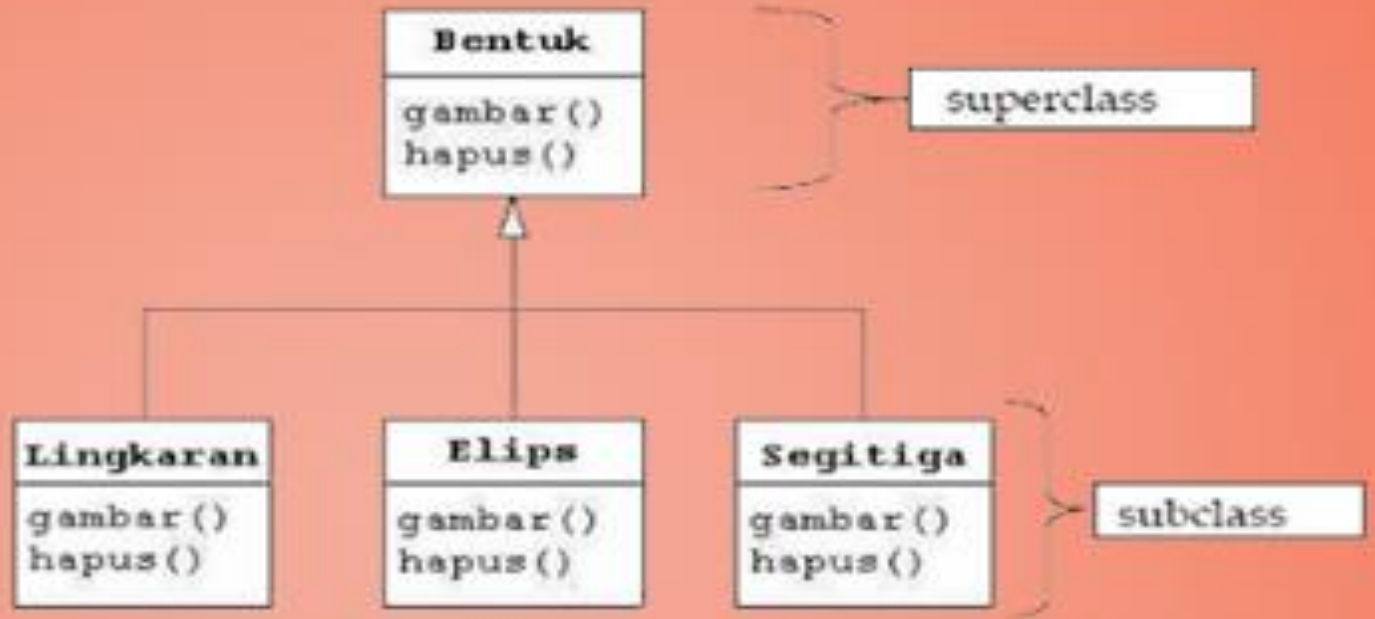


Menerapkan Instruksi Dasar Class dan Operator

INSTRUKSI DASAR CLASS

```
package siswa;

// Author class
 class Siswa {
String nama= "Zahara Ika ";
    public static void main (String[] args)
    {
      Siswa nm = new Siswa();
      System.out.print("nama siswa adalah:" +nm.nama);
    }

}
```

Nama kelas

Variable instance

Mendefinisikan objek dalam kelas(instansiasi)

INSTRUKSI DASAR CLASS

- Instruksi dasar Class

program variable instance dan kelas

Analisa Program

- Variable instance dideklarasikan setelah dekalarasi kelas , tetapi sebelum deklarasi metode. Pada program , setiap objek berasal dari kelas Kucing memiliki variable Nama, Warna, Berat
- Variable Nama dan Warna bertipe data string dan variable Berat interger.
- Membuat objek baru dengan variable instance kucingku
- Operator titik (.) untuk mengakses (instance variable dan method). Kueingku.Nama="cepot"

VARIABEL INSTANT DAN STATIC

- Variabel static menggunakan keyword static
- Variabel static adalah variable yang sama pada semua objek di class tersebut.Perubahan nilai pada variable static akan berpengaruh pada objek lain.

