



PEMROGRMAN

CHAPTER 2

Dasar dan Aturan Penulisan Code Java

Abstract

[Draw your reader in with an engaging abstract. It is typically a short summary of the document.
When you're ready to add your content, just click here and start typing.]

Ahmad Pudoli
[Email address]

CHAPTER 2

Dasar dan Aturan Penulisan Code Java

TUJUAN INTERAKSIONAL

UMUM:

- Mahasiswa mampu menulis pemrograman Java

KHUSUS:

- Mahasiswa mampu menuliskan dan *Mengerti Penggunaan* komentar
- Mahasiswa mampu menuliskan dan *Mengerti Penggunaan* tipe data primitif
- Mahasiswa mampu menuliskan dan *Mengerti Penggunaan* *identifier*
- Mahasiswa mampu menuliskan dan *Mengerti Penggunaan* *Keyword*
- Mahasiswa mampu menuliskan dan *Mengerti Penggunaan* operator
- Mahasiswa mampu menuliskan dan *Mengerti Penggunaan* *casting*

YANG AKAN DIPELAJARI PADA PERTEMUAN INI:

- Komentar Pada Java
- Pernyataan/Statement dan Blok pada Java
- Tipe Data Primitif
- Inisialisasi Variabel dan Konstanta
- Keyword
- Operator
- Casting

A. Komentar Pada Java

Komentar merupakan catatan yang ditulis pada kode dengan tujuan sebagai bahan dokumentasi atau deskripsi tambahan mengenai program. Teks ini bukan bagian dari program dan akan diabaikan oleh compiler, artinya komentar tidak akan melakukan aksi apapun.

Ada tiga jenis komentar yang dapat dilakukan pada Java, yaitu Komentar untuk baris, blok dan komentar khusus untuk Javadoc.

1. Komentar Baris

Komentar baris, akan mengabaikan apapun mulai dari tanda `//` sampai akhir dari baris yang sama. Contoh:

```
// contoh komentar baris pada java
```

2. Komentar Blok

Komentar Blok, akan mengabaikan apapun yang berada diantara tanda `/*` dan `*/`. Contoh:

```
/*  
Contoh komentar blok ...  
Anda dapat membuat komentar di blok ini  
*/
```

3. Komentar Khusus Javadoc

Komentar javadoc khusus digunakan untuk generatisasi dokumentasi HTML untuk program java anda. Anda dapat menciptakan komentar javadoc dengan memulai baris dengan `/**` dan mengakhirinya dengan `*/`. Seperti Komentar blok, ini dapat juga menjangkau beberapa baris. Ini juga dapat terdiri atas tag-tag untuk menambahkan lebih banyak informasi pada komentar anda. Sebagai contoh:

```

/**
This is an example of special java doc comments used
for \n
generating an html documentation. It uses tags like:
@author Florence Balagtas
@version 1.2
*/

```

Berikut contoh keseluruhan dari jenis komentar yang terdapat di Java:

```

Komentar.java *
1
2 // Penulisan komentar pada program dapat dilakukan dengan dua cara
3 // 1. menggunakan komentar multi baris
4 //   komentar multi baris diawali dengan simbol /* dan diakhiri dengan simbol */
5 // 2. menggunakan komentar satu baris
6 //   komentar satu baris menggunakan simbol //
7
8 // Contoh ini komentar multi baris
9 /*
10  * ini komentar baris ke 1
11  * ini komentar baris ke 2
12  * ...
13  * ...
14  * ini komentar baris ke n
15  */
16
17 // Contoh menuliskan komentar satu baris sebagai berikut :
18 // ini komentar satu baris
19
20 // Komentar dapat dituliskan dimanapun di dalam program kita.
21 /* komentar bisa dituliskan di awal baris
22  * komentar dapat dituliskan di dalam class
23  * komentar dapat dituliskan di dalam method
24  * komentar bisa ada di manapun didalam program kita, sesuaikan dengan kebutuhan
25  */

```

B. Pernyataan/Statement dan Block Pada Java

pernyataan adalah satu atau lebih baris kode yang diakhiri dengan semicolon. sebagai contoh untuk pernyataan tunggal adalah:

```
System.out.println("Hello world");
```

