

# TEORI DAN KONSEP PEMROGRAMAN BERBASIS OBJECT

## PERTEMUAN 2

# Penerapan Karakteristik Pemrograman Berorientasi Objek

## a. Enkapsulasi (Encapsulation)

- Karakteristik ini merupakan suatu cara bagaimana menyembunyikan sedemikian rupa suatu proses kedalam sistem, hal ini berguna untuk menghindari interferensi dari luar sistem dan juga lebih untuk menyederhakan sistem itu sendiri.

# Contoh Script Enkapsulasi

Contoh Sederhana

1. Buatlah sebuah Class dengan nama Belajar.Java

```
class Belajar
{
    public String x = "Pintar";
    private String y = "Java";
}
```

2. Buatlah sebuah Class dengan nama Pintar.java

```
public class Pintar{
    public static void main(String[] args){
        Belajar panggil = new Belajar();
        System.out.println("Panggil X : "+panggil.x);
        System.out.println("Panggil Y : "+panggil.y);
    }
}
```

Penjelasan :

Fungsi **public** yang terdapat dalam class Belajar pada variable x, memungkinkan nilai dari variable x dapat diakses oleh class Pintar. Sedangkan variable y yang menggunakan fungsi **private** tidak dapat dipanggil didalam class Pintar.

# Penerapan Karakteristik Pemrograman Berorientasi Objek(lanjutan)

## b. Pewarisan (Inheritance)

Inheritance (pewarisan/penurunan) merupakan penurunan sifat (atribut dan metod) dari induk class kepada subclass atau anak class.

Untuk mendeklarasikan suatu class sebagai subclass atau class turunan dari suatu class dilakukan dengan menambah kata kunci “ extends”

# Contoh Script Pewarisan

Contoh Penerapan Pewarisan pada program

1. buatlah Class dengan nama sepeda.java

```
5 package pewarisan;
6
7 /**
8  *
9  * @author rony
10 */
11
12 @ class Sepeda{
13     int kecepatan = 0;
14     int gir = 0;
15     // method
16     void ubahGir(int pertambahanGir) {
17         gir= gir+ pertambahanGir;
18         System.out.println("Gir:" + gir);
19     }
20     void tambahKecepatan(int pertambahanKecepatan) {
21         kecepatan = kecepatan+ pertambahanKecepatan;
22         System.out.println("Kecepatan:" + kecepatan);
23     }
24 }
25
```

2. buatlah Class dengan nama Sepeda Gunung.java

```
5 package pewarisan;
6
7 /**
8  *
9  * @author rony
10 */
11
12 class SepedaGunung extends Sepeda {
13     int setSadel;
14     public void setSadel(int nilaiSadel) {
15         setSadel = nilaiSadel;
16     }
17     System.out.println("Tinggi Sadel:"+setSadel);
18 }
19
20
21
```

# Contoh Script Pewarisan(lanjutan)

3. buatlah Class dengan nama Sepeda GunungBeraksi.java

```
5  package pewarisan;
6
7  /**
8   *
9   * @author rony
10  */
11  public class SepedaGunungBeraksi {
12      public static void main(String[] args) {
13          // Membuat object
14          SepedaGunung sepedaku = new SepedaGunung();
15          // Memanggil method di object
16          sepedaku.tambahKecepatan(10);
17          sepedaku.ubahGir(2);
18          sepedaku.setSadel(20);
19      }
20  }
```

# **Penerapan Karakteristik Pemrograman Berorientasi Objek(lanjutan)**

## **c. Polymorphism**

**Polimorfisme merupakan Suatu kemampuan dari sebuah object untuk membolehkan mengambil beberapa bentuk yang berbeda agar tidak terjadi duplikasi object (kemampuan untuk mempunyai beberapa bentuk class yang berbeda). Polimorfisme ini terjadi pada saat suatu object bertipe parent class, akan tetai pemanggilan constructornya melalui subclass.**

# Contoh Script Polymorphism

1. Buatlah Sebuah Class dengan Nama Polimorfisme.java

```
class hewan {
    public void suara () {
        System.out.println("Hewan bersuara :");
    }
}
class kuda extends hewan {
    public void suara() {
        System.out.println("Kuda mengikik...");
    }
}
class kucing extends hewan {
    public void suara () {
        System.out.println("Kucing mengeong...");
    }
}
class ayam extends hewan {
    public void suara() {
        System.out.println("Ayam berkokok...");
    }
}
public class Polimorfisme {
    public static void main(String[] args) {
        hewan HEWAN = new hewan();
        kuda KUDA = new kuda();
        kucing KUCING = new kucing();
        ayam AYAM = new ayam();

        HEWAN.suara();

        HEWAN = KUDA;
        HEWAN.suara();

        HEWAN = KUCING;
        HEWAN.suara();

        HEWAN = AYAM;
        HEWAN.suara();
    }
}
```



