Name : Faricha Aulia
Class : 21 / Informatics Engineering
NIM : 2141720155

Jobsheet 1

Introduction to Object Oriented Programming (OOP)

1. Competency

After taking this module, students may be able to:

1. Recognize the difference between object oriented and procedural paradigm in programming.
2. Recognize the basic concept of OOP
3. Implement OOP in java programming

2. Introduction

2.1. Procedural Programming vs Object Oriented Programming

In procedural programming, a program is divided into some sub-program called function. It is difference with OOP, the main concept is a program is divided into object where object consists of state and method.

OOP is more flexible and modular, it can be described that each part of program won't be disturbed if a feature is updated. It is difference with procedural programming where the program may change overall when an update is implemented.

More clearly, this is the sample of procedural programming and object-oriented programming.

Based on those code, object-oriented perspective can be identified more clearly as human language concept than procedural programming. From the first line of object-oriented, a new object uses new followed by the name of the class. It shows that an object is assigned to a variable with some return value, so it can be identified that object oriented programming is more efficient

NB: In this module, we will try to create class/classes, creating object, and calling methods in object. More explanation about class anatomy will be discussed in the next module.

2.2. Basic Concept of OOP

Some aspects are known in OOP, such:

a. Object

Object is a sequence of program which consists of state and behavior. Object in software is modelled to be similar of object in real world. State in object is attribute of the object itself, in example: object Sepeda has state like merek, kecepatan, gear, etc. Behavior is a something that the object can do, in example: in object Sepeda, the behaviors are: tambah kecepatan, pindah gear, kurangi kecepatan, belok, etc.

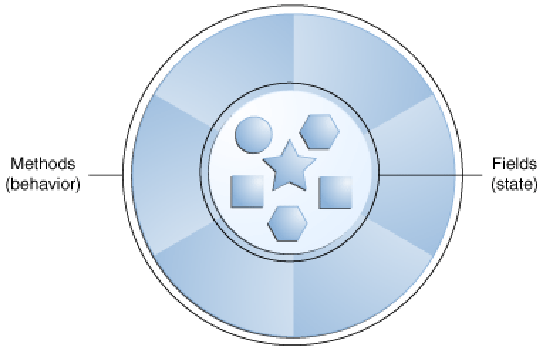


Figure 1. Illustration of object in software

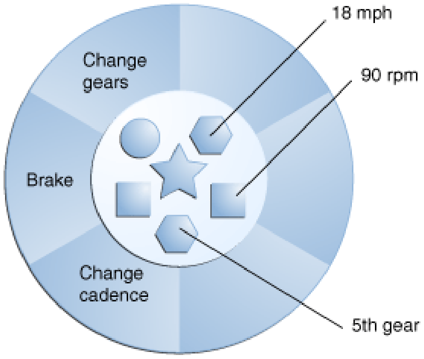


Figure 2. Illustration of state and behavior in object Bicycle

b. Class
Class is a blueprint or prototype from object. Taking example from bicycle object, although there are many kinds of bicycle based on brand and model, the blueprint of bicycle will be the same such component and characteristic. The one you may belong in your home can be defined as instantiation from class Bicycle.

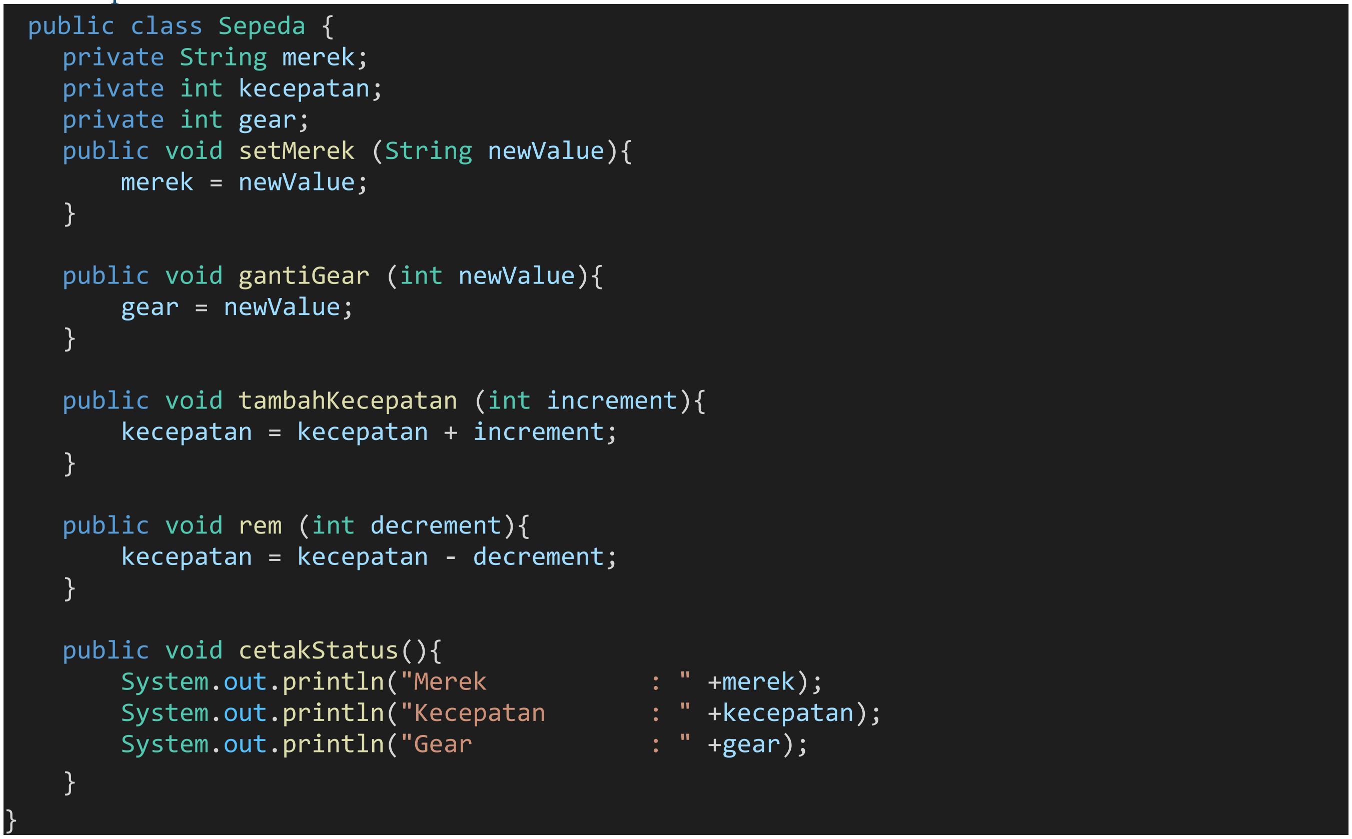
c. Encapsulation
Also known as information-hiding. The complex scheme in program may not be shown inevitably. In bicycle, we can change gear position by pressing gear knob at the handlebar, we may need not to know how the scheme in changing gear works technically.