Block adalah satu atau lebih pernyataan yang terbentang antara kurung kurawal buka dan kurung kurawal tutup yaitu sekumpulan pernyataan sebagai

satu unit kesatuan. Block pernyataan dapat dikumpulkan akan tetapi tidak secara pasti mempunyai keterkaitan fungsi. beberapa jumlah spasi kosong diijinkan terdapat didalamnya, sebagai contoh dari suatu block adalah :

```
public static void main( String[] args ){
    System.out.println("Hello");
    System.out.println("world");
}
```

Pedoman Penulisan Program:

1. Pada saat pembuatan block, anda dapat meletakkan kurung kurawal buka pada baris dengan pernyataan seperti contoh sebagai berikut:

```
public static void main( String[] args ){
```

atau anda dapat meletakkan kurung kurawal pada baris selanjutnya, seperti,

```
public static void main( String[] args ){
```

2. Anda harus memberi jarak (indent) pernyataan selanjutnya setelah awal dari block , seperti contoh berikut,

```
public static void main( String[] args ){
    System.out.println("Hello");
    System.out.println("world");
}
```

C. Java Identifier

Java Identifier adalah suatu tanda yang mewakili nama-nama variabel, method, class dsb.

Pendeklarasian Java adalah case-sensitive. Hal ini berarti bahwa pengidentifikasi : **Hello** tidak sama dengan **hello**. Pengidentifikasi harus dimulai dengan salah satu huruf, underscore “_”, atau tanda dollar “\$”.

Hurufnya dapat berupa huruf besar maupun huruf kecil. Karakter selanjutnya dapat menggunakan nomor 0 sampai 9.

Pengidentifikasi tidak dapat menggunakan kata kunci dalam java seperti class, public, void, dsb. Selanjutnya kita akan berdiskusi lebih banyak tentang kata kunci dalam java.

Pedoman Penulisan Program:

1. Untuk pemberian nama dai class java, diberikan huruf kapital untuk huruf pertama pada nama class. Untuk nama method dan variabel, huruf pertama dari kata harus dimulai dengan huruf kecil. Sebagai contoh:

```
ThisIsAnExampleOfClassName  
thisIsAnExampleOfMethodName
```

2. Pada kasus untuk pengidentifikasi labih dari satu kata, menggunakan huruf kapital untuk mengindikasikan awal dari kata kecuali kata pertama. Sebagai contoh, charArray, fileNumber, ClassName.
3. Hindari menggunakan underscores pada awal pengidentifikasian seperti _read atau _write.

D. Keyword Pada Java

Keyword adalah pengidentifikasi yang telah dipesan untuk didefinisikan sebelumnya oleh java untuk tujuan tertentu. Anda tidak dapat menggunakan keyword sebagai nama variabel anda, class, method dsb. Berikut ini adalah daftar dai kata kunci dalam java (Java Keywords).

CHAPTER 2: Dasar dan Aturan Penulisan Code Java

abstract	continue	for	new	switch
assert***	default	goto*	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum****	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp**	volatile
const*	Float	Native	Super	While
*	not used			
**	added in 1.2			
***	added in 1.4			
****	added in 5.0			

Keyword Pada Java

```
1 package Pertemuan2;
2
3
4 public class LatihanKeyword {
5     public static void main(String[] args){
6         // contoh deklarasi dan inisialisasi variabel tidak menggunakan keyword
7         String nim = "0811503379", nama = "Ahmad";
8         int nilAbsen = 80, nilTugas = 70, nilUAS = 75;
9         int jmlSKS = 3;
10        int biayaPerSKS = 150000;
11        char jk = 'L';
12        final int IF = 9;
13
14        // contoh keyword yang digunakan sebagai variabel
15        final int if() = 9;
16        int package;
17        char break;
18    }
19 }
```

Perbandingan Penggunaan Keyword Pada Varibel

Terlihat pada tiga baris pernyataan/statement terakhir, terindikasi bahwa terjadi error. Hal ini terjadi karena mereka menggunakan keyword yang ada di Java.

