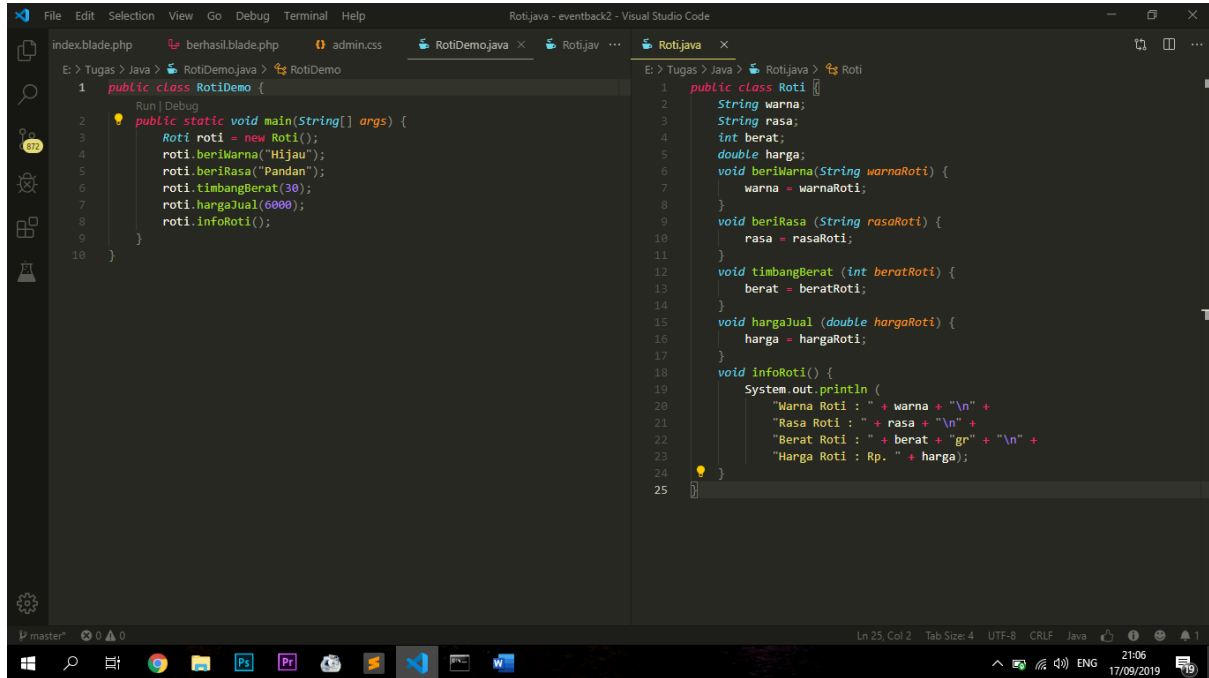


Nama : Wahyu Setyaji Rama Dwijaya  
NIM : L200180065  
Kelas : B

## MODUL 2

### Tugas 2.2

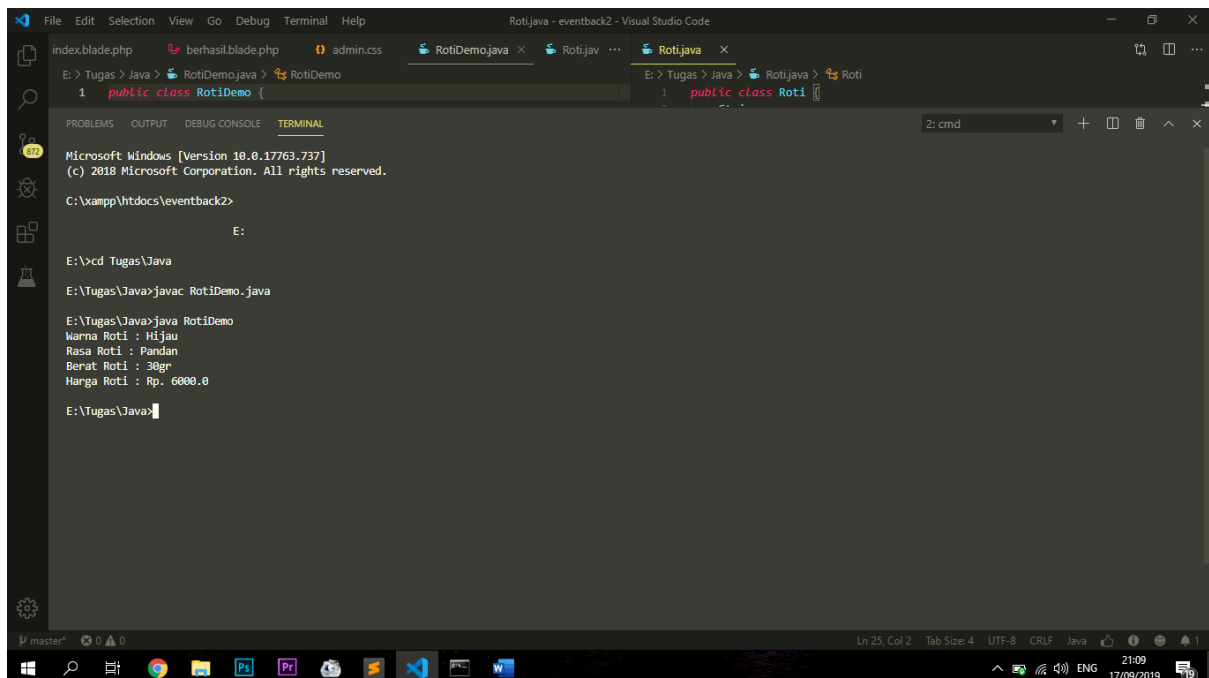


The screenshot shows the Visual Studio Code editor with two Java files open. The left file, `RotiDemo.java`, contains a `main` method that creates a `Roti` object and calls its methods. The right file, `Roti.java`, defines the `Roti` class with attributes for color, taste, weight, and price, and methods to set these attributes and display them.

```
1 public class RotiDemo {  
2     public static void main(String[] args) {  
3         Roti roti = new Roti();  
4         roti.beriWarna("Hijau");  
5         roti.beriRasa("Pandan");  
6         roti.timbangBerat(30);  
7         roti.hargaJual(6000);  
8         roti.infoRoti();  
9     }  
10 }
```

```
1 public class Roti {  
2     String warna;  
3     String rasa;  
4     int berat;  
5     double harga;  
6     void beriWarna(String warnaRoti) {  
7         warna = warnaRoti;  
8     }  
9     void beriRasa (String rasaRoti) {  
10        rasa = rasaRoti;  
11    }  
12    void timbangBerat (int beratRoti) {  
13        berat = beratRoti;  
14    }  
15    void hargaJual (double hargaRoti) {  
16        harga = hargaRoti;  
17    }  
18    void infoRoti() {  
19        System.out.println (  
20            "Warna Roti : " + warna + "\n" +  
21            "Rasa Roti : " + rasa + "\n" +  
22            "Berat Roti : " + berat + "gr" + "\n" +  
23            "Harga Roti : Rp. " + harga);  
24    }  
25 }
```

