. Metode ini menjadi cepat popular karena mendukung layanan layanan yang terdapat pada c++

Pada waktu itu c++ merupakan bahasa pemograman berorientasi obyek yang paling popular

Rumbaugh

Pada tahun 1991 juga rumbaugh memperkenalkan object modelling technique (OMT). Pendekatan yang digunakan tidak jauh berbeda dengan pendekatan yang digunakan coad Yourdon namun dengan notasi yang berbeda. OMT tidak hanya sepenuhnya berbasis pada data driven tapi juga memisahkan proses dari data dengan penggunaan data flow diagram yang terpisah dengan diagram class. OMT juga menggunakan notasi state transition diagram untuk memodelkan aspek dinamis system.

Ivar Jacobshon

Pada tahun 1994 Ivar Jacopshon memperkenalkan konsep use case dan object oriented software engineering. Pada tahun 1994 itu juga yaitu bulan oktober 1994 booch, rumbaugh dan jacobshon mempelopori usaha untuk penyatuan notasi pendekatan berorientasi obyek. Pada tahun 1995 dihasilkan drive pertama dari UML (versi 0.8).

Sejak 1996 pengembangan tersebut di koordinasikan oleh object managemen group (UMG-http://www.OMG.ORG).

UML

Tahun 1997 UML versi 1.1 muncul, Booch, Rumbaugh dan Jacobson menyusun tiga buku serial tentang UML pada tahun 1999. Sejak saat itulah UML telah menjelma menjadi standart bahasa pemodelan untuk aplikasi berorientasi obyek.

Kenapa metodologi berorientasi obyek ?

Menaikkan tingkat keterpakaian kembali (reusability)

Menghilangkan kompleksitas transisi antar tahap pada pengembangan system

Pendekatan terstruktur mendukung abstraksi pada level fungsional

8. KONSEP DALAM BERORIENTASI OBYEK

Teknik baru dalam melihat permasalahan system.

Memandang system yang akan dikembangkan sebagai kumpulan obyek.

9. APAKAH YANG DISEBUT OBYEK?

Sesuatu yang memiliki dunia nyata.

Sesuatu yang mampu menyimpan informasi (status), dan mempunyai operasi (perilaku).

Mempunyai siklus hidup, diciptakan, dimanipulasi dan dihancurkan.

10. APAKAH YANG DISEBUT KELAS

Kumpulan dari obyek yang memiliki karakteristik yang sama.
Definisi static dari himpunan obyek yang sama.
Mempunyai sifat (atribut), perilaku (operasi), hubungan (relasi) dan arti.
Suatu kelas dapat diturunkan dari kelas yang lain.

11. ATRIBUT

Atribut mempresentasikan karakteristik atau keadaan obyek. Pada contoh kasus diatas, sebuah mobil dapat memiliki atribut warna, harga dan pembuat.
Pada tataran implementasi warna dapat direpresentasikan sebagai suatu string (domain nilainya misalya : merah,biru,kuning dll)
Harga dapat berupa bilangan floating point atau bilangan integer.
Sedangkan pembuat dapat bertipe struktur yang terdiri dari nama, identitas, karporat dll

12. METODE

Metode adalah suatu fungsi atau prosedur yang didefinisikan untuk dapat mengakses keadaan internal suatu obyek dari suatu kelas. Tiap fungsi atau prosedur mendefinisikan dan mendeskripsikan perilaku khusus suatu obyek
Sebagai contoh : kelas pegawai memiliki metode hitung gaji.
Metode sebenarnya merupakan antarmuka yang disediakan untuk dapat memanfaatkan perilaku obyek tersebut.
Sebagi contoh : Jika diinginkan dilakukannya perhitungan gaji, maka message ‘hitung gaji’ harus dikirimkan ke obyek pegawai.

13. ENCAPTULATION

Pembukusan atribut data dan layanan (operasi-operasi) yang dipunyai obyek untuk menyembunyikan implementasi dari obyek sehingga obyek lain tidak mengetahui cara kerjanya.
Konsep ini bertujuan untuk menyembunyikan informasi dan karakteristik obyek. Obyek dapat dimanfaatkan hanya dengan cara memanggil metode yang dimiliki obyek tersebut.

