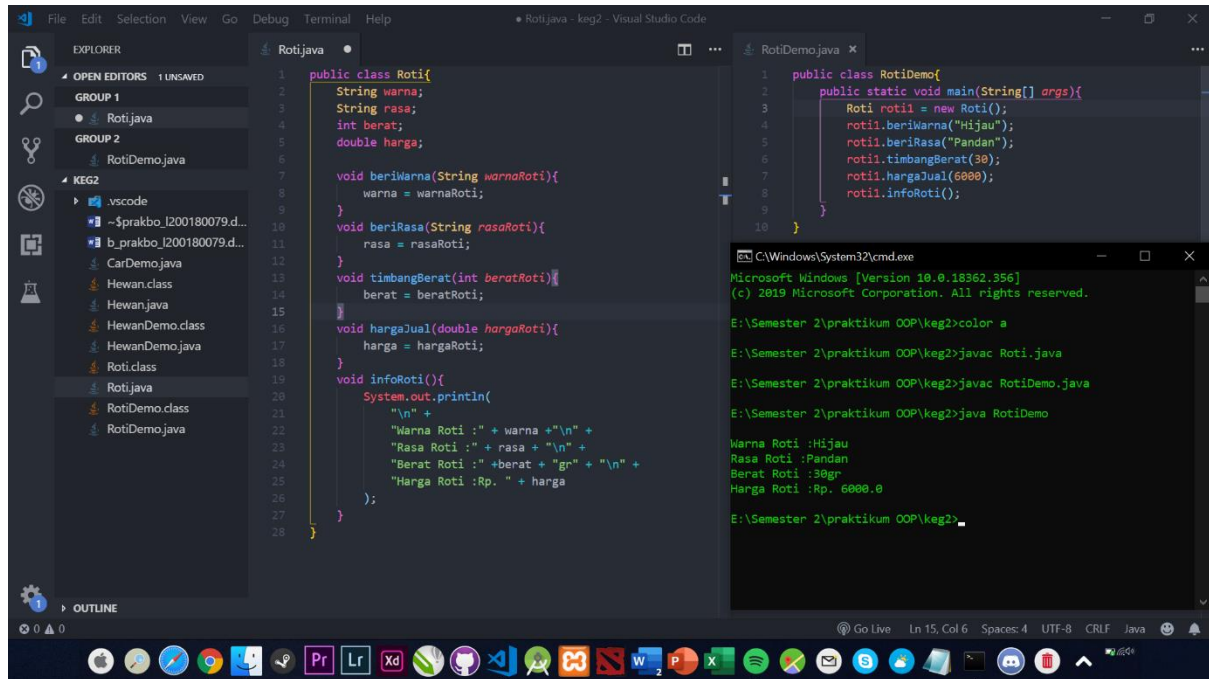


Nama : Rahmat Beny susanto

Nim : L200180079

## MODUL 2

### 1. Pembuatan class



The screenshot shows the Visual Studio Code interface with two Java files open: `Roti.java` and `RotiDemo.java`. The `Roti` class has attributes `warna`, `rasa`, `berat`, and `harga`, and methods `beriWarna`, `beriRasa`, `timbangBerat`, `hargaJual`, and `infoRoti`. The `RotiDemo` class has a `main` method that creates a `Roti` object and calls its methods. A terminal window shows the output of the program.

```
public class Roti{
    String warna;
    String rasa;
    int berat;
    double harga;

    void beriWarna(String warnaRoti){
        warna = warnaRoti;
    }
    void beriRasa(String rasaRoti){
        rasa = rasaRoti;
    }
    void timbangBerat(int beratRoti){
        berat = beratRoti;
    }
    void hargaJual(double hargaRoti){
        harga = hargaRoti;
    }
    void infoRoti(){
        System.out.println(