### Hasil CMD:

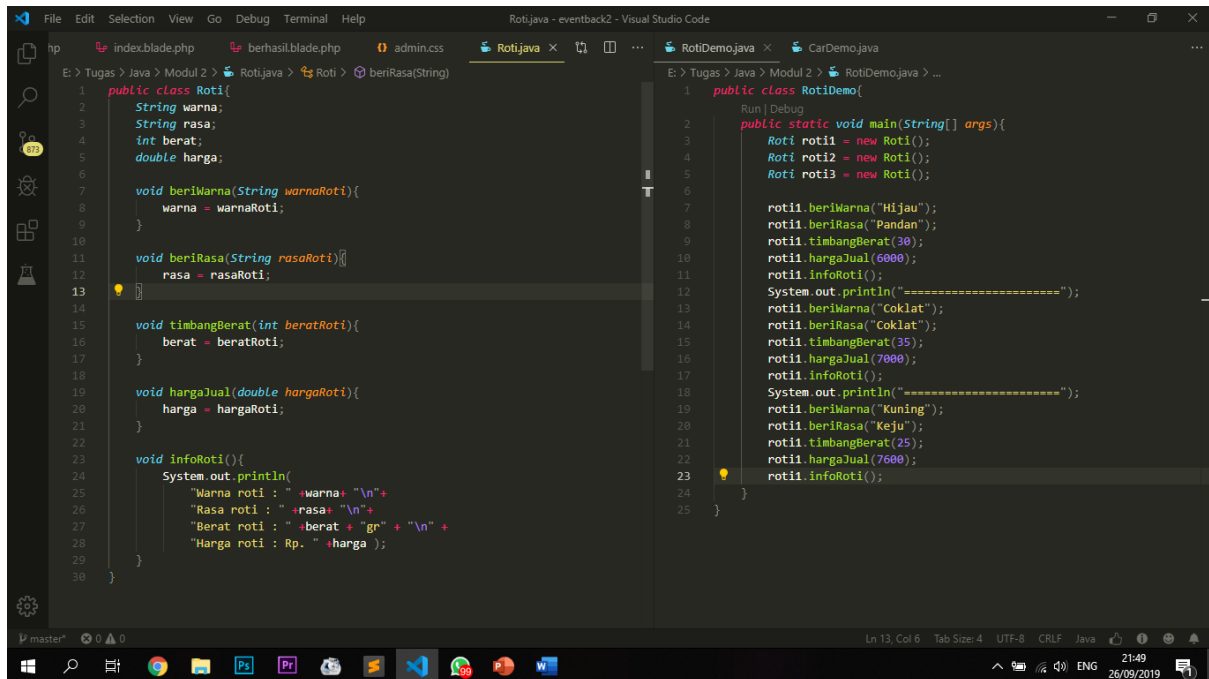


The screenshot shows the Visual Studio Code editor with the `RotiDemo.java` file open. The `TERMINAL` tab is active, displaying the command prompt output for running the Java program. The output shows the color, taste, weight, and price of the `Roti` object.

```
Microsoft Windows [Version 10.0.17763.737]  
(c) 2018 Microsoft Corporation. All rights reserved.  
  
C:\xampp\htdocs\eventback2>  
  
E:  
  
E:\>cd Tugas\Java  
  
E:\Tugas\Java>javac RotiDemo.java  
  
E:\Tugas\Java>java RotiDemo  
Warna Roti : Hijau  
Rasa Roti : Pandan  
Berat Roti : 30gr  
Harga Roti : Rp. 6000.0  
  
E:\Tugas\Java>
```

