

Nama : Wahyu Setyaji Rama Dwijaya

NIM : L200180065

Kelas : B

MODUL 2

Tugas 2.2

The screenshot shows the Visual Studio Code interface with two tabs open: RotiDemo.java and Roti.java. The RotiDemo.java file contains a main method that creates a Roti object and calls its methods. The Roti.java file defines a class with methods for setting color, flavor, weight, and price, and a method to print all details.

```
index.blade.php  berhasil.blade.php  admin.css  RotiDemo.java  Roti.java  ...
E: > Tugas > Java > RotiDemo.java > RotiDemo
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 }
```

```
E: > Tugas > Java > Roti.java > Roti
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 a Windows Command Prompt window titled 'cmd' where the user runs the command 'javac RotiDemo.java'. After compilation, the user runs 'java RotiDemo' which outputs the details of the Roti object.

```
index.blade.php  berhasil.blade.php  admin.css  RotiDemo.java  Roti.java  ...
E: > Tugas > Java > RotiDemo.java > RotiDemo
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 }
```

```
E: > Tugas > Java > Roti.java > Roti
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 }
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\xampp\htdocs\eventback2>

E:
E:\>cd Tugas\Java

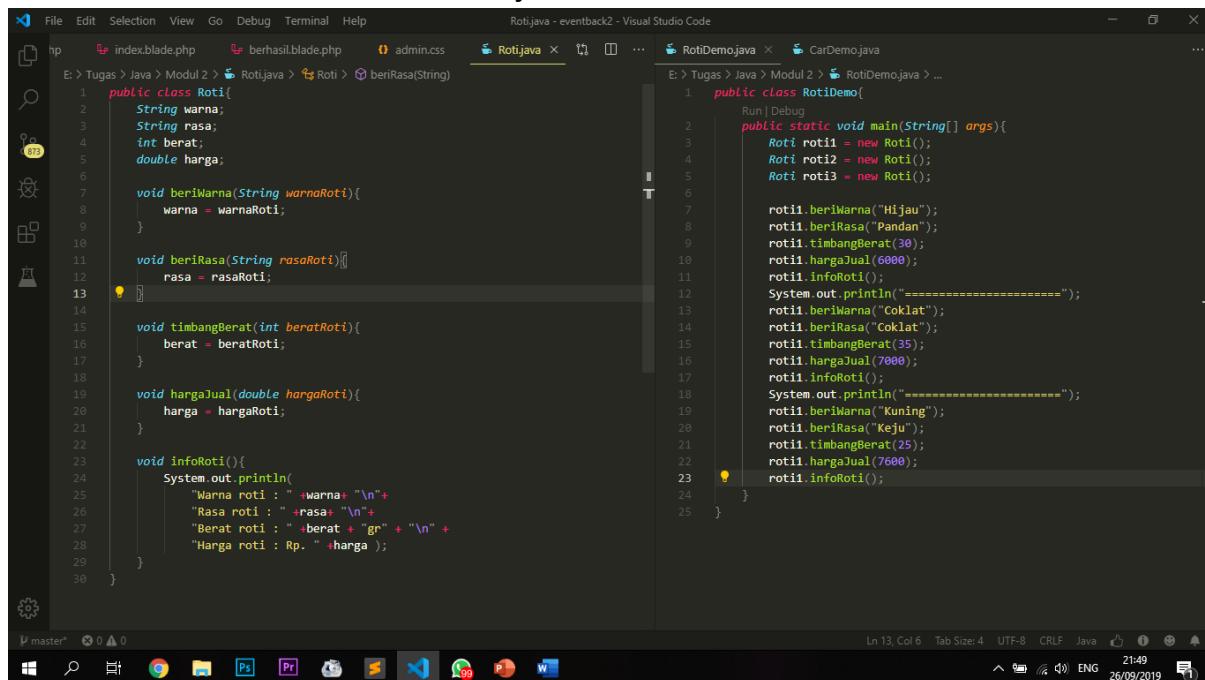
E:\Tugas\Java>javac RotiDemo.java

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

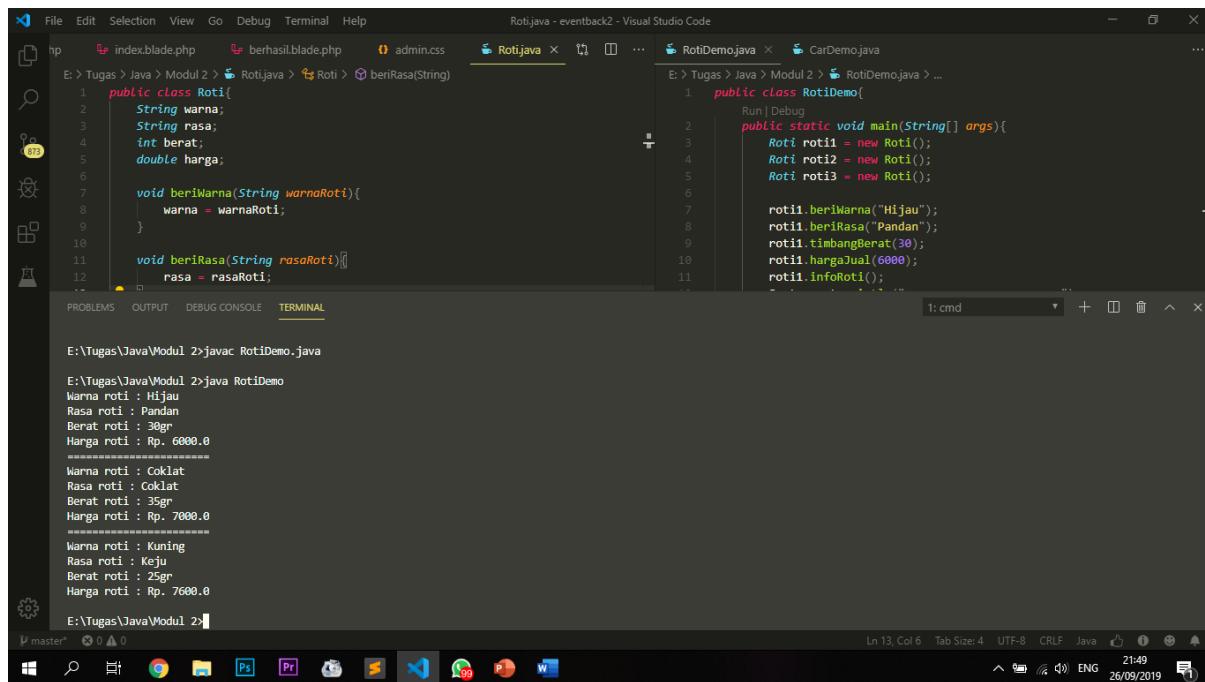
1. Modifikasi Class RotiDemo dan 3 object baru



```
1 public class Roti{
2     String warna;
3     String rasa;
4     int berat;
5     double harga;
6
7     void beriWarna(String warnaRoti){
8         warna = warnaRoti;
9     }
10
11    void beriRasa(String rasaRoti){
12        rasa = rasaRoti;
13    }
14
15    void timbangBerat(int beratRoti){
16        berat = beratRoti;
17    }
18
19    void hargaJual(double hargaRoti){
20        harga = hargaRoti;
21    }
22
23    void infoRoti(){
24        System.out.println(
25            "Warna roti : " +warna+ "\n"+
26            "Rasa roti : " +rasa+ "\n"+
27            "Berat roti : " +berat + "gr" + "\n" +
28            "Harga roti : Rp. " +harga );
29    }
30 }
```

