Pemrograman Berbasis Objek
Class 1

Adam Mukharil Bachtiar

Teknik Informatika UNIKOM

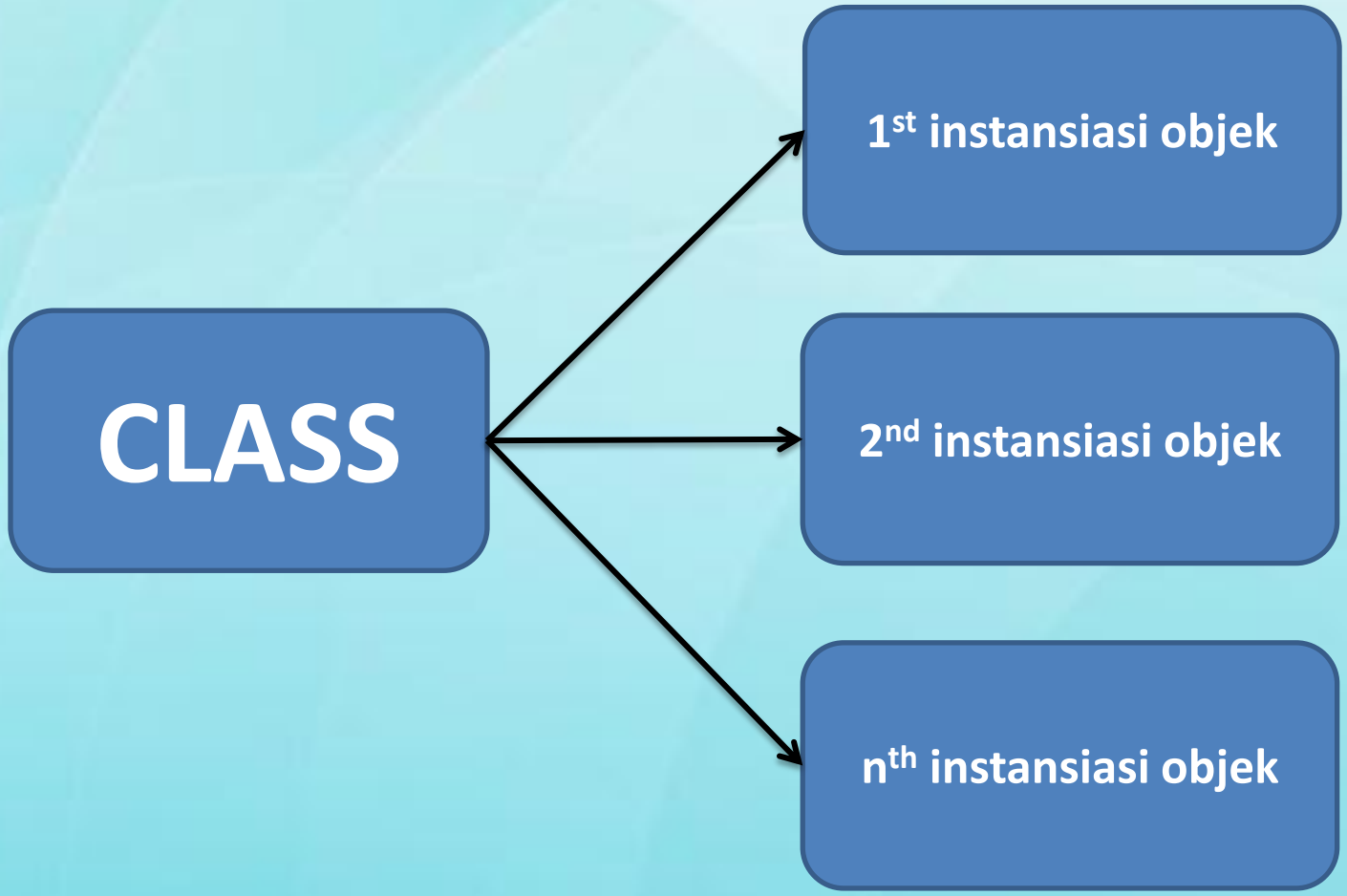
Konsep Dasar PBO

1. Pengertian class
2. Hubungan class dan objek
3. Langkah-langkah pembuatan class
4. Hak akses pada class
5. Encapsulasi
6. Class dan objek C++
7. Class dan objek Java

