Dasar-Dasar Pemrograman

Yudi Adha. ST. MMSI

Bahasa Pemrograman Java

Tujuan

Pada bagian ini, kita akan mendiskusikan mengenai bagian dasar pemrograman Java. Kita akan memulai dengan mencoba menjelaskan bagian dasar dari program Hello.java. Kita juga akan mendiskusikan beberapa pedoman cara menulis script atau petunjuk penulisan kode dalam penulisan program yang lebih efektif dan mudah dibaca.

Pada akhir pembahasan, diharapkan mahasiswa dapat :

- Mengidentifikasi bagian dasar dari program Java
- Membedakan mana yang termasuk ke dalam Java literals, tipe data dasar, tipe variabel, pengidentifikasian dan operator
- Mengembangkan program Java sederhana menggunakan konsep yang dipelajari pada bab ini

Menganalisa program Java Pertama

- Sekarang, kita akan mencoba untuk menganalisa program Java pertama :

```
public class Hello
{
   /**
    * My first java program
    */
   public static void main(String[] args) {
         //menampilkan string ”Hello world” pada layar
      System.out.println("Hello world!");
   }
}
```

Baris pertama kode :

public class Hello

menandakan nama class yaitu Hello. Dalam Java, semua kode seharusnya ditempatkan di dalam deklarasi class. kita melakukannya dengan menggunakan kata kunci class. Sebagai tambahan, class menggunakan access specifier public, yang mengindikasikan bahwa class kita mempunyai akses bebas ke class yang lain dari package yang lain pula .

- Baris berikutnya yaitu yang terdiri atas kurung kurawal { menandakan awal blok.
- Pada kode ini, kita menempatkan kurung kurawal pada baris selanjutnya setelah deklarasi class, bagaimanapun, kita dapat juga meletakkan kurung kurawal ini setelah baris pertama dari kode yang kita tulis. Jadi, kita dapat menulis kode kita sebagai berikut :

```
public class Hello
{
```

atau

```
 public class Hello {
```

•Tiga baris selanjutnya menandakan adanya komentar Java. Komentar adalah sesuatu yang digunakan untuk mendokumentasikan setiap bagian dari kode yang ditulis.

```
/**
* My first java program
*/
```

•Komentar dinyatakan dengan tanda “/*” dan “*/”. Segala sesuatu yang ada diantara tersebut diabaikan oleh compiler Java, dan mereka hanya dianggap sebagai koment

- Baris selanjutnya,

  public static void main(String[] args) {

  atau dapat juga ditulis sebagai berikut,

  public static void main(String[] args)

  {

- mengindikasikan nama suatu method dalam class Hello yang bertindak sebagai method utama. Method utama adalah titik awal dari suatu program Java

- Baris berikutnya :

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

- menampilkan teks “Hello World!” pada layar. Perintah System.out.println(), menampilkan teks yang diapit oleh tanda double pute (“ ”) pada layar.

Komentar pada Java

Komentar adalah catatan yang ditulis pada kode dengan tujuan sebagai bahan dokumentasi. Teks tersebut bukan bagian dari program dan tidak mempengaruhi jalannya program.

Java mendukung tiga jenis komentar :

- C++ style komentar satu baris
- C style beberapa baris
- komentar javadoc khusus

Penulisan Komentar C++ Style

- Komentar C++ style diawali dengan //. Semua teks setelah // dianggap sebagai komentar. Sebagai contoh,

```
// This is a C++ style or single line comments
```

Penulisan Komentar C Style

- Komentar C-style atau juga disebut komentar beberapa baris diawali dengan /* dan diakhiri dengan */. Semua teks yang ada diantara dua tanda tersebut dianggap sebagai komentar.

```
/* this is an example of a
C style or multiline comments */
```

Komentar Khusus javadoc

- Komentar javadoc khusus digunakan untuk men-generate dokumentasi HTML untuk program Java memulai baris dengan /** dan mengakhirinya dengan */.

```
/**
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
*/
```

Java Identifier

Java Identifier adalah suatu tanda yang mewakili nama-nama variabel, method, class, dsb. Contoh dari Identifier adalah : Hello, main, System, out.

