

# **LAPORAN KUIS 2 PRAKTIKUM PBO**

## **“KUIS 2”**



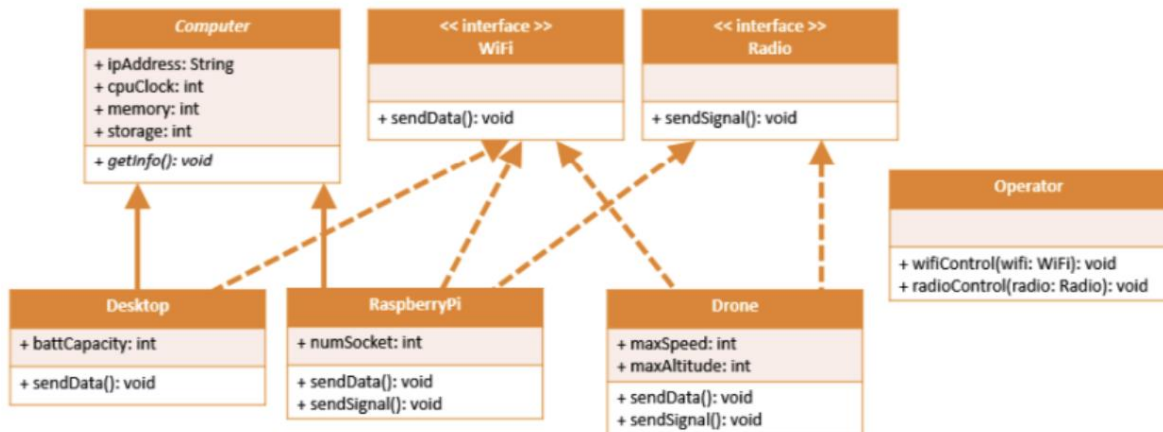
Disusun oleh:

**Farah Zulfa Hamidah/2041720069**

**D4 TEKNIK INFORMATIKA**

**POLITEKNIK NEGERI MALANG**

**2021**



## SOAL!

Buatlah kode program dari class diagram di bawah ini. Terdapat class computer yang berupa abstract class, berikut class turunannya dan class lain yang mengimplementasikan [interface](#) Wifi dan Radio. Seperti biasa, kode program dikumpulkan di github, lakukan pullreq di repo berikut <https://github.com/PBO2122-TI2B/kuis2.git>

## JAWAB!

### Kode Program Class Computer:

```

public abstract class Computer{
    protected String ipAddress;
    protected int cpuClock;
    protected int memory;
    protected int storage;

    public Computer(String ipAddress, int cpuClock, int memory, int storage){
        this.ipAddress = ipAddress;
        this.cpuClock = cpuClock;
        this.memory = memory;
        this.storage = storage;
    }

    public void setIpAddress(String ipAddress){
        this.ipAddress = ipAddress;
    }
    public String getIpAddress(){
        return ipAddress;
    }
    public void setCpuClock(int cpuClock){

```

```

        this.cpuClock = cpuClock;
    }
    public int getCpuClock(){
        return cpuClock;
    }
    public void setMemory(int memory){
        this.memory = memory;
    }
    public int getMemory(){
        return memory;
    }
    public void setStorage(int storage){
        this.storage = storage;
    }
    public int getStorage(){
        return storage;
    }
    public abstract void getInfo();
}

```

#### Kode Program Class Dekstop:

```

public class Dekstop extends Computer implements Wifi{
    private int battCapacity;

    public Dekstop(String ipAddress, int cpuClock, int memory, int storage, int
battCapacity){
        super(ipAddress, cpuClock, memory, storage);
        this.battCapacity = battCapacity;
    }

    @Override
    public void getInfo(){
        System.out.println("ipAddress: " + ipAddress);
        System.out.println("cpuClock: " + cpuClock + " MHz");
        System.out.println("Memory: " + memory + " GB");
        System.out.println("Storage: " + storage + " GB");
        System.out.println("Battery Capicity: " + battCapacity);
    }

    @Override
    public void sendData(){
        System.out.println("Data Dekstop Telah Terkontrol Melalui WIFI");
    }
}