Catatan:

true, false, dan null bukan termasuk kata kunci akan tetapi mereka termasuk kata-kata khusus, jadi anda tidak dapat menggunakan mereka sebagai nama variabel pada program anda.

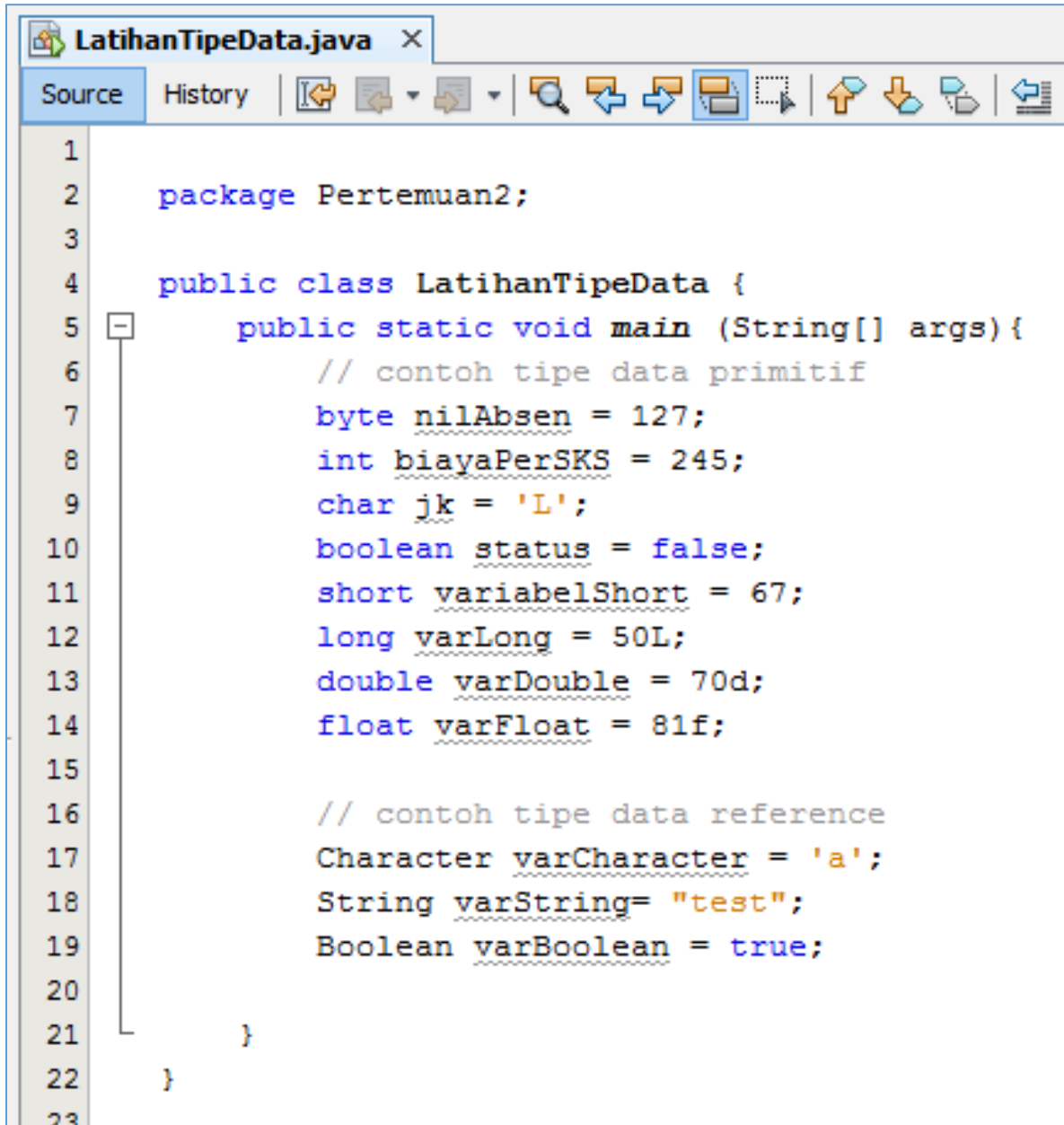
E. Tipe Data Primitif**1. Primitif**

Jenis-jenis nya sbb :