# OPERATOR PADA BAHASA JAVA

## a.Operator Aritmatika

**Operator adalah simbol atau karakter yang biasa dilibatkan dalam program untuk melakukan sesuatu operasi atau manipulasi, seperti penjumlahan, pengurangan dan lain-lain.**

Operator	Keterangan	Contoh (x1=4, x2=3)
+	Penjumlahan	$x1 + x2 = 7$
-	Pengurangan	$x1 - x2 = 1$
*	Perkalian	$x1 * x2 = 12$
/	Pembagian	$x1 / x2 = 1,33$
%	Modulus atau sisa pembagian	$x1 \% x2 = 1$

Berikut contoh penggunaan operator aritmatika, bisa anda ikuti seperti dibawah ini.

```
1  /**
2   *
3   * @author rony
4   */
5  public class OperatorAritmatika {
6
7      public static void main ( String [] args ) {
8          // Deklarasi dan inisialisasi variabel
9          int i = 9 ;
10         int j = 4 ;
11         // Menampilkan nilai variabel
12         System.out.println ( " Nilai variabel : " ) ;
13         System.out.println ( " i = " + i ) ;
14         System.out.println ( " j = " + j ) ;
15         // Menggunakan operator penjumlahan
16         System.out.println ( " Penjumlahan : " ) ;
17         System.out.println ( " i + j = " + ( i + j ) ) ;
18         // Menggunakan operator pengurangan
19         System.out.println ( " Pengurangan : " ) ;
20         System.out.println ( " i - j = " + ( i - j ) ) ;
21         // Menggunakan operator perkalian
22         System.out.println ( " Perkalian : " ) ;
23         System.out.println ( " i * j = " + ( i * j ) ) ;
24         // Menggunakan operator pembagian
25         System.out.println ( " Pembagian : " ) ;
26         System.out.println ( " i / j = " + ( i / j ) ) ;
27         // Menggunakan operator modulus
28         System.out.println ( " Modulus : " ) ;
29         System.out.println ( " i % j = " + ( i % j ) ) ;
30
31     }
32 }
```

## **b. ekspresi aritmatika**

**Penulisan suatu ekspresi aritmatika pada program java, sangat berkaitan dengan pernyataan pemberi nilai. Karena hasil dari ekspresi aritmatika akan ditampung kedalam suatu variabel.**

# ekspresi aritmatika(lanjutan)

**Bentuk umum penulisan ekspresi aritmatika, seperti dibawah ini :**

- a. LValue (Left Value), merupakan berupa variabel tunggal sebagai penampung hasil dari ekspresi Aritmatika**
- b. RValue (Right Value), merupakan Ekspresi Aritmatika, bisa berupa unary, binary atau ternary dan variabel lainnya.**

## **c.Hierarki Operator Aritmatik**

**Didalam suatu penulisan ekspresi aritmatika sering kita jumpai menggunakan beberapa operator aritmatika yang berbeda secara bersamaan. Maka dalam prosenya akan berbeda, tergantung dari urutan atau tingkatan operator tersebut.**

## c. Hierarki Operator Aritmatik(lanjutan)

MODUL\_D3\_JAVA - Microsoft Word

Home Insert Page Layout References Mailings Review View

Print Layout Full Screen Reading Layout Web Layout Outline Draft Document Views

Ruler Gridlines Message Bar Document Map Thumbnails Show/Hide

Zoom 100% One Page Two Pages Page Width

New Window Arrange All Split View Side by Side Synchronous Scrolling Reset Window Position Window Switch Windows

Macros

Berikut contoh penggunaan hierarki operator aritmatika, bisa anda ikuti seperti dibawah ini.

```
Source History
1
2 /**
3  *
4  * @author rony
5  */
6 public class HierarkiOperator {
7     public static void main(String[] args)
8     {
9
10    int A, B;
11    A = 8 + 2 * 3 / 6;
12    B = (8 + 2) * 3 / 6;
13    System.out.println("Hasil dari A = " + A);
14    System.out.println("Hasil dari B = " + B);
15
16    }
17 }
18
```

Output yang dihasilkan dari program HierarkiOperator.java diatas, seperti dibawah ini :

Page: 22 of 55 Words: 31/7,912 116%

start DOSEN MODUL\_D3\_JAVA - M... SAP\_D3\_BUSINESS... Microsoft PowerPoint ... 10:02 AM

## d.Operator Pemberi Nilai

Sebelumnya, kita telah mengenal operator pemberi nilai (Assignment Operator), yaitu menggunakan tanda sama dengan " = ", sebagai contoh  $A = A + 1$ . Dari ekspresi  $A = A + 1$ , bisa disederhanakan bentuk penulisan ekspresinya, yaitu menjadi  $A += 1$ .

