

Rekayasa Perangkat Lunak

Mata Pelajaran : Pemrograman Berorientasi Objek

Kelas : XI

Daftar Isi

A. Kompetensi Dasar	1
B. Tujuan Pembelajaran	1
C. Indikator Pencapaian Kompetensi	1
D. Uraian Materi	1
1. Pertemuan 1	1
1.1. Penulisan Komentar	1
1.2. Identifiers	2
1.3. Kata-kata kunci	3
1.4. Tipe data primitive, Inisialisasi dan nilai default	3
2. Pertemuan 2	8
3. Pertemuan 3	8
4. Pertemuan 4	8

A. Kompetensi Dasar

- Memahami dasar-dasar dan aturan pemrograman berorientasi obyek
- Menyajikan aturan dan dasar-dasar pemrograman berorientasi obyek

B. Tujuan Pembelajaran

- Melalui studi literatur siswa dapat memahami dasar-dasar aturan pemrograman berorientasi obyek
- Melalui praktikum siswa dapat mengimplementasikan dasar-dasar aturan pemrograman berorientasi obyek

C. Indikator Pencapaian Kompetensi

- Dapat menuliskan komentar pada bahasa pemrograman berorientasi obyek
- Mampu membuat penamaan identifiers yang benar dan tepat.
- Mengenali kata-kata kunci bahasa pemrograman berorientasi obyek
- Mampu menyebutkan dan menjelaskan 8 tipe data primitif beserta nilai defaultnya
- Mampu mendeklarasikan dan meng-inisialisasi variabel

D. Uraian Materi

1. Pertemuan 1

1.1. Penulisan Komentar

Komentar adalah baris code program yang tidak dieksekusi oleh compiler. Biasanya komentar digunakan untuk dokumentasi pemrograman, agar jika orang lain yang melihat kode program paham tujuan penulisan dari baris atau blok code program tersebut. Pada bahasa pemrograman java komentar dapat ditulis dengan 3 cara.

1) End Of Line Comment (//)

Komentar dengan tanda garis miring ganda, digunakan untuk membuat komentar hanya satu baris saja.

```
11 | //System.out.println digunakan untuk mencetak kalimat ke layar  
12 | System.out.println("Assalamualaikum");
```

2) Multiple Line Comment (/* */)

Komentar dengan tanda garis miring dan asterik (/ * */) disebut dengan Multiple-Line Comment, setiap baris kata atau kalimat yang diapit oleh tanda tersebut akan dianggap sebagai komentar. Biasanya digunakan untuk penulisan komentar lebih dari satu baris. Sebagai contoh :

```
12 System.out.println("Assalamualaikum");
13 /*
14     Penggunaan garis miring dan asterisk
15     setiap baris dalam tanda apit tersebut
16     adalah
17     komentar
18 */
```

3) Javadoc Comment (/** */)

Komentar dengan tanda garis miring ganda didepan dan asterik (/** */) disebut dengan Javadoc Comment. Penggunaannya sama seperti Multiple Line Comment, akan tetapi penggunaannya untuk dokumentasi-dokumentasi didalam program. Javadoc comment terkadang dibuat otomatis oleh IDE Pemrograman. Contoh Javadoc comment

```
7 package latihan0;
8 /**
9  *
10 * @author Safrizal
11 */
```

1.2. Identifiers

Identifiers adalah nama untuk mengidentifikasi objek,class, variable, properties, prosedur dan function. Syarat penulisan identifiers :

- 1) Membedakan huruf besar dan huruf kecil (case sensitive).
- 2) Nama identifier bisa terdiri dari satu atau beberapa karakter yang terdiri dari angka (0 – 9), huruf (A – Z, a – z), dollar (\$), garis-bawah/underscore(_).
- 3) Penamaan identifier harus dimulai dengan huruf, underscore(_), atau karakter dollar (\$). Tidak boleh dimulai dengan angka.
- 4) Simbol-simbol operator seperti (*), (+), (/), (-) tidak boleh digunakan.
- 5) Dalam penamaan identifier tidak boleh menggunakan spasi.

- 6) Tidak diperbolehkan menggunakan Kata-kata keyword dalam java (misalnya: public, class, static, dsb).
- 7) Panjang nama identifier tidak dibatasi.