Tipe Data	Range	Ukuran Memory (1 byte = 8 bit)	Nilai Default
byte	-128 s/d 127	8-bit signed	0
short	-32.768 s/d 32.767	16-bit signed	0
int	-2,147,483,648 s/d 2,147,483,647	32-bit signed	0
long	-9,223,372,036,854,775,808 s/d 9,223,372,036,854,775,807	64-bit signed	0L
float	-3.4E38 s/d 3.4E38	32-bit IEEE 754	0.0f
double	-1.7E308 s/d 1.7E308	64-bit IEEE 754	0.0d
Char	character ASCII	8-bit signed	'\u0000'
Boolean	nilai true atau false	8-bit signed	false

2. Reference

Tipe data berupa class, dimana class ini memiliki berbagai fungsi yang bisa digunakan untuk kebutuhan pemrograman. Jenis-jenis nya antara lain : **Byte, Short, Integer, Long, Float, Double, Boolean, Character, String** dan lain sebagainya.



```
1
2 package Pertemuan2;
3
4 public class LatihanTipeData {
5     public static void main (String[] args){
6         // contoh tipe data primitif
7         byte nilAbsen = 127;
8         int biayaPerSKS = 245;
9         char jk = 'L';
10        boolean status = false;
11        short variabelShort = 67;
12        long varLong = 50L;
13        double varDouble = 70d;
14        float varFloat = 81f;
15
16        // contoh tipe data reference
17        Character varCharacter = 'a';
18        String varString= "test";
19        Boolean varBoolean = true;
20
21    }
22 }
23
```

F. Inisialisasi Variabel

Variabel merupakan sebuah tempat dimemori yang dialokasikan untuk menampung data yang memiliki sifat sementara, yang digunakan hanya pada saat program masih berjalan dan dihilangkan/dihapus dari memori setelah program berakhir.

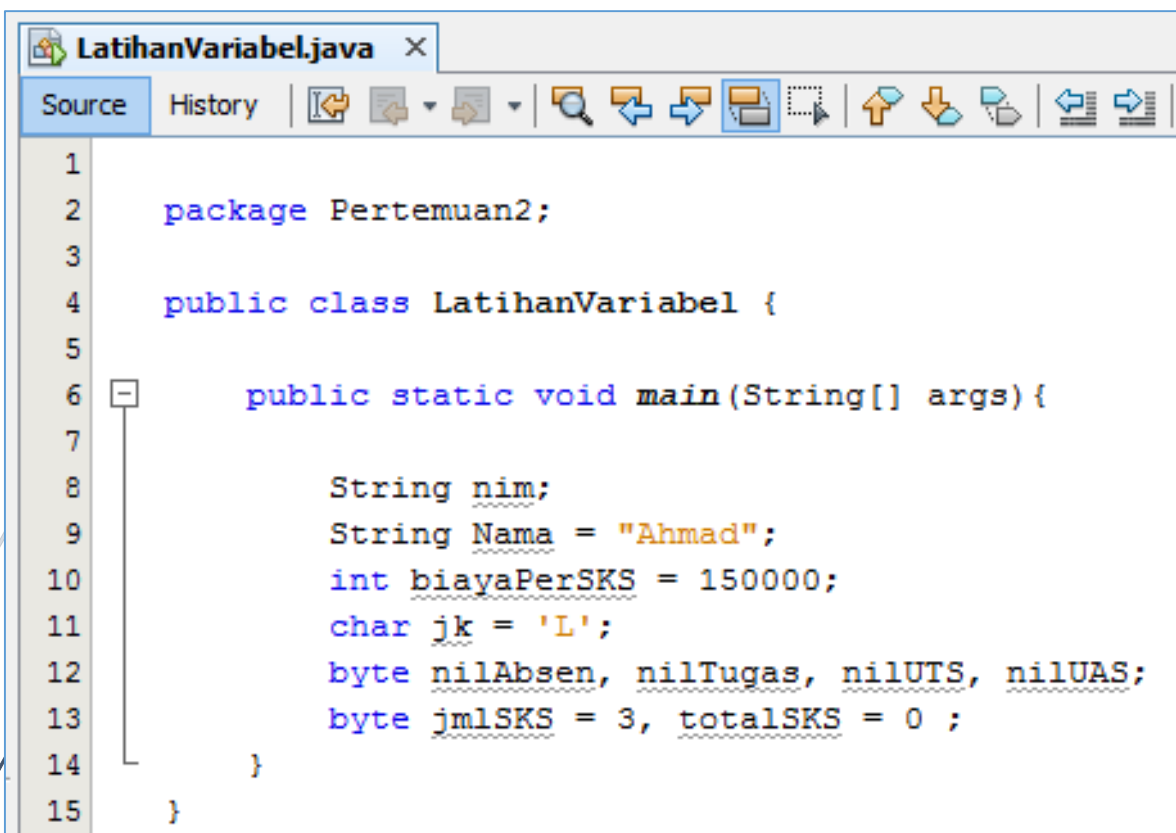
variabel memiliki tipe data dan nama. tipe data mengindikasikan tipe dari nilai yang dapat dibentuk oleh variabel itu sendiri. nama variabel harus mengikuti aturan untuk pengidentifikasiannya.