            "\n" +
            "Warna Roti : " + warna + "\n" +
            "Rasa Roti : " + rasa + "\n" +
            "Berat Roti : " + berat + "gr" + "\n" +
            "Harga Roti :Rp. " + harga
        );
    }
}
```

```
public class RotiDemo{
    public static void main(String[] args){
        Roti roti1 = new Roti();
        roti1.beriWarna("Hijau");
        roti1.beriRasa("Pandan");
        roti1.timbangBerat(30);
        roti1.hargaJual(6000);
        roti1.infoRoti();
    }
}
```

```
Microsoft Windows [Version 10.0.18362.356]
(c) 2019 Microsoft Corporation. All rights reserved.

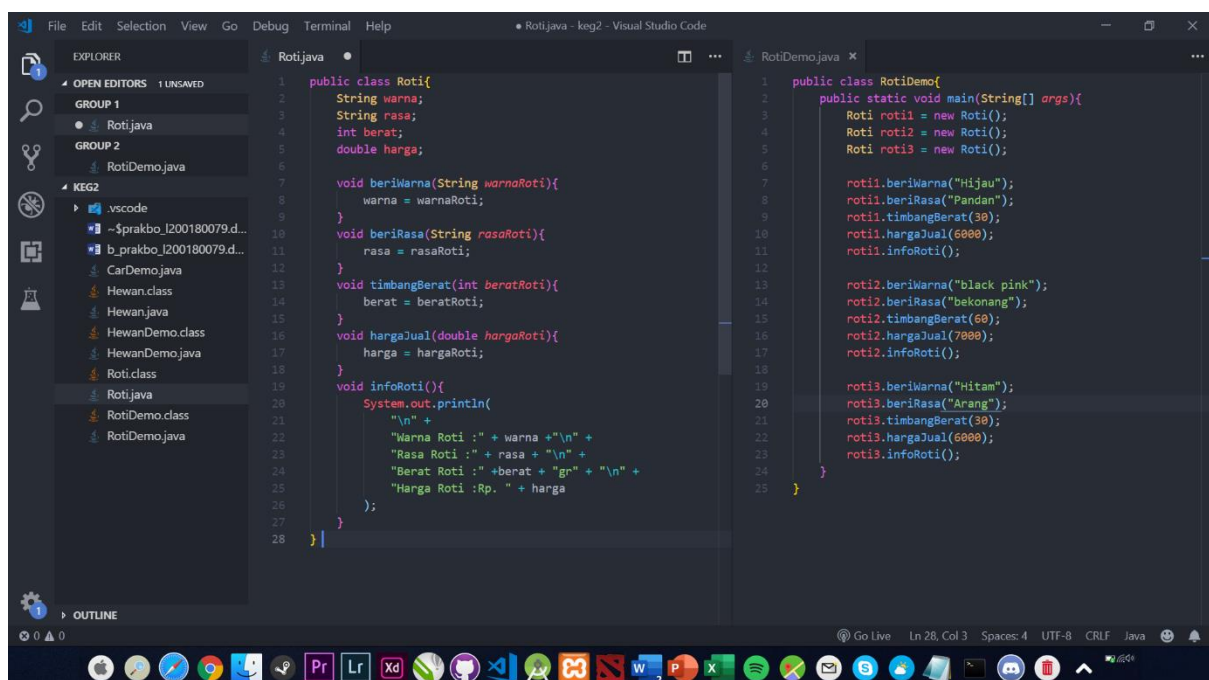
E:\Semester 2\praktikum OOP\keg2>color a
E:\Semester 2\praktikum OOP\keg2>javac Roti.java
E:\Semester 2\praktikum OOP\keg2>javac RotiDemo.java
E:\Semester 2\praktikum OOP\keg2>java RotiDemo

Warna Roti :Hijau
Rasa Roti :Pandan
Berat Roti :30gr
Harga Roti :Rp. 6000.0

E:\Semester 2\praktikum OOP\keg2>
```

