

**LAPORAN PRAKTIKUM  
PEMROGRAMAN BERORIENTASI OBYEK  
MODUL 2**



**DISUSUN OLEH:**

<b>NIM</b>	<b>L200220277</b>
<b>NAMA</b>	<b>MHD. FARHAN LUBIS</b>
<b>KELAS</b>	<b>F</b>

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

**2023**

# DAFTAR ISI

<b>DAFTAR ISI</b> .....	2
<b>LATIHAN</b> .....	4
<b>1. Modifikasi class RotiDemo dan buatlah 3 object baru di dalamnya</b> .....	4
a. Class Roti .....	4
b. Class RotiDemo .....	5
c. Output RotiDemo .....	6
<b>2. Class diagram dari class RotiDemo</b> .....	6
<b>3. Class baru yang bisa digunakan sebagai template/blueprint dari class CarDemo (tanpa fungsi main())</b> .....	7
a. Class Car .....	7
b. Class CarDemo .....	8
c. Output CarDemo .....	9
<b>4. 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()</b> .....	10
a. Class Kucing .....	10
b. Class KucingMain .....	11
c. Output KucingMain .....	11
<b>5. Aplikasi PBO yang umum dapat berupa aplikasi keuangan. Bank Account (Rekening Bank) adalah salah satu hal yang dapat dijadikan sebagai suatu object di dalam PBO</b> .....	12
a. Class yang dapat merepresentasikan Object Rekening tersebut. Variabel dari object ini adalah saldo, no_rekening, nama dan method berupa cek_saldo(), menabung(), menarik(), dan transfer() .....	12
b. Class yang memiliki sebuah fungsi main() yang digunakan untuk mendemonikan pembuatan object tersebut. ....	12
<b>6. Daftar variable dan fungsi/method yang dimiliki oleh Class 'String' dalam dokumentasi Java.</b> 12	
a. Variables dalam Class 'String' .....	12
b. Methods dalam Class 'String' .....	12
<b>TUGAS</b> .....	14
<b>1. Class Hewan dan object dari class tersebut sehingga dapat dibuat berbagai macam hewan dengan karakternya masing-masing.</b> .....	14

a.	<i>Class Hewan</i> .....	14
b.	<i>Class HewanMain</i> .....	14
c.	<i>Output HewanMain</i> .....	15
2.	<b><i>Class berdasarkan class diagram</i></b> .....	16
a.	<i>Class Dosen</i> .....	16
b.	<i>Class Karyawan</i> .....	16
c.	<i>Class Mahasiswa</i> .....	16
d.	<i>Class UniversitasMain</i> .....	16
e.	<i>Output UniversitasMain</i> .....	16
d.	<i>Class Kucing</i> .....	16