```

```
}
```

#### Kode Program Class Drone:

```
public class Drone implements Radio, Wifi{
    private int maxSpeed;
    private int maxAltitude;

    public Drone(int maxSpeed, int maxAltitude){
        this.maxSpeed = maxSpeed;
        this.maxAltitude = maxAltitude;
    }

    public void getInfo2(){
        System.out.println("Max Speed: "+ maxSpeed +" fps");
        System.out.println("Max Altitude: "+ maxAltitude +" meter");
    }

    @Override
    public void sendSignal(){
        System.out.println("Signal Drone Telah terkontrol Melalui Radio");
    }
    @Override
    public void sendData(){
        System.out.println("Data Drone Telah Terkontrol Melalui WIFI");
    }
}
```

#### Kode Program Class RaspberryPi:

```
public class RaspberryPi extends Computer implements Radio, Wifi{
    private int numSocket;

    public RaspberryPi(String ipAddress, int cpuClock, int memory, int storage,
int numSocket){
        super(ipAddress, cpuClock, memory, storage);
        this.numSocket = numSocket;
    }

    @Override
    public void getInfo(){
        System.out.println("ipAddress: "+ ipAddress);
    }
}
```

```

        System.out.println("cpuClock: " + cpuClock + " MHz");
        System.out.println("Memory: " + memory + " GB");
        System.out.println("Storage: " + storage + " GB");
        System.out.println("Number Socket: " + numSocket + " A");
    }
    @Override
    public void sendSignal(){
        System.out.println("Signal Raspberry PI Telah terkontrol Melalui Radio");
    }
    @Override
    public void sendData(){
        System.out.println("Data Raspberry PI Telah Terkontrol Melalui WIFI");
    }
}

```

#### Kode Program Class Wifi:

```

public interface Wifi {
    public abstract void sendData();
}

```

#### Kode Program Class Radio:

```

public interface Radio {
    public abstract void sendSignal();
}

```

#### Kode Program Class Operator:

```

public class Operator {
    public void wifiControl(Wifi wifi){
        wifi.sendData();
    }
    public void radioControl(Radio radio){
        radio.sendSignal();
    }
}

```

### Kode Program Main Class:

```
public class Main {  
    public static void main(String[] args){  
        Dekstop dk = new Dekstop("74.110.208.65", 1, 4, 8, 60);  
        RaspberryPi ra = new RaspberryPi("10.8.0.12", 700, 1, 32, 3);  
        Drone dr = new Drone(60, 150);  
        Operator op = new Operator();  
  
        //Dekstop  
        System.out.println("=====");  
;  
        System.out.println("-----SPESIFIKASI DEKSTOP-----"  
");  
        dk.getInfo();  
        op.wifiControl(dk);  
  
        //Raspberry PI  
        System.out.println("");  
        System.out.println("-----SPESIFIKASI RASPBERRY PI-----"  
");  
        ra.getInfo();  
        op.wifiControl(ra);  
        op.radioControl(ra);  
  
        //Drone  
        System.out.println("");  
        System.out.println("-----SPESIFIKASI DRONE-----"  
");  
        dr.getInfo2();  
        op.wifiControl(dr);  
        op.radioControl(dr);  
        System.out.println("=====");  
;  
    }  
}
```

## Output:

```
=====
-----SPESIFIKASI DEKSTOP-----
ipAddress: 74.110.208.65
cpuClock: 1 MHz
Memory: 4 GB
Storage: 8 GB
Battery Capacity: 60
Data Dekstop Telah Terkontrol Melalui WIFI

-----SPESIFIKASI RASPBERRY PI-----
ipAddress: 10.8.0.12
cpuClock: 700 MHz
Memory: 1 GB
Storage: 32 GB
Number Socket: 3 A
Data Raspberry PI Telah Terkontrol Melalui WIFI
Signal Raspberry PI Telah terkontrol Melalui Radio

-----SPESIFIKASI DRONE-----
Max Speed: 60 fps
Max Altitude: 150 meter
Data Drone Telah Terkontrol Melalui WIFI
Signal Drone Telah terkontrol Melalui Radio
=====
PS C:\Users\LENOVO IP3\OneDrive\Documents\Kuliah\PBO\Kuis2\KuisPrakPBO2> █
```