

## LAPRAK PERTEMUAN 11

NAMA: Elfan Pradita Rusmin

NIM : 20230801068

Mapel : PBO

1. **Pegawai:** Kelas induk yang menyimpan informasi umum pegawai seperti nama, ID, dan gaji, serta metode menampilkan() yang menampilkan informasi tersebut.

### Kodingan

```
● ● ●  
1  public class Kasir extends Pegawai {  
2      @Override  
3      public void menampilkan() {  
4          System.out.println("Nama: " + nama);  
5          System.out.println("Id Pegawai: " + id_pegawai);  
6          System.out.println("Gaji: " + gaji);  
7      }  
8      // Metode untuk tugas kasir  
9      public void tugas() {  
10         System.out.println("Tugas: Melakukan transaksi dengan pembeli");  
11         System.out.println("-----");  
12     }  
13 }  
14 }
```

2. **Manager:** Kelas turunan dari Pegawai yang mengoverride metode menampilkan() dan menambahkan tugas manajerial khusus dalam metode tugas().

### Kodingan

```
● ● ●  
1  public class Manager extends Pegawai {  
2      @Override  
3      public void menampilkan() {  
4          System.out.println("Nama: " + nama);  
5          System.out.println("Id Pegawai: " + id_pegawai);  
6          System.out.println("Gaji: " + gaji);  
7      }  
8      // Metode untuk tugas manager  
9      public void tugas() {  
10         System.out.println("Tugas: Melakukan manajemen untuk franchise");  
11         System.out.println("-----");  
12     }  
13 }  
14 }
```

3. **Kasir:** Kelas turunan dari Pegawai yang mengoverride metode menampilkan() dan menambahkan tugas kasir khusus dalam metode tugas().

### Kodingan

```
● ● ●

1  public class Kasir extends Pegawai {
2      @Override
3      public void menampilkan() {
4          System.out.println("Nama: " + nama);
5          System.out.println("Id Pegawai: " + id_pegawai);
6          System.out.println("Gaji: " + gaji);
7      }
8
9      // Metode untuk tugas kasir
10     public void tugas() {
11         System.out.println("Tugas: Melakukan transaksi dengan pembeli");
12         System.out.println("-----");
13     }
14 }
```

4. **Koki:** Kelas turunan dari Pegawai yang mengoverride metode menampilkan() dan menambahkan tugas memasak dan membuat minuman dalam metode tugas().

### Kodingan

```
● ● ●

1  public class Koki extends Pegawai {
2      @Override
3      public void menampilkan() {
4          System.out.println("Nama: " + nama);
5          System.out.println("Id Pegawai: " + id_pegawai);
6          System.out.println("Gaji: " + gaji);
7      }
8
9      // Metode untuk tugas koki
10     public void tugas() {
11         System.out.println("Tugas: Memasak makanan dan membuat minuman");
12         System.out.println("-----");
13     }
14 }
```

5. **Pelayan:** Kelas turunan dari Pegawai yang mengoverride metode menampilkan() dan menambahkan tugas melayani dan menyajikan pesanan dalam metode tugas().

### Kodingan

```
● ● ●

1  public class Pelayan extends Pegawai {
2      @Override
3      public void menampilkan() {
4          System.out.println("Nama: " + nama);
5          System.out.println("Id Pegawai: " + id_pegawai);
6          System.out.println("Gaji: " + gaji);
7      }
8
9      // Metode untuk tugas pelayan
10     public void tugas() {
11         System.out.println("Tugas: Melayani dan menyajikan pesanan pembeli");
12         System.out.println("-----");
13     }
14 }
```

6. **Satpam:** Kelas turunan dari Pegawai yang mengoverride metode menampilkan() dan menambahkan tugas menjaga keamanan dalam metode tugas().

