

Pemrograman I

Literatur

- H.M Deitel, P.J Deitel, **Small Java How to Program**-sixth Edition, Pearson Prentice Hall, 2005
- Elliot B. Koffman, Paul A.T. Wolfgang, **Objects, Abstraction, Data Structures and Design Using Java**, John Wiley & Sons.Inc, 2005
- Ian F. Darwin, **Java Cookbook**, O'Reilly, 2001
- Mark Allen Weiss, **Data Structures & Algorithm Analysis in Java**, Addison-Wesley, 1999
- Moh.Sjukani, **Algoritma & Struktur Data dengan C, C++ dan Java**, Mitra Wacana Media, Agustus 2005
- Rangsang Purnama, **Tuntunan Pemrograman Java jilid- 1**, Prestasi Pustaka Publisher, Januari 2003
- Rangsang Purnama, **Tuntunan Pemrograman Java jilid- 2**, Prestasi Pustaka Publisher, Juli 2003
- Rangsang Purnama, **Tuntunan Pemrograman Java jilid 3**, Prestasi Pustaka Publisher, Maret 2003
- Ariesto Hadi Sutopo, Fajar Masya, **Pemrograman Berorientasi Objek dengan Java**, Graha Ilmu, 2005
- Indrajani, Martin, **Pemrograman Berorientasi Objek dengan Java**, Elex Media Komputindo, 2004
- Melvin Antonius, Damian Bayu Imam Santoso, Carneles, **Membuat Animasi dengan Java**, Elex Media Komputindo, 2004

Materi

1. **Pengertian Java**
2. **Setup /Instalasi Java**
3. **Version Control (Git)**
4. Anatomi aplikasi Java
5. Classpath
6. **Variabel dan Tipe Data**
7. Operator
8. Control Flow (If.... Else, For/While)
9. **Class & Object**
10. **Method**
11. **Exception**
12. Konsep OOP
13. **Inheritance**
14. Encapsulation
15. **Polymorphism**
16. Abstract Class & Interface
17. **Composition & Aggregation**
18. Studi Kasus & Presentasi

OPERATOR, STRUKTUR KONTROL (FOR, IF, WHILE)

Operator Logika

Operator	Fungsi
&&	Logika AND
 	Logika OR
!	Logika NOT
xor	Logika xor

Operator

- Macam Operator
 1. Assignment : Isi Data
 2. Aritmatik
 3. Relasi
 4. Bitwise
 5. BitShift
 6. Logika

Operator :1.Assignment

- Assignment : Isi Datanya dapat berupa
 - ☐ `int x`
 - ☐ `x =5`
 - ☐ `x = x + 2` atau `x+=2`

Operator : 1. Assignment

- Contoh Operator

```
//Perintah Operator
```

```
package Latihan01.sesi1.bin;
```

```
public class CobaOperator01
```

```
{
```

```
    public static void main(String[] Xx)
```

```
    {
```

```
        int x=7;
```

```
        System.out.println("Nilai awal X adalah " +x);
```

```
        System.out.println("Nilai X++      adalah " + x++);
```

```
        System.out.println("Nilai X        adalah " +x);
```

```
        System.out.println();
```

```
        int y=25;
```

```
        System.out.println("Nilai awal Y adalah " +y);
```

```
        System.out.println("Nilai ++Y      adalah " + ++y);
```

```
        System.out.println("Nilai Y        adalah " +y);
```

```
    }
```

```
}
```

C:\windows\system32\cmd.exe

```
E:\Data Sep2010\06 Kuliah Gasal 1314\04 Pemrograman01\Murid\Latihan>javac -d bin  
-cp bin src/Operator/CobaOperator01.java
```

```
E:\Data Sep2010\06 Kuliah Gasal 1314\04 Pemrograman01\Murid\Latihan>java -cp bin  
Latihan01.sesi1.bin.CobaOperator01
```

```
Nilai awal X adalah 7
```

```
Nilai X++      adalah 7
```

```
Nilai X        adalah 8
```

```
Nilai awal Y adalah 25
```

```
Nilai ++Y      adalah 26
```

```
Nilai Y        adalah 26
```



```

/**
 * class increment.
 * @author (Adichan)
 * @version (op-02)
 */
public class increment
{
    public static void main(String[] args)
    {
        int a;
        a=120;
        System.out.println ("contoh postfix increment :");
        System.out.println (a);
        System.out.println (a++);
        System.out.println (a);
        a=120;
        System.out.println ("contoh prefix increment :");
        System.out.println (a);
        System.out.println (++a);
        System.out.println (a);
        a=120;
        System.out.println ("contoh postfix decrement :");
        System.out.println (a);
        System.out.println (a--);
        System.out.println (a);
        a=120;
        System.out.println ("contoh prefix decrement :");
        System.out.println (a);
        System.out.println (--a);
        System.out.println (a);

    }
}

