

## LAPORAN PRAKTIKUM PBO MODUL 2

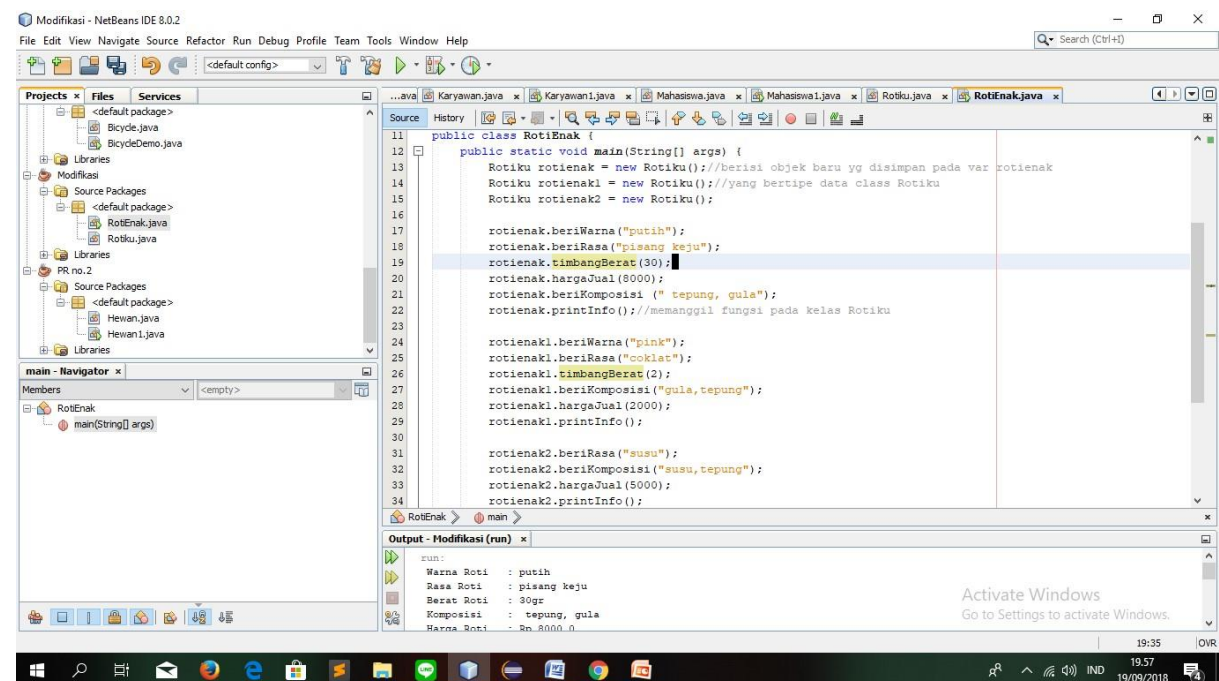
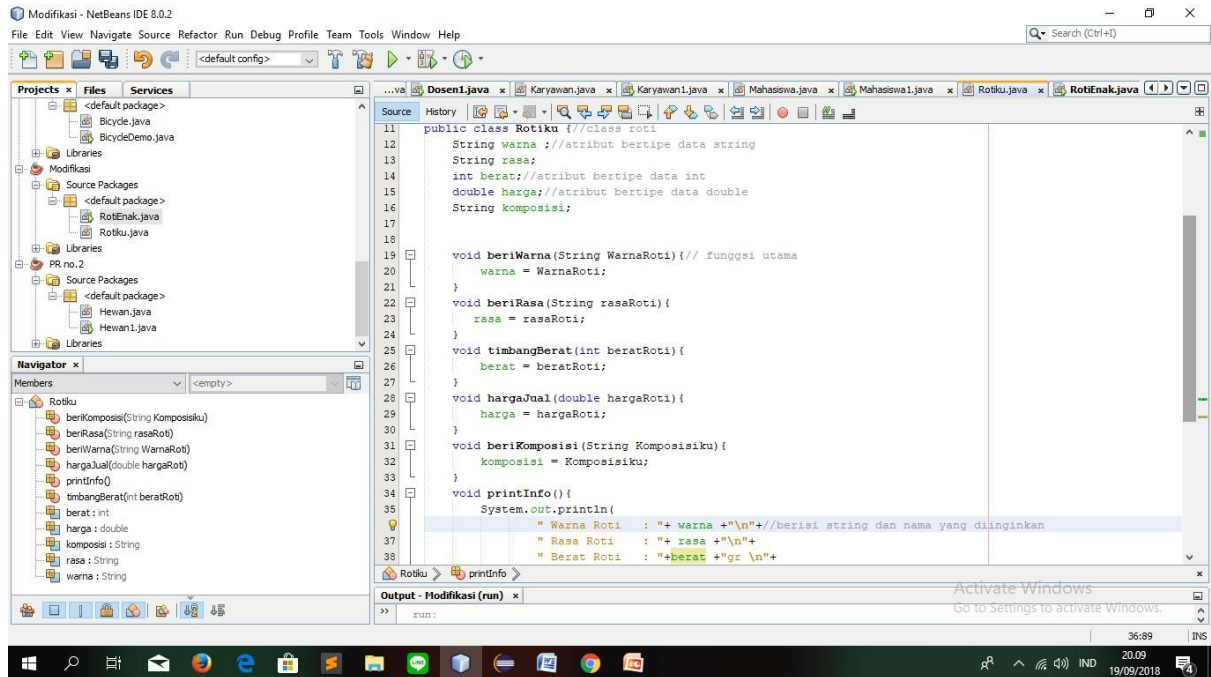
NAMA : Agus Satrio Rukmana