## LATIHAN

### 1. Modifikasi *class* RotiDemo dan buatlah 3 *object* baru di dalamnya

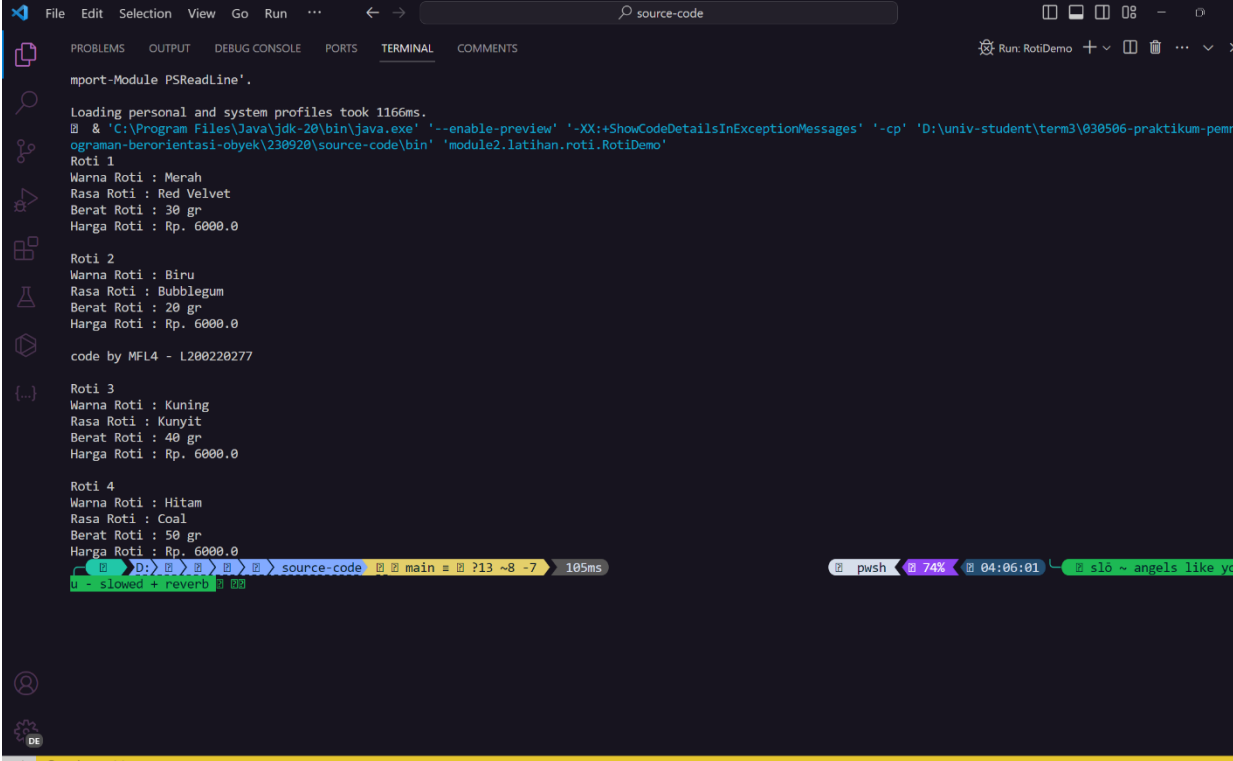
#### a. *Class* Roti

```
1  package module2.latihan.roti;
2
3  public class Roti {
4      String warna;
5      String rasa;
6      int berat;
7      double harga;
8
9      void beriWarna(String warnaRoti) {
10         warna = warnaRoti;
11     }
12
13     void beriRasa(String rasaRoti) {
14         rasa = rasaRoti;
15     }
16     void timbangBerat(int beratRoti) {
17         berat = beratRoti;
18     }
19     void hargaJual(double hargaRoti) {
20         harga = hargaRoti;
21     }
22     void infoRoti() {
23         System.out.println(
24             "Warna Roti : " + warna + "\n" +
25             "Rasa Roti : " + rasa + "\n" +
26             "Berat Roti : " + berat + " gr" + "\n" +
27             "Harga Roti : Rp. " + harga);
28     }
29 }
30
```

b. *Class RotiDemo*

```
1  package module2.latihan.roti;
2
3  public class RotiDemo {
4      public static void main(String[] args) {
5          Roti roti = new Roti();
6          Roti roti2 = new Roti();
7          Roti roti3 = new Roti();
8          Roti roti4 = new Roti();
9
10
11         System.out.println("Roti 1");
12         roti.beriWarna("Merah");
13         roti.beriRasa("Red Velvet");
14         roti.timbangBerat(30);
15         roti.hargaJual(6000);
16         roti.infoRoti();
17         System.out.print("\n");
18
19         System.out.println("Roti 2");
20         roti2.beriWarna("Biru");
21         roti2.beriRasa("Bubblegum");
22         roti2.timbangBerat(20);
23         roti2.hargaJual(6000);
24         roti2.infoRoti();
25         System.out.print("\n");
26         System.out.println("code by MFL4 - L200220277\n");
27
28         System.out.println("Roti 3");
29         roti3.beriWarna("Kuning");
30         roti3.beriRasa("Kunyit");
31         roti3.timbangBerat(40);
32         roti3.hargaJual(6000);
33         roti3.infoRoti();
34         System.out.print("\n");
35
36         System.out.println("Roti 4");
37         roti4.beriWarna("Hitam");
38         roti4.beriRasa("Coal");
39         roti4.timbangBerat(50);
40         roti4.hargaJual(6000);
41         roti4.infoRoti();
42     }
43 }
```