d. Inheritance
Inheritance allow us to organize the structure of program naturally. We can expand the functionality of program without make a large modification in that program. In example, bicycle can be inherited to another model like mountain bike and road bike where each model has additional component which is not owned by common bike. In another word, mountain bike and road bike inherit bicycle.

e. Polymorphism
Polymorphism imitates real world object characteristics where an object may have different form. In example, plane object can be inherited to be jet plane and helicopter which have ability to increase the speed. But, the method to increase the speed is different between those models because of different engines used technically.

3. Experiment Activity

3.1.Experiment Act 1



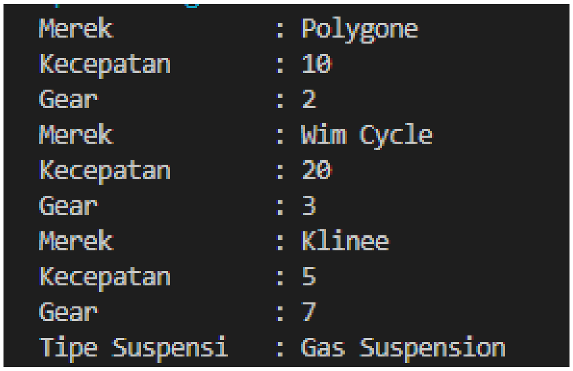
```
 public class Sepeda {
   private String merek;
   private int kecepatan;
   private int gear;
   public void setMerek (String newValue){
       merek = newValue;
   }

   public void gantiGear (int newValue){
       gear = newValue;
   }

   public void tambahKecepatan (int increment){
       kecepatan = kecepatan + increment;
   }

   public void rem (int decrement){
       kecepatan = kecepatan - decrement;
   }

   public void cetakStatus(){
       System.out.println("Merek           : " +merek);
       System.out.println("Kecepatan       : " +kecepatan);
       System.out.println("Gear            : " +gear);
   }
}
```

```
public class Demo {
    public static void main(String[] args) {
        Sepeda spd1 = new Sepeda();
        Sepeda spd2 = new Sepeda();
        SepedaGunung spd3 = new SepedaGunung();

        spd1.setMerek("Polygone");
        spd1.tambahKecepatan(10);
        spd1.gantiGear(2);
        spd1.cetakStatus();

        spd2.setMerek("Wim Cycle");
        spd2.tambahKecepatan(10);
        spd2.gantiGear(2);
        spd2.tambahKecepatan(10);
        spd2.gantiGear(3);
        spd2.cetakStatus();

        spd3.setMerek("Klinee");
        spd3.tambahKecepatan(5);
        spd3.gantiGear(7);
        spd3.setTipeSuspensi("Gas Suspension");
        spd3.cetakStatus();
    }
}
```

3.2. Experiment Activity 2

```
public class SepedaGunung extends Sepeda {
    private String tipeSuspensi;

    public void setTipeSuspensi(String newValue){
        tipeSuspensi = newValue;
    }

    public void cetakStatus(){
        super.cetakStatus();
        System.out.println("Tipe Suspensi   : " +tipeSuspensi);
    }
}
```

3.3. Experiment Running



```
Merek           : Polygone
Kecepatan       : 10
Gear            : 2
Merek           : Wim Cycle
Kecepatan       : 20
Gear            : 3
Merek           : Klinee
Kecepatan       : 5
Gear            : 7
Tipe Suspensi   : Gas Suspension
```

4. Conclusion

Based on the activity, we have demonstrated about what is the paradigm of OOP and the implementation into a simple program. Inheritance also one of the importance features in OOP, we have demonstrated it by implementing class BicycleGunung.

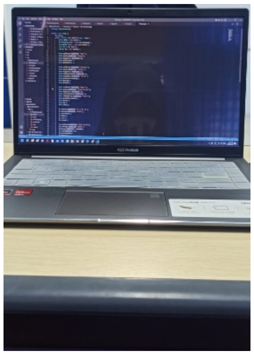
We also know that the Mountain Bike class is similar with the Bicycle class (having gear and speed state, also some behaviors like increasing speed, braking, changing gear, etc.) but with additional state named suspension type. So we may not necessarily create the Bicycle class from the scratch, we use extends or just inherits from the Bicycle class, then we add a new feature which is not owned by the Bicycle class before. This is the characteristic of OOP which is not having in procedural programming.

5. Question Test

1. Explain the difference between object and class!
   Answers:
   - Class collection of a function that is made to represent something / a particular purpose. The parable is like a mold
   - Objects are uses of classes that are ready to be used. The parable is like a printed thing
2. State your reason why color and engine type can be classified as attribute for car object!
   Answers: Because it has characteristics
3. State one of OOP better point than procedural programming!
   Answers: Because OOP breaks programming tasks into objects, which encapsulate data and methods. So the approach is easier to learn, develop and maintain
4. Is it allowed to define two attributes in one line code such "public String name,address;"?
   Answers: Yes, as long as they have the same data type
5. In BicycleGunung class, state your reason why brand, speed, and gear attributes are not written again in this class!
   Answers: Because we have extended the 'Sepeda' class that already has these attributes. So it will definitely be passed down to his child

6. Assignment

1. Follow these instructions to make your practical assignment is performed systematically:
   a. Take 4 photographs of objects around you, 2 objects must be implementation of inheritance concept, example: refrigerator, chair, living room table, desk! As we know that living room table and desk are inherited by table class.