### 2. Latihan

#### 2.1 memodifikasi class RotiDemo dan menambah 3 object



The screenshot shows the Visual Studio Code interface with the same two Java files. The `RotiDemo` class has been modified to create three `Roti` objects (`roti1`, `roti2`, and `roti3`) and call their methods. The terminal window shows the output of the program.

```
public class Roti{
    String warna;
    String rasa;
    int berat;
    double harga;

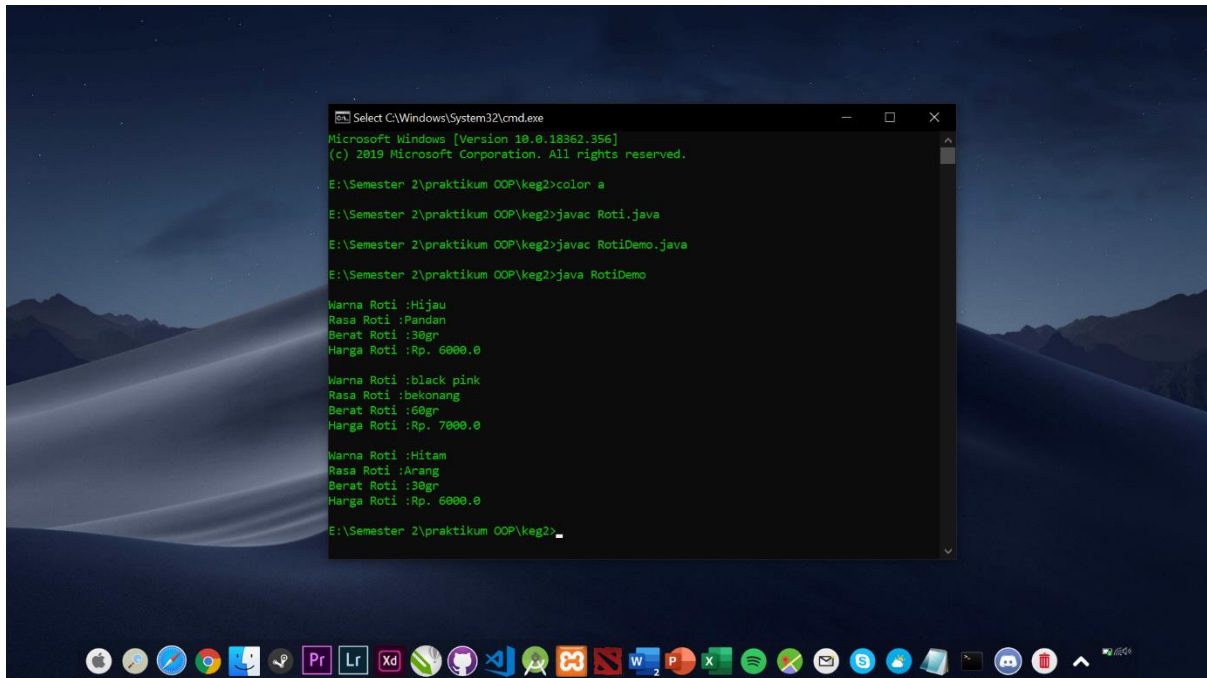
    void beriWarna(String warnaRoti){
        warna = warnaRoti;
    }
    void beriRasa(String rasaRoti){
        rasa = rasaRoti;
    }
    void timbangBerat(int beratRoti){
        berat = beratRoti;
    }
    void hargaJual(double hargaRoti){
        harga = hargaRoti;
    }
    void infoRoti(){
        System.out.println(
            "\n" +
            "Warna Roti : " + warna + "\n" +
            "Rasa Roti : " + rasa + "\n" +
            "Berat Roti : " + berat + "gr" + "\n" +
            "Harga Roti :Rp. " + harga
        );
    }
}
```

```
public class RotiDemo{
    public static void main(String[] args){
        Roti roti1 = new Roti();
        Roti roti2 = new Roti();
        Roti roti3 = new Roti();

        roti1.beriWarna("Hijau");
        roti1.beriRasa("Pandan");
        roti1.timbangBerat(30);
        roti1.hargaJual(6000);
        roti1.infoRoti();

        roti2.beriWarna("black pink");
        roti2.beriRasa("bekonang");
        roti2.timbangBerat(60);
        roti2.hargaJual(7000);
        roti2.infoRoti();

        roti3.beriWarna("Hitam");
        roti3.beriRasa("Arang");
        roti3.timbangBerat(30);
        roti3.hargaJual(6000);
        roti3.infoRoti();
    }
}
```



## 2.2 gambar class diagram dari class RotiDemo

RotiDemo
#nama : String #rasa : String #berat : integer #harga : integer
#beriWarna() #beriRasa() #timbangBerat() #hargaJual()

2.2 membuat satu class baru yang bisa digunakan sebagai template/blueprint dari class CarDemo, tidak memiliki fungsi main

