

UTS
Pemrograman Berorientasi Objek



Muhamad Salman Adhim Baqy

A11.2020.12641

Fakultas Ilmu Komputer

Program Studi Teknik Informatika

Universitas Dian Nuswantoro

2022

1. a. - Class adalah template / cetak biru yang menggambarkan perilaku / keadaan yang menjadi objek dukungan jeniannya
Contoh : public class Mobil {

}
- Object adalah sebuah code yang memiliki keadaan dan perilaku.
Object adalah turunan dari sebuah class, sebuah objek bisa didefinisikan setelah suatu class ada terlebih dahulu
Contoh : Mobil memiliki warna dan bisa bergerak

```
public class Mobil {  
    String Warna; // atribut dan Objek  
    String Merk; // atribut dan Objek  
    void Maju () {  
        } // method dan Objek  
    void Mundur () {  
        } // method dan Objek  
}
```

```
public class Mobil Demo {  
    public static void main (String[] args) {  
        Mobil m1 = new Mobil();  
    }  
}
```

3
- Atribut adalah nilai data yang terdapat pada suatu objek yang berasal dari class

Contoh : Mobil memiliki warna merah
public class Mobil {
 String warna = "Merah";
}

3
- Method adalah sesuatu yang dapat dilakukan objek

Contoh : public class Mobil {
 int gigi; // atribut
 void Maju () {
 } // method
 String Gigi () {
 return gigi ++;
 } // method
}

- b. Enkapsulasi adalah suatu mekanisme yang menyembunyikan atau memproteksi suatu proses dari kemungkinan intervensi dari luar sistem.

Contoh :

```
public class Mobil {
    public String warna;
    public int tahunProduksi;
    public void setTahunProduksi (int tp) {
        tahunProduksi = tp;
    }
    public int getTahunProduksi () {
        return tahunProduksi;
    }
}

public class MobilDemo {
    public static void main (String[] args) {
        Mobil m1 = new Mobil ();
        m1.setTahunProduksi (2018);
        System.out.println ("tahun = " + m1.getTahunProduksi());
    }
}
```

- c. Inheritance adalah suatu class yang dapat mewariskan atribut dan method pada class lain ;

contoh : public class Mobil extends MobilAvanza {

- d. Overloading adalah beberapa method dengan nama yang sama , tapi dibedakan dari jumlah dan/atau tipe parameter

contoh : public class Gabut {

```
    int tambah (int nil1, int nil2) {
        return nil1 + nil2 ;
    }
    String tambah (String nilai1, String nil2) {
        return nilai1 + "tambah" + nil2 ;
    }
}
```


2. a. Konstruktor

- Super Class (int p, int q, int r) {

a = p;

b = q;

c = r;

}

- class subclass adalah inheritance dari superclass

SubClass (int l, int m, int n, int o) {

super(l, m, n); // memanggil konstruktor superclass

d = o;

}

b. Inheritance

class SubClass extends SuperClass {

}

c. Overloading

- class SuperClass {

void Show (String s) {
 System.out.println (s);
}

}

- class SubClass extends SuperClass {

void Show (String x) {
 System.out.println (x);

}

}

SuperClass (int p, int q, int r) {

a = p;

b = q;

...

}

SubClass (int l, int m, int n, int o) {

super (l, m, n);

d = o;

}

Overriding

- class SuperClass {

void Show () {
 System.out.println ("a = " + a);
}

}

- class SubClass extends SuperClass {

void Show () {
 super.Show ("class Superclass");

}

}

}

d. Class Super Class

a = 4

b = 7

c = 1

class SubClass

a = 4

b = 7

c = 1

d = 33

e.

Super Class

a : int

b : int

c : int

SuperClass (p : int, q : int, r : int)

show () : void

show (s : String) : void

↑ extends

Sub Class

d : int

SubClass (l : int, m : int, n : int, o : int)

show () : void

show (x : String) : void

+ main () : void

5. a. Resource.java

```
public class Resource {
```

```
    private String title;
```

```
    private Boolean borrowed_status;
```

```
    private String date - borrowed;
```

```
    private String due_date;
```

```
    public String getTitle () {  
        return title; }  
    }
```

```
    public void setTitle (String ResourceTitle) {  
        title = ResourceTitle; }  
    }
```

```
    public Boolean getBorrowedStatus () {  
        return borrowed_status; }  
    }
```

```
    public void setBorrowedStatus (String status) {  
        borrowed_status = status; }  
    }
```

```

    public String getDateBorrowed() {
        return date - borrowed;
    }
    public void setDateBorrowed(String date) {
        date - borrowed = date;
    }
    public void setDueDate(Date borrowedDate) {
        due-date : borrowedDate;
    }
}

```

Print.java

```

public class Print extends Resource {
    private String author;
    private String publisher;
    private String category;
    private int isbn;

    public String getAuthor() {
        return author;
    }
    public void setAuthor(String name) {
        author = name;
    }
    public String getPublisher() {
        return publisher;
    }
    public void setPublisher(String publisherName) {
        publisher = publisherName;
    }
    public String getCategory() {
        return category;
    }
    public void setCategory(String categoryName) {
        category = categoryName;
    }
    public String getISBN() {
        return isbn;
    }
    public void setISBN(int number) {
        isbn = number;
    }
}

```



```

    public String dueDate (Date borrowedDate) {
        return borrowedDate;
    }
}

```

Magazine.java

```

public class Magazine extends Print {
    private int ISSN;
    private int Issue;

    public int getISSN () {
        return ISSN;
    }

    public void setISSN (int ISSN_num) {
        ISSN = ISSN_num;
    }

    public int getIssue () {
        return Issue;
    }

    public int setIssue (int Issue_num) {
        Issue = Issue_num;
    }
}
}

```

```

b. public class EncyclopediaDemo {
    public static void main (String [] args) {
        Encyclopedia bE = new Encyclopedia ();
        bE.setTitle ("Book of Power");
        bE.setAuthor ("Salman Adhim");
        bE.setISBN (12641);
        bE.setVolume (8);

        System.out.println ("Title\t: " + bE.getTitle());
        System.out.println ("Author\t: " + bE.getAuthor());
        System.out.println ("ISBN\t: " + bE.getISBN());
        System.out.println ("Volume\t: " + bE.getVolume());
    }
}
}

```