Notasi  $+=$ , ini dikenal dengan operator pemberi nilai aritmatika. Java menyediakan beberapa notasi pemberi nilai.



## d.Operator Pemberi Nilai(lanjutan)

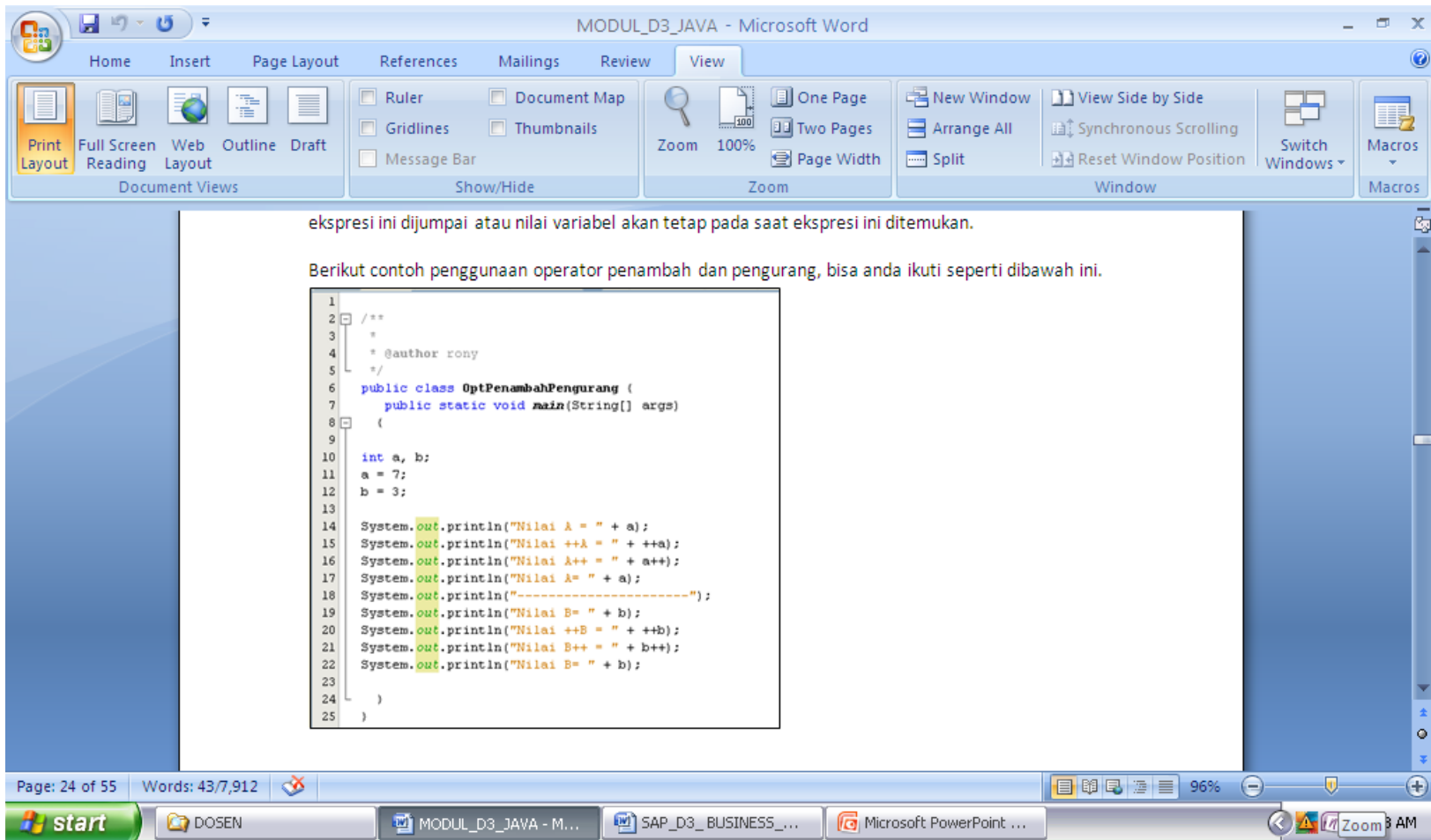
Contoh penggunaan Operator Pemberi Nilai

```
1
2  /**
3   *
4   * @author rony
5   */
6  public class OptPemberiNilai {
7      public static void main(String[] args)
8  {
9      int a, c, d, e;
10     double b;
11     a = 3; b = 5; c = 13; d = 4; e = 6;
12
13     //penggunaan assignment operator
14     a *=2;
15     b /=2;
16     c %=2;
17     d +=2;
18     e -=2;
19
20     System.out.println("Nilai a*=2 adalah:"+a);
21     System.out.println("Nilai b/=2 adalah:"+b);
22     System.out.println("Nilai c%=2 adalah:"+c);
23     System.out.println("Nilai d+=2 adalah:"+d);
24     System.out.println("Nilai e-=2 adalah:"+e);
25
26 }
27 }
```