The screenshot shows the Visual Studio Code interface with two Java files open: CarDemo.java and Car.java. The Explorer sidebar on the left shows a project structure with files like Roti.java, CarDemo.java, Car.java, and various class files. The CarDemo.java file contains a main method that creates two Car objects and calls methods like changeCadence, speedUp, changeGear, and printInfo. The Car.java file is a new class with attributes cadence, speed, and gear, and methods changeCadence, speedUp, changeGear, and printInfo. A terminal window at the bottom shows the execution of javac and java commands, displaying the output of the CarDemo program.

```
public class CarDemo{
    public static void main(String[] args){
        Car car1 = new Car();
        Car car2 = new Car();

        car1.changeCadence(50);
        car1.speedUp(20);
        car1.changeGear(2);
        car1.printInfo();

        car2.changeCadence(30);
        car2.speedUp(10);
        car2.changeGear(1);
        car2.printInfo();
    }
}

public class Car{
    int cadence;
    int speed;
    int gear;

    void changeCadence(int cdc){
        cadence = cdc;
    }
    void speedUp(int spd){
        speed = spd;
    }
    void changeGear(int gear){
        this.gear = gear;
    }
    void printInfo(){
        System.out.println(
            "Irama : " + cadence + "\n" +
            "Kecepatan : " + speed + "\n" +
            "Gear : " + gear + "\n"
        );
    }
}
```

```
E:\Semester 2\praktikum OOP\keg2>javac Car.java
E:\Semester 2\praktikum OOP\keg2>javac CarDemo.java
E:\Semester 2\praktikum OOP\keg2>java CarDemo
Irama :50
Kecepatan :20
Gear :2

Irama :30
Kecepatan :10
Gear :1

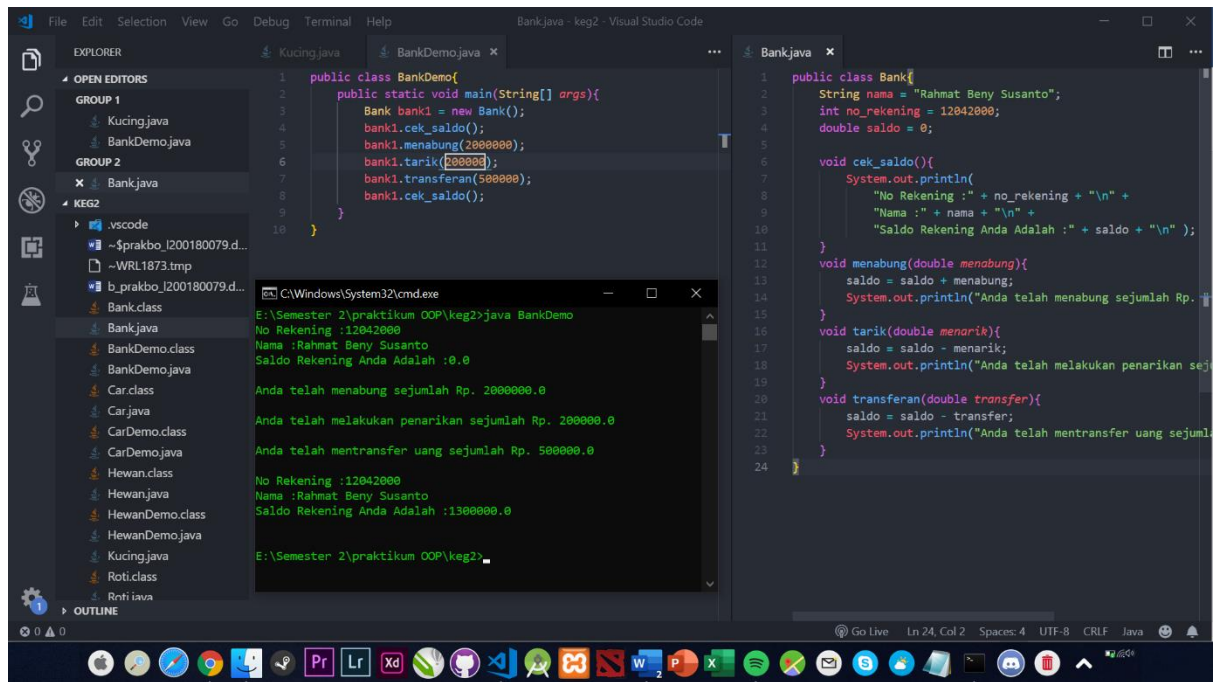
E:\Semester 2\praktikum OOP\keg2>
```