### Kodingan

```
● ● ●

1  public class Satpam extends Pegawai {
2      @Override
3      public void menampilkan() {
4          System.out.println("Nama: " + nama);
5          System.out.println("Id Pegawai: " + id_pegawai);
6          System.out.println("Gaji: " + gaji);
7      }
8
9      // Metode untuk tugas satpam
10     public void tugas() {
11         System.out.println("Tugas: Menjaga keamanan di dalam dan di luar franchise");
12         System.out.println("-----");
13     }
14 }
```

## Output

```
Menampilkan nama, id pegawai, gaji dan tugas.
-----
Nama: Sifa
Id Pegawai: 1
Gaji: 7 Juta
Tugas: Melakukan manajemen untuk franchise
-----
Nama: Aldi
Id Pegawai: 2
Gaji: 1,2 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: Aldi
Id Pegawai: 5
Gaji: 1 Juta
Tugas: Menjaga keamanan di dalam dan di luar franchise
```

7. **Animal:** Kelas induk yang memiliki properti name dan metode speak(), yang menampilkan pesan "Animal speaks" saat dipanggil.

## Kodingan

```
● ● ●

1  public class Animal {
2      // Properti
3      String name;
4
5      // Constructor
6      public Animal(String name) {
7          this.name = name;
8      }
9
10     // Metode
11     public void speak() {
12         System.out.println("Animal speaks");
13     }
14 }
```

8. **Dog:** Kelas turunan dari Animal yang mengoverride metode speak() untuk menampilkan pesan berupa nama anjing dan suara "Woof".

### Kodingan

```
● ● ●

1 public class Dog extends Animal {
2
3     // Constructor
4     public Dog(String name) {
5         super(name); // Memanggil constructor dari superclass
6     }
7
8     // Override metode speak
9     @Override
10    public void speak() {
11        System.out.println(name + " says Woof");
12    }
13 }
```

9. **AnimalTester:** Kelas utama yang membuat objek Dog dan memanggil metode speak(), yang menghasilkan output sesuai dengan implementasi speak() di kelas Dog.

### Kodingan

```
● ● ●

1 public class AnimalTester {
2     public static void main(String[] args) {
3         // Membuat objek dari subclass Dog
4         Dog dog = new Dog("Buddy");
5
6         // Memanggil metode speak
7         dog.speak(); // Output: Buddy says Woof
8     }
9 }
```

## Output

Buddy says Woof

10. **Vehicle**: Kelas induk yang memiliki properti brand dan konstruktor untuk menginisialisasi merek kendaraan, serta metode displayInfo() untuk menampilkan informasi merek kendaraan.

## Kodingan

```
● ● ●

1 public class Vehicle {
2     String brand;
3
4     // Constructor
5     public Vehicle(String brand) {
6         this.brand = brand;
7     }
8
9     // Metode untuk menampilkan informasi kendaraan
10    public void displayInfo() {
11        System.out.println("Brand: " + brand);
12    }
13 }
```

11. **Car**: Kelas turunan dari Vehicle yang menambahkan properti doors (jumlah pintu) dan mengoverride metode displayInfo() untuk menampilkan informasi jenis kendaraan dan jumlah pintu bersama dengan merek.

## Kodingan



```
1 public class Car extends Vehicle {  
2     int doors;  
3  
4     // Constructor  
5     public Car(String brand, int doors) {  
6         super(brand); // Memanggil constructor superclass  
7         this.doors = doors;  
8     }  
9  
10    // Overloading metode displayInfo  
11    public void displayInfo(String type) {  
12        super.displayInfo();  
13        System.out.println("Type: " + type + ", Doors: " + doors);  
14    }  
15 }
```

12. **VehicleTester**: Kelas utama yang membuat objek Car dan memanggil metode displayInfo() untuk menampilkan informasi kendaraan, termasuk merek, jenis, dan jumlah pintu.

### Kodingan



```
1 public class VehicleTester {  
2     public static void main(String[] args) {  
3         // Membuat objek Car  
4         Car car = new Car("Toyota", 4);  
5  
6         // Menampilkan informasi kendaraan  
7         car.displayInfo("Sedan"); // Output: Brand: Toyota, Type: Sedan, Doors: 4  
8     }  
9 }
```

### Output

Brand: Toyota  
Type: Sedan, Doors: 4