## **E.Operator Penambah dan Pengurang**

**Masih berkaitan dengan operator pemberi nilai, Java menyediakan operator penambah dan pengurang, yaitu digunakan untuk menambah satu dan mengurangi satu dari nilai pada dirinya sendiri. Dari contoh penulisan operator pemberi nilai sebagai penyederhanaannya dapat digunakan operator penambah dan pengurang.**

# E.Operator Penambah dan Pengurang(lanjutan)



MODUL\_D3\_JAVA - Microsoft Word

Home Insert Page Layout References Mailings Review View

Print Layout Full Screen Reading Web Layout Outline Draft Document Views

Ruler Document Map Gridlines Thumbnails Message Bar Show/Hide

Zoom 100% One Page Two Pages Page Width

New Window View Side by Side Arrange All Synchronous Scrolling Split Reset Window Position Switch Windows Macros

ekspresi ini dijumpai atau nilai variabel akan tetap pada saat ekspresi ini ditemukan.

Berikut contoh penggunaan operator penambah dan pengurang, bisa anda ikuti seperti dibawah ini.

```
1  /**
2   *
3   * @author rony
4   */
5  public class OptPenambahPengurang {
6      public static void main(String[] args)
7      {
8
9
10     int a, b;
11     a = 7;
12     b = 3;
13
14     System.out.println("Nilai A = " + a);
15     System.out.println("Nilai ++A = " + ++a);
16     System.out.println("Nilai A++ = " + a++);
17     System.out.println("Nilai A = " + a);
18     System.out.println("-----");
19     System.out.println("Nilai B = " + b);
20     System.out.println("Nilai ++B = " + ++b);
21     System.out.println("Nilai B++ = " + b++);
22     System.out.println("Nilai B = " + b);
23
24 }
25 }
```

Page: 24 of 55 Words: 43/7,912 96%

start DOSEN MODUL\_D3\_JAVA - M... SAP\_D3\_BUSINESS... Microsoft PowerPoint ... Zoom 3 AM

## F. Operator Pembandingan (Comparison)

Berikut contoh penggunaan operator pembandingan, bisa anda ikuti seperti dibawah ini.

```
1
2  /**
3   *
4   * @author rony
5   */
6  public class OptPembandingan {
7      public static void main(String[] args)
8      {
9          int a, b, c, d, e, f;
10         boolean aa, bb, cc, dd, ee, ff;
11
12         a = 5; b = 6; c = 3; d = 9; e = 8; f = 2;
13
14         aa=a== b;
15         bb=b> c;
16         cc=c< d;
17         dd=d>= e;
18         ee=e<= f;
19         ff=b!= a;
20
21         System.out.println("Apakah"+a+"== "+ b +"? jawabannya " + aa);
22         System.out.println("Apakah"+b+"> "+ c +"? jawabannya " + bb);
23         System.out.println("Apakah"+c+"< "+ d +"? jawabannya " + cc);
24         System.out.println("Apakah"+d+">= "+ e +"? jawabannya " + dd);
25         System.out.println("Apakah"+e+"<= "+ f +"? jawabannya " + ee);
26         System.out.println("Apakah"+b+"!= "+ a +"? jawabannya " + ff);
27     }
28 }
```

## **G. Operator Logika**

**Operator Relasi digunakan untuk menghubungkan dua buah operasi relasi menjadi sebuah ungkapan kondisi. Hasil dari operator logika ini menghasilkan nilai boolean True atau False.**

- a).Operator Logika AND**
- b). Operator Logika OR**
- c). Operator Logika NOT**

## G. Operator Logika(lanjutan)

Contoh Penerapan Operator Logika dalam kode program

```
1
2  /**
3   *
4   * @author rony
5   */
6  public class OptLogika {
7      public static void main(String[] args) {
8
9          System.out.println("Operasi AND");
10         System.out.println("true && true = " + (true && true));
11         System.out.println("true && false = " + (true && false));
12         System.out.println("false && true = " + (false && true));
13         System.out.println("false && false = " + (false && false));
14
15         System.out.println("\nOperasi OR");
16         System.out.println("true || true = " + (true || true));
17         System.out.println("true || false = " + (true || false));
18         System.out.println("false || true = " + (false || true));
19         System.out.println("false || false = " + (false || false));
20
21         System.out.println("\nOperasi XOR");
22         System.out.println("true ^ true = " + (true ^ true));
23         System.out.println("true ^ false = " + (true ^ false));
24         System.out.println("false ^ true = " + (false ^ true));
25         System.out.println("false ^ false = " + (false ^ false));
26
27         System.out.println("\nOperasi NOT");
28         System.out.println("!true = " + (!true));
29         System.out.println("!false = " + (!false));
30     }
31 }
```