

OPERATOR

Bahasa Java

OPERATOR PENUGASAN

=

+=

*=

-=

/=

%=

Operator Penugasan

1. Operator “=” digunakan untuk memberikan nilai kepada variabel

Contoh:

```
int nilai; //disini variabel nilai belum memiliki nilai
```

Jika variabel nilai dicetak, maka akan error atau muncul peringatan karena disini dibuat variabel tapi tidak digunakan.

```
nilai = 5; //disini variabel nilai diberi nilai 5 menggunakan operator “=”
```

Jika variabel nilai dicetak, maka akan menghasilkan 5.



Operator Penugasan Penjumlahan

2. Operator “+=” digunakan untuk menambahkan dan mengupdate nilai suatu variabel

contoh:

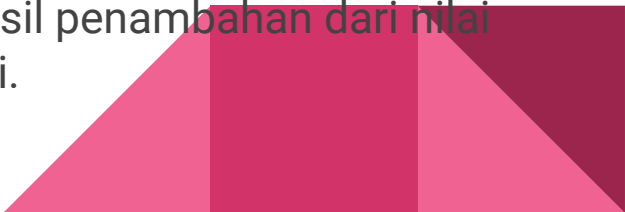
```
int nilai = 3
```

```
    nilai += 2
```

ini sama dengan :

```
int nilai = nilai + 3
```

Jika dicetak, maka hasilnya adalah 5 yang merupakan hasil penambahan dari nilai awal 3 dengan nilai 2 kemudian disimpan ke variabel nilai.



Operator Penugasan Pengurangan

3. Operator “-=” digunakan untuk mengurangi nilai variabel dan mengupdate nilai dari variabel.

contoh :

```
int  nilai = 5;
```

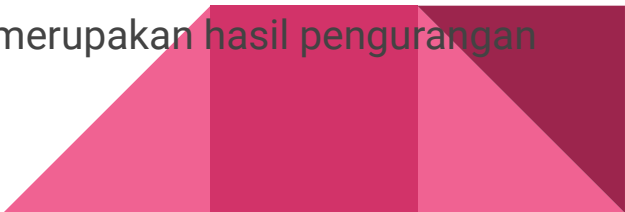
```
    nilai-= 3;
```

sama dengan

```
int nilai = 5
```

```
    nilai = nilai - 3
```

Jika dicetak, maka keduanya akan menghasilkan nilai **2** yang merupakan hasil pengurangan nilai awal yaitu 5 dikurang dengan 3



Operator Penugasan Perkalian

4. Operator “*=” digunakan untuk mengalikan nilai variabel dan mengupdate nilai dari variabel.

contoh :

```
int nilai = 5;
```

```
    nilai*= 3;
```

sama dengan

```
int nilai = 5
```

```
    nilai = nilai * 3
```

Jika dicetak, maka keduanya akan menghasilkan nilai **15** yang merupakan hasil perkalian nilai awal yaitu 5 dengan 3

Operator Penugasan Pembagian

5. Operator “/=” digunakan untuk membagi nilai variabel dan mengupdate nilai dari variabel.

contoh :

```
double    nilai = 5;
```

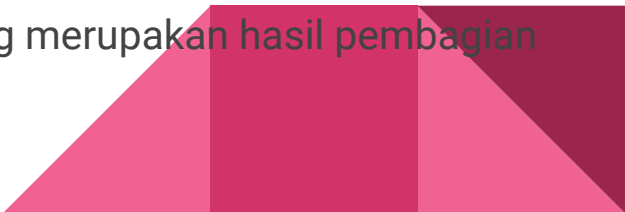
```
    nilai/= 2;
```

sama dengan

```
double nilai = 5
```

```
    nilai = nilai / 2
```

Jika dicetak, maka keduanya akan menghasilkan nilai **2.5** yang merupakan hasil pembagian nilai awal yaitu 5 dibagi dengan 2



