

KELAS, OBJEK, METHOD

Komputer dan Pemrograman

Endina Putri Purwandari
Prodi Sistem Informasi & Informatika
Universitas Bengkulu



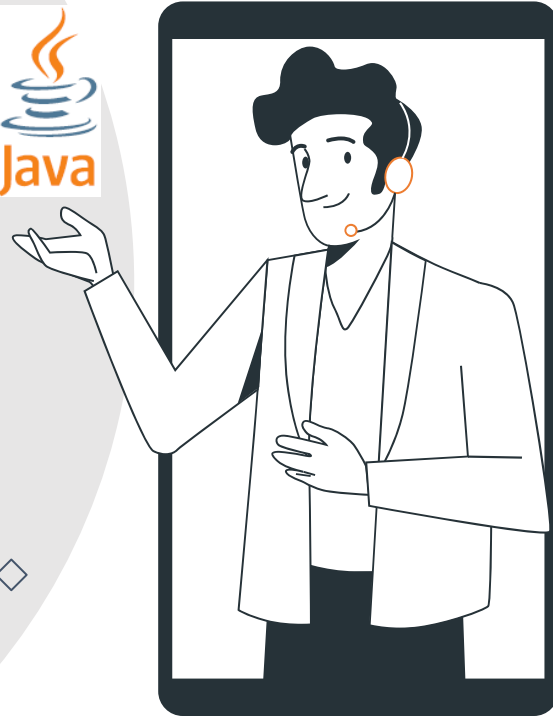
Pokok Bahasan

01

Kelas

02

Objek





01

Kelas

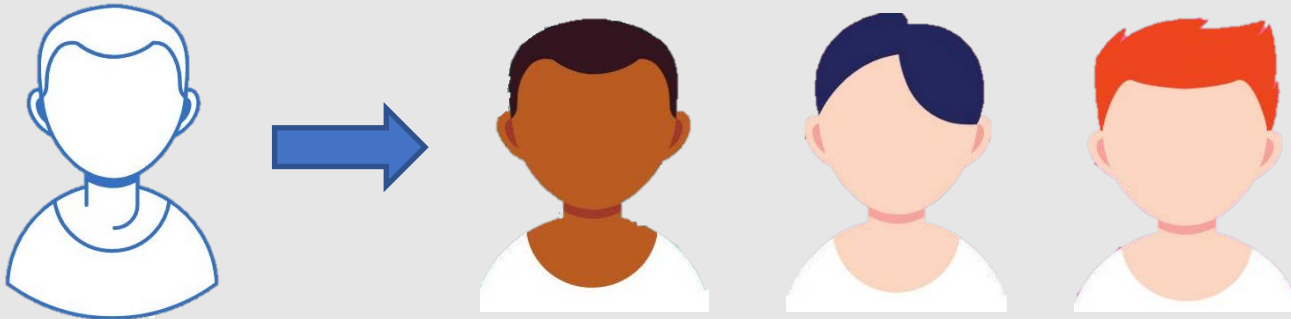


Kelas

Apa itu kelas?

- Entitas keadaan dan perilaku dari objek
 - Sekelompok objek yang memiliki sifat umum
 - Kumpulan atribut variabel, fungsi, dan method dari sebuah objek
- Setelah kompilasi tersimpan pada file *.class (bytecode dari kelas)

Apakah manusia termasuk kelas?



Deklarasi Kelas

```
[1] public class Manusia {  
[2]     // deklarasi variabel  
[3]     String nama;  
[4]     char jenisKel;  
[5]     String kulit;  
[6]     String rambut;  
  
[7]     // deklarasi constructor  
[8]     public Manusia() {  
[9]         System.out.println("Kelas Manusia tanpa nama");  
[10]    }  
[12] }
```



Ketentuan Kelas

```
[1] public class Manusia {    }
```

Modifier menentukan akses

public = member dapat diakses dari mana saja, terbuka publik

private = member hanya dapat diakses oleh kelas itu sendiri.

protected = member mengakses class itu sendiri, class turunan **dalam satu package**

Modifier **protected** hanya bisa diberikan pada member bukan pada class

Isi dari kelas yang **diawali dan diakhiri** dengan **tanda kurung kurawal { }**

Format penulisan: CapitalizeEachWord setiap kata diawali huruf kapital

Misal: `class Manusia`, `class ManusiaDewa`, `class ManusiaBiasa`

```
[7]    // deklarasi constructor
```

```
[8]    public Manusia() {    }
```