```

Contoh increment dan drecrement apabila suatu nilai awal = 120 SOURCE

```
contoh postfix increment :  
120  
120  
121  
contoh prefix increment :  
120  
121  
121  
contoh postfix decrement :  
120  
120  
119  
contoh prefix decrement :  
120  
119  
119
```

Contoh
increment dan
drecrement
apabila suatu nilai
awal = 120
OUTPUT

Operator :2.Aritmatik

- Aritmatik dapat berupa

☐ +

☐ -

☐ /

☐ *

☐ % (mod)

Operator Aritmatika

Standar Operator dari yang tertinggi sampai yang terendah	Keterangan
*	Multiplication / Perkalian
/	Division / Pembagian
%	Remainder / Sisa Hasil Bagi
+	Addition / Penjumlahan
-	Subtraction / Pengurangan

Operator :2.Aritmatik

- Contoh Aritmatik

```
//Perintah Operator  
package Latihan01.sesi1.bin;
```

```
public class CobaOperator02 {  
    public static void main(String[] Xx) {  
        int x=2012;  
        int kbt1 = x %4;  
        System.out.println("Tahun : " +x);  
        System.out.println("Kabisat=0/Bukan Kabisat=1, Hasilnya " +kbt1);  
        System.out.println();  
        System.out.println();  
        int y=2013;  
        int kbt2 = y %4;  
        System.out.println("Tahun : " +y);  
        System.out.println("Kabisat=0/Bukan Kabisat=1, Hasilnya " +kbt2);  
        System.out.println();  
    }  
}
```

```
C:\windows\system32\cmd.exe  
E:\Data Sep2010\06 Kuliah Gasal 1314\04 Pemrograman01\Murid\Latihan>javac -d bin  
-cp bin src/Operator/CobaOperator02.java  
E:\Data Sep2010\06 Kuliah Gasal 1314\04 Pemrograman01\Murid\Latihan>java -cp bin  
Latihan01.sesi1.bin.CobaOperator02  
Tahun : 2012  
Kabisat=0/Bukan Kabisat=1, Hasilnya 0  
  
Tahun : 2013  
Kabisat=0/Bukan Kabisat=1, Hasilnya 1
```

Coba Anda kerjakan !!!!

Lalu push ke Github

Operator	Contoh cara penulisan pada java	Keterangan	Hasil
Contoh : apabila c=3, d=5, e=4, f=6, g=12			
+=	c += 7	c = c + 7	...?
- =	d - = 4	d = d - 4	...?
* =	e * = 5	e = e * 5	...?
/ =	f / = 5	f = f / 5	...?
% =	g % = 7	g = g % 7	...?

Operator :3.Relasi

- Relasi
 - ❑ Hanya berlaku untuk tipe data primitif (int, long)
 - ❑ Untuk data obyek membandingkan dengan menggunakan perintah equal
→ Integer x=3 dan Integer y=20 →
maka perintah membandingkannya adalah → x equal (y) → False

Operator Relasi

Operator	By Java	Contoh	Keterangan
=	==	$x == y$	x sama dengan y
≠	!=	$x != y$	x tidak sama dengan y
>	>	$x > y$	x lebih besar dari y
<	<	$x < y$	x lebih besar dari y
≥	>=	$x >= y$	x lebih besar atau sama dengan y
≤	<=	$x <= y$	x lebih kecil atau sama dengan y

Operator :4.Bitwise

- Bilangan bulat adalah bertype Integer, type data dilihat dari kapasitasnya dari yang terkecil ke terbesar, yaitu :
 - byte
 - short
 - int
 - long
- Data terbagi menjadi basis, diantaranya adalah :

- Binary (basis 2)
- Octal (basis 8)
- Decimal (basis 10)
- Hexadecimal (basis 16)

	Bin	Oct	Dec	Hex
1 + 1	10	2	2	2
6 + 2	1000	10	8	8
6 + 4	1010	12	10	10
9 + 7	10000	20	16	10