Operator Penugasan Pembagian

6. Operator “%=” digunakan untuk mencari sisa bagi nilai variabel dan mengupdate nilai dari variabel.

contoh :

```
int  nilai = 5;
```

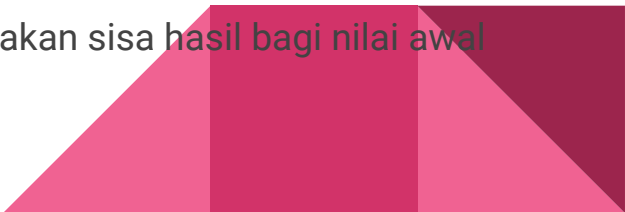
```
    nilai %= 3;
```

sama dengan

```
int  nilai = 5;
```

```
    nilai = nilai % 3;
```

Jika dicetak, maka keduanya akan menghasilkan nilai **2** yang merupakan sisa hasil bagi nilai awal yaitu 5 dengan 3



OPERATOR PEMBANDING

==

!=

<

>=

>

<=

APAKAH SAMA DENGAN (==)

1. Operator “==” digunakan untuk mengecek apakah nilai sebelah kiri dan sebelah kanan operator nilainya sama. Hasil dari perbandingan ini adalah nilai boolean (true atau false).

contoh :

```
3 == 5;
```

Jika dicetak akan menghasilkan false (salah) karena 3 tidak sama dengan 5.

Contoh 2:

```
int a =2;
```

```
nilai ==2;
```

Jika dicetak maka akan menghasilkan true (benar) karena nilai variabel a adalah 2 dan dibandingkan dengan nilai 2. Karena keduanya memiliki nilai 2 maka hasilnya adalah benar.

APAKAH TIDAK SAMA DENGAN (!=)

2. Operator “!=” digunakan untuk mengecek apakah nilai sebelah kiri dan sebelah kanan operator tidak sama. Hasil dari perbandingan ini adalah nilai boolean (true atau false).

contoh :

```
3 != 5;
```

Jika dicetak akan menghasilkan true (benar) karena benar bahwa 3 tidak sama dengan 5.

Contoh 2:

```
int a =2;
```

```
nilai ==2;
```

Jika dicetak maka akan menghasilkan false (salah) karena nilai variabel a adalah 2 dibandingkan dengan nilai 2. Karena keduanya memiliki nilai yang sama tidak benar (false) kalau a tidak sama dengan 2.

APAKAH LEBIH BESAR (>)

3. Operator ">" digunakan untuk mengecek apakah nilai sebelah kiri lebih besar dari pada nilai sebelah kanan operator. Hasil dari pembadingan ini adalah nilai boolean (true atau false).

contoh :

```
3 > 5;
```

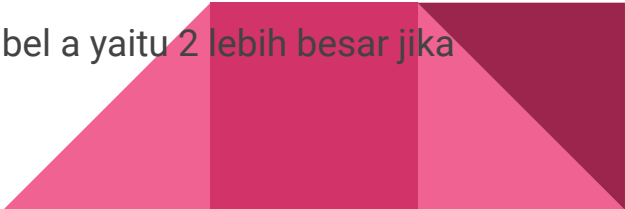
Jika dicetak akan menghasilkan false (salah) karena tidak benar bahwa 3 lebih besar dari pada 5.

Contoh 2:

```
int a =2;
```

```
nilai > 1;
```

Jika dicetak maka akan menghasilkan true (benar) karena nilai variabel a yaitu 2 lebih besar jika dibandingkan dengan 1.



APAKAH LEBIH KECIL(<)

4. Operator "<" digunakan untuk mengecek apakah nilai sebelah kiri lebih besar dari pada nilai sebelah kanan operator. Hasil dari pembadingan ini adalah nilai boolean (true atau false).

contoh :

```
3 < 5;
```

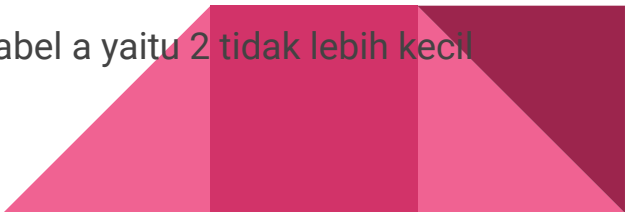
Jika dicetak akan menghasilkan true (benar) karena benar bahwa 3 lebih kecil dari pada 5.

Contoh 2:

```
int a =2;
```

```
nilai < 1;
```

Jika dicetak maka akan menghasilkan false (salah) karena nilai variabel a yaitu 2 tidak lebih kecil jika dibandingkan dengan 1.