2.3 membuat suatu class yang dapat merepresentasikan sifat-sifat dari object Kucing

The screenshot shows the Visual Studio Code interface with a single Java file open: Kucing.java. The Explorer sidebar on the left shows a project structure with files like Kucing.java, Car.class, Car.java, CarDemo.class, CarDemo.java, Hewan.class, Hewan.java, HewanDemo.class, HewanDemo.java, Roti.class, Roti.java, RotiDemo.class, and RotiDemo.java. The Kucing.java file contains a class definition with attributes umur and bulu, and methods meong, umur, and a setter for umur.

```
public class Kucing{
    int umur;
    String bulu;

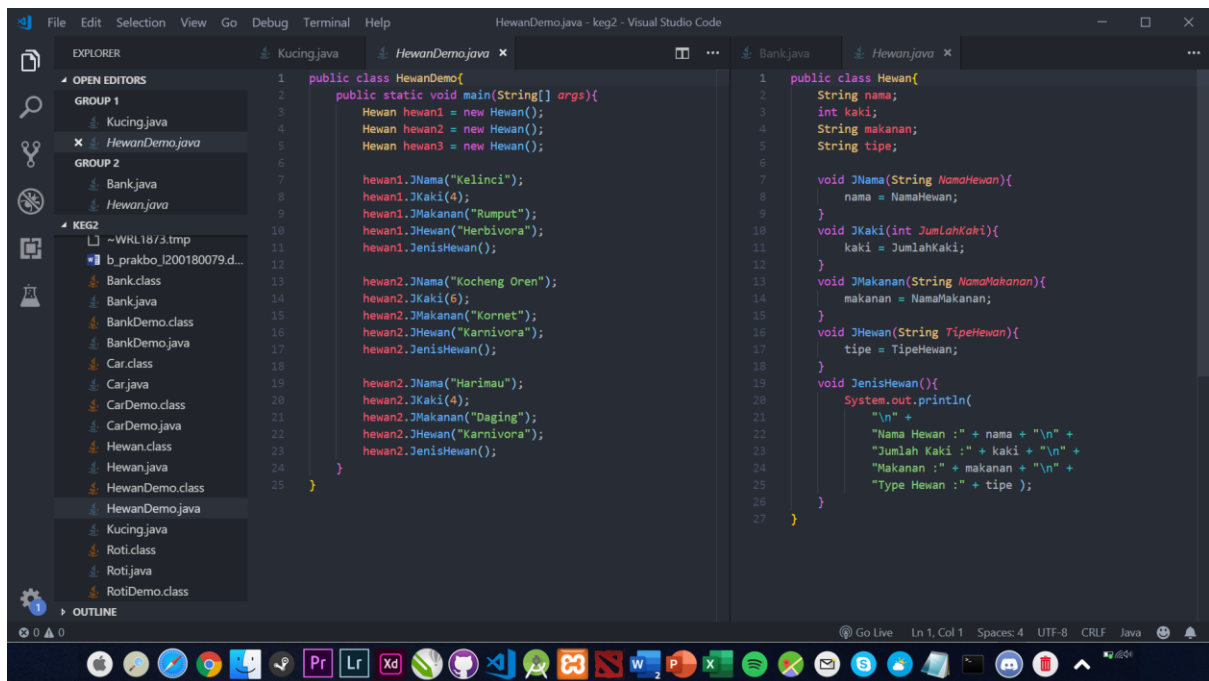
    void meong(String ngeong){
        bulu = ngeong;
    }
    void umur(int umur){
        this.umur = umur;
    }
}
```