Operator :4.Bitwise

- Contoh misalkan membandingkan 8 & 10

```
1000
1010
----- &
1000
```

- Contoh misalkan membandingkan 8 || 10

```
1000
1010
----- ||
1010
```

- Contoh misalkan membandingkan 8 XOR 10

```
1000
1010
----- xor
0010
```

```
C:\windows\system32\cmd.exe

E:\Data Sep2010\06 Kuliah Gasal 1314\04 Pemrograman01\Murid\Latihan>javac -d bin
-cp bin src/Operator/CobaOperator04.java

E:\Data Sep2010\06 Kuliah Gasal 1314\04 Pemrograman01\Murid\Latihan>java -cp bin
Latihan01.sesi1.bin.CobaOperator04
Nilai Binary      [bin] 100= 4
Nilai Octal       [Oct] 100= 64
Nilai Decimal     [Dec] 100= 100
Nilai Hexadecimal[Hex] 100= 256
```

- Contoh Bitwise

```
//Perintah Operator -- BITWISE

package Latihan01.sesi1.bin;

public class CobaOperator04
{
    public static void main(String[] XX)
    {
        int x=100;
        System.out.println("Nilai Binary      [bin] 100= " +Integer.parseInt("100",2));
        System.out.println("Nilai Octal       [Oct] 100= " +Integer.parseInt("100",8));
        System.out.println("Nilai Decimal     [Dec] 100= " +Integer.parseInt("100",10));
        System.out.println("Nilai Hexadecimal[Hex] 100= " +Integer.parseInt("100",16));
    }
}
```

Operator :5.Bitshift

- BitShift → merupakan perintah geser kiri (dikali 2) dan geser kanan (dibagi 2)
 - 1000 → (angka desimal (8), akan digeser kekiri 1 hasilnya pasti 16
 - 10000 → Nilai jadi 16 → Apabila geser kiri
 - 100 → Nilai jadi 4 → Apabila geser kanan

Operator :5.Bitshift

- Penggunaan untuk akses level dengan kemungkinan :

Hapus	Edit	Lihat
0	0	1
0	1	0
1	0	0

Operator :5.Bitshif

Hapus	Edit	Lihat
0	0	1
0	1	0
1	0	0

- Penggunaan untuk akses level dengan kemungkinan :

- Boleh lihat saja $\rightarrow 001 \rightarrow 1$
- Boleh Edit saja $\rightarrow 010 \rightarrow 2$
- Boleh hapus saja $\rightarrow 100 \rightarrow 4$
- Boleh Lihat atau Edit $\rightarrow L | E \rightarrow 011$

```

001
010
----- |
011
    
```

- Boleh lihat atau hapus $\rightarrow L | H \rightarrow 101 \rightarrow 5$

```

001
100
----- |
101
    
```

- Boleh lihat/edit/hapus $\rightarrow L | E | H \rightarrow 111 \rightarrow 7$

```

001
010
100
----- |
111
    
```

Operator :5.Bitshift

- Dalam studi kasus permission, digunakan untuk ijin (pakai operator OR / |)
- Dalam studi kasus permission, digunakan memeriksa ijin (pakai operator AND/&)
 - a. User susi, mempunyai perm = 4,
 - b. User budi, mempunyai perm = 6,

Operator :5.Bitshif

Hapus	Edit	Lihat
0	0	1
0	1	0
1	0	0

- User susi, mempunyai perm = 4,
 - apakah dia bisa lihat (L & E) == lihat
001
010
----- &
000
Hasil dari perhitungan 000 dibanding kembali dengan perm lihat [001], apabila tidak sama → kesimpulannya dia tidak bisa lihat
 - apakah dia bisa hapus (H & E) == hapus
100
010
----- &
000
Hasil dari perhitungan 000 dibanding kembali dengan perm hapus [100], apabila tidak sama → kesimpulannya dia tidak bisa hapus

Operator :5.Bitshift

b. User budi, mempunyai perm = 6,

- apakah dia bisa lihat (L & P) == lihat

001

110

----- &

000

Hasil dari perhitungan 000 dibanding kembali dengan perm lihat [001], apabila tidak sama → kesimpulannya dia tidak bisa lihat

- apakah dia bisa edit (E & P) == edit

010

110

----- &

010

Hasil dari perhitungan 010 dibanding kembali dengan perm edit [010], apabila sama → kesimpulannya dia bisa edit

- apakah dia bisa hapus (H & P) == hapus

100

110

----- &

100

Hasil dari perhitungan 100 dibanding kembali dengan perm hapus [100], apabila sama → kesimpulannya dia bisa hapus

Operator :6.Logika

LOGIKA			
Op1	Op2	AND &&	OR
T	T	T	T
T	F	F	T
F	T	F	T
F	F	F	F