

**PRAKTIKUM PEMROGAMAN BERORIENTASI OBJEK**

**MODUL 2 : CLASS DAN OBJECT**



**Disusun oleh:**

**AFIFAH GH AISANI IMANA**

**L200190198**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS KOMUNIKASI DAN INFORMATIKA**

**UNIVERSITAS MUHAMMADIYAH SURAKARTA**

**TAHUN 2019/2020**

## 2.4 LATIHAN

1. Silakan modifikasi class RotiDemo dan buatlah 3 object baru di dalamnya

```
package package4;

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

    void jenisRoti(String jenisRoti) {
        jenis = jenisRoti;
    }

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

(Kode dari class Roti yang berisi blueprint)

```

package package4;

public class RotiDemo {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        Roti roti = new Roti();
        roti.jenisRoti("Roti Pandan");
        roti.beriWarna("Hijau");
        roti.beriRasa("Pandan");
        roti.timbangBerat(30);
        roti.hargaJual(5000);
        roti.infoRoti();

        //objek tambahan 1
        Roti rotiTawar = new Roti();
        rotiTawar.jenisRoti("Roti Tawar");
        rotiTawar.beriWarna("Putih");
        rotiTawar.beriRasa("Vanilla");
        rotiTawar.timbangBerat(20);
        rotiTawar.hargaJual(2000);
        rotiTawar.infoRoti();

```

```

        //objek tambahan 2
        Roti rotiSelai = new Roti();
        rotiSelai.jenisRoti("Roti Selai");
        rotiSelai.beriWarna("Ungu");
        rotiSelai.beriRasa("Blueberry");
        rotiSelai.timbangBerat(40);
        rotiSelai.hargaJual(7000);
        rotiSelai.infoRoti();

        //objek tambahan 3
        Roti rotiSobek = new Roti();
        rotiSobek.jenisRoti("Roti Sobek");
        rotiSobek.beriWarna("Coklat");
        rotiSobek.beriRasa("Kacang");
        rotiSobek.timbangBerat(30);
        rotiSobek.hargaJual(10000);
        rotiSobek.infoRoti();
    }
}

```

(Kode dari class RotiDemo sebagai main class)

```
Output - package4 (run) ×
run:
Jenis Roti : Roti Pandan
Warna Roti : Hijau
Rasa Roti : Pandan
Berat Roti : 30gr
Harga Roti : Rp.5000.0

Jenis Roti : Roti Tawar
Warna Roti : Putih
Rasa Roti : Vanilla
Berat Roti : 20gr
Harga Roti : Rp.2000.0