Daftar variable dan fungsi/method yang dimiliki oleh Class String

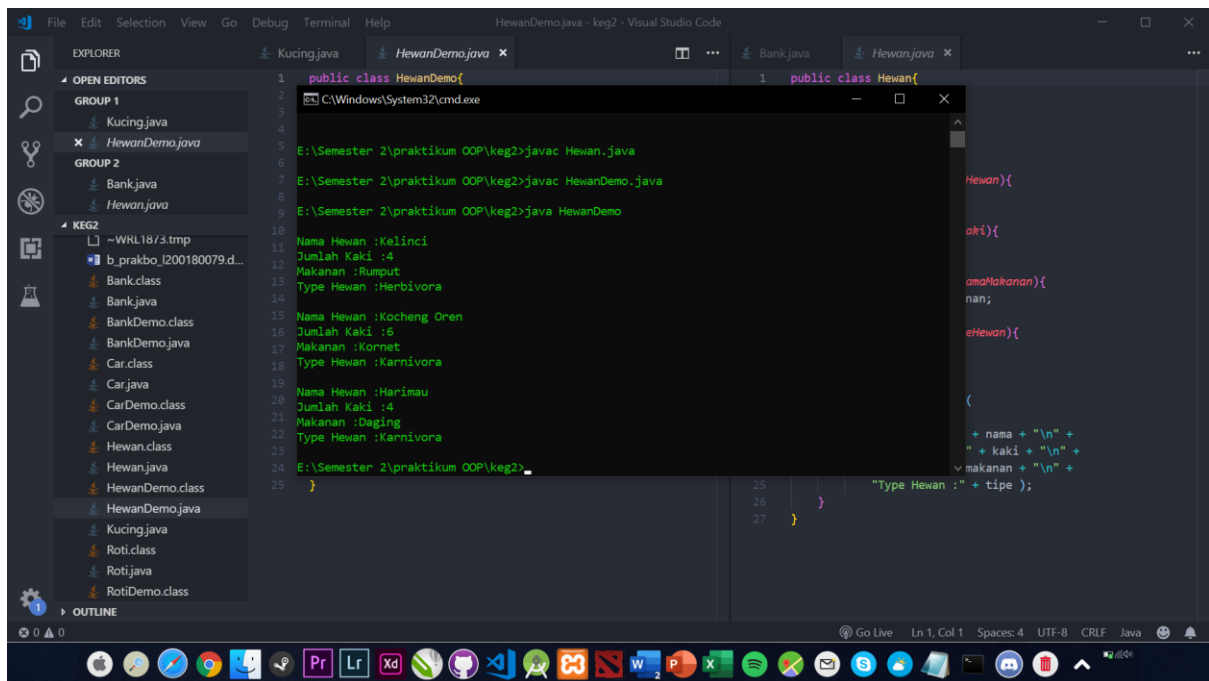
- String dataString = "Data didalam String"
- codePointAt(int) = mengembalikan code ASCII dari sebuah char yang di ambil dari String yang bersangkutan
- charAt(int) = mengembalikan Karakter yang ada di sebuah string bersangkutan sesuai dengan index yang di masukkan.
- codePointBefore(int) = mengembalikan code ASCII dari karakter yang di ambil dari sebuah String