   b. Observe those objects to define the attribute and method!
   Kursi
   - Attribut : merek,warna,kapasitas
   - Method : diduduki

   Kursi Goyang
   - Attribut : jumlah kaki, jenis kayu

- Method : goyang, pindah

Kursi Putar

- Attribut : jumlah roda, tipe tuas
- Method : geser,up,down

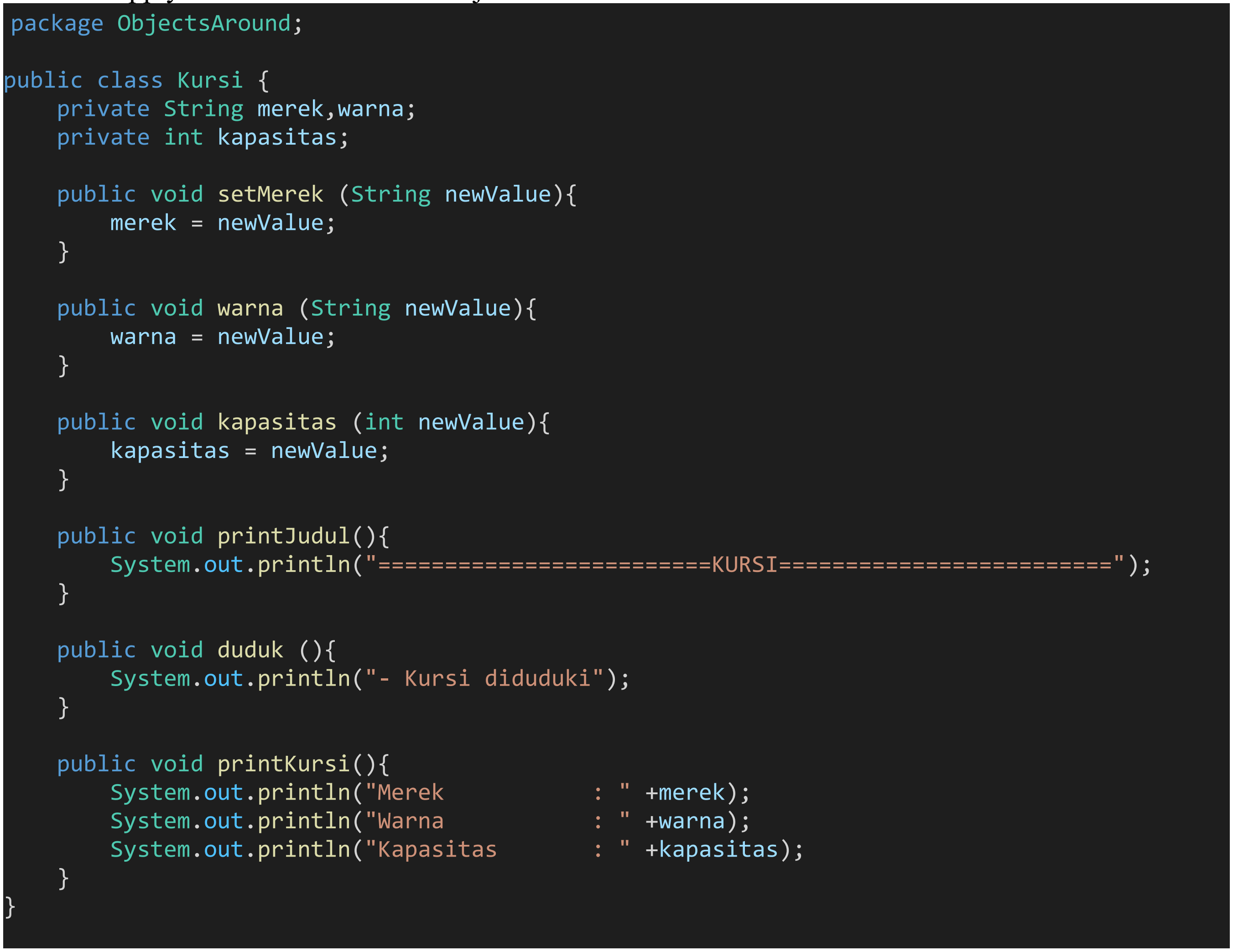
Laci

- Attribut : ukuran,bentuk
- Method : buka,isi,tutup

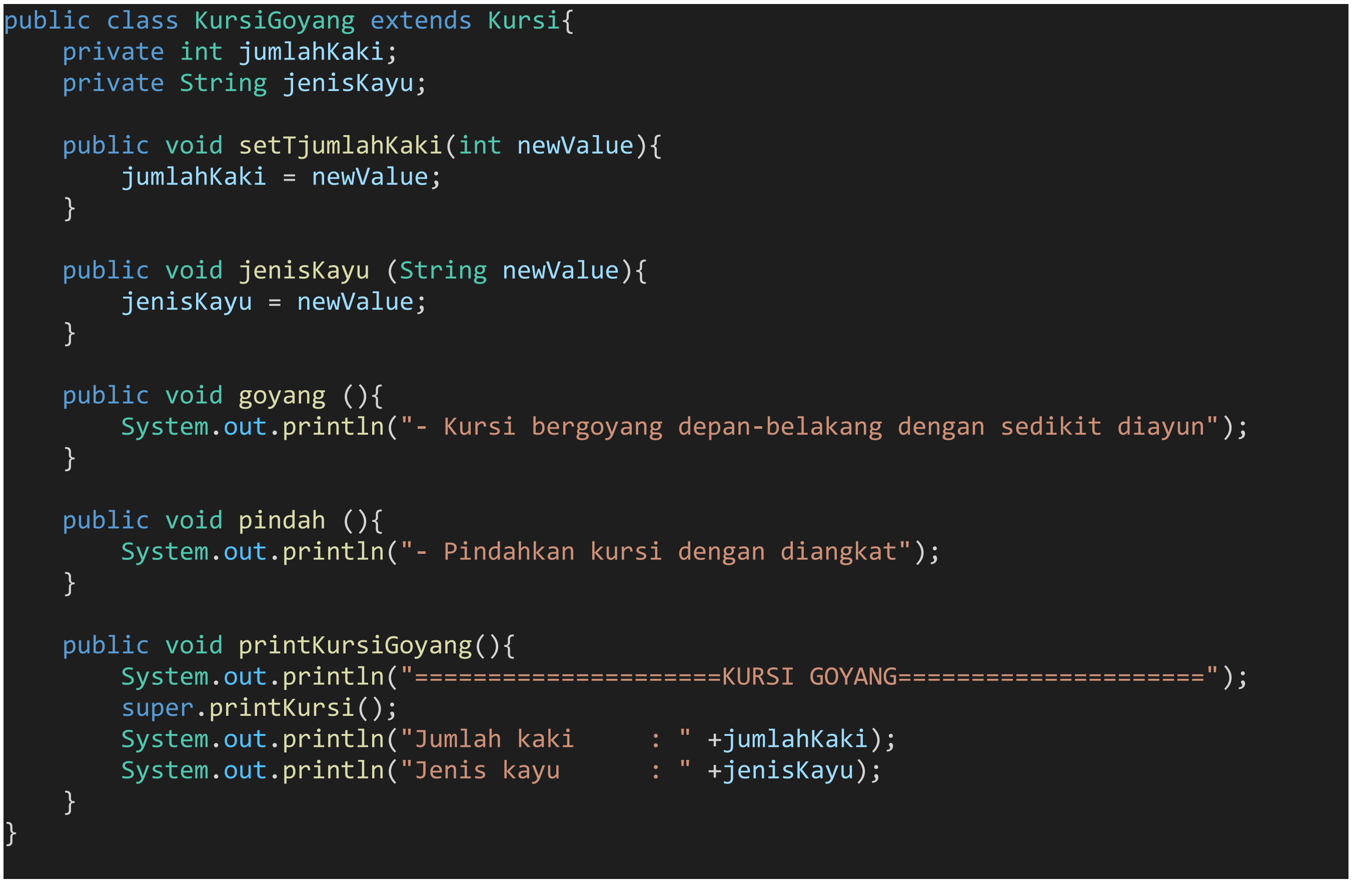
Laptop

- Attribut : brand,tipe,ram,harga
- Method : nyala,bunyi,mati