## Latihan 2.4

### 1. Modifikasi Class RotiDemo dan 3 object baru



The screenshot shows the Visual Studio Code editor with two Java files open: `Roti.java` and `RotiDemo.java`. The `Roti.java` file contains the `Roti` class with attributes `warna`, `rasa`, `berat`, and `harga`, and methods `beriWarna`, `beriRasa`, `timbangBerat`, `hargaJual`, and `infoRoti`. The `RotiDemo.java` file contains the `RotiDemo` class with a `main` method that creates three `Roti` objects: `roti1` (Hijau, Pandan, 30g, 6000), `roti2` (Coklat, Coklat, 35g, 7000), and `roti3` (Kuning, Keju, 25g, 7600). The `infoRoti` method is called for each object to display its details.

```
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(
            "Warna roti : " + warna + "\n" +
            "Rasa roti : " + rasa + "\n" +
            "Berat roti : " + berat + "gr" + "\n" +
            "Harga roti : Rp. " + harga );
    }
}

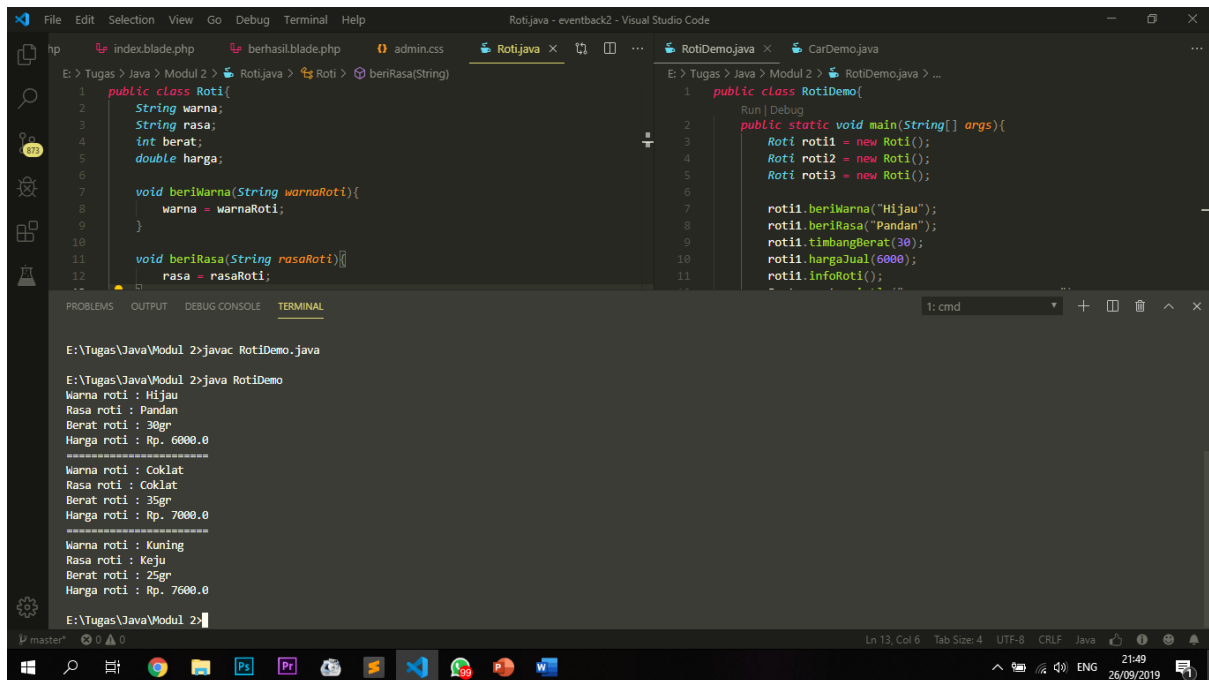
public class RotiDemo {
    public static void main(String[] args) {
        Roti roti1 = new Roti();
        Roti roti2 = new Roti();
        Roti roti3 = new Roti();

        roti1.beriWarna("Hijau");
        roti1.beriRasa("Pandan");
        roti1.timbangBerat(30);
        roti1.hargaJual(6000);
        roti1.infoRoti();
        System.out.println("=====");

        roti1.beriWarna("Coklat");
        roti1.beriRasa("Coklat");
        roti1.timbangBerat(35);
        roti1.hargaJual(7000);
        roti1.infoRoti();
        System.out.println("=====");

        roti1.beriWarna("Kuning");
        roti1.beriRasa("Keju");
        roti1.timbangBerat(25);
        roti1.hargaJual(7600);
        roti1.infoRoti();
    }
}
```