## Pekerjaan Rumah



```
1 public class HewanDemo{
2     public static void main(String[] args){
3         Hewan hewan1 = new Hewan();
4         Hewan hewan2 = new Hewan();
5         Hewan hewan3 = new Hewan();
6
7         hewan1.JNama("Kelinci");
8         hewan1.JKaki(4);
9         hewan1.JMakanan("Rumput");
10        hewan1.JHewan("Herbivora");
11        hewan1.JenisHewan();
12
13        hewan2.JNama("Kocheng Oren");
14        hewan2.JKaki(6);
15        hewan2.JMakanan("Kornet");
16        hewan2.JHewan("Karnivora");
17        hewan2.JenisHewan();
18
19        hewan2.JNama("Harimau");
20        hewan2.JKaki(4);
21        hewan2.JMakanan("Daging");
22        hewan2.JHewan("Karnivora");
23        hewan2.JenisHewan();
24    }
25 }
```

```
1 public class Hewan{
2     String nama;
3     int kaki;
4     String makanan;
5     String tipe;
6
7     void JNama(String NamaHewan){
8         nama = NamaHewan;
9     }
10    void JKaki(int JumlahKaki){
11        kaki = JumlahKaki;
12    }
13    void JMakanan(String NamaMakanan){
14        makanan = NamaMakanan;
15    }
16    void JHewan(String TipeHewan){
17        tipe = TipeHewan;
18    }
19    void JenisHewan(){
20        System.out.println(
21            "\n" +
22            "Nama Hewan : " + nama + "\n" +
23            "Jumlah Kaki : " + kaki + "\n" +
24            "Makanan : " + makanan + "\n" +
25            "Type Hewan : " + tipe );
26    }
27 }
```



```
1 public class HewanDemo{
2     public static void main(String[] args){
3         Hewan hewan1 = new Hewan();
4         Hewan hewan2 = new Hewan();
5         Hewan hewan3 = new Hewan();
6
7         hewan1.JNama("Kelinci");
8         hewan1.JKaki(4);
9         hewan1.JMakanan("Rumput");
10        hewan1.JHewan("Herbivora");
11        hewan1.JenisHewan();
12
13        hewan2.JNama("Kocheng Oren");
14        hewan2.JKaki(6);
15        hewan2.JMakanan("Kornet");
16        hewan2.JHewan("Karnivora");
17        hewan2.JenisHewan();
18
19        hewan2.JNama("Harimau");
20        hewan2.JKaki(4);
21        hewan2.JMakanan("Daging");
22        hewan2.JHewan("Karnivora");
23        hewan2.JenisHewan();
24    }
25 }
```

```
1 public class Hewan{
2     String nama;
3     int kaki;
4     String makanan;
5     String tipe;
6
7     void JNama(String NamaHewan){
8         nama = NamaHewan;
9     }
10    void JKaki(int JumlahKaki){
11        kaki = JumlahKaki;
12    }
13    void JMakanan(String NamaMakanan){
14        makanan = NamaMakanan;
15    }
16    void JHewan(String TipeHewan){
17        tipe = TipeHewan;
18    }
19    void JenisHewan(){
20        System.out.println(
21            "\n" +
22            "Nama Hewan : " + nama + "\n" +
23            "Jumlah Kaki : " + kaki + "\n" +
24            "Makanan : " + makanan + "\n" +
25            "Type Hewan : " + tipe );
26    }
27 }
```

```
C:\Windows\System32\cmd.exe
E:\Semester 2\praktikum OOP\keg2>javac Hewan.java
E:\Semester 2\praktikum OOP\keg2>javac HewanDemo.java
E:\Semester 2\praktikum OOP\keg2>java HewanDemo
Nama Hewan :Kelinci
Jumlah Kaki :4
Makanan :Rumput
Type Hewan :Herbivora
Nama Hewan :Kocheng Oren
Jumlah Kaki :6
Makanan :Kornet
Type Hewan :Karnivora
Nama Hewan :Harimau
Jumlah Kaki :4
Makanan :Daging
Type Hewan :Karnivora
E:\Semester 2\praktikum OOP\keg2>
```



The screenshot shows the Visual Studio Code interface with two Java files open. The Explorer sidebar on the left shows a project structure with files like Kucing.java, Dosen.java, Mahasiswa.java, and Karyawan.java. The main editor area displays the code for these files.

```
1 public class Dosen{
2     String nama;
3     int nik;
4     String pendidikan;
5     Date tgllahir;
6
7     void tampilkanNama(String name){
8         nama = name;
9     }
10    void tampilkanNik(int nik){
11        this.nik = nik;
12    }
13    void tampilkanTglLahir(Date tgl){
14        tgllahir = tgl;
15    }
16 }
```

```
1 public class Mahasiswa{
2     String nama;
3     String alamat;
4     int nim;
5     int semester;
6
7     void tampilkanNama(String name){
8         nama = name;
9     }
10    void tampilkanAlamat(String almt){
11        alamat = almt;
12    }
13    void tampilkanNim(int Nim){
14        nim = Nim;
15    }
16    void tampilkanSemester(int smtr){
17        semester = smtr;
18    }
19 }
```

The screenshot shows the Visual Studio Code interface with the Karyawan.java file open. The Explorer sidebar on the left shows a project structure with files like Mahasiswa.java, Karyawan.java, Kucing.java, and others. The main editor area displays the code for Karyawan.java.

```
1 public class Karyawan{
2     String nama;
3     String alamat;
4     String jabatan;
5     int gaji;
6
7     void tampilkanNama(String name){
8         nama = name;
9     }
10    void tampilkanAlamat(String almt){
11        alamat = almt;
12    }
13    void tampilkanJabatan(String jbtn){
14        jabatan = jbtn;
15    }
16    void tampilkanGaji(int gji){
17        gaji = gji;
18    }
19 }
```