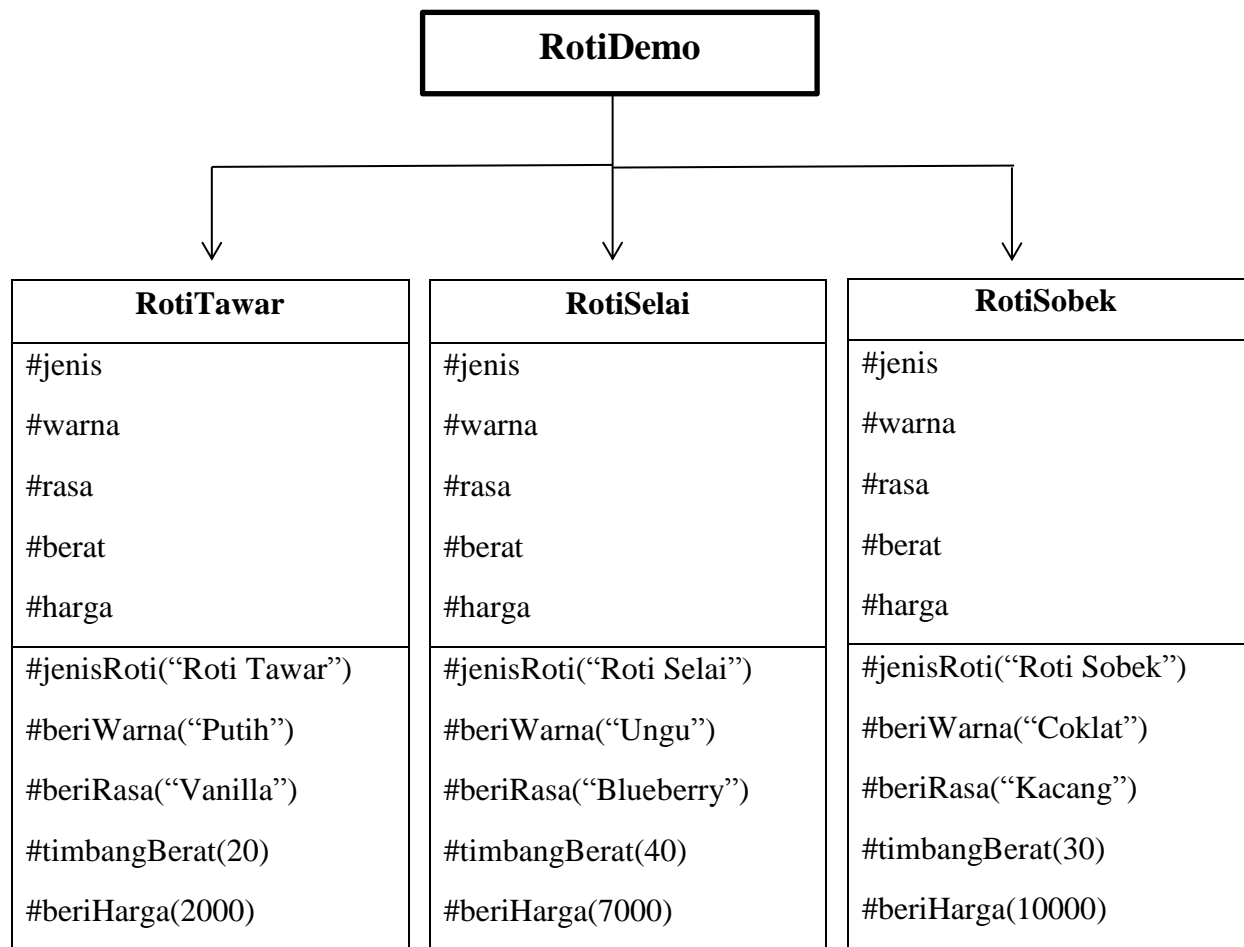
Jenis Roti : Roti Selai
Warna Roti : Ungu
Rasa Roti : Blueberry
Berat Roti : 40gr
Harga Roti : Rp.7000.0

Jenis Roti : Roti Sobek
Warna Roti : Coklat
Rasa Roti : Kacang
Berat Roti : 30gr
Harga Roti : Rp.10000.0

BUILD SUCCESSFUL (total time: 1 second)
```

*(Hasil yang muncul setelah kode di run)*

2. Gambarkan class diagram dari class RotiDemo



3. Buatlah satu class baru yang bisa digunakan sebagai template/blueprint dari class CarDemo seperti terlihat pada Program 3. Class baru tersebut tidak memiliki fungsi main().

```
package package5;

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();
    }
}
```

*(Kode dari class CarDemo sebagai main class)*

```

package package5;
public class Car {
    int cadence;
    int speed;
    int gear;

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

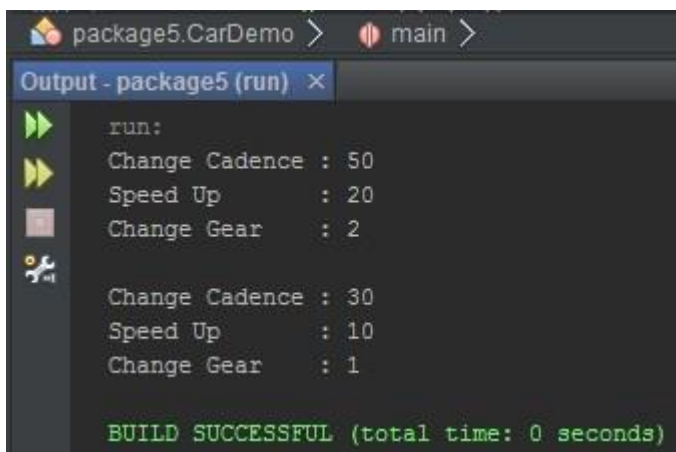
    void speedUp(int speedUp) {
        speed = speedUp;
    }

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

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

```

(Kode dari class Car yang berisi blueprint dari class CarDemo)



```

package5.CarDemo > main >
Output - package5 (run) x
run:
Change Cadence : 50
Speed Up      : 20
Change Gear    : 2

Change Cadence : 30
Speed Up      : 10
Change Gear    : 1

BUILD SUCCESSFUL (total time: 0 seconds)

```

(Hasil yang muncul setelah kode di run)

4. Buatlah suatu class yang dapat merepresentasikan sifat-sifat dari object Kucing. Object ini memiliki field/variable/properties berupa umur, warna bulu dan method berupa meong() dan umur().