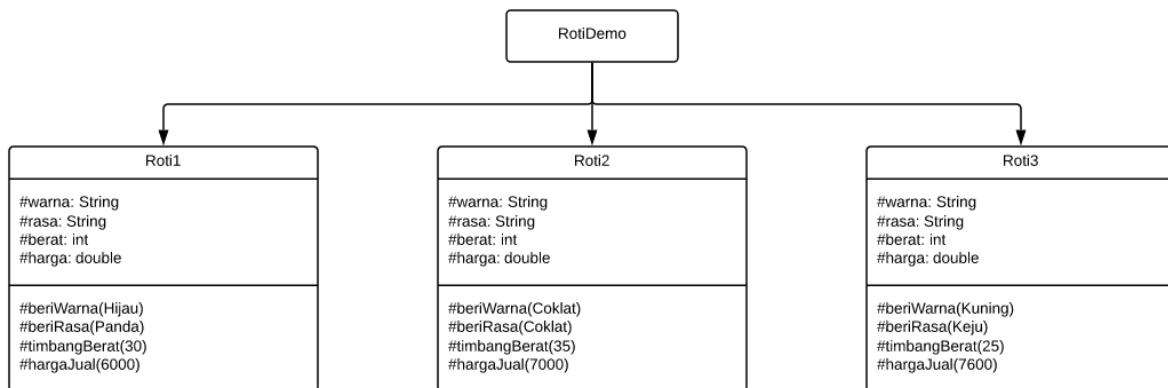
```
1 public class RotiDemo{
2
3     Roti roti1 = new Roti();
4     Roti roti2 = new Roti();
5     Roti roti3 = new Roti();
6
7     roti1.beriWarna("Hijau");
8     roti1.beriRasa("Pandan");
9     roti1.timbangBerat(30);
10    roti1.hargaJual(6000);
11    roti1.infoRoti();
12    System.out.println("=====");
13    roti1.beriWarna("Coklat");
14    roti1.beriRasa("Kuning");
15    roti1.timbangBerat(35);
16    roti1.hargaJual(7000);
17    roti1.infoRoti();
18    System.out.println("=====");
19    roti1.beriWarna("Keju");
20    roti1.timbangBerat(25);
21    roti1.hargaJual(7600);
22    roti1.infoRoti();
23 }
24
25 }
```

Hasil CMD



```
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 : Kuning
Berat roti : 35gr
Harga roti : Rp. 7000.0
=====
Warna roti : Keju
Rasa roti : Keju
Berat roti : 25gr
Harga roti : Rp. 7600.0
```

2. Diagram RotiDemo



3. CarDemo

The screenshot shows a Visual Studio Code interface with multiple tabs open. The active tab is 'CarDemo.java' which contains the following Java code:

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

    void changeCadence(int changeCan){
        cadence = changeCan;
    }

    void speedUp(int upSpeed){
        speed = upSpeed;
    }

    void changeGear(int GearChange){
        gear = GearChange;
    }

    void printInfo(){
        System.out.println(
            "Cadence : "+cadence+"\n"+
            "Speed : "+speed+"\n"+
            "Gear : "+gear );
    }
}

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();

        System.out.println(
            "===="
        );

        car1.changeCadence(30);
        car1.speedUp(10);
        car1.changeGear(1);
        car1.printInfo();
    }
}
```

Hasil CMD:

The screenshot shows a Visual Studio Code interface with several tabs open: web.php, index.blade.php, berhasil.blade.php, Car.java, admin.css, and CarDemo.java. The Car.java and CarDemo.java files are displayed in the main editor area. The Car.java file contains a Car class with methods changeCadence and speedUp. The CarDemo.java file creates two Car objects, calls their methods, and prints the results. Below the editor is a terminal window showing the command-line output of running the Java code.

```
File Edit Selection View Go Debug Terminal Help
CarDemo.java - eventback2 - Visual Studio Code

E: Tugas > Java > Modul 2 > Car.java
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    }
}

E: Tugas > Java > Modul 2 > CarDemo.java
1 public class CarDemo{
2     public static void main(String[] args){
3         Car car1 = new Car();
4         Car car2 = new Car();
5
6         car1.changeCadence(50);
7         car1.speedUp(20);
8         car1.changeGear(2);
9         car1.printInfo();
10
11         System.out.println(
12             "====="
13         );
14     }
15 }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
E:

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

4. Kucing

The screenshot shows a Visual Studio Code interface with the title bar "Kucing.java - eventback2 - Visual Studio Code". The code editor contains the following Java code:

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

The status bar at the bottom right shows "Ln 1, Col 1 Tab Size: 4 UTF-8 CRLF Java 22:18 ENG 26/09/2019".