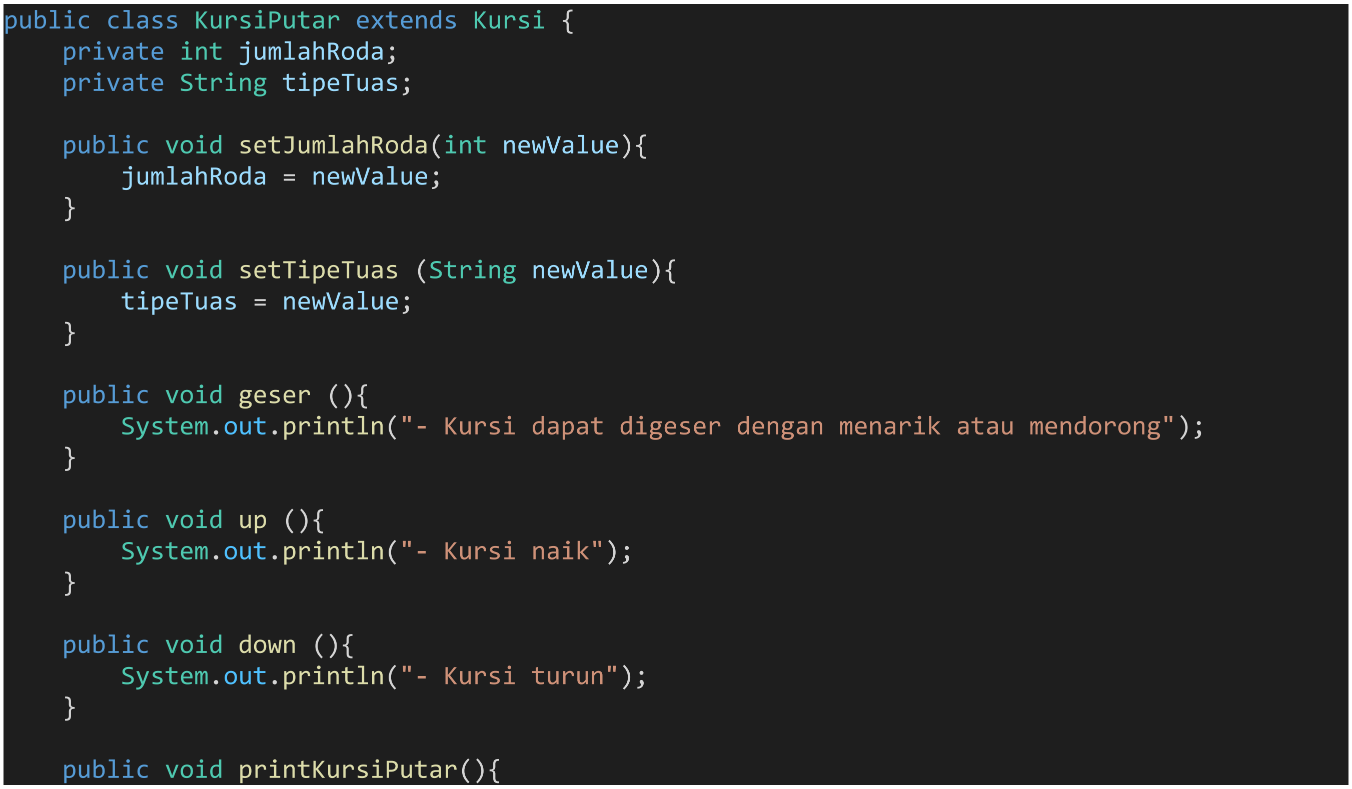
c. Convert those objects into four classes in Java programming!
d. Add one additional class as a class which inherits its attributes and methods to living room table class and desk class!
e. Add two attributes for each class!
f. Add three methods for each class including a method for showing the information!
g. Add one class named Demo for main class!
h. Instance an object for each class!
i. Apply each method for each object in main class!