Class

1. Design, template, atau Blue-Print.
2. Struktur data dari sebuah objek.
3. Dari sebuah class bisa dibuat banyak objek.

Hubungan Class dengan Objek



Langkah-Langkah Pembuatan Class

1. Write your class

2. Write a tester (TestDrive) class

3. In your tester, make an object and access the object's variables and methods

Tapi sebelum itu.....

Kita bongkar hak akses class!!!

Hak Akses Pada Class

1. **Private**

   variabel/method hanya dapat diakses oleh kelas itu sendiri.

2. **Protected**

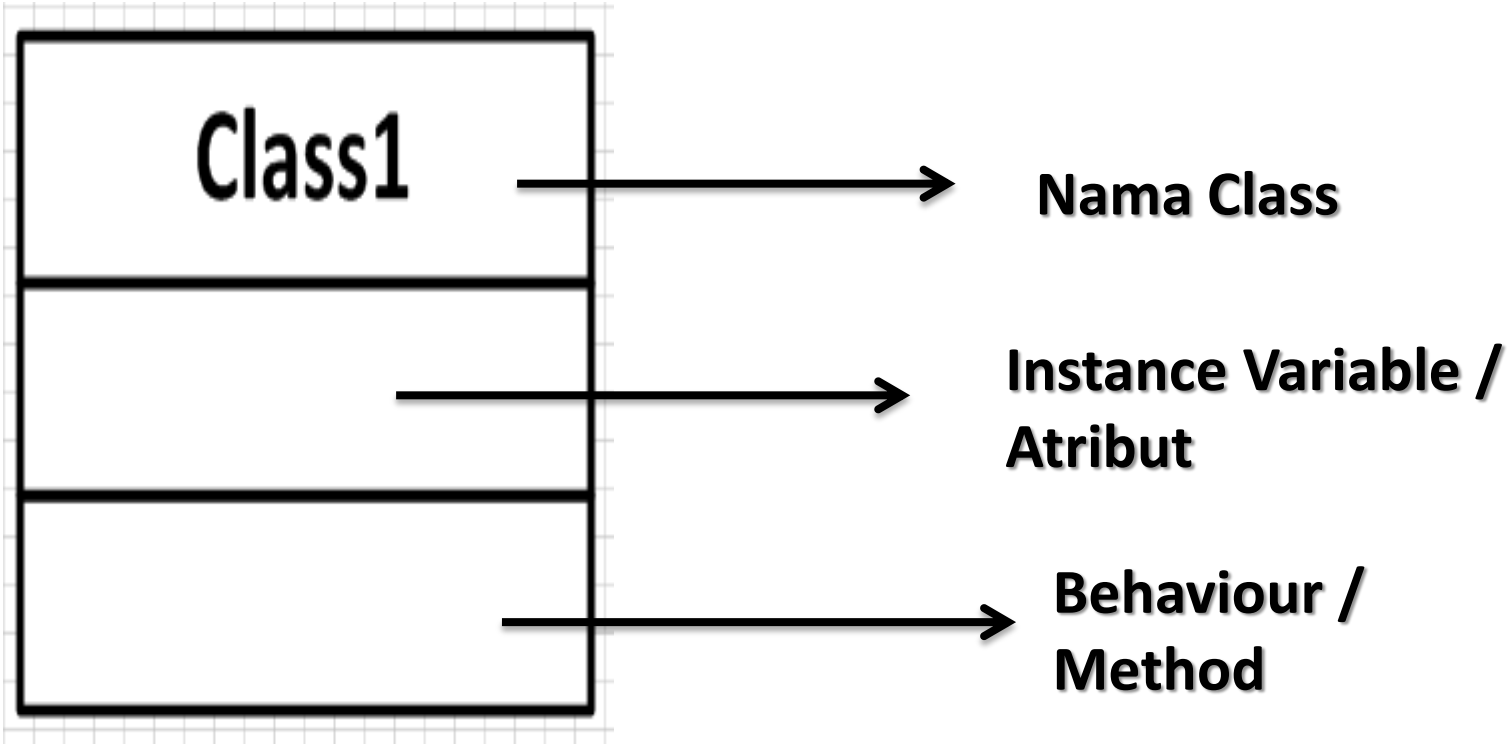
   a. Variabel/method dapat diakses oleh semua kelas turunan.