5. BankDemo

The screenshot shows a Visual Studio Code interface with the title bar "BankDemo.java - eventback2 - Visual Studio Code". The code editor contains two Java files: "Bank.java" and "BankDemo.java".

Bank.java:

```
1 public class Bank{  
2     double saldo = 0;  
3     int no_rekening = 200130065;  
4     String nama = "Rama Sullivan";  
5  
6     void cek_saldo(){  
7         System.out.println(  
8             "Nama : "+nama+"\n"+  
9             "No. Rekening : "+no_rekening+"\n"+  
10            "Saldo : Rp. "+saldo );  
11    }  
12    void menabung(double jumlahNabung){  
13        saldo = saldo+jumlahNabung;  
14        System.out.println(  
15            "Anda menabung sebesar : Rp. "+jumlahNabung+"\n"+  
16            "Saldo anda sekarang : Rp. "+saldo);  
17    }  
18    void menarik(double jumlahNarik){  
19        saldo = saldo-jumlahNarik;  
20        System.out.println(  
21            "Anda menarik sebesar : Rp. "+jumlahNarik+"\n"+  
22            "Saldo anda sekarang : Rp. "+saldo);  
23    }  
24    void transfer(double jumlahTransfer){  
25        saldo = saldo-jumlahTransfer;  
26        System.out.println(  
27            "Anda menarik sebesar : Rp. "+jumlahTransfer+"\n"+  
28            "Saldo anda sekarang : Rp. "+saldo);  
29    }  
30 }
```

BankDemo.java:

```
1 public class BankDemo{  
2     public static void main(String[] args){  
3         Bank bank1 = new Bank();  
4  
5         bank1.cek_saldo();  
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 }
```

The status bar at the bottom right shows "Ln 1, Col 1 Tab Size: 4 UTF-8 CRLF Java 22:19 ENG 26/09/2019".

Hasil CMD:

The screenshot shows the Visual Studio Code interface with two tabs open: 'Bank.java' and 'BankDemo.java'. The 'Bank.java' tab contains a class definition with methods for checking balance and depositing money. The 'BankDemo.java' tab contains the main method for creating an instance of the Bank class and performing a withdrawal. Below the code editor is a terminal window showing the output of the Java command-line application. The terminal output includes the initial account details (Name: Rama Sullivan, Account Number: 200180065, Balance: Rp. 0.0), a deposit of Rp. 100000.0, a withdrawal of Rp. 30000.0, another withdrawal of Rp. 40000.0, and the final balance of Rp. 30000.0.

```
index.blade.php berhasil.blade.php Carjava admin.css Bank.java BankDemo.java

E:\Tugas> Java > Modul 2 > Bank.java
1 public class Bank{
2     double saldo = 0;
3     int no_rekening = 200180065;
4     String nama = "Rama Sullivan";
5
6     void cek_saldo(){
7         System.out.println(
8             "Nama : "+nama+"\n"+
9             "No. Rekening : "+no_rekening+"\n"+
10            "Saldo : Rp. "+saldo );
11    }
12    void menabung(double jumlahNabung){}
13    void menarik(double jumlahNtarik){}

E:\Tugas> Java > Modul 2 > BankDemo.java > BankDemo
1 public static void main(String[] args){
2     Bank bank1 = new Bank();
3
4     bank1.cek_saldo();
5     System.out.println("-----");
6     bank1.menabung(100000);
7     System.out.println("-----");
8     bank1.menarik(30000);
9     System.out.println("-----");
10    bank1.transfer(40000);
11

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
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 compareIgnoreCase (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)