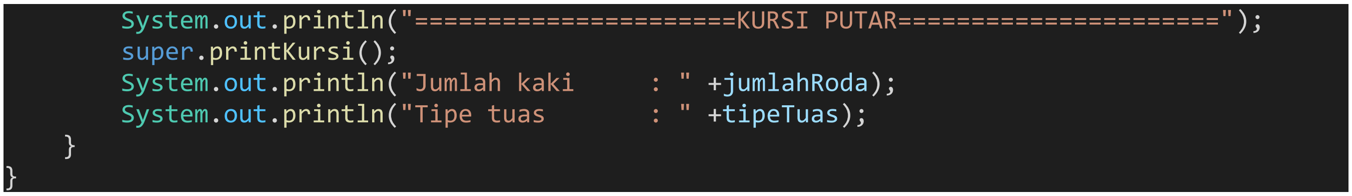
```java
 package ObjectsAround;

public class Kursi {
    private String merek,warna;
    private int kapasitas;

    public void setMerek (String newValue){
        merek = newValue;
    }

    public void warna (String newValue){
        warna = newValue;
    }

    public void kapasitas (int newValue){
        kapasitas = newValue;
    }

    public void printJudul(){
        System.out.println("=========================KURSI=========================");
    }

    public void duduk (){
        System.out.println("- Kursi diduduki");
    }

    public void printKursi(){
        System.out.println("Merek           : " +merek);
        System.out.println("Warna           : " +warna);
        System.out.println("Kapasitas       : " +kapasitas);
    }
}
```



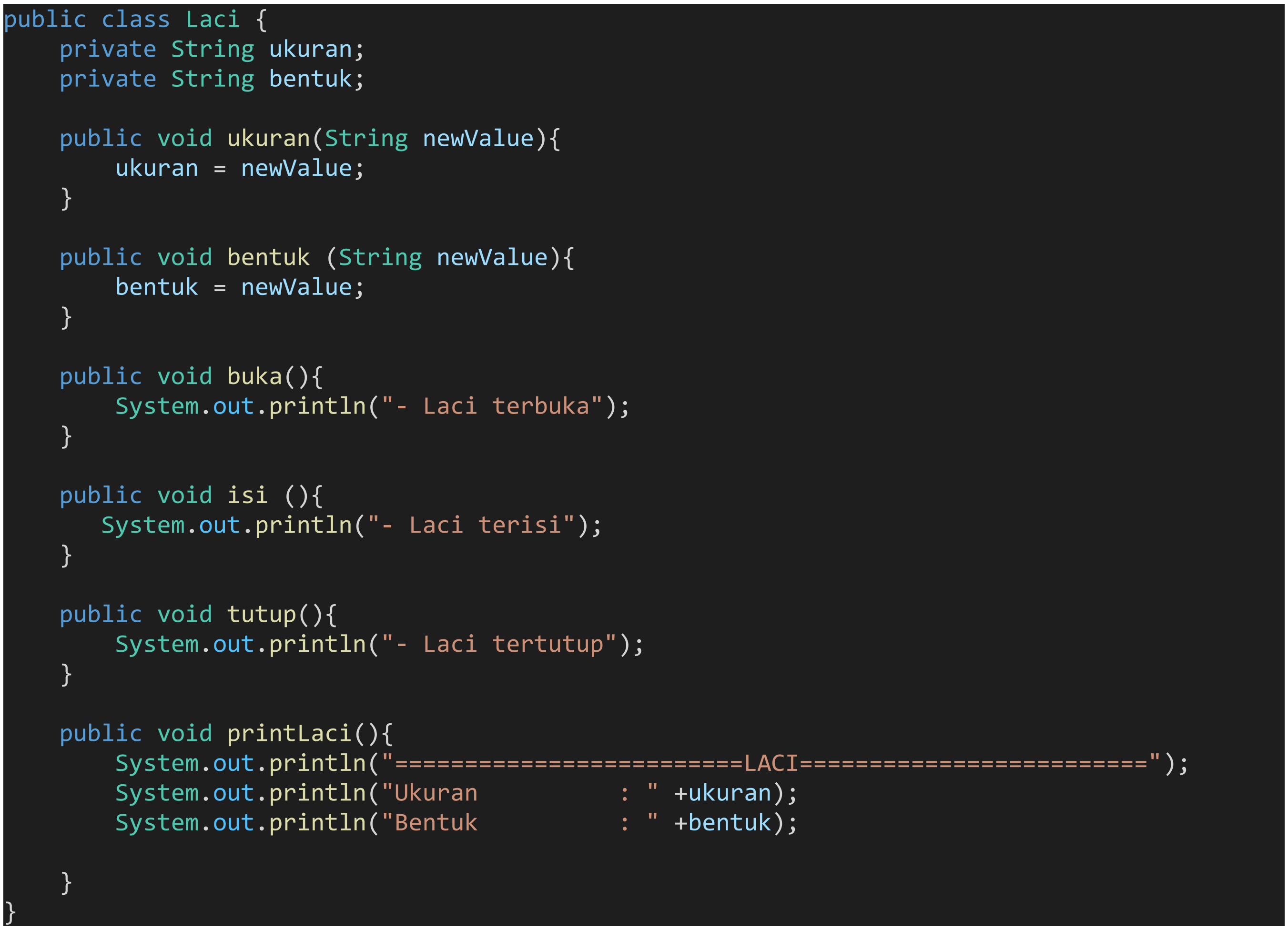
```
public class KursiGoyang extends Kursi{
    private int jumlahKaki;
    private String jenisKayu;

    public void setTjumlahKaki(int newValue){
        jumlahKaki = newValue;
    }

    public void jenisKayu (String newValue){
        jenisKayu = newValue;
    }

    public void goyang (){
        System.out.println("- Kursi bergoyang depan-belakang dengan sedikit diayun");
    }

    public void pindah (){
        System.out.println("- Pindahkan kursi dengan diangkat");
    }

    public void printKursiGoyang(){
        System.out.println("=====================KURSI GOYANG=====================");
        super.printKursi();
        System.out.println("Jumlah kaki     : " +jumlahKaki);
        System.out.println("Jenis kayu      : " +jenisKayu);
    }
}
```