NIM : L200170117

KELAS : C

### LATIHAN

1.



2.

RotiEnak	
Warna	:String
Rasa	:String
Komposisi	:string
Harga	: double
Berat	: int
beriWarna() beriRasa() timbangBearat() hargaJual() beriKomposisi()	

3.

```

11 public class Bicycle { //class bicycle
12     int Cadance; //line 12-14 adalah sebuah atribut
13     int speedUp;
14     int Gear ;
15
16     void changeCadance(int cadancel){
17         Cadance = cadancel;
18     }
19
20     void speedUp(int speedUp1){
21         speedUp = speedUp1;
22     }
23     void changeGear(int gear1){
24         Gear = gear1;
25     } //line 16,20,23,26 merupakan sebuah fungsi utama
26     void printInfo(){
27         System.out.println( //line 27 untuk memanggil perintah di bawah ini
28             " cadance :"+ Cadance +"\n"+
29             " SpeedUp :"+ speedUp +"\n"+
30             " Gear :"+ Gear);
31     } //line 28-30 merupakan string dan atribut guna nya untuk memanggil atribut di atas
32 }

```

Output - latihan No.3 (run)

```

run:
cadance :50
SpeedUp :20
Gear :2
cadance :40
SpeedUp :10
Gear :1

```

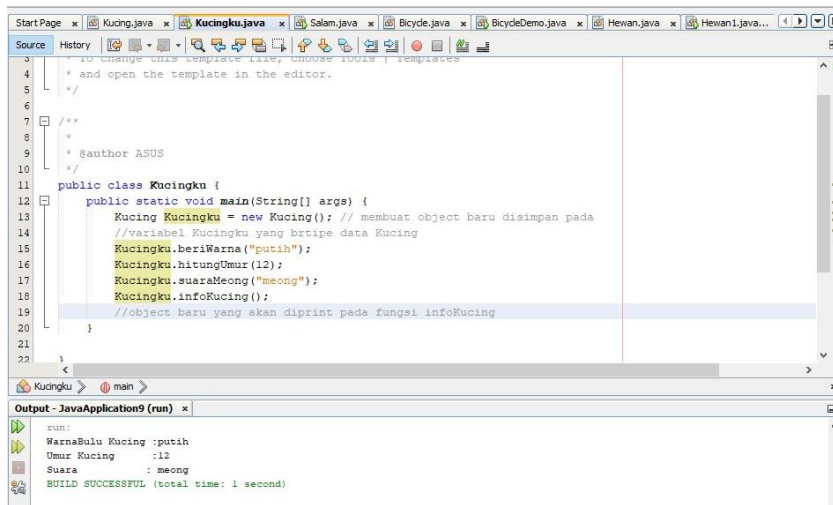
4. class kucing

```

11 public class Kucing { //kelas kucing
12     String WarnaBulu; //line 12,13,14 ini merupakan sebuah atribut
13     int umur; //jika menyangkut masalah angka maka ditambahkan tipe datanya
14     String Suara;
15
16     void beriWarna (String warnaBulu){//berupa fungsi warna
17         WarnaBulu = warnaBulu; //fungsinya untuk memanggil atribut
18     }
19
20     void hitungUmur (int umurKucing){ // berupa fungsi umur
21         umur = umurKucing; //untuk memanggil atribut umur
