

PRAKTIKUM PEMROGRAMAN BERBASIS OBJECT

MODUL 2

CLASS DAN OBJECT



Disusun Oleh :

MUHAMMAD WAHYU SYAFI'UDDIN

L200210056

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA**

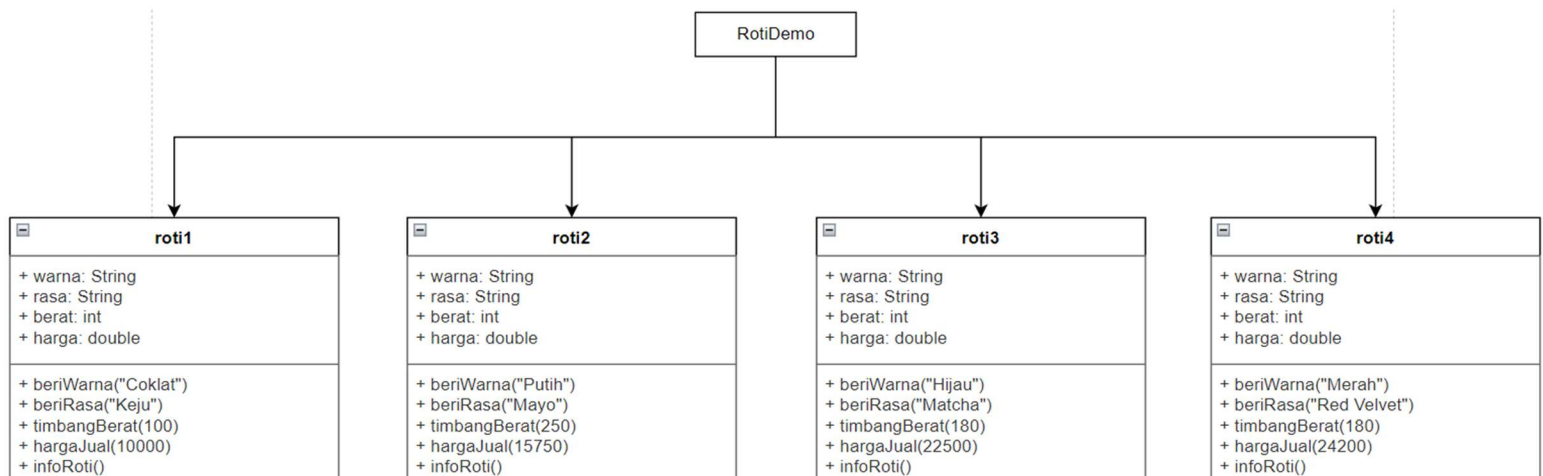
TAHUN 2021/2022

A. Latihan

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

```
KucingDemo.java  RotiDemo.java X  Kucing.java  Roti.java  CarDemo.java
RotiDemo.java > ...
1  public class RotiDemo {
2      Run | Debug
3      public static void main(String[] args) {
4          Roti roti1 = new Roti();
5          roti1.beriWarna(warnaRoti: "Coklat");
6          roti1.beriRasa(rasaRoti: "Keju");
7          roti1.timbangBerat(beratRoti: 100);
8          roti1.hargaJual(hargaRoti: 10000);
9          roti1.infoRoti();
10
11         Roti roti2 = new Roti();
12         roti2.beriWarna(warnaRoti: "Putih");
13         roti2.beriRasa(rasaRoti: "Mayo");
14         roti2.timbangBerat(beratRoti: 250);
15         roti2.hargaJual(hargaRoti: 15750);
16         roti2.infoRoti();
17
18         Roti roti3 = new Roti();
19         roti3.beriWarna(warnaRoti: "Hijau");
20         roti3.beriRasa(rasaRoti: "Matcha");
21         roti3.timbangBerat(beratRoti: 180);
22         roti3.hargaJual(hargaRoti: 22500);
23         roti3.infoRoti();
24
25         Roti roti4 = new Roti();
26         roti4.beriWarna(warnaRoti: "Merah");
27         roti4.beriRasa(rasaRoti: "Red Velvet");
28         roti4.timbangBerat(beratRoti: 180);
29         roti4.hargaJual(hargaRoti: 24200);
30         roti4.infoRoti();
31
32         System.out.println(x: "=====");
33     }
34 }
35 }
```