### c. *Output* RotiDemo



```
File Edit Selection View Go Run ... source-code
PROBLEMS OUTPUT DEBUG CONSOLE PORTS TERMINAL COMMENTS
Run: RotiDemo + - - - - -
import Module PSReadLine'.
Loading personal and system profiles took 1166ms.
C:\Program Files\Java\jdk-20\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'D:\univ-student\term3\030506-praktikum-pemrograman-berorientasi-obyek\230920\source-code\bin' 'module2.latihan.roti.RotiDemo'
Roti 1
Warna Roti : Merah
Rasa Roti : Red Velvet
Berat Roti : 30 gr
Harga Roti : Rp. 6000.0
Roti 2
Warna Roti : Biru
Rasa Roti : Bubblegum
Berat Roti : 20 gr
Harga Roti : Rp. 6000.0
code by MFL4 - L200220277
Roti 3
Warna Roti : Kuning
Rasa Roti : Kuningit
Berat Roti : 40 gr
Harga Roti : Rp. 6000.0
Roti 4
Warna Roti : Hitam
Rasa Roti : Coal
Berat Roti : 50 gr
Harga Roti : Rp. 6000.0
D:\> source-code main = ?13 ~8 -7 105ms
pwsh 74% 04:06:01 sl6 ~ angels like yo
u - slowed + reverb
```

## 2. *Class diagram* dari class RotiDemo

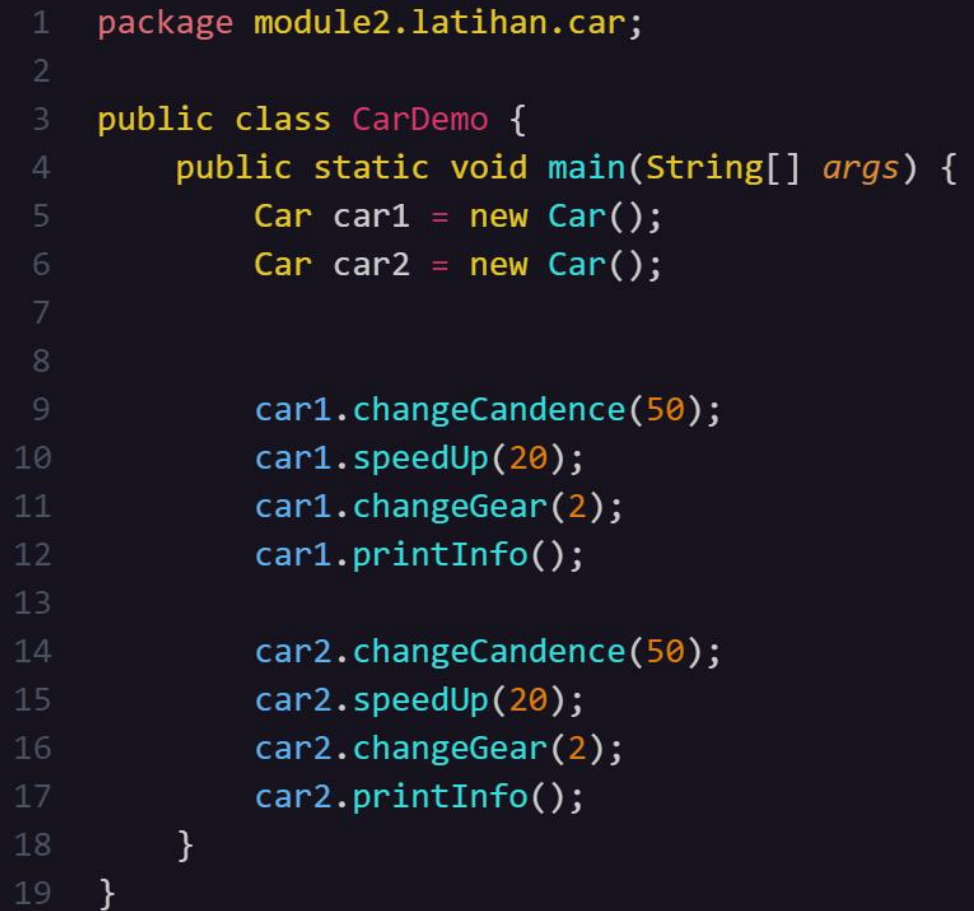


3. *Class* baru yang bisa digunakan sebagai *template/blueprint* dari *class* CarDemo (tanpa fungsi `main()`)