Format Deklarasi :

TipeData **namaVariabel** [= inisialisasiNilaiAwal];

Contoh :

```
int nilaiAbsen;
int nilaiTugas, nilaiUTS, nilaiUAS;
String namaDepan = "Tariq"
String namaTengah="Athar", namaBelakang="Kisan";
char nilaiHuruf='A' ;
```

A screenshot of an IDE window titled "LatihanVariabel.java". The window shows a Java code editor with a toolbar at the top. The code is as follows:

```
1
2 package Pertemuan2;
3
4 public class LatihanVariabel {
5
6     public static void main(String[] args){
7
8         String nim;
9         String Nama = "Ahmad";
10        int biayaPerSKS = 150000;
11        char jk = 'L';
12        byte nilAbsen, nilTugas, nilUTS, nilUAS;
13        byte jmlSKS = 3, totalSKS = 0 ;
14    }
15 }
```

Contoh Program Inisialisasi Variabel

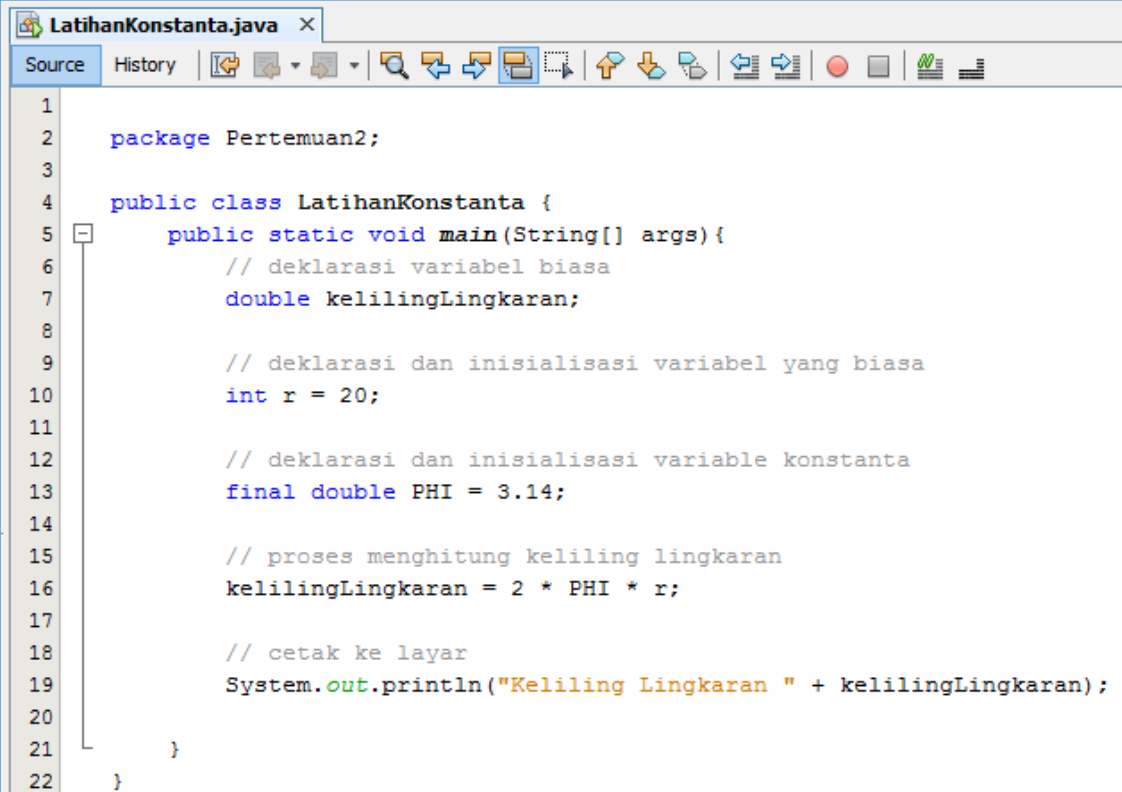
G. Inisialisasi Konstanta

Konstanta Pada prinsipnya merupakan sebuah variabel. Bedanya variabel ini menyimpan suatu nilai yang tidak bisa diubah. Variabel yang biasa boleh tidak diinisialisasi, sedangkan konstanta harus diinisialisasi dan nilai inisialisasi tersebut tidak dapat diubah.

Deklarasi konstanta mirip dengan deklarasi variabel seperti umumnya. Tetapi diawali dengan kata kunci/keyword **final** sebelum tipe datanya. Biasanya cara penulisan variabel konstanta harus dibuat dengan huruf kapital.

Berikut contoh penamaan konstanta:

```
final double PI = 3.14;
```



```

1
2  package Pertemuan2;
3
4  public class LatihanKonstanta {
5      public static void main(String[] args){
6          // deklarasi variabel biasa
7          double kelilingLingkaran;
8
9          // deklarasi dan inisialisasi variabel yang biasa
10         int r = 20;
11
12         // deklarasi dan inisialisasi variable konstanta
13         final double PHI = 3.14;
14
15         // proses menghitung keliling lingkaran