APAKAH LEBIH BESAR ATAU SAMA DENGAN (\geq)

5. Operator " \geq " digunakan untuk mengecek apakah nilai sebelah kiri lebih besar atau sama dengan nilai sebelah kanan operator. Hasil dari perbandingan ini adalah nilai boolean (true atau false).

contoh :

```
3 >= 5;
```

Jika dicetak akan menghasilkan false (salah) karena tidak benar bahwa 3 lebih besar atau sama dengan 5.

Contoh 2:

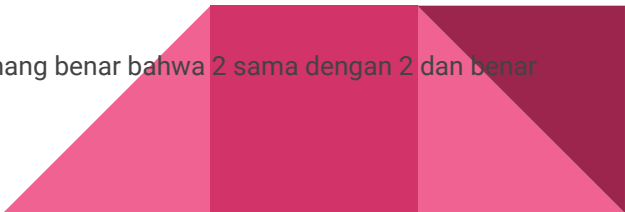
```
int a =2;
```

```
nilai >= 2;
```

Contoh 3:

```
2 >= 1;
```

Baik contoh 2 maupun contoh 3 Jika dicetak maka akan menghasilkan true (benar) karena memang benar bahwa 2 sama dengan 2 dan benar bahwa 2 lebih besar dari pada 1



APAKAH LEBIH KECIL ATAU SAMA DENGAN (<=)

6. Operator "<=" digunakan untuk mengecek apakah nilai sebelah kiri lebih kecil atau sama dengan nilai pada sebelah kanan operator. Hasil dari perbandingan ini adalah nilai boolean (true atau false).

contoh :

```
3 <= 5; //true
```

```
3 <= 3; //true
```

Jika dicetak akan menghasilkan true (benar) karena benar bahwa 3 lebih kecil dari pada 5 dan benar bahwa 3 sama dengan 3

Contoh 2:

```
int a =2;
```

```
nilai <= 1;
```

Jika dicetak maka akan menghasilkan false (salah) karena nilai variabel a yaitu 2 tidak lebih kecil jika dibandingkan dengan 1 dan tidak benar bahwa 2 sama dengan 1.



&&

OPERATOR LOGIKA

!

||

LOGIKA “DAN” (&&)

1. Operator “&&” digunakan untuk membandingkan 2 nilai boolean. Hanya akan bernilai true (benar), Jika nilai kiri dan kanan bernilai true, selain itu maka nilainya adalah false.

contoh :

```
boolean x = 3 <= 5 ; //true  
boolean y = 3 <= 3; //true
```

x && y;

Jika dicetak akan menghasilkan true (benar) karena nilai x dan y adalah true dengan prinsip operator && bahwa jika nilai kiri dan kanan adalah true maka nilainya adalah true

Contoh 2:

```
boolean a = false;  
boolean b = 1 <= 1;
```

Jika dicetak maka akan menghasilkan false (salah) karena sesuai prinsip operator && bahwa hanya akan bernilai benar jika kedua nilai yang dibandingkan bernilai benar.

LOGIKA “ATAU” (II)

1. Operator “||” digunakan untuk membandingkan 2 nilai boolean. Hanya akan bernilai false (salah) jika nilai kiri dan kanan bernilai false, selain itu maka nilainya adalah true.

contoh :

```
boolean x = 3 <= 5 ; //true  
boolean y = 3 <= 3; //true
```

x || y;

Jika dicetak akan menghasilkan true (benar) karena nilai x dan y bernilai true dengan prinsip operator || bahwa hanya akan bernilai false jika kedua nilai yang dibandingkan bernilai false.

Contoh 2:

```
boolean a = false;  
boolean b = 2 <= 1;
```

Jika dicetak maka akan menghasilkan false (salah) karena sesuai prinsip operator “||” bahwa akan bernilai false jika kedua nilai yang dibandingkan bernilai false.

LOGIKA “NOT” (!)

1. Operator “!” digunakan untuk membalikkan nilai boolean. Jika nilai true dibalikkan akan menghasilkan false, sebaliknya jika nilai false dibalikkan maka akan menghasilkan nilai true.

contoh :

```
boolean y = 3 <= 5 ; //true  
!y;
```

Jika dicetak akan menghasilkan false (salah) karena nilai y adalah true, sesuai dengan prinsip operator ! akan membalikkan nilai yang diberikan.

Contoh 2:

```
boolean a = false;  
!a;
```

Jika dicetak maka akan menghasilkan true (benar) karena nilai a adalah false sehingga ketika dibalikkan maka nilainya akan bernilai true.