Contoh penamaan identifier yang benar:

Nama

\$Nama

NamaSiswa

NomorInduk3

namaSiswa2

nomor8Induk

nomor_induk

Contoh penamaan identifier yang salah:

1Nama

/NomorInduk

Nama Siswa

1.3. Kata-kata kunci

Keyword (kata kunci) adalah kata-kata khusus dalam Java yang digunakan untuk pemrograman. Berikut adalah beberapa keyword yang digunakan dalam bahasa pemrograman Java:

abstract	boolean	break	byte
case	catch	char	class
const	continue	default	do
double	else	extends	final
finally	float	for	future
goto	if	implements	import
instanceof	int	interface	long

native	new	package	private
protected	public	return	short
static	super	switch	synchronized
this	throw	throws	transient
try	void	volatile	while

1.4. Tipe data primitive, Inisialisasi dan nilai default

1) Byte

- Memiliki ukuran 8 bit
- Memiliki ukuran minimal nilainya -128 (-2^7)
- Memiliki default nilai 0
- Gunakan tipe data ini kalau range datanya kecil, dengan tujuan untuk menghemat memori dibanding int, terutama kalau datanya berupa array
- Contoh : byte a = 100; byte b = -50;

2) Short

- Memiliki ukuran 16 bit
- Memiliki ukuran minimal nilainya -32,768
- Memiliki ukuran maksimal nilainya 32,768
- Memiliki default nilai 0
- Sama dengan byte, gunakan kalau perlu optimasi penggunaan memory, terutama kalau datanya berupa array
- Contoh : short s = 10000; short r = -20000;

3) Int

- Memiliki ukuran 32 bit
- Memiliki ukuran minimal nilainya -2,147,483,648
- Memiliki ukuran maksimal nilainya 2,147,483,648
- Memiliki default nilai 0
- int adalah tipe data yang paling sering digunakan untuk representasi angka bulat. Int adalah pilihan default jika tidak diperlukan optimasi memori. Kalau jangkauan angkanya kurang lebar gunakan long.
- Contoh : int i = 100000; int b = -200000;

4) Long

- Memiliki ukuran 64 bit
- Memiliki ukuran minimal nilainya -9223372036854775808
- Memiliki ukuran maksimal nilainya 9223372036854775807
- Memiliki default nilai 0L
- Gunakan tipe data ini jika jangkauanya lebih lebar dari pada int
- Contoh : long l = 100000L; long b = -200000L;

5) Float

- Memiliki ukuran 32 bit
- Memiliki default nilai 0.0f
- float adalah tipe data pecahan yang didefinisikan oleh standar IEEE 754.
- Digit di belakang koma tidak bisa diatur, jika ingin merepresentasikan angka yang bisa diatur jumlah digit di depan dan di belakang koma, gunakan BigDecimal
- tipe data ini tidak pernah digunakan untuk perhitungan uang
- Contoh : float f1 = 234.5f;

6) Double

- Memiliki ukuran 64 bit
- Memiliki default nilai 0.0d
- double adalah tipe data pecahan yang didefinisikan oleh standar IEEE 754.
- Digit di belakang koma tidak bisa diatur, jika ingin merepresentasikan angka yang bisa diatur jumlah digit di depan dan di belakang koma, gunakan BigDecimal
- Contoh : double d = 234.5;

7) Boolean

- Tipe data boolean hanya mempunyai dua nilai, true dan false
- Memiliki Default nilai false.

- Contoh : `boolean b = false;`

8) Char

- Memiliki ukuran 64 bit
- Memiliki ukuran minimal nilainya `'\u0000'` atau 0
- Memiliki ukuran maksimal nilainya `'\uffff'` atau 65535
- `char` adalah tipe data yang bisa menyimpan nilai unicode. Digunakan untuk merepresentasikan karakter unicode yang didalamnya terdapat abjad, angka, simbol dan karakter lainnya. Table unicode bisa dilihat pada link berikut
<http://www.tamasoft.co.jp/en/general-info/unicode.html>
- Contoh : `char c = 'A';`

Inisialisasi (Initialized) adalah proses atau perintah memberikan nilai ke dalam variable. Inisialisasi bisa dilakukan pada saat pendeklarasian variable maupun diluar pendeklarasian.

Contoh Deklarasi Variable, dan Inisialisasi Nilai Default pada java :

```

16 byte tipeByte; //Nama Variable tipeByte, tipe data byte
17 short tipeShort; //Nama Variable tipeShort, tipe data short
18 int tipeInt; //Nama Variable tipeInt, tipe data int
19 long tipeLong; //Nama Variable tipeLong, tipe data long
20 float tipeFloat; //Nama Variable tipeFloat, tipe data float
21 double tipeDouble; //Nama Variable tipeDouble, tipe data double
22 boolean tipeBoolean; //Nama Variable tipeBoolean, tipe data boolean
23 char tipeChar; //Nama Variable tipeChar, tipe data char;
24
25 /*Inisialisasi nilai pada variable
26     Inisialisasi adalah memberikan nilai kedalam variable
27     Inisialisasi bisa dilakukan pada saat pendeklarasian variable diatas
28     contoh :
29     byte tipeByte=0;
30 */
31 //INISIALISASI dengan nilai default
32 tipeByte=0;
33 tipeShort=0;
34 tipeInt=0;
35 tipeLong=0l; //angka 0 dan huruf l (EL)
36 tipeFloat=0.0f;
37 tipeDouble=0.0d;
38 tipeBoolean=false;
39 tipeChar='\u0000';
40 System.out.println("Nilai Default byte =" + tipeByte);
41 System.out.println("Nilai Default short =" + tipeShort);
42 System.out.println("Nilai Default int =" + tipeInt);
43 System.out.println("Nilai Default long =" + tipeLong);
44 System.out.println("Nilai Default float =" + tipeFloat);
45 System.out.println("Nilai Default double =" + tipeDouble);
46 System.out.println("Nilai Default boolean =" + tipeBoolean);
47 System.out.println("Nilai Default char =" + tipeChar);