16         kelilingLingkaran = 2 * PHI * r;
17
18         // cetak ke layar
19         System.out.println("Keliling Lingkaran " + kelilingLingkaran);
20
21     }
22 }

```

Latihan Menggunakan Variabel Konstanta

H. Operator

Dalam Java, ada beberapa tipe operator. Ada operator aritmatika, operator relasi, operator logika, dan operator kondisi. Operator ini mengikuti macam-

macam prioritas yang pasti jadi compilernya akan tahu yang mana operator untuk dijalankan lebih dulu dalam kasus beberapa operator yang dipakai bersama-sama dalam satu pernyataan.

1. Operator Aritmatika

Berikut ini adalah dasar operator arithmatika yang dapat digunakan untuk membuat suatu program java,

Operator	Penggunaan	Keterangan
+	op1 + op2	Menjumlahkan op1 dengan op2
*	op1 * op2	Mengalikan op1 dengan op2
/	op1 / op2	Membagi ope1 dengan op2
%	op1 % op2	Menghitung sisa pembagian (modulus) op1 dengan op2
-	op1 - op2	Pengurangan nilai op1 oleh op2

2. Operator Increment dan Decrement

Dari sisi operator dasar aritmatika, java juga terdiri atas operator unary increment (++) dan operator unary decrement (--). operator increment dan decrement menambah dah mengurangi nilai yang tersimpan dalm bentuk variabel angka terhadap nilai 1. Sebagai contoh, pernyataan,

```
count = count + 1; //increment nilai count dengan nilai 1
```

pernyataan tersebut ekivalen dengan,

```
count++;
```

Operator	Penggunaan	Keterangan
----------	------------	------------

++	op++	Increment op sebanyak 1; mengevaluasi nilai op sebelum bertambah
++	++op	Increment op sebanyak 1; mengevaluasi nilai op setelah itu bertambah
--	op--	Decrement op sebanyak 1; mengevaluasi nilai op sebelum bertambah
--	--op	Decrement op sebanyak 1; mengevaluasi nilai op setelah itu bertambah

Operator increment dan decrement dapat ditempatkan sebelum atau sesudah operand.