22     }
23
24     void suaraMeong (String SuaraMeong){ //fungsi meong
25         Suara = SuaraMeong; //memanggil atribut
26     }
27     void infoKucing(){ //berupa fungsi
28         System.out.println( //untuk memanggil perintah di bawah ini
29             "WarnaBulu Kucing :"+WarnaBulu+"\n"+
30             "Umur Kucing :"+umur+"\n"+
31             "Suara : "+Suara);
32     } //line 29,30,31 yaitu berisi string lalu panggil sesuai dg atribut yg
33     //sudah dibuat diatas
34 }
35
36
37

```

- Class kucingKu



The screenshot shows an IDE with several open files. The active file is `Kucingku.java`, which contains the following code:

```

3  // To change this template file, choose Tools | Templates
4  // and open the template in the editor.
5  */
6
7  /**
8   *
9   * @author ASUS
10  */
11  public class Kucingku {
12      public static void main(String[] args) {
13          Kucing Kucingku = new Kucing(); // membuat object baru disimpan pada
14          //variabel Kucingku yang brtipe data Kucing
15          Kucingku.beriWarna("putih");
16          Kucingku.hitungUmur(12);
17          Kucingku.suaraMeong("meong");
18          Kucingku.infoKucing();
19          //object baru yang akan diprint pada fungsi infoKucing
20      }
21  }
22

```

The output window at the bottom shows the following text:

```

run:
WarnaBulu Kucing :putih
Umur Kucing      :12
Suara             : meong
BUILD SUCCESSFUL (total time: 1 second)

```

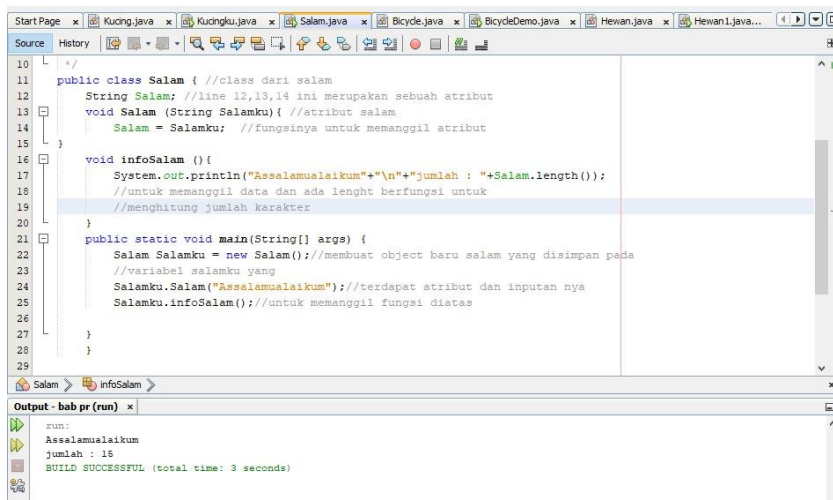
6. Berikut adalah daftar metode yang didukung oleh kelas String:

- 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)