```

Hasil kompilasi kode program diatas :

```

Nilai Default byte =0
Nilai Default short =0
Nilai Default int =0
Nilai Default long =0
Nilai Default float =0.0
Nilai Default double =0.0
Nilai Default boolean =false
Nilai Default char =

```

Latihan :

- 1) Pilih 5 kata kunci yang terdapat pada bahasa pemrograman java, kemudian cari 5 kegunaan kata kunci tersebut !
- 2) Buatlah contoh penulisan identifiers yang benar (5 buah) dan yang salah (5 buah) !

- 3) Buatlah 8 buah variable dengan 8 tipe data yang berbeda, inisialisasi variabel tersebut dengan nilai yang berbeda-beda, kemudian tampilkan kelayar, sertakan komentar penjelasan dari setiap baris kode yang ditulis !

Catatan : nama variable harus jelas, mewakili tujuan dari dibuatnya variable tersebut.

2. Pertemuan 2

1.1. Casting

Untuk tipe data primitif yang berupa angka, kita bisa melakukan pertukaran data. Proses pertukaran data antar tipe data yang berbeda disebut dengan casting. Ada casting yang dilakukan secara implisit, ada juga casting yang dilakukan secara eksplisit.

Casting secara implisit dilakukan tanpa adanya kode tambahan, biasanya terjadi kalau kita memasukkan tipe data dengan ukuran yang lebih kecil ke tipe data yang lebih besar, misalnya dari byte ke int

contoh :

```
byte z = 100;  
int i = z;  
System.out.println(i);
```

Casting secara eksplisit dilakukan jika kita memasukkan data dengan ukuran lebih besar ke ukuran lebih kecil, misalnya dari int ke byte. Perlu diperhatikan bahwa presisi data akan berubah dari ukuran ke besar ke ukuran lebih kecil, sehingga kalau nilainya melebihi jangkauan, maka terjadi pemotongan nilai dan hasilnya bisa tidak terduga.

contoh :

```
int i = 100;  
byte z = (byte) i;  
System.out.println(z);
```

Contoh di atas tidak terjadi pemotongan nilai karena 100 masih di dalam jangkauan byte, kalau contoh di bawah ini akan terjadi pemotongan nilai :

contoh :

```
int i = 1000;  
byte z = (byte) i;  
System.out.println(z);
```

nilai variabel z tidak akan 1000, karena jangkauan byte hanya sampai 127 saja.

1.2. Promotion

Promotion terjadi pada saat operasi aritmatik dimana kompiler berusaha mencari tipe data yang sesuai dengan tipe data operan yang berbeda-beda

```
byte a = 20;
byte b = 40;
byte c = 10;
int total = a*b+c;
System.out.println(total);
```

Meskipun variabel a,b dan c masing masing, bertipe byte, namun di dalam ekspresi diatas, variabel variabel tersebut akan dipromosikan ke dalam tipe int, tanpa harus melakukan typecasting terlebih dahulu

1.3. Operator

Operator Aritmatika

Operator	Keterangan
+	Penjumlahan
-	Pengurangan
*	Perkalian
/	Pembagian
%	Sisa hasil bagi

Operator Penugasan/Assignment

Operator	Keterangan
=	Pemberian Nilai
+=	Penambahan Bilangan
-=	Pengurangan Bilangan
/=	Pebagian Bilangan

Contoh :

```
byte a ;
byte b = 40;
byte c = 20;
byte d=30;

a=10;
b+=2;
c-=5;
d/=5;

System.out.println(a);
System.out.println(b);
System.out.println(c);
System.out.println(d);
```

Operator Perbandingan

Operator	Keterangan
==	Sama dengan
<	Lebih kecil
<=	Lebih kecil sama dengan
>	Lebih besar
>=	Lebih besar sama dengan
!=	Tidak sama

Operator Logika

Nama Operator	Simbol	Kegunaan
AND	&&	Dua Kondisi Harus Dipenuhi
OR		Dari dua kondisi, akan benar bila ada salah satu atau lebih kondisi yang dipenuhi
NOT	!	Invers dari kondisi yang diberikan

3. Pertemuan 3

Pengambilan keputusan

4. Pertemuan 4

Perulangan