Ketika digunakan sebelum operand, akan menyebabkan variabel diincrement atau didecrement oleh nilai 1, dan kemudian nilai baru digunakan dalam pernyataan dimana dia ditambahkan. Sebagai contoh,

```
int i = 10;
int j = 3;
int k = 0;
k = ++j + i; //akan menghasilkan k = 4+10 = 14
```

Ketika operator increment atau decrement ditempatkan setelah operand, nilai variabel yang lama akan digunakan lebih dulu dioperasikan lebih dulu terhadap pernyataan dimana dia ditambahkan. Sebagai contoh,

```
int i = 10;
int j = 3;
k = j++ + i; //akan menghasilkan k = 3+10 = 13
```

Catatan:

Selalu jaga pernyataan yang mengandung operator increment dan decrement untuk dipahami secara mudah dan sederhana.

3. Operator Relasi

Operator Relasi membandingkan dua nilai dan menentukan keterhubungan diantara nilai-nilai tersebut. Hasil keluarannya berupa **nilai boolean** yaitu true atau false.

Operator	Penggunaan	Keterangan
>	op1 > op2	op1 bernilai lebih besar dari op2
>=	op1 >= op2	op1 bernilai lebih besar atau sama dengan nilai op2
<	op1 < op2	op1 bernilai lebih kecil dari nilai op2
<=	op1 <= op2	op1 bernilai lebih kecil atau sama dengan nilai op2
==	op1 == op2	Nilai op1 sama dengan nilai op2
!=	op1 != op2	Nilai op1 tidak sama dengan op2

3. Operator Logika

Memiliki beberapa jenis simbol yaitu : &, &&, |, ||, ^, !

4. Operator Kondisi

Memiliki bentuk sbb : **exp1?exp2:exp3**

```

LatihanOperator.java x
Source History
1
2 package Pertemuan2;
3
4 public class LatihanOperator {
5
6     public static void main (String[] args){
7         int tambah, kali, bagi;
8         int a = 5, b = 10, c = 15;
9         // cetak nilai a, b dan c
10        System.out.println("cetak nilai a, b dan c");
11        System.out.println("a = " + a);
12        System.out.println("b = " + b);
13        System.out.println("c = " + c);
14        System.out.println("=====\n");
15
16        // contoh penggunaan operator Aritmatika
17        tambah = a + b;
18        kali = a * b;
19        bagi = kali / a;
20        System.out.println("Hasil Operator Aritmatika");
21        System.out.println("Penjumlahan a dan b = " + tambah);
22        System.out.println("perkalian a dan b = " + kali);
23        System.out.println("pembagian hasil perkalian dengan variabel a="+ bagi);
24        System.out.println("=====\n");
25
26        // contoh penggunaan operator increment dan decrement
27        tambah++;
28        kali--;
29        System.out.println("Hasil Increment dan Decrement");
30        System.out.println("Hasil Increment tambah = " + tambah);
31        System.out.println("Hasil Decrement kali = " + kali);
32
33        int incrementDidepan = ++tambah + a;
34
35        System.out.println("Hasil Increment tambah = " + tambah);
36        System.out.println("Nilai dahuluiIncrement = " + incrementDidepan);
37
38        int incrementDibelakang = tambah++ + a;
39
40        System.out.println("Hasil Increment tambah = " + tambah);
41        System.out.println("Nilai incrementDibelakang = " + incrementDibelakang);
42
43        System.out.println("=====\n");
44
45        // contoh penggunaan operator relasi dan kondisi
46        System.out.println("Apakah nilai variabel bagi >= 10 : " + ((bagi>=10)? "Benar" : "salah"));
47
48        // contoh penggunaan operator logika
49        System.out.println("Apakah a>b dan b>c = " + ( a>b && b>c ));
50        System.out.println("Apakah a>b atau b>c = " + ( a>b || b>c ));
51        System.out.println("Apakah c>b atau b>c = " + ( a>b || b>c ));
52    }
53
54 }

```

I. Konversi dan Casting

Pada pemrograman sering sekali terjadi pemindahan nilai dari satu variabel ke variabel lainnya dengan tipe data yang berbeda. Dalam pemindahan tersebut ada dua jenis pemindahan yang dilakukan yaitu [Konversi dan Casting](#).