## PEKERJAAN RUMAH :

### 1. class salam



```

10  */
11  public class Salam { //class dari salam
12      String Salam; //line 12,13,14 ini merupakan sebuah atribut
13      void Salam (String Salamku) { //atribut salam
14          Salam = Salamku; //fungsinya untuk memanggil atribut
15      }
16      void infoSalam () {
17          System.out.println("Assalamualaikum"+"\n"+"jumlah : "+Salam.length());
18          //untuk memanggil data dan ada lenght berfungsi untuk
19          //menghitung jumlah karakter
20      }
21      public static void main(String[] args) {
22          Salam Salamku = new Salam(); //membuat object baru salam yang disimpan pada
23          //variabel salamku yang
24          Salamku.Salam("Assalamualaikum"); //terdapat atribut dan inputan nya
25          Salamku.infoSalam(); //untuk memanggil fungsi diatas
26      }
27  }
28  }
29

```

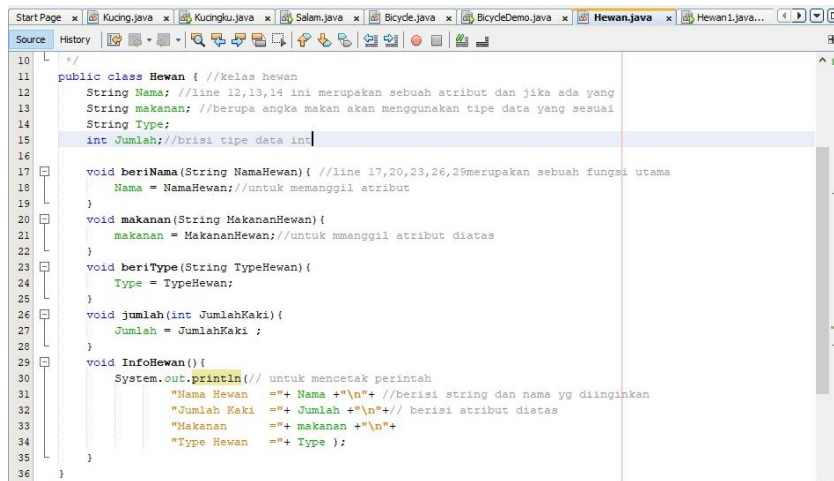
Output - bab pr (run) x

```

RUN:
Assalamualaikum
jumlah : 15
BUILD SUCCESSFUL (total time: 3 seconds)

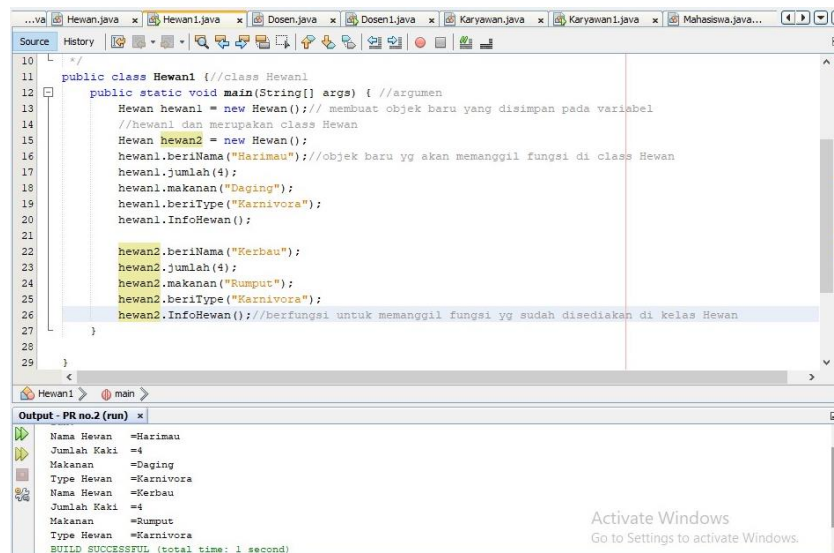
```