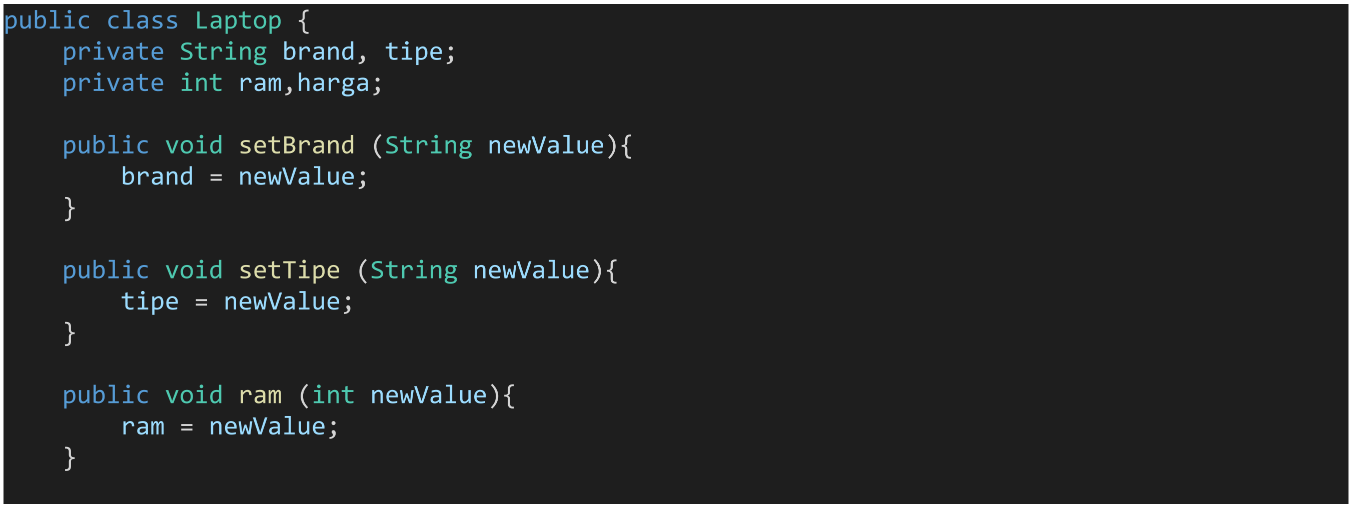
```
public class KursiPutar extends Kursi {
    private int jumlahRoda;
    private String tipeTuas;

    public void setJumlahRoda(int newValue){
        jumlahRoda = newValue;
    }

    public void setTipeTuas (String newValue){
        tipeTuas = newValue;
    }

    public void geser (){
        System.out.println("- Kursi dapat digeser dengan menarik atau mendorong");
    }

    public void up (){
        System.out.println("- Kursi naik");
    }

    public void down (){
        System.out.println("- Kursi turun");
    }

    public void printKursiPutar(){
```



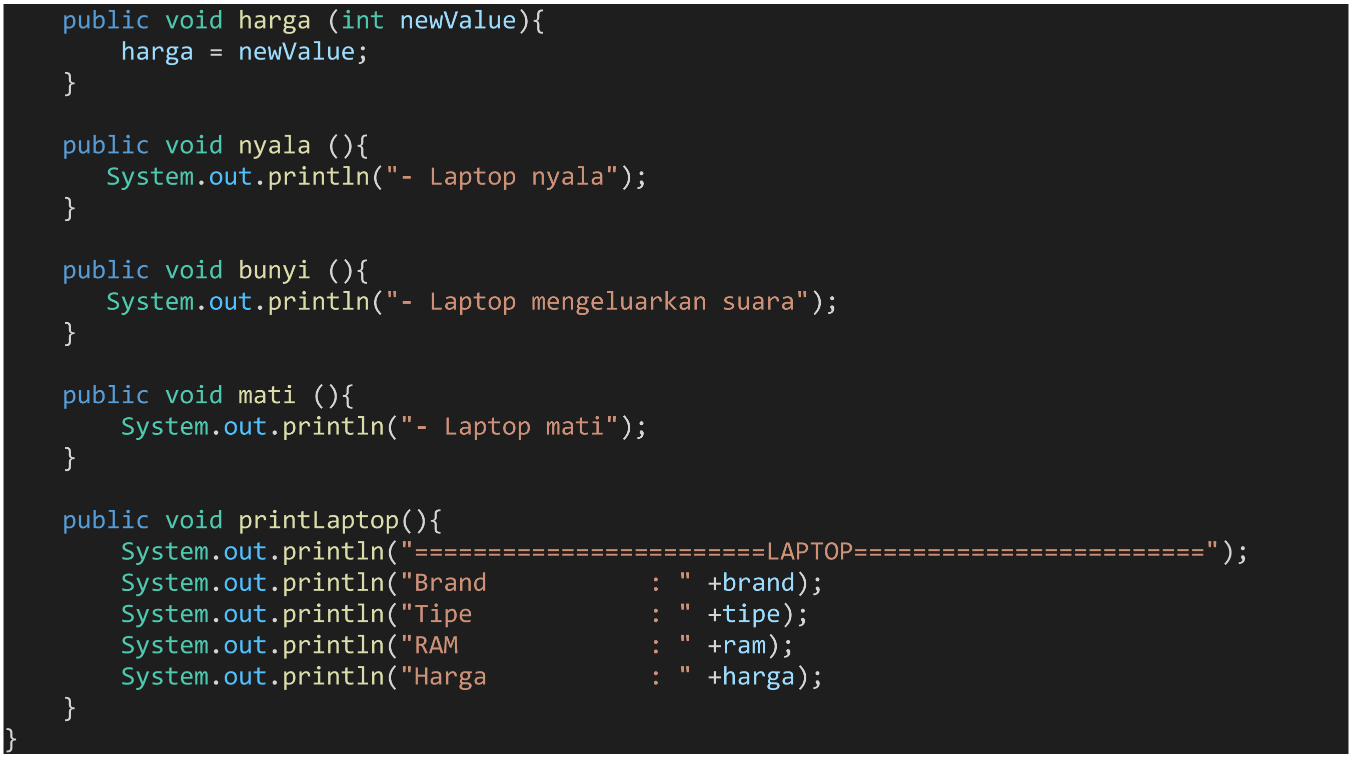
```java
        System.out.println("======================KURSI PUTAR======================");
        super.printKursi();
        System.out.println("Jumlah kaki     : " +jumlahRoda);
        System.out.println("Tipe tuas       : " +tipeTuas);
    }
}
```