Satu Class memiliki satu constructors utama (**nama constructor = nama class**)





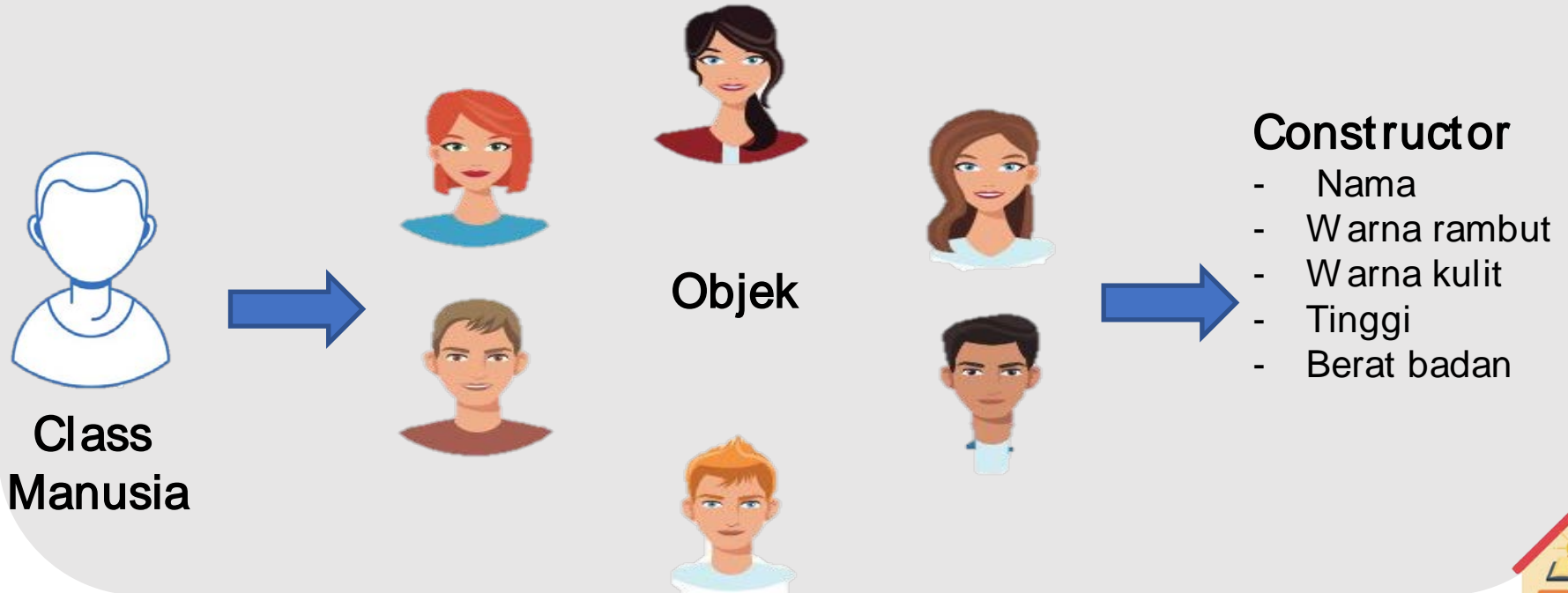
02

OBJEK



Objek

Objek adalah wujud turunan dari kelas berupa representasi entitas dari dunia nyata, seperti benda, orang, tempat, dll.



Deklarasi Objek

```
[1] public class Manusia {
[2]     // deklarasi constructor dengan parameter variabel
[3]     public Manusia(String nama, char jenisKel, String kulit,
[4]         String rambut) {
[5]         System.out.println(" Nama Manusia ini = " + nama +
[6]             "\n Jenis Kelamin = " + jenisKel +
[7]             "\n Warna Kulit = " + kulit +
[8]             "\n Warna Rambut = " + rambut);
[9]     }
[10]
[11]     //deklarasi method
[12]     public static void main( String[] args) {
[13]         Manusia satu = new Manusia("Putri", 'P', "coklat", "hitam");
[14]     }
[15] }
```

```
Nama Manusia ini = Putri
Jenis Kelamin = P
Warna Kulit = coklat
Warna Rambut = hitam
```