## 2.berisi codingan awal class Hewan



```
10  */
11  public class Hewan { //kelas hewan
12      String Nama; //line 12,13,14 ini merupakan sebuah atribut dan jika ada yang
13      String makanan; //berupa angka makan akan menggunakan tipe data yang sesuai
14      String Type;
15      int Jumlah; //brisi tipe data int
16
17      void beriNama(String NamaHewan) { //line 17,20,23,26,29 merupakan sebuah fungsi utama
18          Nama = NamaHewan; //untuk memanggil atribut
19      }
20      void makanan(String MakananHewan) {
21          makanan = MakananHewan; //untuk memanggil atribut diatas
22      }
23      void beriType(String TypeHewan) {
24          Type = TypeHewan;
25      }
26      void jumlah(int JumlahKaki) {
27          Jumlah = JumlahKaki;
28      }
29      void InfoHewan() {
30          System.out.println(// untuk mencetak perintah
31              "Nama Hewan      ="+ Nama +"\n"+ //berisi string dan nama yg diinginkan
32              "Jumlah Kaki    ="+ Jumlah +"\n"+ //berisi atribut diatas
33              "Makanan        ="+ makanan +"\n"+
34              "Type Hewan      ="+ Type );
35      }
36  }
```

- Berisi objek baru dan hasil dari keseluruhan codingan class Hewan1



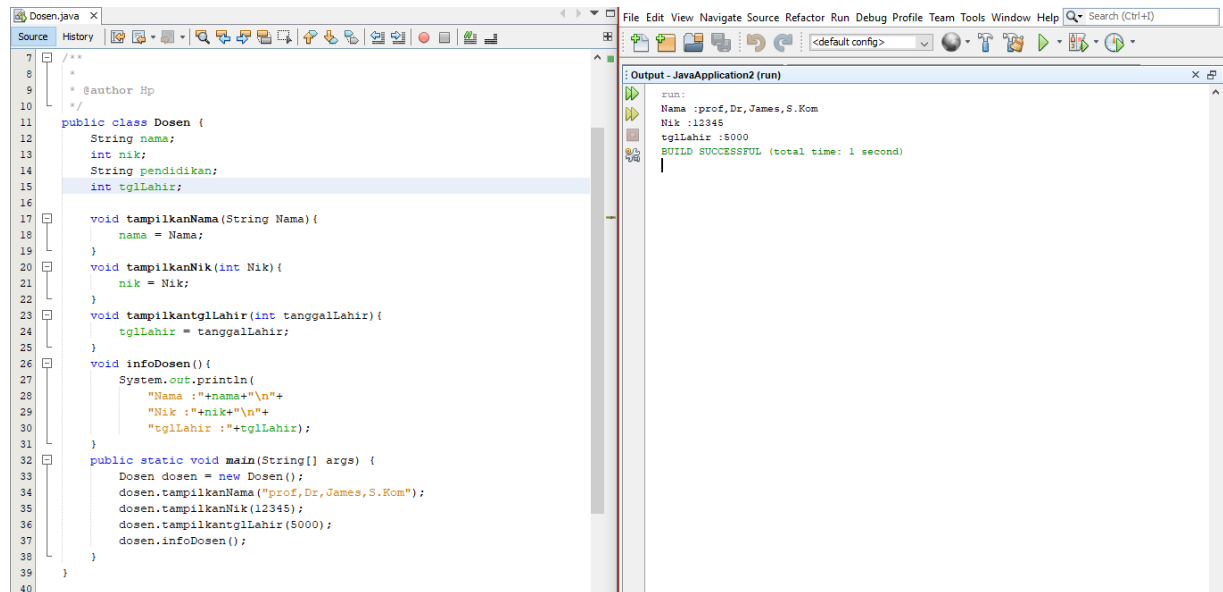
```
10  */
11  public class Hewan1 { //class Hewan1
12      public static void main(String[] args) { //argumen
13          Hewan hewan1 = new Hewan(); // membuat objek baru yang disimpan pada variabel
14          //hewan1 dan merupakan class Hewan
15          Hewan hewan2 = new Hewan();
16          hewan1.beriNama("Harimau"); //objek baru yg akan memanggil fungsi di class Hewan
17          hewan1.jumlah(4);
18          hewan1.makanan("Daging");
19          hewan1.beriType("Karnivora");
20          hewan1.InfoHewan();
21
22          hewan2.beriNama("Kerbau");
23          hewan2.jumlah(4);
24          hewan2.makanan("Rumput");
25          hewan2.beriType("Karnivora");
26          hewan2.InfoHewan(); //berfungsi untuk memanggil fungsi yg sudah disediakan di kelas Hewan
27      }
28  }
29  }
```