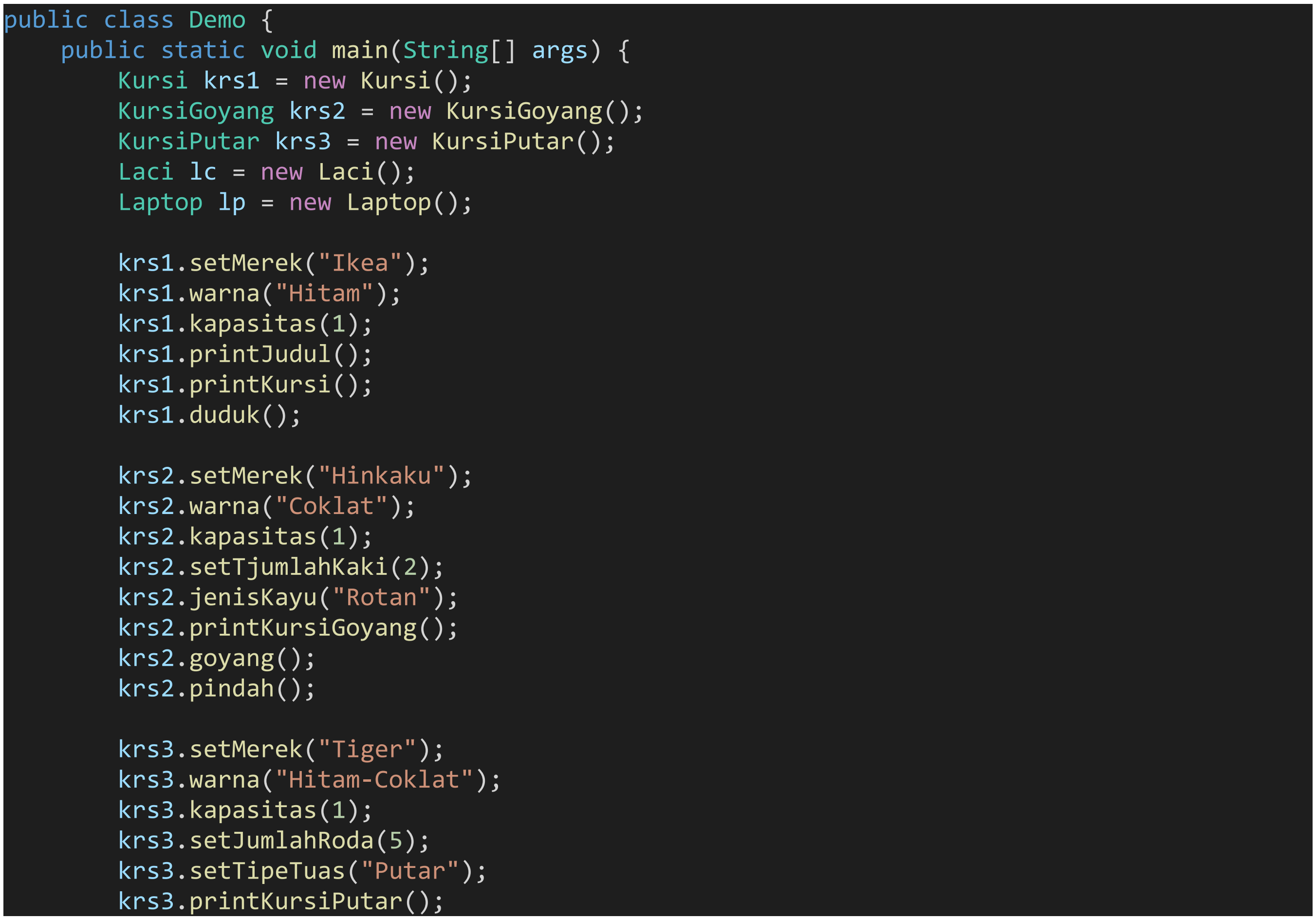
```java
public class Laci {
    private String ukuran;
    private String bentuk;

    public void ukuran(String newValue){
        ukuran = newValue;
    }

    public void bentuk (String newValue){
        bentuk = newValue;
    }

    public void buka(){
        System.out.println("- Laci terbuka");
    }

    public void isi (){
       System.out.println("- Laci terisi");
    }

    public void tutup(){
        System.out.println("- Laci tertutup");
    }

    public void printLaci(){
        System.out.println("=========================LACI=========================");
        System.out.println("Ukuran          : " +ukuran);
        System.out.println("Bentuk          : " +bentuk);

    }
}
```