```
package kucing;
public class Kucing {
    String buluKucing;
    int umurKucing;

    void meong(String meong) {
        buluKucing = meong;
    }

    void umur(int umur) {
        umurKucing = umur;
    }

    void printInfo() {
        System.out.println(
            "Warna Bulu : " + buluKucing + "\n" +
            "Umur Kucing : " + umurKucing + "\n"
        );
    }
}
```

5. Salah satu aplikasi PBO yang sangat umum adalah berupa aplikasi keuangan. Bank Account (Rekening Bank) adalah salah satu hal yang dapat dijadikan sebagai suatu object di dalam PBO.



```

package bank.account;
public class BankAccount {
    double saldo = 0;
    String no_rekening = "087654123";

    void cek_saldo() {
        System.out.println(
            "No. Rekening          : " + no_rekening + "\n" +
            "Jumlah Saldo Anda Saat Ini Rp.: " + saldo + "\n"
        );
    }

    void menabung(double nabung) {
        saldo = saldo + nabung;
        System.out.println(
            "Anda Menabung Sejumlah      : Rp." + nabung + "\n" +
            "Jumlah Saldo Anda Saat Ini    : Rp." + saldo + "\n"
        );
    }

    void menarik(double tarik) {
        saldo = saldo - tarik;
        System.out.println(
            "Anda Menarik Sejumlah        : Rp." + tarik + "\n" +
            "Jumlah Saldo Anda Saat Ini    : Rp." + saldo + "\n"
        );
    }

    void transfer(double tf) {
        saldo = saldo - tf;
        System.out.println(
            "Anda Mentransfer Sejumlah     : Rp." + tf + "\n" +
            "Jumlah Saldo Anda Saat Ini    : Rp." + saldo + "\n"
        );
    }
}

```

(Kode dari class BankAccount yang berisi blueprint dari class BankAccountDemo)

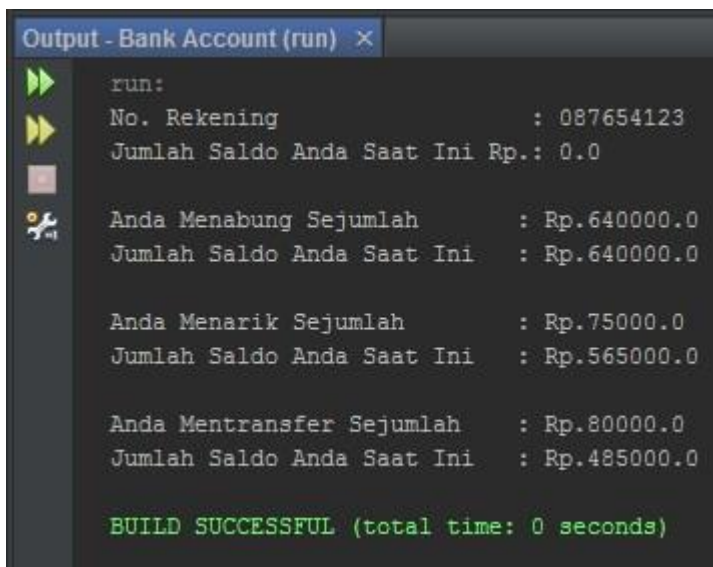
```

package bank.account;
public class BankAccountDemo {
    public static void main(String[] args) {
        BankAccount nasabah1 = new BankAccount();

        nasabah1.cek_saldo();
        nasabah1.menabung(640000);
        nasabah1.menarik(75000);
        nasabah1.transfer(80000);
    }
}

```

*(Kode dari class BankAccountDemo sebagai main class)*



```

Output - Bank Account (run) x
run:
No. Rekening           : 087654123
Jumlah Saldo Anda Saat Ini Rp.: 0.0

Anda Menabung Sejumlah : Rp.640000.0
Jumlah Saldo Anda Saat Ini : Rp.640000.0

Anda Menarik Sejumlah  : Rp.75000.0
Jumlah Saldo Anda Saat Ini : Rp.565000.0

Anda Mentransfer Sejumlah : Rp.80000.0
Jumlah Saldo Anda Saat Ini : Rp.485000.0

BUILD SUCCESSFUL (total time: 0 seconds)

```

*(Hasil yang muncul setelah kode di run)*

6. Perhatikan Class String yang ada di dalam dokumentasi Java. Sebutkan daftar variable dan fungsi/method yang dimiliki oleh Class String tersebut.

- 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)