

Nama : Arditya Adjie Rosandi

NIM : 20230801274

Pemrograman Berorientasi Objek



Bagian 1: Hewan

```
// Superclass
class Animal {
    // Properti
    String name;

    // Constructor
    public Animal(String name) {
        this.name = name;
    }

    // Metode
    public void speak() {
        System.out.println("SUARA HEWAN");
    }
}
```

Superclass: Animal adalah superclass yang memiliki properti name dan metode speak(). Metode speak() akan mencetak "SUARA HEWAN" ke layar.

```
// Subclass
class Dog extends Animal {

    // Constructor
    public Dog(String name) {
        super(name); // Memanggil constructor dari superclass
    }

    // Override metode speak
    @Override
    public void speak()
    {
        System.out.println(name + " BILANG GUKGUKGUK");
    }
}
```

Subclass: Dog adalah subclass yang extends Animal. Subclass ini memiliki constructor yang memanggil constructor dari superclass menggunakan kata kunci super.

Override metode: Subclass Dog override metode speak() dari superclass Animal. Metode speak() pada subclass Dog akan mencetak "DIKDOG BILANG GUKGUKGUK" ke layar.

```

public class HewanTester {
    Run | Debug
    public static void main(String[] args)
    {
        System.out.print("\033[H\033[2J");
        Dog dog = new Dog(name: "DIKDOG");
        dog.speak();
    }
}

```

Penggunaan: Pada class HewanTester, objek Dog dibuat dengan nama "DIKDOG" dan metode speak() dipanggil. Karena metode speak() telah di-override pada subclass Dog, maka yang akan dicetak adalah "DIKDOG BILANG GUKGUKGUK".

Bagian 2: Kendaraan

```

class Vehicle {
    String brand;

    // Constructor
    public Vehicle(String brand) {
        this.brand = brand;
    }

    // Metode untuk menampilkan informasi kendaraan
    public void displayInfo() {
        System.out.println("BRAND : " + brand);
    }
}

```

Superclass: Vehicle adalah superclass yang memiliki properti brand dan metode displayInfo() untuk menampilkan informasi kendaraan.

```
// Subclass
class Car extends Vehicle {
    int doors;

    // Constructor
    public Car(String brand, int doors) {
        super(brand);
        this.doors = doors;
    }

    // Overloading metode displayInfo
    public void displayInfo(String type)
    {
        super.displayInfo();
        System.out.println("TIPE : " + type);
        System.out.println("PINTU : " + doors);
    }
}
```

Subclass: Car adalah subclass yang extends Vehicle. Subclass ini memiliki properti doors dan constructor yang memanggil constructor dari superclass menggunakan kata kunci super.

Overloading metode: Subclass Car memiliki metode displayInfo() yang overload metode displayInfo() dari superclass Vehicle. Metode displayInfo() pada subclass Car memiliki parameter tambahan type dan menampilkan informasi tambahan tentang kendaraan, seperti tipe dan jumlah pintu.

```
public class KendaraanTester {
    Run | Debug
    public static void main(String[] args)
    {
        System.out.print("\033[H\033[2J");

        Car car = new Car(brand:"TOYOTA", doors:4);
        car.displayInfo(type:"SEDAN");
    }
}
```

Penggunaan: Pada class KendaraanTester, objek Car dibuat dengan brand "TOYOTA" dan jumlah pintu 4. Metode displayInfo() dipanggil dengan parameter "SEDAN" untuk menampilkan informasi tentang kendaraan.

Bagian 3: Pegawai

```
public class Pegawai {
    String nama;
    int id_pegawai;
    String gaji;

    public void menampilkan(){
        System.out.println("MENAMPILKAN NAMA, ID PEGAWAI, GAJI DAN TUGAS.");
        System.out.println("-----");
    }
}
```

Kelas Pegawai: Kelas Pegawai memiliki beberapa atribut (property) yang merepresentasikan informasi tentang pegawai, yaitu:

- nama: nama pegawai
- id_pegawai: ID pegawai
- gaji: gaji pegawai

Metode menampilkan: Kelas Pegawai memiliki metode menampilkan() yang digunakan untuk menampilkan informasi tentang pegawai. Namun, pada contoh ini, metode menampilkan() hanya menampilkan teks yang menyatakan bahwa informasi tentang pegawai akan ditampilkan, tetapi tidak benar-benar menampilkan informasi tentang pegawai.

```
public class Kasir extends Pegawai {

    @Override
    public void menampilkan()
    {
        System.out.println("NAMA : " + nama);
        System.out.println("ID PEGAWAI : " + id_pegawai);
        System.out.println("GAJI : " + gaji);
    }

    public void tugas() {
        System.out.println("TUGAS : MELAKUKAN TRANSAKSI DENGAN PEMBELI.");
        System.out.println();
    }
}
```

Kelas Kasir: Kelas Kasir adalah subclass dari kelas Pegawai. Artinya, kelas Kasir memiliki semua atribut dan metode yang dimiliki oleh kelas Pegawai.

Override metode menampilkan: Kelas Kasir override metode menampilkan() dari kelas Pegawai. Metode menampilkan() pada kelas Kasir digunakan untuk menampilkan informasi tentang kasir, seperti nama, ID pegawai, dan gaji.

Metode tugas: Kelas Kasir memiliki metode tugas() yang digunakan untuk menampilkan tugas yang dilakukan oleh kasir, yaitu melakukan transaksi dengan pembeli.

```

public class Koki extends Pegawai {

    @Override
    public void menampilkan()
    {
        System.out.println("NAMA : " + nama);
        System.out.println("ID PEGAWAI : " + id_pegawai);
        System.out.println("GAJI : " + gaji);
    }

    public void tugas() {
        System.out.println("TUGAS : MEMASAK MAKANAN DAN MEMBUAT MINUMAN.");
        System.out.println();
    }
}

```

Kelas Koki: Kelas Koki adalah subclass dari kelas Pegawai. Artinya, kelas Koki memiliki semua atribut dan metode yang dimiliki oleh kelas Pegawai.

Override metode menampilkan: Kelas Koki override metode menampilkan() dari kelas Pegawai. Metode menampilkan() pada kelas Koki digunakan untuk menampilkan informasi tentang koki, seperti nama, ID pegawai, dan gaji.

Metode tugas: Kelas Koki memiliki metode tugas() yang digunakan untuk menampilkan tugas yang dilakukan oleh koki, yaitu memasak makanan dan membuat minuman.

```

public class Manager extends Pegawai {

    @Override
    public void menampilkan()
    {
        System.out.println("NAMA : " + nama);
        System.out.println("ID PEGAWAI : " + id_pegawai);
        System.out.println("GAJI : " + gaji);
    }

    public void tugas() {
        System.out.println("TUGAS : MELAKUKAN MANAJEMEN UNTUK FRANCHISE.");
        System.out.println();
    }
}

```

Kelas Manager: Kelas Manager adalah subclass dari kelas Pegawai. Artinya, kelas Manager memiliki semua atribut dan metode yang dimiliki oleh kelas Pegawai.

Override metode menampilkan: Kelas Manager override metode menampilkan() dari kelas Pegawai. Metode menampilkan() pada kelas Manager digunakan untuk menampilkan informasi tentang manager, seperti nama, ID pegawai, dan gaji.

Metode tugas: Kelas Manager memiliki metode tugas() yang digunakan untuk menampilkan tugas yang dilakukan oleh manager, yaitu melakukan manajemen untuk franchise.

```

public class Pelayan extends Pegawai {

    @Override
    public void menampilkan()
    {
        System.out.println("NAMA : " + nama);
        System.out.println("ID PEGAWAI : " + id_pegawai);
        System.out.println("GAJI : " + gaji);
    }

    public void tugas() {
        System.out.println("Loading... MELAYANI DAN MENYAJIKAN PESANAN PEMBELI.");
        System.out.println();
    }
}

```

Kelas Pelayan: Kelas Pelayan adalah subclass dari kelas Pegawai. Artinya, kelas Pelayan memiliki semua atribut dan metode yang dimiliki oleh kelas Pegawai.

Override metode menampilkan: Kelas Pelayan override metode menampilkan() dari kelas Pegawai. Metode menampilkan() pada kelas Pelayan digunakan untuk menampilkan informasi tentang pelayan, seperti nama, ID pegawai, dan gaji.