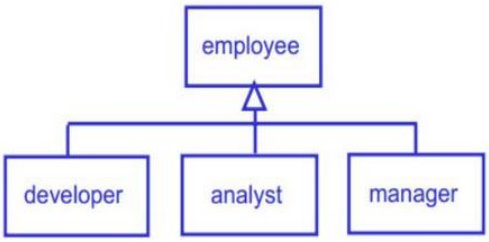
14. INHERITTANCE

Mekanisme yang memungkinkan suatu kelas obyek mewarisi sebagian atau seluruh definisi kepada kelas obyek lain.
Konsep pewarisan memungkinkan suatu obyek dibangun dari obyek lain. Pada konsep ini akan ada kelas yang bertindak sebagai parent classatau dikenal sebagai super kelas. Sedangkan kelas turunannya menjadi subkelas. Kelas yang diturunkan dari kelas lain akan memiliki karakteristik dan perilaku yang dimiliki superkelasnya.

15. HIRAKI KELAS

Sistem berorientasi obyek mengorganisasi kelas kedalam hiraki subclass-superclass
Perbedaan karakteristik dan perilaku digunakan sebagai dasar penilaian untuk membedakan antara kelas dan sub kelas.

Gambar menunjukkan Hiraki kelas. Employee adalah super kelas, sedangkan developer, analyst dan manager adalah sub kelas.



16. MESSAGE

Message pada dasarnya adalah pemanggilan fungsi. Namun message berbeda dari pemanggilan subrutin.

Dengan message yang sama dua obyek berbeda dapat melakukan operasi yang berbeda pula.

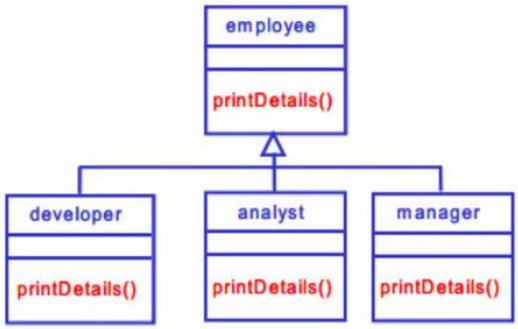
Konsep ini dikenal sebagai Polymorphisme

17. POLYMORPHISME

Kemampuan suatu kelas obyek digunakan dibanyak tujuan yang berbeda dengan pernyataan nama yang sama. Konsep ini memungkinkan suatu metode yang berada pada beberapa kelas yang berbeda dapat memliki perilaku berbeda.

Dengan konsep ini memudahkan untuk menulis suatu kode yang memiliki reusability tinggi.

Contoh: **Polymorphisme**



TUGAS INDIVIDU

PERTEMUAN 1

1. Apa yang dimaksud dengan Obyek?
2. Apa yang dimaksud dengan Kelas?
3. Apa yang dimaksud dengan Atribut?
4. Apa yang dimaksud dengan Service?
5. Jelaskan mengenai Klasifikasi Obyek!
6. Apa yang dimaksud dengan Metodologi?
7. Jelaskan pengertian Metodologi Berorientasi Obyek!
8. Sebutkan metode berorientasi obyek!
9. Sebutkan dan jelaskan keuntungan metodologi berorientasi obyek!
10. Mengapa metodologi berorientasi obyek?

PERTEMUAN 2

1. Jelaskan pengertian dasar Analisis!
2. Jelaskan analisis berorientasi obyek!
3. Jelaskan tujuan analisis!
4. Sebutkan tahapan analisis!
5. Sebutkan tahapan pelaksanaan analisis berorientasi obyek!
6. Jelaskan Identifikasi kelas dan obyek!
7. Jelaskan Identifikasi atribut dan layanan!
8. Jelaskan pengertian dasar Perancangan!
9. Jelaskan perancangan berorientasi obyek!
10. Jelaskan tujuan perancangan berorientasi obyek!

PERTEMUAN 3

1. Apa yang dimaksud dengan Class Diagram?
2. Sebutkan dan jelaskan elemen-elemen Class Diagram!
3. Apa yang dimaksud dengan Visibilitas Kelas?
4. Apa yang dimaksud dengan Multiplicity Kelas?
5. Apa yang dimaksud dengan Paket (Package)?
6. Apa yang dimaksud dengan Sequence Diagram?
7. Jelaskan tujuan Sequence Diagram!
8. Gambarkan dan jelaskan simbol komponen Sequence Diagram!
9. Jelaskan apa yang dimaksud dengan Partisipan dan Message!
10. Jelaskan apa yang dimaksud dengan Lifeline, Activation, Time dan Return!