### Hasil CMD

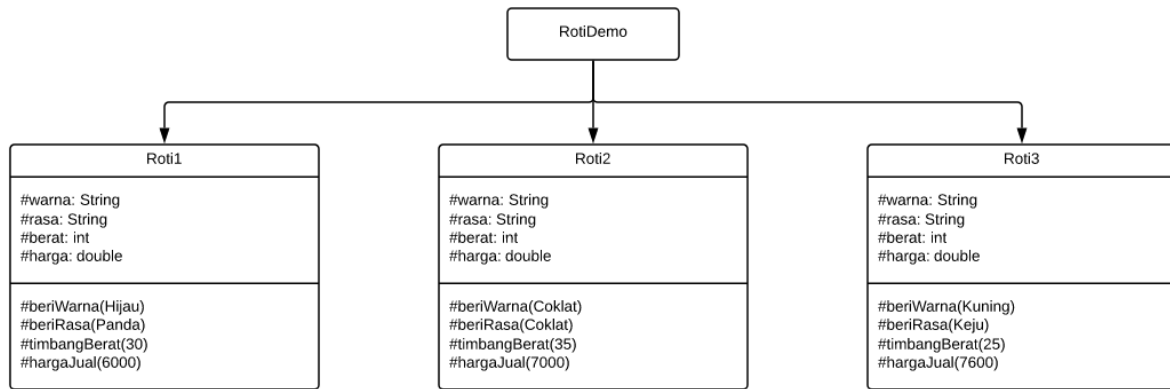


The screenshot shows the Visual Studio Code editor with the `RotiDemo.java` file open. The `main` method is executed, and the output is displayed in the terminal window. The output shows the details of three `Roti` objects: `roti1` (Hijau, Pandan, 30g, 6000), `roti2` (Coklat, Coklat, 35g, 7000), and `roti3` (Kuning, Keju, 25g, 7600). The output is formatted with line breaks and a separator line between each object's details.

```
E:\Tugas\Java\Modul 2>javac RotiDemo.java

E:\Tugas\Java\Modul 2>java RotiDemo
Warna roti : Hijau
Rasa roti : Pandan
Berat roti : 30gr
Harga roti : Rp. 6000.0
=====
Warna roti : Coklat
Rasa roti : Coklat
Berat roti : 35gr
Harga roti : Rp. 7000.0
=====
Warna roti : Kuning
Rasa roti : Keju
Berat roti : 25gr
Harga roti : Rp. 7600.0
E:\Tugas\Java\Modul 2>
```

## 2. Diagram RotiDemo



## 3. CarDemo

```
1 public class Car {
2     int cadence;
3     int speed;
4     int gear;
5
6     void changeCadence(int changeCan){
7         cadence = changeCan;
8     }
9
10    void speedUp(int upSpeed){
11        speed = upSpeed;
12    }
13
14    void changeGear(int gearChange){
15        gear = gearChange;
16    }
17
18    void printInfo(){
19        System.out.println(
20            "Cadence : "+cadence+"\n"+
21            "Speed : "+speed+"\n"+
22            "Gear : "+gear );
23    }
24 }
```

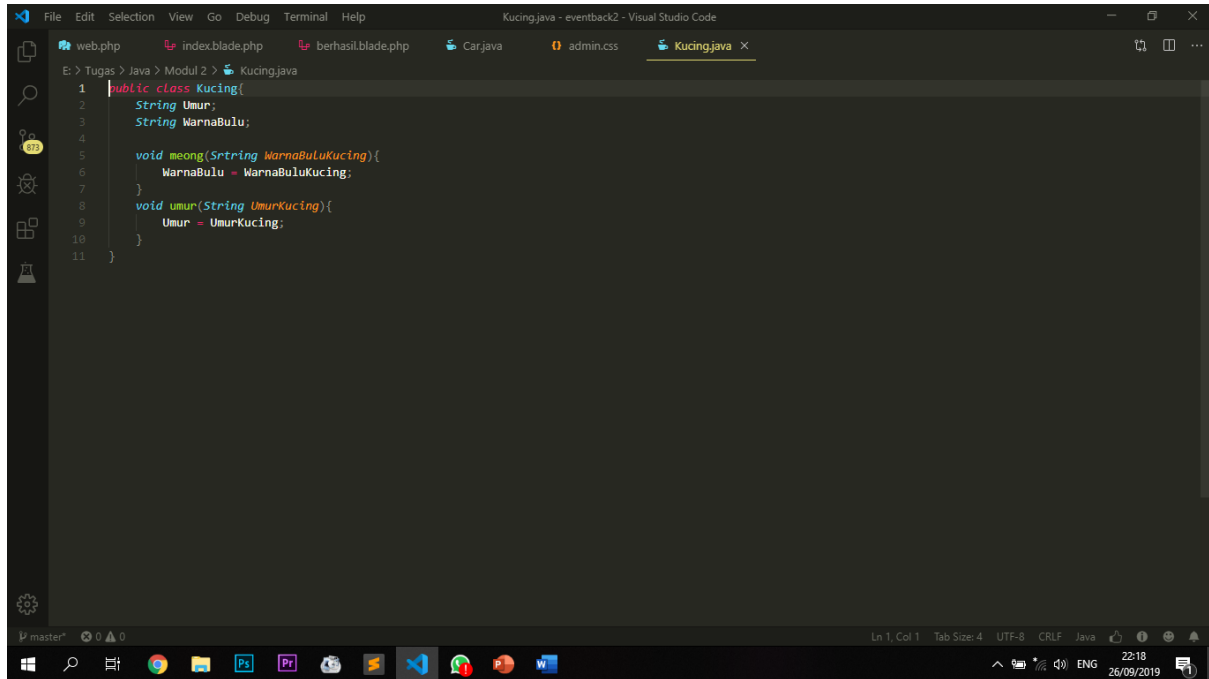
The screenshot shows the **CarDemo.java** file in Visual Studio Code. The code defines a **Car** class with attributes `cadence`, `speed`, and `gear`. It includes methods `changeCadence`, `speedUp`, `changeGear`, and `printInfo`. The `printInfo` method uses `System.out.println` to display the current state of the car.