a. *Class* Car

```
1 package module2.latihan.car;
2
3 public class Car {
4     private int speed;
5     private int gear;
6     private int candence;
7
8     public void changeCandence(int candence) {
9         this.candence = candence;
10    }
11
12    public void speedUp(int speed) {
13        this.speed = speed;
14    }
15
16    public void changeGear(int gear) {
17        this.gear = gear;
18    }
19
20    public void printInfo() {
21        System.out.println("Speed: " + speed + " Gear: " + gear + " Candence: " + candence);
22        System.out.println("code by MFL4 - L200220277\n");
23    }
24 }
```

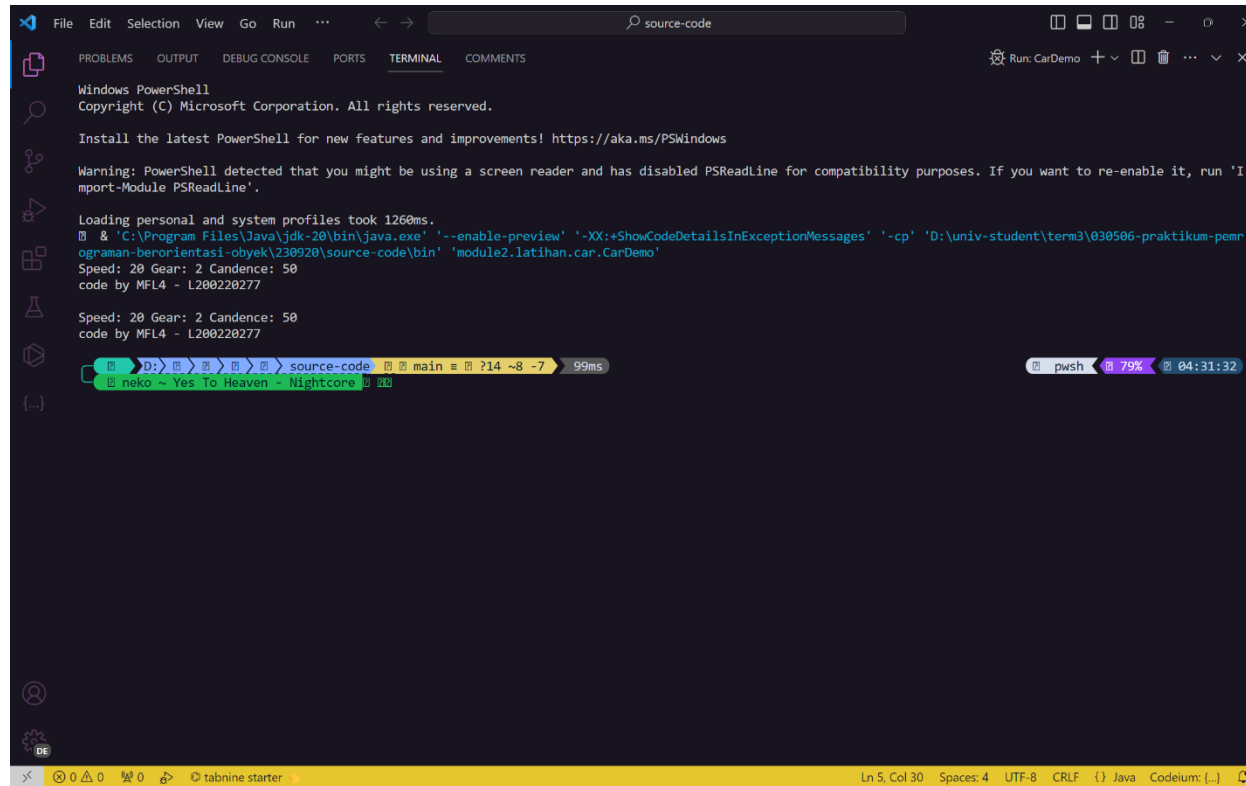
b. *Class* CarDemo



```
1  package module2.latihan.car;
2
3  public class CarDemo {
4      public static void main(String[] args) {
5          Car car1 = new Car();
6          Car car2 = new Car();
7
8
9          car1.changeCandence(50);
10         car1.speedUp(20);
11         car1.changeGear(2);
12         car1.printInfo();
13
14         car2.changeCandence(50);
15         car2.speedUp(20);
16         car2.changeGear(2);
17         car2.printInfo();
18     }
19 }
```