Output - PR no.2 (run)

```
Nama Hewan      =Harimau
Jumlah Kaki     =4
Makanan        =Daging
Type Hewan      =Karnivora
Nama Hewan      =Kerbau
Jumlah Kaki     =4
Makanan        =Rumput
Type Hewan      =Karnivora
BUILD SUCCESSFUL (total time: 1 second)
```

3.

## DOSEN

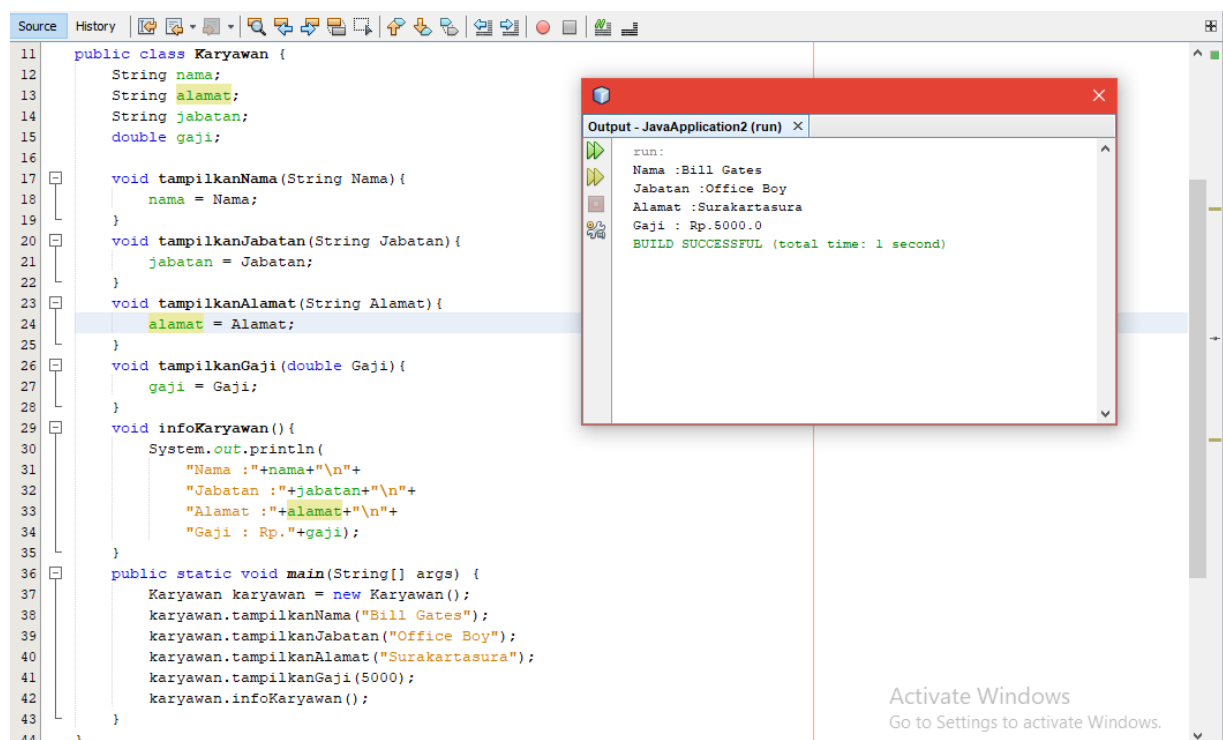


```
7  /**
8   *
9   * @author Hp
10  */
11  public class Dosen {
12      String nama;
13      int nik;
14      String pendidikan;
15      int tglLahir;
16
17      void tampilkanNama(String Nama) {
18          nama = Nama;
19      }
20      void tampilkanNik(int Nik) {
21          nik = Nik;
22      }
23      void tampilkanTglLahir(int tanggalLahir) {
24          tglLahir = tanggalLahir;
25      }
26      void infoDosen() {
27          System.out.println(
28              "Nama :"+nama+"\n"+
29              "Nik :"+nik+"\n"+
30              "tglLahir :"+tglLahir);
31      }
32      public static void main(String[] args) {
33          Dosen dosen = new Dosen();
34          dosen.tampilkanNama("prof,Dr,James,S.Kom");
35          dosen.tampilkanNik(12345);
36          dosen.tampilkanTglLahir(5000);
37          dosen.infoDosen();
38      }
39  }
40
```