## Hasil CMD:

```
E:\>cd Tugas\JavaModul 2
E:\Tugas\Java\Modul 2>javac CarDemo.java
E:\Tugas\Java\Modul 2>java CarDemo
Cadence : 50
Speed : 20
Gear : 2
-----
Cadence : 30
Speed : 10
Gear : 1
E:\Tugas\Java\Modul 2>
```

The screenshot shows the terminal output of the **CarDemo** program. The program is executed in the directory `E:\Tugas\Java\Modul 2`. The output shows the initial state of the car (Cadence: 50, Speed: 20, Gear: 2) and the state after a series of operations (Cadence: 30, Speed: 10, Gear: 1).

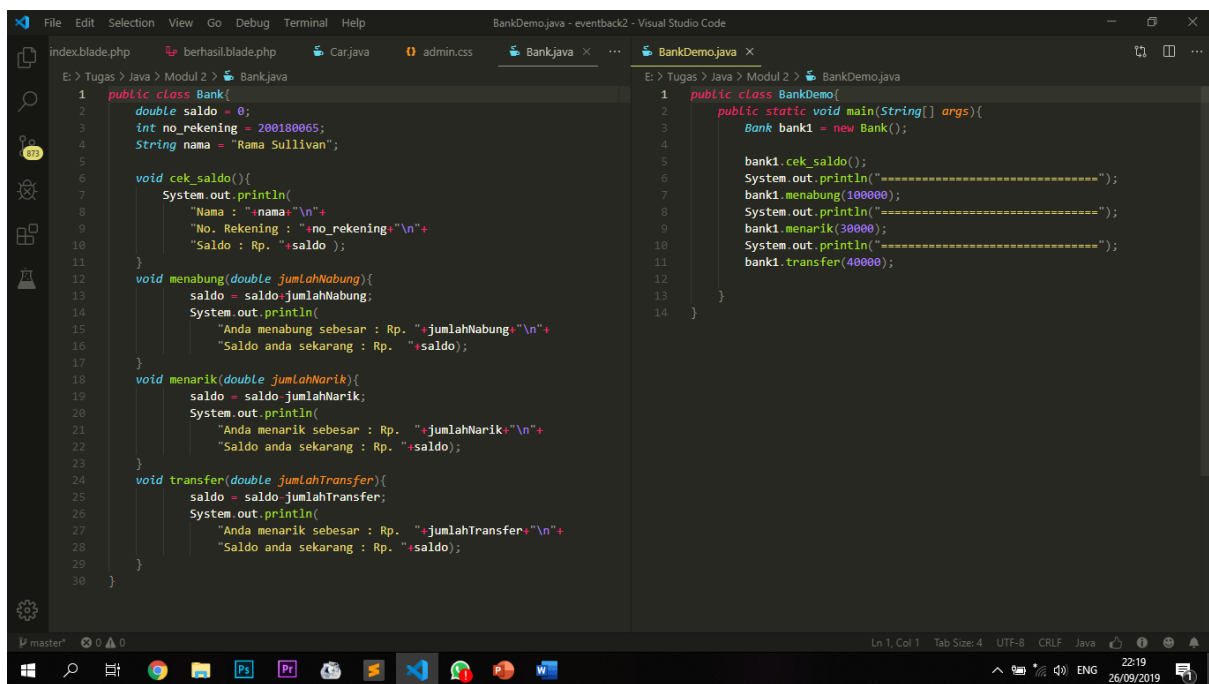
## 4. Kucing



The screenshot shows the Visual Studio Code editor with the file `Kucing.java` open. The code defines a `Kucing` class with attributes `Umur` and `WarnaBulu`, and methods `meong` and `umur`.

```
1 public class Kucing{
2     String Umur;
3     String WarnaBulu;
4
5     void meong(String WarnaBuluKucing){
6         WarnaBulu = WarnaBuluKucing;
7     }
8
9     void umur(String UmurKucing){
10        Umur = UmurKucing;
11    }
```