### c. *Output* CarDemo



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

Warning: PowerShell detected that you might be using a screen reader and has disabled PSReadLine for compatibility purposes. If you want to re-enable it, run 'Import-Module PSReadLine'.

Loading personal and system profiles took 1260ms.
PS C:\Program Files\Java\jdk-20\bin> java.exe --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp 'D:\univ-student\term3\030506-praktikum-pemrograman-berorientasi-obyek\230920\source-code\bin' 'module2.latihan.car.CarDemo'
Speed: 20 Gear: 2 Candence: 50
code by MFL4 - L200220277

PS C:\Program Files\Java\jdk-20\bin>
```

4. 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().

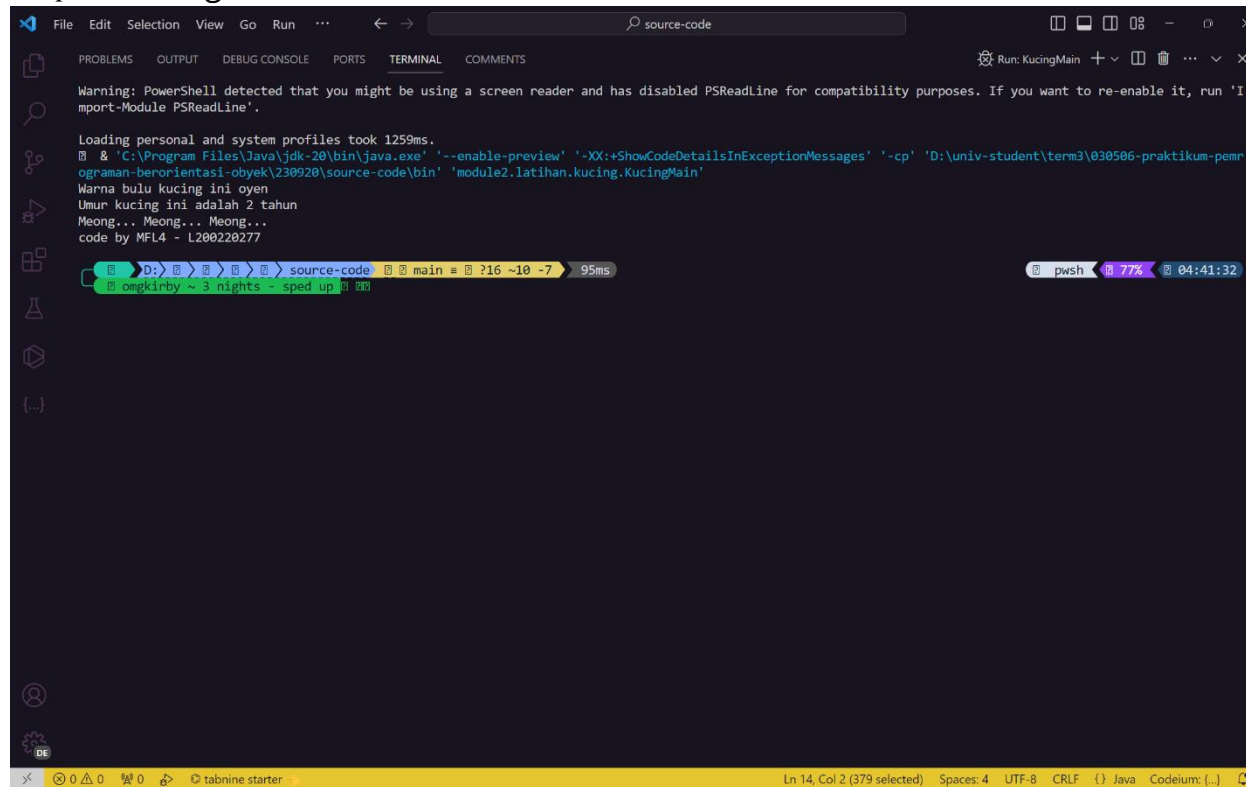
a. Class Kucing