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().

```
KucingDemo.java  RotiDemo.java  Kucing.java  Roti.java  CarDemo.java  Car.java X
car > Car.java > ...
1  package car;
2
3  public class Car {
4      int cadence;
5      int speed;
6      int gear;
7
8      void changeCadence(int valueCadence){
9          cadence = valueCadence;
10     }
11
12     void changeGear(int valueGear){
13         gear = valueGear;
14     }
15
16     void speedUp(int valueGear){
17         gear = valueGear;
18     }
19
20     void printInfo(){
21         System.out.println("Cadence: " + cadence + "\n" +
22                             "Speed: " + speed + "\n" +
23                             "Gear: " + gear);
24     }
25 }
26
```

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()

```
KucingDemo.java  RotiDemo.java  Kucing.java X  Roti.java  CarDemo.java  Car.java
katz > Kucing.java > ...
2
3  public class Kucing {
4      int umur;
5      String warnaBulu;
6
7
8      void meong(){
9          System.out.println("Meong!!!!!!!!!!!!!!");
10     }
11
12     void umur(int umurKucing){
13         umur = umurKucing;
14     }
15
16     void warnaBulu(String warnaBuluKucing){
17         warnaBulu = warnaBuluKucing;
18     }
19
20     void kucingApa(){
21         System.out.println("Umur Kucing: " + umur + "\n" +
22                             "Warna Bulu Kucing: " + warnaBulu);
23     }
24 }
25
26
```

```
KucingDemo.java X  RotiDemo.java  Kucing.java  Roti
katz > KucingDemo.java > ...
2
3  public class KucingDemo {
4      Run | Debug
5      public static void main(String[] args) {
6          Kucing katz1 = new Kucing();
7
8          katz1.umur(umurKucing: 3);
9          katz1.warnaBulu(warnaBuluKucing: "Hitam");
10         katz1.meong();
11         katz1.kucingApa();
12     }
13
14
```

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

Modifier and Type	Method and Description
Char	<code>charAt(int index)</code> Returns the char value at the specified index.
Int	<code>codePointAt(int index)</code> Returns the character (Unicode code point) at the specified index.
Int	<code>codePointBefore(int index)</code> Returns the character (Unicode code point) before the specified index.
Int	<code>codePointCount(int beginIndex, int endIndex)</code> Returns the number of Unicode code points in the specified text range of this String.
Int	<code>compareTo(String anotherString)</code> Compares two strings lexicographically.
Int	<code>compareToIgnoreCase(String str)</code> Compares two strings lexicographically, ignoring case differences.
String	<code>concat(String str)</code> Concatenates the specified string to the end of this string.
Boolean	<code>contains(CharSequence s)</code> Returns true if and only if this string contains the specified sequence of char values.
Boolean	<code>contentEquals(CharSequence cs)</code> Compares this string to the specified CharSequence.
Boolean	<code>contentEquals(StringBuffer sb)</code> Compares this string to the specified StringBuffer.
static String	<code>copyValueOf(char[] data)</code> Equivalent to <code>valueOf(char[])</code> .
static String	<code>copyValueOf(char[] data, int offset, int count)</code> Equivalent to <code>valueOf(char[], int, int)</code> .
Boolean	<code>endsWith(String suffix)</code> Tests if this string ends with the specified suffix.
Boolean	<code>equals(Object anObject)</code> Compares this string to the specified object.
Boolean	<code>equalsIgnoreCase(String anotherString)</code> Compares this String to another String, ignoring case considerations.
static String	<code>format(Locale l, String format, Object... args)</code> Returns a formatted string using the specified locale, format string, and arguments.
static String	<code>format(String format, Object... args)</code> Returns a formatted string using the specified format string and arguments.
byte[]	<code>getBytes()</code> Encodes this String into a sequence of bytes using the platform's default charset, storing the result into a new byte array.
byte[]	<code>getBytes(Charset charset)</code> Encodes this String into a sequence of bytes using the given charset , storing the result into a new byte array.