- Jika variabel tidak menggunakan keyword static disebut variable instant
- Perubahan nilai variable instant di satu objek tidak berpengaruh pada variable instant di objek yang berbeda

METHOD VOID DAN NON VOID

- **Method void tidak mengembalikan nilai, sedangkan method non void mengembalikan nilai**

METHOD VOID DAN NON VOID

Penambahan method void pada deklarasi kelas Karyawan

```
public class Karyawan {
   String ID, nama, divisi;
   Double gaji;
   void cetakData() {
      System.out.println("Data Karyawan ");
      System.out.println("ID     : " + ID);
      System.out.println("Nama   : " + nama);
      System.out.println("Divisi : " + divisi);
      System.out.println("Gaji   : " + gaji);
   }}
```

Contoh pemanggilan method void

```
public class ImplementasiMethodVoid {
   public static void main(String[] args) {
      //instansiasi objek karyawan
      Karyawan Karyawan001 = new
Karyawaan();
      //mengisi data pada objeck karyawan
      Karyawan001.ID = "K001";
      Karyawan001.nama = "Agus Ramadhan";
      Karyawan001.divisi = "Keuangan";
      Karyawan001.gaji = 1850000;
      //memanggil method cetakData();
      Karyawan001.cetakData();
   }
}
```

Method void tidak mengembalikan nilai, sedangkan method non void mengembalikan nilai

CONSTRUCTOR

- Constructor adalah method khusus yang didefinisikan di dalam Class dan akan dipanggil secara otomatis tiap kali terjadi instansiasi objek. Constructor itu sendiri berfungsi untuk melakukan inisialisasi nilai terhadap data-data yang terdapat pada Class yang bersangkutan. Jika kita tidak mendefinisikan constructor pada Class yang kita buat, maka secara otomatis Java akan membuatkannya untuk kita. Constructor semacam ini dinamakan dengan default constructor

CONTOH CONSTRUCTOR

Deklarasi kelas Karyawan dengan constructor

```
public class Karyawan {
     String ID, nama, divisi;
     double gaji;

     //constructor kelas karyawan
     Karyawan() {
          ID = "k001";
          nama = "Budi";
          divisi = "Produksi";
          gaji = "1750000";
     }

     void cetakData() {
          System.out.println("Data
Karyawan   :");
          System.out.println("ID
: " + ID);
          System.out.println("Nama
: " + nama);

System.out.println("Divisi : " +
divisi);
          System.out.println("Gaji
: " + gaji);
     }

     double hitungSumbanganZakat()
{
          double zakat = gaji *
0.025;
          return zakat;
     }
}
```

Deklarasi kelas Karyawan dengan constructor dinamis

```
public class Karyawan {

     String ID, nama, divisi;
     double gaji;

//constructor kelas Karyawan
     Karyawan(String kode, String
Nama, String Div, double Gaji) {
          ID = kode;
          nama = Nama;
          divisi = Div;
          gaji = Gaji;
     }

     void cetakData() {
          System.out.println("Data
Karyawan   :");
          System.out.println("ID
: " + ID);
          System.out.println("Nama
: " + nama);

System.out.println("Divisi : " +
divisi);
          System.out.println("Gaji
: " + gaji);
     }

     double hitungSumbanganZakat()
{
          double zakat = gaji *
0.025;
          return zakat;
     }
                    }
```

KONVERSI TIPE DATA

- Tipe data : String, Integer, Float, Double dsb
- Jika menggunakan komponen textfield, label, textarea maka semua data yang menggunakan komponen tersebut akan menjadi tipe data String
- Perlu dilakukan konversi tipe data dari string ke double, integer, atau tipe data numerik lain.

KONVERSI TIPE DATA


```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    Karyawan satu=new Karyawan();
     Double gaji,takat1;
    gaji= Double.parseDouble(jTextField3.getText());
    takat1=satu.hitungzakat();

     // TODO add your handling code here:
     jLabel3.setText("Id Karyawan:  " + satu.id + "Nama Karyawan:  " + satu.nama
    + "Gaji:  " + gaji.toString());
    jLabel4.setText("Zakat :  " + String.valueOf(zakat1));
}
```

PENGGUNAAN OPERATOR

Operator di Java

- Operator Unary

Arti Operator	Operator	Contoh Pemakaian
Pre-Increment	++operand	int i = 8 ; int j = ++i; i bernilai 9, j bernilai 9