```
1 package module2.latihan.kucing;
2
3 public class Kucing {
4     int umur;
5     String warnaBulu;
6
7     void meong() {
8         System.out.println("Meong... Meong... Meong...");
9     }
10    void umur(int umur) {
11        this.umur = umur;
12        System.out.println("Umur kucing ini adalah " + umur + " tahun");
13    }
14 }
```

## b. Class KucingMain

```
1 package module2.latihan.kucing;
2
3 public class KucingMain {
4     public static void main(String[] args) {
5         Kucing kucing = new Kucing();
6
7         kucing.warnaBulu = "oyen";
8         System.out.println("Warna bulu kucing ini " + kucing.warnaBulu);
9         kucing.umur(2);
10        kucing.meong();
11
12        System.out.println("code by MFL4 - L200220277\n");
13    }
14 }
```

## c. Output KucingMain



```
Warning: PowerShell detected that you might be using a screen reader and has disabled PSReadLine for compatibility purposes. If you want to re-enable it, run 'Import-Module PSReadLine'.

Loading personal and system profiles took 1259ms.
& 'C:\Program Files\Java\jdk-20\bin\java.exe' '-enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'D:\univ-student\term3\030506-praktikum-pemrograman-berorientasi-obyek\230920\source-code\bin' 'module2.latihan.kucing.KucingMain'