Deklarasi Constructor

```
// deklarasi constructor dengan parameter
```

```
[3] public Manusia(String nama, char jenisKel, String kulit, String rambut)
```

Karakteristik Constructor:

- 1) constructor **tidak** memiliki nilai **return** seperti method.
- 2) **Constructor dengan parameter**
→ dapat menentukan value yang berbeda-beda saat pemanggilan class
- 3) Constructor **tanpa parameter**
→ value selalu tetap, kecuali diubah dengan method lain
- 4) **Satu Class** dapat memiliki **banyak constructor** (*Constructor Overloading*)
Nama **constructor sama**, namun **beda tipe dan jumlah parameter**
public Manusia(String nama, char jenisKel, String kulit, String rambut)
public Manusia(String namaLengkap, int tinggi, int berat)



Deklarasi Objek

```
[11] Manusia satu = new Manusia("Putri", 'P', "coklat", "hitam");
```

Bagian deklarasi Objek

1) Deklarasi variabel → Manusia satu

dengan nama variabel = satu sebagai sebuah tipe objek dari kelas

2) Instansiasi → new

Kata kunci new untuk menciptakan sebuah objek

3) Inisialisasi → Manusia("Putri", 'P', "coklat", "hitam");

Pemanggilan konstruktor dengan mengisi nilai variabel

Manusia(isi variabel mengikuti pola variabel konstruktor)



Terima Kasih

PROSES adalah INTI dari HASIL yang kita dapatkan



KELAS, OBJEK, METHOD

Komputer dan Pemrograman

Endina Putri Purwandari
Prodi Sistem Informasi & Informatika
Universitas Bengkulu