## 5. BankDemo

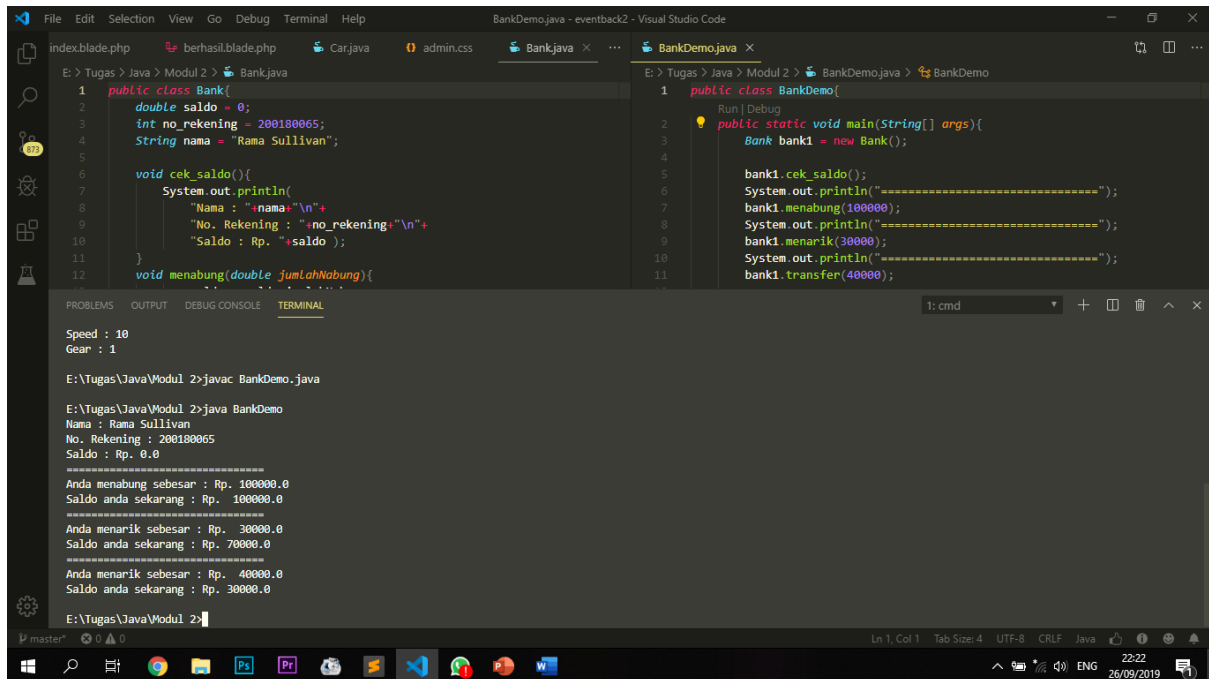


The screenshot shows the Visual Studio Code editor with two files open: `Bank.java` and `BankDemo.java`. `Bank.java` defines a `Bank` class with attributes `saldo`, `no_rekening`, and `nama`, and methods `cekSaldo`, `menabung`, `menarik`, and `transfer`. `BankDemo.java` contains a `main` method that creates a `Bank` object and performs several operations.

```
1 public class Bank{
2     double saldo = 0;
3     int no_rekening = 2001800065;
4     String nama = "Rama Sullivan";
5
6     void cekSaldo(){
7         System.out.println(
8             "Nama : "+nama+"\n"+
9             "No. Rekening : "+no_rekening+"\n"+
10            "Saldo : Rp. "+saldo);
11    }
12
13    void menabung(double jumlahNabung){
14        saldo = saldo+jumlahNabung;
15        System.out.println(
16            "Anda menabung sebesar : Rp. "+jumlahNabung+"\n"+
17            "Saldo anda sekarang : Rp. "+saldo);
18    }
19
20    void menarik(double jumlahNarik){
21        saldo = saldo-jumlahNarik;
22        System.out.println(
23            "Anda menarik sebesar : Rp. "+jumlahNarik+"\n"+
24            "Saldo anda sekarang : Rp. "+saldo);
25    }
26
27    void transfer(double jumlahTransfer){
28        saldo = saldo-jumlahTransfer;
29        System.out.println(
30            "Anda menarik sebesar : Rp. "+jumlahTransfer+"\n"+
31            "Saldo anda sekarang : Rp. "+saldo);
32    }
33 }
```

```
1 public class BankDemo{
2     public static void main(String[] args){
3         Bank bank1 = new Bank();
4
5         bank1.cekSaldo();
6         System.out.println("=====");
7         bank1.menabung(100000);
8         System.out.println("=====");
9         bank1.menarik(30000);
10        System.out.println("=====");
11        bank1.transfer(40000);
12    }
13 }
14 }
```