Warna bulu kucing ini oyen
Umur kucing ini adalah 2 tahun
Meong... Meong... Meong...
code by MFL4 - L200220277
```

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

- a. *Class* yang dapat merepresentasikan *Object* Rekening tersebut. Variabel dari *object* ini adalah `saldo`, `no_rekening`, `nama` dan *method* berupa `cek_saldo()`, `menabung()`, `menarik()`, dan `transfer()`.
- b. *Class* yang memiliki sebuah fungsi `main()` yang digunakan untuk mendemonstrasikan pembuatan *object* tersebut.

**6. Daftar *variable* dan *fungsi/method* yang dimiliki oleh *Class* ‘*String*’ dalam dokumentasi Java.**

- a. Variables dalam Class ‘String’
  - Variable Utama
    - `private final char value[]`: Menyimpan karakter-karakter dalam string.
- b. Methods dalam Class ‘String’
  - Methods Pengambilan Informasi
    - `public int length()`: Mengembalikan panjang (jumlah karakter) dari string.
    - `public char charAt(int index)`: Mengembalikan karakter pada posisi tertentu dalam string.
    - `public int indexOf(int ch)`: Mengembalikan indeks dari karakter pertama yang cocok dengan karakter yang diberikan.
    - `public int lastIndexOf(int ch)`: Mengembalikan indeks dari karakter terakhir yang cocok dengan karakter yang diberikan.
    - `public boolean isEmpty()`: Memeriksa apakah string kosong (panjangnya 0).
  - Methods Pembandingan dan Pemeriksaan
    - `public boolean equals(Object anObject)`: Memeriksa apakah string sama dengan objek lain.
    - `public boolean equalsIgnoreCase(String anotherString)`: Membandingkan string tanpa memperhatikan perbedaan huruf besar-kecil.
    - `public boolean startsWith(String prefix)`: Memeriksa apakah string dimulai dengan substring yang diberikan.
    - `public boolean endsWith(String suffix)`: Memeriksa apakah string berakhir dengan substring yang diberikan.
    - `public boolean contains(CharSequence sequence)`: Memeriksa apakah string berisi urutan karakter tertentu.
    - `public boolean matches(String regex)`: Memeriksa apakah string cocok dengan ekspresi reguler yang diberikan.
  - Methods Manipulasi String

- `public String concat(String str):` Menggabungkan string dengan string lainnya.
- `public String replace(CharSequence target, CharSequence replacement):` Mengganti semua kemunculan suatu karakter atau substring dengan karakter atau substring lain.
- `public String substring(int beginIndex):` Mengembalikan substring mulai dari indeks tertentu hingga akhir string.
- `public String substring(int beginIndex, int endIndex):` Mengembalikan substring mulai dari indeks awal hingga indeks akhir yang diberikan.
- `public String trim():` Menghapus spasi kosong dari awal dan akhir string.
- `public String[] split(String regex):` Memecah string menjadi array substring berdasarkan ekspresi reguler.
- **Methods Pengubahan Format**
  - `public String toLowerCase():` Mengubah string menjadi huruf kecil.
  - `public String toUpperCase():` Mengubah string menjadi huruf besar.
- **Methods Konversi dan Encoding**
  - `public byte[] getBytes():` Mengonversi string ke dalam bentuk byte menggunakan karakter encoding default.
- **Methods Penggabungan dan Format String**
  - `public static String valueOf(Object obj):` Mengonversi objek menjadi string.
  - `public static String join(CharSequence delimiter, CharSequence... elements):` Menggabungkan array elemen menjadi string dengan delimiter tertentu.

## TUGAS

1. *Class* Hewan dan *object* dari class tersebut sehingga dapat dibuat berbagai macam hewan dengan karakternya masing-masing.

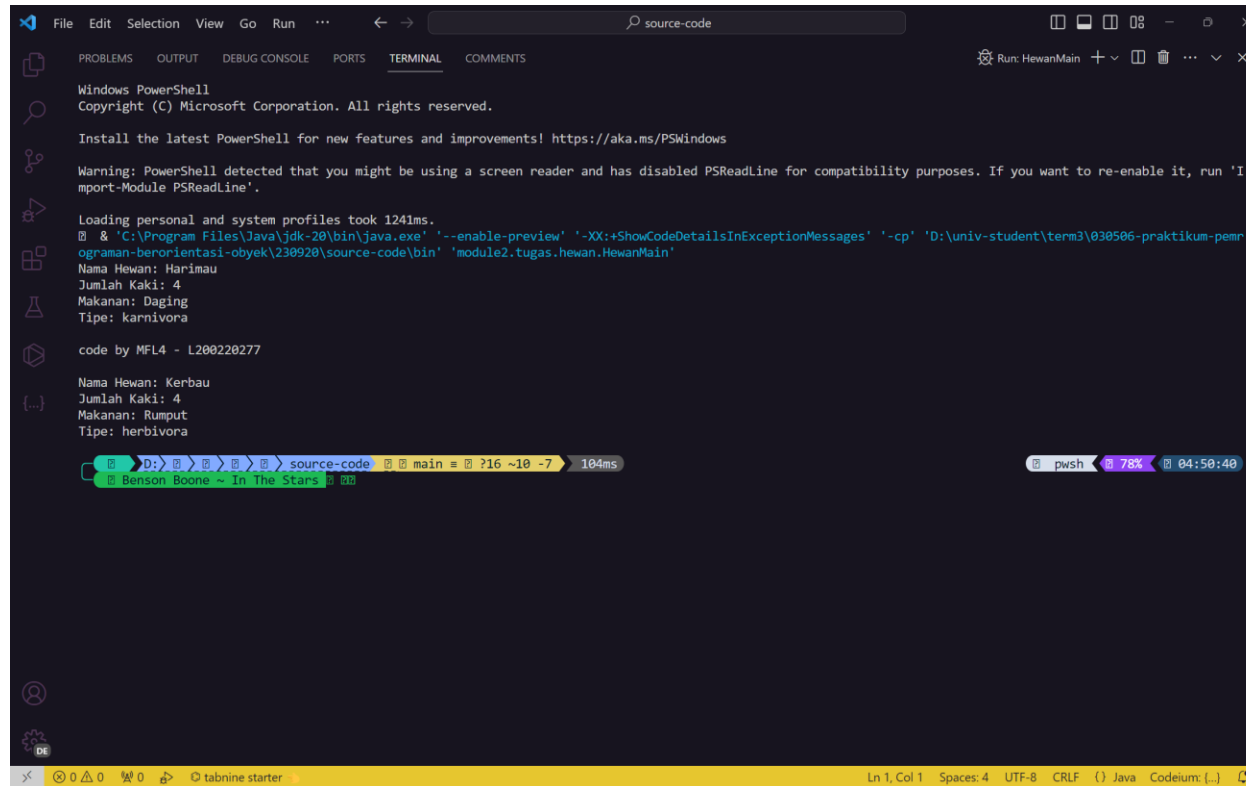
- a. *Class* Hewan