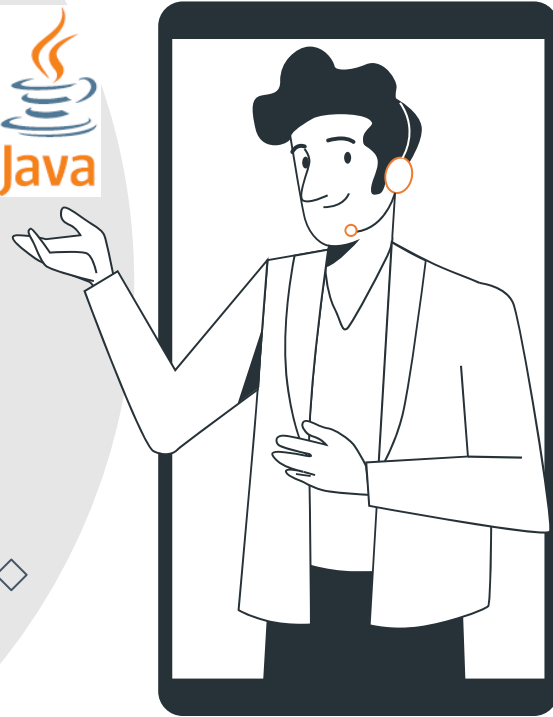
Pokok Bahasan

03

Method

04

Extend





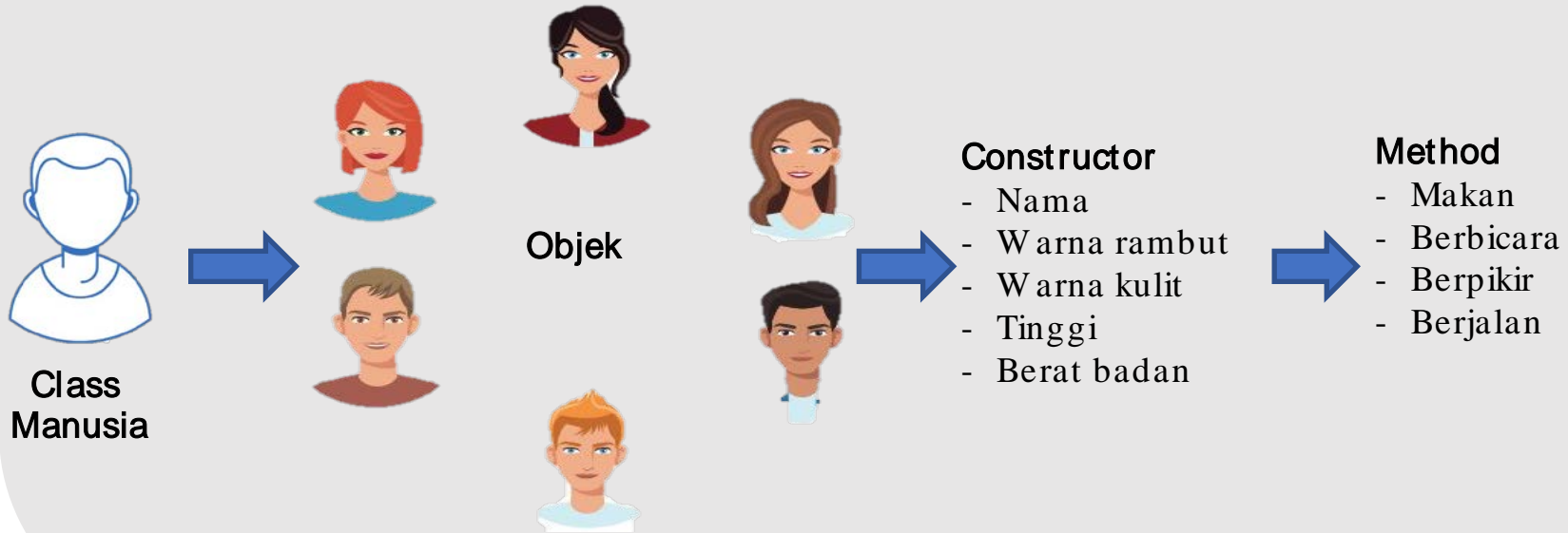
03

Method



Method

Method adalah kegiatan/ aksi/ perilaku (*behavior*) yang dilakukan objek.



Struktur Main Method

Main Method

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

↓ ↓ ↓ ↓ ↓

modifier return type fungsi parameter

- 1) Modifier berupa public, private, protected, dan tanpa access modifier (default).
- 2) Non access modifier, tidak berpengaruh pada aksesibilitas, namun memberikan sifat khusus berupa static, final, dan abstract.
- 3) Return type → `void` = tidak memiliki keluaran, void berarti kosong/hampa.
- 4) Nama method, `main` dapat dipanggil dari tempat lain pada program.
- 5) Parameters, opsional, dapat berjumlah lebih dari satu



Struktur Method 1

Method Tanpa Return Value

`public static void main(int, string, double, char)`

Diagram illustrating the structure of a method without a return value:

- `public static` are labeled as **modifier** (indicated by two blue arrows pointing to them).
- `void` is labeled as **return type** (indicated by a blue arrow pointing to it).
- `main` is labeled as **fungsi** (indicated by a blue arrow pointing to it).
- `(int, string, double, char)` are labeled as **Parameter tipe data** (indicated by a blue arrow pointing to the opening parenthesis).

- 1) Modifier berupa public, private, protected, dan tanpa access modifier (default).
- 2) Non access modifier, tidak berpengaruh pada aksesibilitas, namun memberikan sifat khusus berupa static, final, dan abstract.
- 3) Return type → `void` = tidak memiliki keluaran, void berarti kosong/hampa.
- 4) Nama method, `main` dapat dipanggil dari tempat lain pada program.
- 5) Parameters, opsional, dapat berjumlah lebih dari satu



Struktur Method 2

Method Dengan Return Value

modifier return type fungsi parameter

↓ ↓ ↓ ↓

```
public static double makan (int tinggi) {  
    return (tinggi-100) - ((tinggi-100)*10/100);  
}
```

- 1) Return type dalam tipe **double** maka keluaran dalam bilangan pecahan
- 2) Method punya satu parameter **int**

```
double ideal = satu.makan(155);
```

pemanggilan luaran method dalam tipe **double** (**ideal**)

pemanggilan input method dalam tipe **int** - makan(155)