## Hasil CMD:



The screenshot shows the Visual Studio Code editor with two Java files open: `Bank.java` and `BankDemo.java`. The `Bank.java` file contains a `Bank` class with attributes `saldo`, `no_rekening`, and `nama`, and methods `cek_saldo()` and `menabung()`. The `BankDemo.java` file contains a `BankDemo` class with a `main` method that creates a `Bank` object and calls its methods. The terminal window at the bottom shows the output of running the program, displaying the account details and the results of the deposit and withdrawal operations.

```
Speed : 10
Gear : 1

E:\Tugas\Java\Modul 2>javac BankDemo.java

E:\Tugas\Java\Modul 2>java BankDemo
Nama : Rama Sullivan
No. Rekening : 200180065
Saldo : Rp. 0.0

=====
Anda menabung sebesar : Rp. 100000.0
Saldo anda sekarang : Rp. 100000.0

=====
Anda menarik sebesar : Rp. 30000.0
Saldo anda sekarang : Rp. 70000.0

=====
Anda menarik sebesar : Rp. 40000.0
Saldo anda sekarang : Rp. 30000.0

E:\Tugas\Java\Modul 2>
```

## 6. Berikut adalah daftar metode yang didukung oleh kelas String:

- `Char charAt (int index)`
- `int compareTo (Object o)`
- `int compareTo (String anotherString)`
- `int compareToIgnoreCase (String str)`
- `String concat (String str)`
- `contentEquals boolean (StringBuffer sb)`
- `statis String copyValueOf (char [] data)`
- `statis String copyValueOf (char [] data, int offset, int count)`
- `boolean endsWith (String suffix)`
- `boolean equals (Object anObject)`
- `boolean equalsIgnoreCase (String anotherString):`
- `getBytes byte ()`
- `byte [] getBytes (String charsetName)`
- `int hashCode()`
- `int indexOf (int ch)`
- `int indexOf (int ch, int fromIndex)`
- `int indexOf (String str)`
- `int indexOf (String str, int fromIndex)`
- `String intern ()`
- `int lastIndexOf (int ch)`
- `int lastIndexOf (int ch, int fromIndex)`
- `int lastIndexOf (String str)`
- `int panjang ()`
- `String [] split (String regex)`

- String [] split (String regex, batas int)
- boolean startsWith (String prefix, int toffset)
- String substring (int beginIndex)
- String substring (int beginIndex, int endIndex)
- char di [] toCharArray ()
- String toLowerCase ()
- String toLowerCase (lokal Lokal)
- String toString ()
- String toUpperCase ()
- String toUpperCase (lokal Lokal)
- String lis ()
- statis String valueOf (primitif tipe data x)