B. Pekerjaan Rumah

1. Buatlah class Hewan dan juga ciptakan object dari class tersebut sehingga kita bisa membuat berbagai macam object hewan dengan karakternya masing-masing! Kode program akan menghasilkan output seperti berikut.

```
KucingDemo.java  Hewan.java X  HewanDemo.java  RotiDemo.java
hewan > Hewan.java > ...
1  package hewan;
2
3  public class Hewan {
4      String namaHwn;
5      int jmlKaki;
6      String makanan;
7      String tipeHwn;
8
9      void kasihNama(String namaHewan){
10         namaHwn = namaHewan;
11     }
12
13     void kasihJmlKaki(int jmlKakiHewan){
14         jmlKaki = jmlKakiHewan;
15     }
16
17     void kasihMakanan(String makananHewan){
18         makanan = makananHewan;
19     }
20
21     void kasihTipe(String tipeHewan){
22         tipeHwn = tipeHewan;
23     }
24
25     void hewanApa(){
26         System.out.println("Nama Hewan: " + namaHwn + "\n" +
27             "Jumlah Kaki Hewan: " + jmlKaki + "\n" +
28             "Makanan Hewan: " + makanan + "\n" +
29             "Tipe Hewan: " + tipeHwn + "\n");
30     }
31 }
32
33
```

```
KucingDemo.java  Hewan.java  HewanDemo.java X  Roti
hewan > HewanDemo.java > ...
1  package hewan;
2
3  public class HewanDemo {
4      Run | Debug
5      public static void main(String[] args) {
6          Hewan harimau = new Hewan();
7          Hewan kerbau = new Hewan();
8
9          harimau.kasihNama(namaHewan: "Harimau");
10         harimau.kasihJmlKaki(jmlKakiHewan: 4);
11         harimau.kasihMakanan(makananHewan: "Daging");
12         harimau.kasihTipe(tipeHewan: "Karnivora");
13         harimau.hewanApa();
14
15         kerbau.kasihNama(namaHewan: "Kerbau");
16         kerbau.kasihJmlKaki(jmlKakiHewan: 4);
17         kerbau.kasihMakanan(makananHewan: "Rumput");
18         kerbau.kasihTipe(tipeHewan: "Karnivora");
19         kerbau.hewanApa();
20     }
21 }
22
23
24
25
26
```

2. Buatlah class berdasarkan class diagram pada Gambar 2.4. berikut ini!

```
Dosen.java X  Mahasiswa.java  Karyawan.java
univ > Dosen.java > ...
1  package univ;
2
3  public class Dosen {
4      String nama;
5      int nik;
6      String pendidikan;
7      String tgllahir;
8
9      void tampilkanNama(String namaDosen){
10         nama = namaDosen;
11     }
12
13     void tampilkanTgllahir(String tgllahirDosen){
14         tgllahir = tgllahirDosen;
15     }
16
17     void tampilkanNik(int nikDosen){
18         nik = nikDosen;
19     }
20
21     void tampilkanPendidikan(String pendidikanDosen){
22         pendidikan = pendidikanDosen;
23     }
24
25     void panggilDosen(){
26         System.out.println("Nama Dosen: " + nama + "\n" +
27             "NIK Dosen: " + nik + "\n" +
28             "Pendidikan Dosen: " + pendidikan + "\n" +
29             "Tanggal Lahir Dosen: " + tgllahir + "\n");
30     }
31 }
32
```

```
Dosen.java  Mahasiswa.java X  Karyawan.java
univ > Mahasiswa.java > ...
1  package univ;
2
3  public class Mahasiswa {
4      String nama;
5      String nim;
6      String alamat;
7      int semester;
8
9      void tampilkanNama(String namaMhs){
10         nama = namaMhs;
11     }
12
13     void tampilkanNim(String nimMhs){
14         nim = nimMhs;
15     }
16
17     void tampilkanAlamat(String alamatMhs){
18         alamat = alamatMhs;
19     }
20
21     void tampilkanSemester(int semesterMhs){
22         semester = semesterMhs;
23     }
24
25     void panggilMahasiswa(){
26         System.out.println("Nama Mahasiswa: " + nama + "\n" +
27             "NIM Mahasiswa: " + nim + "\n" +
28             "Alamat Mahasiswa: " + alamat + "\n" +
29             "Semester Mahasiswa: " + semester + "\n");
30     }
31 }
32
```

```
Dosen.java  Mahasiswa.java  Karyawan.java X
univ > Karyawan.java > ...
1  package univ;
2
3  public class Karyawan {
4      String nama;
5      String alamat;
6      String jabatan;
7      double gaji;
8
9      void tampilkanNama(String namaKaryawan){
10         nama = namaKaryawan;
11     }
12
13     void tampilkanJabatan(String jabatanKaryawan){
14         jabatan = jabatanKaryawan;
15     }
16
17     void tampilkanAlamat(String alamatKaryawan){
18         alamat = alamatKaryawan;
19     }
20
21     void tampilkanGaji(double gajiKaryawan){
22         gaji = gajiKaryawan;
23     }
24
25     void panggilKaryawan(){
26         System.out.println("Nama Karyawan: " + nama + "\n" +
27             "Alamat Karyawan: " + alamat + "\n" +
28             "Jabatan Karyawan: " + jabatan + "\n" +
29             "Gaji Karyawan: " + gaji + "\n");
30     }
31 }
32
```