- 3) Method mengembalikan hasil (output) dengan kata kunci **return**



Deklarasi Method

```
[1] public class Manusia {
[2]     String nama;
[3]     public Manusia() {
[4]         nama = "Putri";
[5]     }
[6]     // method dari kelas Manusia
[7]     void tampil() {    // method void tanpa return value
[8]         System.out.println(" Nama Manusia ini "+ nama);
[9]     }
[10]    public static int makan(int berat, int nasi) {    // method dengan return value
[11]        return berat+nasi;
[12]    }
[13]    public static double makan(int tinggi) {    // method dengan return value
[14]        return (tinggi-100) - ((tinggi-100)*10/100);
[15]    }
[16]    public static void main( String[] args) {
[17]        Manusia satu = new Manusia();
[18]        // memanggil method
[19]        satu.tampil();
[20]        int makanNasi = satu.makan(45,2);
[21]        double ideal = satu.makan(155);
[22]        System.out.println("\n Dia setiap makan nasi beratnya naik "+ makanNasi + " kg" );
[23]        System.out.println("\n Dia berat badan idealnya "+ ideal + " kg" );
[24]    }
```





04

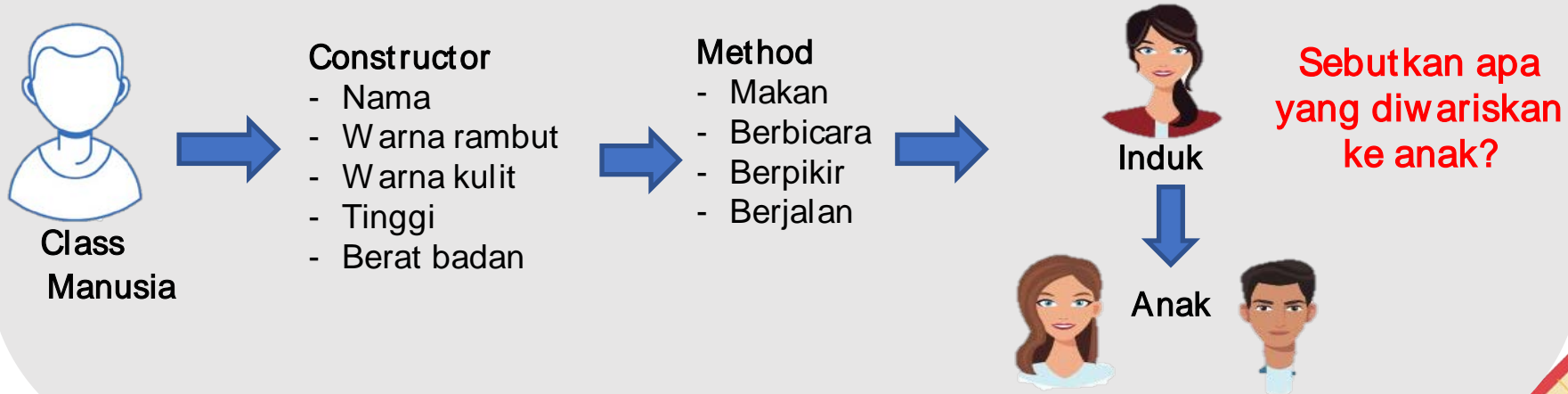
Extends



Extends

Extends adalah pewarisan dari semua fungsi, variabel, dan method dari kelas induk

Satu anak hanya boleh merujuk ke satu induk.



Deklarasi Extends

```
public class Ortu {    //membuat kelas induk
    void sukaMenonton(String a) {    //method induk spesifik
        System.out.println("Nonton " + a);
    }
    void sukaMembaca(String a) { //method induk umum bisa diubah anak
        System.out.println("Suka Baca " + a);
    }
}

public static void main(String [] args) {
    System.out.println("Sifat Orang Tua :");
    Ortu objekO = new Ortu(); //memanggil objek induk
    objekO.sukaMenonton("Berita");    //memanggil sifat spesifik induk
    objekO.sukaMembaca("Koran"); //memanggil method dengan variabel dapat diubah

    System.out.println("\n Sifat Anak :");
    Anak objekA = new Anak(); //memanggil objek anak
    objekA.sukaMenonton(9, "Film Drakor");    //memanggil sifat spesifik anak yang diturunkan induk
    objekA.sukaMembaca("Komik One Piece"); //memanggil method induk otomatis diturunkan ke anak
} }

class Anak extends Ortu {    //membuat kelas anak
    void sukaMenonton(int a, String b) { //method kelas anak
        System.out.println("Nonton Jam " + a + " Malam " + b);
    } }
}
```



Terima Kasih

PROSES adalah INTI dari HASIL yang kita dapatkan