   b. Variabel tidak dapat diakses dalam pola use (bukan sebagai inheritance).

3. **Public**

   variabel/method dapat diakses oleh semua kelas.

Masih ingat dengan ini???



Sebetulnya ada konsep dasar OOP di balik gambar ini!!!

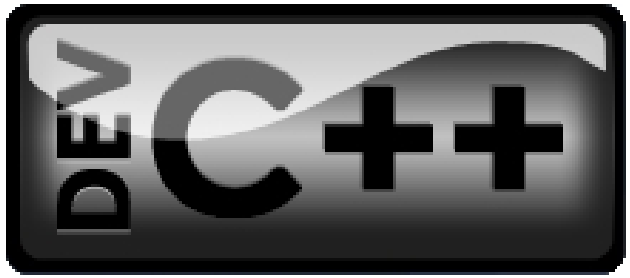
Konsep itu adalah .....

ENKAPSULASI

Enkapsulasi

1. Pembungkusan variabel dan method dalam sebuah object yang terlindungi serta menyediakan interface untuk mengakses variabel tersebut.
2. Variabel dan method yang dipunyai sebuah object bisa ditentukan hak aksesnya.

Class dan Objek C++



1 Write your class

```
#include <iostream>
#include <string.h>

using namespace std;

class orang
{
  private:
        char nama[21];
        int umur;
  public:
        void setNama(char *s){
           strcpy(nama,s);
        }

        char *getNama(){return nama;}
        int getUmur(){return umur;}
};
```

Nama kelas

Kata kunci class mengawali pendeklarasian kelas

Nama anggota class

Hak akses

2 Write a tester (TestDrive) class

```
main()
{
    //Kodemu nanti di sini


    system("PAUSE");
    return EXIT_SUCCESS;
}
```

```
orang x;

main(){
        orang y;

        x.setNama("Adam");
        x.setUmur(23);
        y.setNama("Dian");
        y.setUmur(23);

        cout<<"x.getNama()<<" "<<x.getUmur()<<endl;
        cout<<"y.getNama();<<" "<<y.getUmur()<<endl;

        system("PAUSE");
}
```

Instansiasi objek global

Instansiasi objek lokal

GETTER & SETTER

3 In your tester, make an object and access the object's variables and methods

Wait!!!

Apa itu getter dan setter???

Getter dan Setter

1. Penggunaan hak akses PRIVATE sangat diperlukan untuk atribut di dalam sebuah class (efek samping: menyebabkan atribut tersebut).
2. Untuk bisa tetap “menggunakan” atribut tersebut digunakan getter dan setter → **method**.
3. Getter untuk mendapatkan nilai atribut sedangkan setter untuk memberikan nilai atribut tersebut (secara tidak langsung).