Post-Increment	operand++	int i = 8;
		int j = i++;
		i bernilai 9, j bernilai 8

Pre-Decrement	--operand	int i = 8 ; int j = --i; i bernilai 7 , j bernilai 7
Post-Increment	operand--	int i=8; int j = i--; i bernilai 7, j bernilai 8

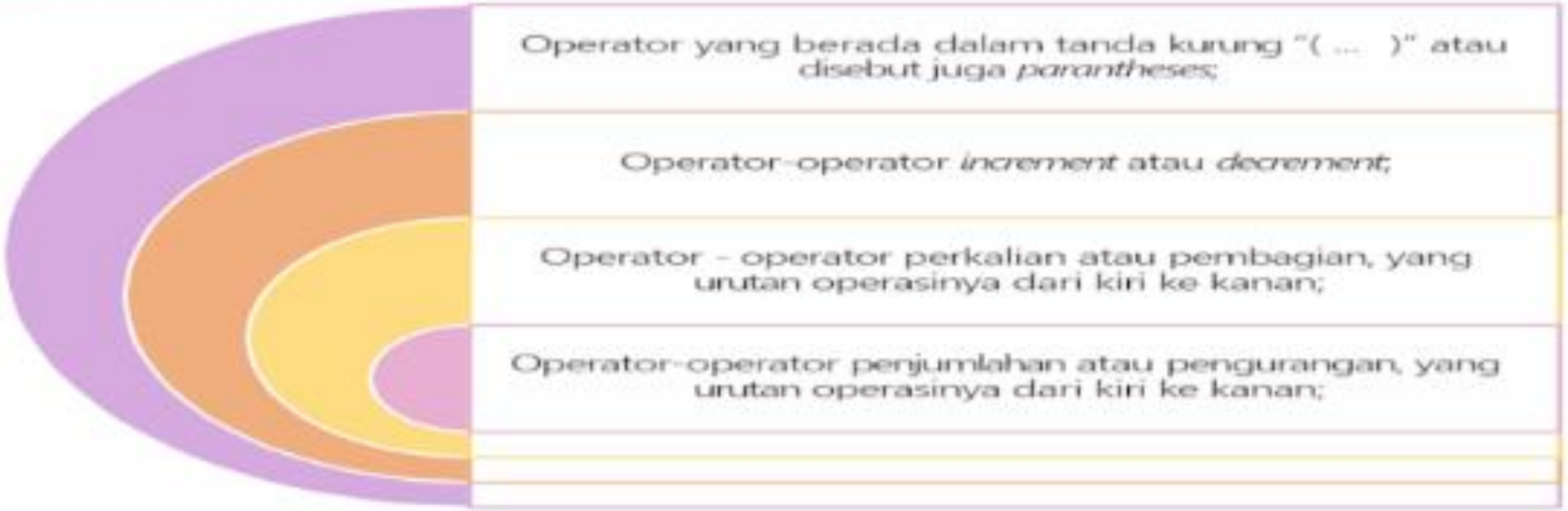
PENGGUNAAN OPERATOR

- Operator Binary

Arti Operator	Operator	Contoh Pemakaian	Keterangan
Penjumlahan	+	sum=num1 + num2	
Pengurangan	-	diff=num1 - num2	
Perkalian	*	prod=num1 * num2	
Pembagian	/	quot=num1 / num2	jika num1 dan num2 adalah integer, pembagian akan menghasilkan nilai integer tanpa mengikutsertakan sisa, jika terdapat sisa.
Sisa (modulus)	%	mod=num1 % num2	Hasil operasi modulus adalah sisa dari operasi num1 / num2 Hasil operasi modulus memiliki tanda (+/-) yang sama dengan operand pertama

PENGGUNAAN OPERATOR

Prioritas Operator



PENGGUNAAN OPERATOR

Prioritas Operator

- Contoh

```
int c = 12 * 3 +5 / (7 - 2) ;
```

Maka urutan operasinya adalah sebagai berikut :

```
int c = 12 * 3 + 5 / 5 ;
int c = 36 + 5 / 5;
int c = 36 + 1 ;
int c = 37;
```

OPERATOR LOGIKA

- Operator logika memiliki satu atau lebih operand boolean yang menghasilkan nilai boolean. Operator Logika diantaranya: && (logika AND), || (logika OR), | (boolean logika inclusive OR), ^ (boolean logika exclusive OR), dan ! (logika NOT).