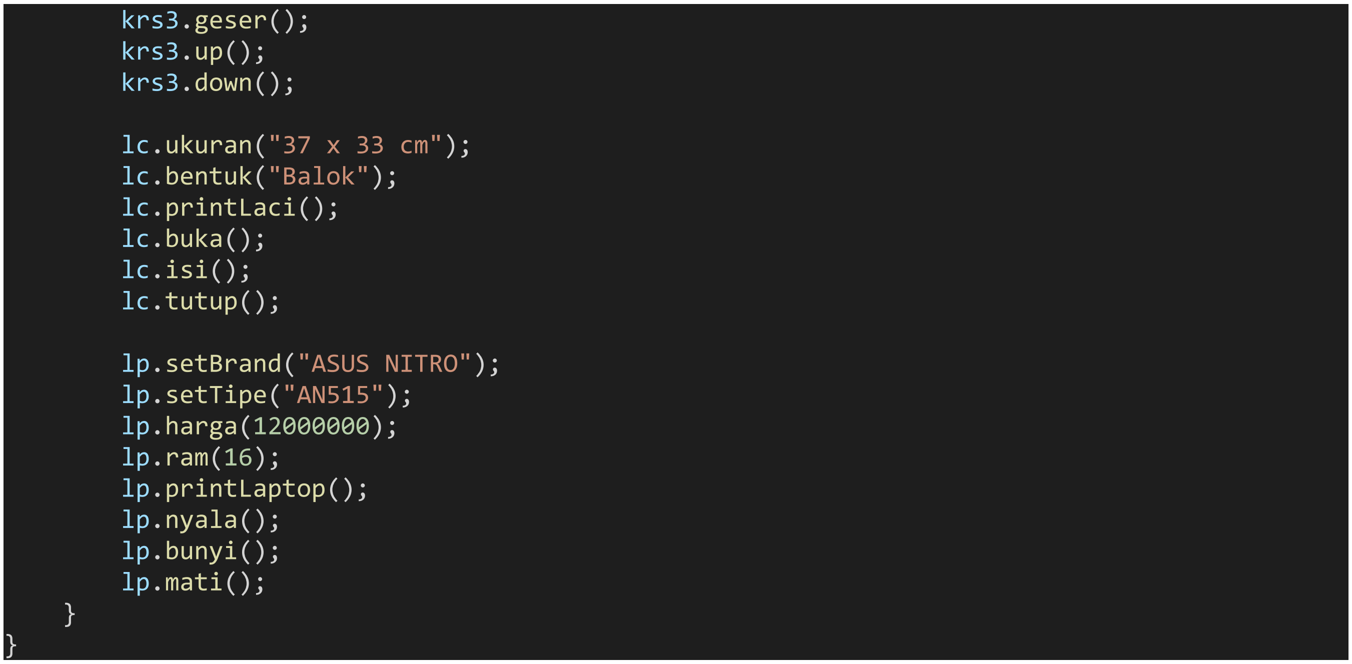
```java
public class Laptop {
    private String brand, tipe;
    private int ram,harga;

    public void setBrand (String newValue){
        brand = newValue;
    }

    public void setTipe (String newValue){
        tipe = newValue;
    }

    public void ram (int newValue){
        ram = newValue;
    }
```



```java
    public void harga (int newValue){
        harga = newValue;
    }

    public void nyala (){
       System.out.println("- Laptop nyala");
    }

    public void bunyi (){
       System.out.println("- Laptop mengeluarkan suara");
    }

    public void mati (){
        System.out.println("- Laptop mati");
    }

    public void printLaptop(){
        System.out.println("========================LAPTOP========================");
        System.out.println("Brand           : " +brand);
        System.out.println("Tipe            : " +tipe);
        System.out.println("RAM             : " +ram);
        System.out.println("Harga           : " +harga);
    }
}
```

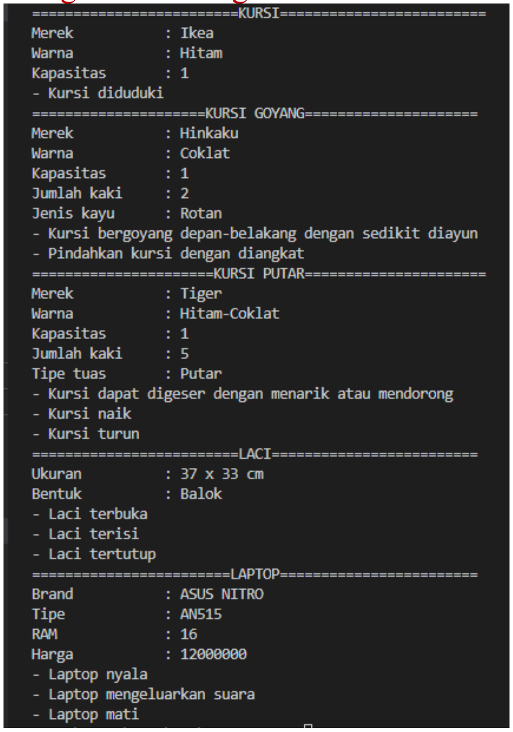


```java
public class Demo {
    public static void main(String[] args) {
        Kursi krs1 = new Kursi();
        KursiGoyang krs2 = new KursiGoyang();
        KursiPutar krs3 = new KursiPutar();
        Laci lc = new Laci();
        Laptop lp = new Laptop();

        krs1.setMerek("Ikea");
        krs1.warna("Hitam");
        krs1.kapasitas(1);
        krs1.printJudul();
        krs1.printKursi();
        krs1.duduk();

        krs2.setMerek("Hinkaku");
        krs2.warna("Coklat");
        krs2.kapasitas(1);
        krs2.setTjumlahKaki(2);
        krs2.jenisKayu("Rotan");
        krs2.printKursiGoyang();
        krs2.goyang();
        krs2.pindah();

        krs3.setMerek("Tiger");
        krs3.warna("Hitam-Coklat");
        krs3.kapasitas(1);
        krs3.setJumlahRoda(5);
        krs3.setTipeTuas("Putar");
        krs3.printKursiPutar();
```



```
        krs3.geser();
        krs3.up();
        krs3.down();

        lc.ukuran("37 x 33 cm");
        lc.bentuk("Balok");
        lc.printLaci();
        lc.buka();
        lc.isi();
        lc.tutup();

        lp.setBrand("ASUS NITRO");
        lp.setTipe("AN515");
        lp.harga(12000000);
        lp.ram(16);
        lp.printLaptop();
        lp.nyala();
        lp.bunyi();
        lp.mati();
    }
}
```

Assigment running:



```
=========================KURSI=========================
Merek           : Ikea
Warna           : Hitam
Kapasitas       : 1
- Kursi diduduki
=====================KURSI GOYANG=====================
Merek           : Hinkaku
Warna           : Coklat
Kapasitas       : 1
Jumlah kaki     : 2
Jenis kayu      : Rotan
- Kursi bergoyang depan-belakang dengan sedikit diayun
- Pindahkan kursi dengan diangkat
======================KURSI PUTAR======================
Merek           : Tiger
Warna           : Hitam-Coklat
Kapasitas       : 1
Jumlah kaki     : 5
Tipe tuas       : Putar
- Kursi dapat digeser dengan menarik atau mendorong
- Kursi naik
- Kursi turun
=========================LACI=========================
Ukuran          : 37 x 33 cm
Bentuk          : Balok
- Laci terbuka
- Laci terisi
- Laci tertutup
========================LAPTOP========================
Brand           : ASUS NITRO
Tipe            : AN515
RAM             : 16
Harga           : 12000000
- Laptop nyala
- Laptop mengeluarkan suara
- Laptop mati
```