Class dan Objek JAVA

1 Write your class

```
class orang
{
        private String nama;
        private int umur;

    public void setNama(String *nama){
        this.nama=nama;
        }

        public String getNama(){return nama;}
        public int getUmur(){return umur;}
};
```

Nama kelas

Kata kunci class mengawali pendeklarasian kelas

Nama anggota class

Hak akses

2 Write a tester (TestDrive) class

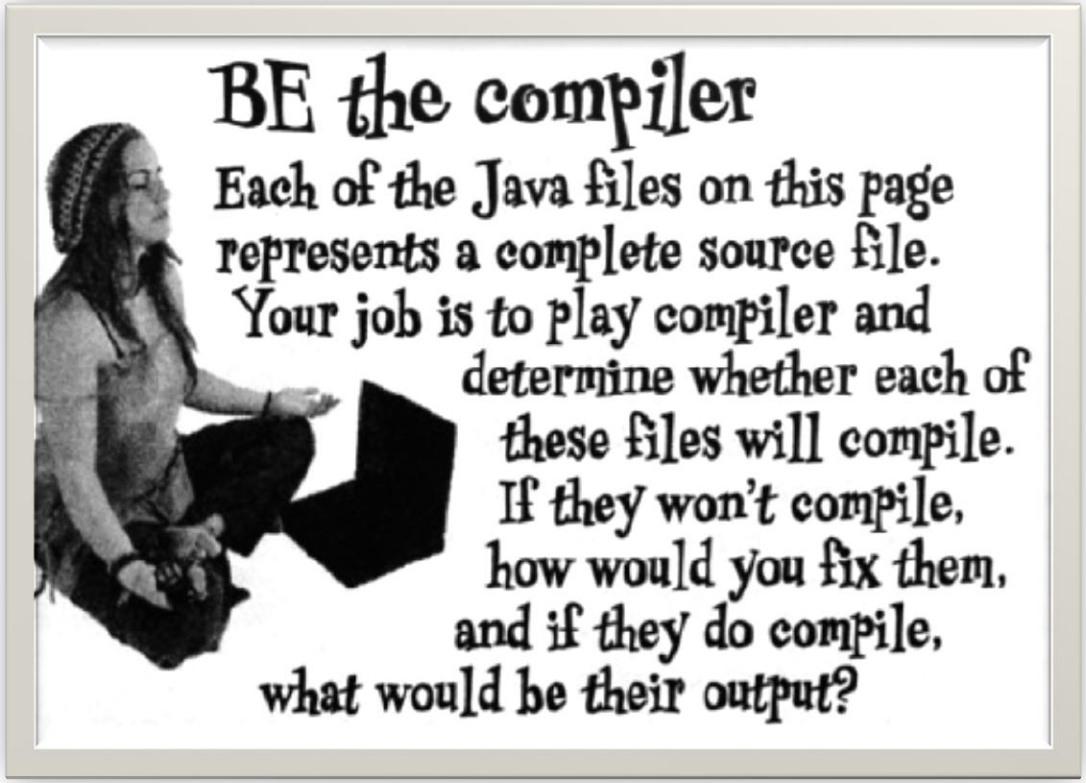
```
public class OrangTes{
  public static void main(String args[]){
     //Kodemu nanti di sini

  }
}
```

```
public class OrangTes{
  public static void main(String args[]){
      orang org = new orang();
      org.setNama("Amir");
      System.out.println("Nama: "+org.getNama());
      System.out.println("Umur: "+org.getUmur());
  }
}
```

Instansiasi objek

3 In your tester, make an object and access the object's variables and methods

Brain Storming



Brain Storming

A

```
class TapeDeck {

  boolean canRecord = false;

  void playTape() {
    System.out.println("tape playing");
  }

  void recordTape() {
    System.out.println("tape recording");
  }
}

class TapeDeckTestDrive {
  public static void main(String [] args) {

    t.canRecord = true;
    t.playTape();

    if (t.canRecord == true) {
      t.recordTape();
    }
  }
}
```

B

```
class DVDPlayer {

  boolean canRecord = false;

  void recordDVD() {
    System.out.println("DVD recording");
  }
}

class DVDPlayerTestDrive {
  public static void main(String [] args) {

    DVDPlayer d = new DVDPlayer();
    d.canRecord = true;
    d.playDVD();

    if (d.canRecord == true) {
      d.recordDVD();
    }
  }
}
```

FINISH!!!