Output - JavaApplication2 (run)

```
run:
Nama :prof,Dr,James,S.Kom
Nik :12345
tglLahir :5000
BUILD SUCCESSFUL (total time: 1 second)
```

## KARYAWAN



```
11  public class Karyawan {
12      String nama;
13      String alamat;
14      String jabatan;
15      double gaji;
16
17      void tampilkanNama(String Nama) {
18          nama = Nama;
19      }
20      void tampilkanJabatan(String Jabatan) {
21          jabatan = Jabatan;
22      }
23      void tampilkanAlamat(String Alamat) {
24          alamat = Alamat;
25      }
26      void tampilkanGaji(double Gaji) {
27          gaji = Gaji;
28      }
29      void infoKaryawan() {
30          System.out.println(
31              "Nama :"+nama+"\n"+
32              "Jabatan :"+jabatan+"\n"+
33              "Alamat :"+alamat+"\n"+
34              "Gaji : Rp."+gaji);
35      }
36      public static void main(String[] args) {
37          Karyawan karyawan = new Karyawan();
38          karyawan.tampilkanNama("Bill Gates");
39          karyawan.tampilkanJabatan("Office Boy");
40          karyawan.tampilkanAlamat("Surakartasura");
41          karyawan.tampilkanGaji(5000);
42          karyawan.infoKaryawan();
43      }
44  }
```

Output - JavaApplication2 (run)

```
run:
Nama :Bill Gates
Jabatan :Office Boy
Alamat :Surakartasura
Gaji : Rp.5000.0
BUILD SUCCESSFUL (total time: 1 second)
```

Activate Windows  
Go to Settings to activate Windows.

## MAHASISWA

The screenshot shows an IDE window with a Java source file and an output console. The source file defines a `Mahasiswa` class with attributes `nama`, `nim`, `alamat`, and `semester`. It includes setter methods `tampilkanNama`, `tampilkanNim`, `tampilkanAlamat`, and `tampilkanSemester`, along with an `infoMahasiswa` method that prints the student's details. The `main` method creates a `Mahasiswa` object and calls these methods with specific values. The output console shows the results of the `run` operation, displaying the student's information and a success message.

```
11 public class Mahasiswa {
12     String nama;
13     String nim;
14     String alamat;
15     int semester;
16
17     void tampilkanNama(String Nama) {
18         nama = Nama;
19     }
20     void tampilkanNim(String Nim) {
21         nim = Nim;
22     }
23     void tampilkanAlamat(String Alamat) {
24         alamat = Alamat;
25     }
26     void tampilkanSemester(int Semester) {
27         semester = Semester;
28     }
29     void infoMahasiswa() {
30         System.out.println(
31             "Nama : "+nama+"\n"+
32             "Nim : "+nim+"\n"+
33             "Alamat : "+alamat+"\n"+
34             "Semester : "+semester);
35     }
36     public static void main(String[] args) {
37         Mahasiswa mahasiswa = new Mahasiswa();
38         mahasiswa.tampilkanNama("Bill Gates");
39         mahasiswa.tampilkanNim("Office Boy");
40         mahasiswa.tampilkanAlamat("Surakartasura");
41         mahasiswa.tampilkanSemester(3);
42         mahasiswa.infoMahasiswa();
43     }
44 }
```

Output - JavaApplication2 (run) x

```
run:
Nama :Bill Gates
Nim :Office Boy
Alamat :Surakartasura
Semester :3
BUILD SUCCESSFUL (total time: 1 second)
```

Activate Windows  
Go to Settings to activate Windows.