Metode tugas: Kelas Pelayan memiliki metode tugas() yang digunakan untuk menampilkan tugas yang dilakukan oleh pelayan, yaitu melayani dan menyajikan pesanan pembeli.

```

public class Satpam extends Pegawai {

    public void menampilkan()
    {
        System.out.println("NAMA : " + nama);
        System.out.println("ID PEGAWAI : " + id_pegawai);
        System.out.println("GAJI : " + gaji);
    }

    public void tugas() {
        System.out.println("TUGAS : MENJAGA KEAMANAN DI DALAM DAN DI LUAR FRANCHISE.");
        System.out.println();
    }
}

```

Kelas Satpam: Kelas Satpam adalah subclass dari kelas Pegawai. Artinya, kelas Satpam memiliki semua atribut dan metode yang dimiliki oleh kelas Pegawai.

Metode menampilkan: Kelas Satpam memiliki metode menampilkan() yang digunakan untuk menampilkan informasi tentang satpam, seperti nama, ID pegawai, dan gaji.

Metode tugas: Kelas Satpam memiliki metode tugas() yang digunakan untuk menampilkan tugas yang dilakukan oleh satpam, yaitu menjaga keamanan di dalam dan di luar franchise.

```
public class Main {  
    Run | Debug  
    public static void main(String[] args) {  
  
        System.out.print("\033[H\033[2J");  
  
        // Membuat objek berdasarkan class  
        Manager manager = new Manager();  
        Kasir kasir = new Kasir();  
        Koki koki = new Koki();  
        Pelayan pelayan = new Pelayan();  
        Satpam satpam = new Satpam();  
  
        // Memasukkan nilai variabel menggunakan objek  
        manager.nama = "SYIFA";  
        manager.id_pegawai = 1;  
        manager.gaji = "7 JUTA";  
  
        kasir.nama = "ALDI";  
        kasir.id_pegawai = 2;  
        kasir.gaji = "3 JUTA";  
  
        koki.nama = "REZA";  
        koki.id_pegawai = 3;  
        koki.gaji = "2 JUTA";  
  
        pelayan.nama = "DEAN";  
        pelayan.id_pegawai = 4;  
        pelayan.gaji = "1,2 JUTA";
```

```
        satpam.nama = "DIKI";  
        satpam.id_pegawai = 5;  
        satpam.gaji = "3 JUTA";  
  
        // Menjalankan fungsi pada superclass  
  
        manager.menampilkan();  
        manager.tugas();  
  
        kasir.menampilkan();  
        kasir.tugas();  
  
        koki.menampilkan();  
        koki.tugas();  
  
        pelayan.menampilkan();  
        pelayan.tugas();  
  
        satpam.menampilkan();  
        satpam.tugas();  
    }  
}
```

Membuat objek: Kode tersebut membuat objek-objek berdasarkan kelas Manager, Kasir, Koki, Pelayan, dan Satpam.

Memasukkan nilai variabel: Kode tersebut memasukkan nilai variabel ke dalam objek-objek yang telah dibuat.

Menjalankan fungsi: Kode tersebut menjalankan fungsi menampilkan() dan tugas() pada setiap objek untuk menampilkan informasi tentang pegawai dan tugas yang dilakukan oleh mereka.

Output:

```
NAMA : SYIFA
ID PEGAWAI : 1
GAJI : 7 JUTA
TUGAS : MELAKUKAN MANAJEMEN UNTUK FRANCHISE.

NAMA : ALDI
ID PEGAWAI : 2
GAJI : 3 JUTA
TUGAS : MELAKUKAN TRANSAKSI DENGAN PEMBELI.

NAMA : REZA
ID PEGAWAI : 3
GAJI : 2 JUTA
TUGAS : MEMASAK MAKANAN DAN MEMBUAT MINUMAN.

NAMA : DEAN
ID PEGAWAI : 4
GAJI : 1,2 JUTA
TUGAS : MELAYANI DAN MENYAJIKAN PESANAN PEMBELI.

NAMA : DIKI
ID PEGAWAI : 5
GAJI : 3 JUTA
TUGAS : MENJAGA KEAMANAN DI DALAM DAN DI LUAR FRANCHISE.
```