Konversi adalah pemberian nilai kepada variabel yang berbeda tipe datanya. Dalam konversi melibatkan *widening conversation* yaitu tipe data tujuan harus mempunyai range yang lebih besar dari pada tipe data aslinya. Misalnya dari byte ke short, short ke long, atau int ke double. Konversi tidak dapat dilakukan pada tipe data boolean.

Casting adalah saat programmer melakukan konversi antar tipe yang tipenya tidak kompatibel, tidak dapat mengonversi secara otomatis maka dibutuhkan casting. Casting juga melibatkan

narrowing conversation yaitu bila tipe tujuan lebih kecil dari tujuan asal maka akan ada informasi yang hilang. Contoh casting saat pemberian nilai dari tipe data double ke int yang memiliki range lebih kecil dari double.

Format proses casting :

```
i = (int) d;
```

i = variabel tipe data int

d = variabel tipe data double


```
Konversi.java *
1  /*
2   *  Konversi atau promotion
3   *
4   */
5
6  public class Konversi{
7      public static void main (String[] args){
8
9          char c = 'a';
10         byte b = 12;
11         short s = 32767;
12         int i = 2147483647;
13         long l = 72036854775807L;
14         float f = 3.324243532f;
15         double d = 3.23442323445242;
16
17         System.out.println("Default value c = "+c);
18         System.out.println("Default value b = "+b);
19         System.out.println("Default value s = "+s);
20         System.out.println("Default value i = "+i);
21         System.out.println("Default value l = "+l);
22         System.out.println("Default value f = "+f);
23         System.out.println("Default value d = "+d);
24
25         d = f;
26         f = l;
27         l = i;
28         i = s;
29         s = b;
30         //c = i;
31
32         System.out.println("\nAfter Promotion proses : ");
33
34         System.out.println("\nDefault value c(int)           = "+c);
35         System.out.println("Default value b(short)      = "+b);
36         System.out.println("Default value s(byte)       = "+s);
37         System.out.println("Default value i(short)      = "+i);
38         System.out.println("Default value l (int)       = "+l);
39         System.out.println("Default value f (long)      = "+f);
40         System.out.println("Default value d (float)     = "+d);
41     }
42 }
43
44
```

```
Casting.java *
1  /*
2  * Casting
3  *
4  */
5
6  public class Casting{
7      public static void main (String args[]){
8
9          char c = 'a';
10         byte b = 12;
11         short s = 32767;
12         int i = 2147483647;
13         long l = 72036854775807L;
14         float f = 3.324243532f;
15         double d = 3.23442323445242;
16
17         System.out.println("Default value c = "+c);
18         System.out.println("Default value b = "+b);
19         System.out.println("Default value s = "+s);
20         System.out.println("Default value i = "+i);
21         System.out.println("Default value l = "+l);
22         System.out.println("Default value f = "+f);
23         System.out.println("Default value d = "+d);
24
25         f = (float)d;
26         l = (long)f;
27         i = (int)l;
28         s = (short)i;
29         b = (byte)s;
30         c = (char)i;
31
32         System.out.println("\nAfter Casting proses: ");
33
34         System.out.println("\nDefault value c(int)      = "+c);
35         System.out.println("Default value b(short)   = "+b);
36         System.out.println("Default value s(int)     = "+s);
37         System.out.println("Default value i(long)    = "+i);
38         System.out.println("Default value l (float)  = "+l);
39         System.out.println("Default value f (double) = "+f);
40         System.out.println("Default value d (double) = "+d);
41     }
42 }
43
```