```
1 package module2.tugas.hewan;
2
3 public class Hewan {
4     String nama;
5     int jumlahKaki;
6     String makanan;
7     String tipe;
8
9     void setDataHewan(String nama, int jumlahKaki, String makanan, String tipe) {
10         this.nama = nama;
11         this.jumlahKaki = jumlahKaki;
12         this.makanan = makanan;
13         this.tipe = tipe;
14     }
15     String showDataHewan() {
16         return ("Nama Hewan: " + nama + "\nJumlah Kaki: " + jumlahKaki + "\nMakanan: " + makanan + "\nTipe: " + tipe + "\n");
17     }
18 }
```

- b. *Class* HewanMain

```
1 package module2.tugas.hewan;
2
3 public class HewanMain {
4     public static void main(String[] args) {
5         Hewan harimau = new Hewan();
6         Hewan kerbau = new Hewan();
7         harimau.setDataHewan("Harimau", 4, "Daging", "karnivora");
8         kerbau.setDataHewan("Kerbau", 4, "Rumput", "herbivora");
9
10         System.out.println(harimau.showDataHewan());
11         System.out.println("code by MFL4 - L200220277\n");
12         System.out.println(kerbau.showDataHewan());
13     }
14 }
```

### c. *Output* HewanMain



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

Warning: PowerShell detected that you might be using a screen reader and has disabled PSReadLine for compatibility purposes. If you want to re-enable it, run 'Import-Module PSReadLine'.

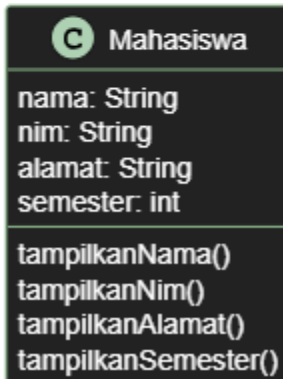
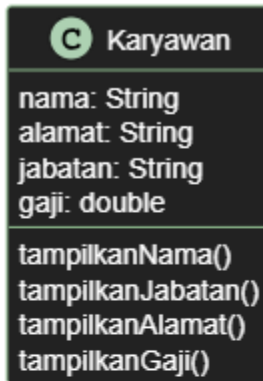
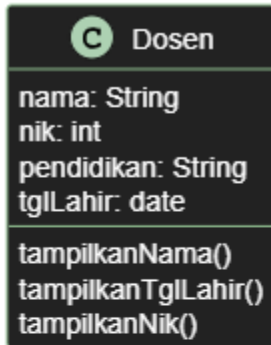
Loading personal and system profiles took 1241ms.
PS C:\Program Files\Java\jdk-20\bin> java.exe --enable-preview -XX:ShowCodeDetailsInExceptionMessages -cp 'D:\univ-student\term3\030506-praktikum-pemrograman-berorientasi-obyek\230920\source-code\bin' 'module2.tugas.hewan.HewanMain'
Nama Hewan: Harimau
Jumlah Kaki: 4
Makanan: Daging
Tipe: karnivora

code by MFL4 - L200220277

Nama Hewan: Kerbau
Jumlah Kaki: 4
Makanan: Rumpit
Tipe: herbivora

D:\univ-student\term3\030506-praktikum-pemrograman-berorientasi-obyek\230920\source-code\bin> java.exe --enable-preview -XX:ShowCodeDetailsInExceptionMessages -cp 'D:\univ-student\term3\030506-praktikum-pemrograman-berorientasi-obyek\230920\source-code\bin' 'module2.tugas.hewan.HewanMain'
```

## 2. Class berdasarkan *class diagram*



- Class* Dosen
- Class* Karyawan
- Class* Mahasiswa
- Class* UniversitasMain
- Output* UniversitasMain