Pendeklarasian Java adalah case-sensitive. Hal ini berarti bahwa Identifier : **H**ello tidak sama dengan **h**ello. Identifier harus dimulai dengan salah satu huruf, underscore "_", atau tanda dollar "$". Hurufnya dapat berupa huruf besar maupun huruf kecil. Karakter selanjutnya dapat menggunakan nomor 0 smpai 9.

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

Pemberian Nama dari Class Java

Untuk pemberian nama dari 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. Sebagi contoh:

***T**his**I**s**A**n**E**xample**O**f**C**lass**N**ame*

***t**his**I**s**A**n**E**xample**O**f**M**ethod**N**ame*

Pada kasus untuk identifier lebih dari satu kata, menggunakan huruf kapital untuk mengindikasikan awal dari kata kecuali kata pertama. Sebagai contoh, charArray, fileNumber, ClassName.

Hindari menggunakan underscores pada awal identifier seperti _read atau _write.

Keyword dalam 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

Tipe Data

Bahasa Pemrograman Java

Tipe Data Primitif

Bahasa pemrograman Java mendefinisikan delapan tipe data primitif, diantaranya adalah boolean (untuk bentuk logika), char (untuk bentuk tekstual), byte, short, int, long (integral), double and float (floating point).

logika – boolean

Tipe data boolean diwakili oleh dua pernyataan : true dan false. Sebagai contoh adalah,

```
boolean result = true;
```

teksual – char

Tipe data character (char), diwakili oleh karakter single Unicode. Tipe data ini harus memiliki ciri berada dalam tanda *single quotes*(' '). Sebagai contoh,

'a' //Huruf a

'\t' //A tab

Untuk menampilkan karakter khusus seperti ' (single quotes) atau " (double quotes), menggunakan karakter escape \. Sebagai contoh,

'\'' //untuk single quotes

'\"' //untuk double quotes

Integral – byte, short, int & long

Tipe data integral dalam Java menggunakan tiga bentuk- yaitu desimal, oktal atau heksadesimal. Contohnya,

2 //nilai desimal 2

077//angka 0 pada awal pernyataan mengindikasikan nilai oktal

0xBACC //karakter 0x mengindikasikan nilai heksadesimal

Floating Point – float dan double

Tipe Floating point memiliki double sebagai default tipe datanya. Floating-point literal termasuk salah satunya desimal point atau salah satu dari pilihan berikut ini

E or e //(add exponential value)

F or f //(float)

D or d //(double)

Contohnya adalah,

```
3.14 //nilai floating-point sederhana (a double)
6.02E23      //A nilai floating-point yang besar
2.718F       //A nilai float size sederhana
123.4E+306D  //A nilai double yang besar dengan nilai redundant D
```

Variabel

Variabel *adalah item yang digunakan data untuk menyimpan pernyataan objek.*

Variabel memiliki ***tipe data*** *dan* ***nama****. Tipe data menandakan tipe nilai yang dapat dibentuk oleh variabel itu sendiri.* ***Nama variabel*** *harus mengikuti aturan untuk identifier.*

Deklarasi dan Inisialisasi Variabel

Untuk deklarasi variabel adalah sebagai berikut,

<data tipe> <name> [=initial value];

Catatan: Nilainya berada diantara <> adalah nilai yang disyaratkan, sementara nilai dalam tanda [] bersifat optional.

Berikut ini adalah contoh program yang mendeklarasikan dan menginisialisasi beberapa variabel,

```
public class VariableSamples
{
public static void main( String[] args ){
   //deklarasi tipe data dengan nama variable
   // result dan tipe data boolean boolean     result;
    //deklarasi tipe data dengan nama variabel
   // option dan tipe data char
   char option;
   option = 'C';              //menandai 'C' sebagai option

   //deklarasi tipe data dengan nama variabel
   //grade, double tipe data dan telah di inisialisasi
   //to 0.0
   double grade = 0.0;
}
}
```

Menampilkan Data Variabel

Untuk mengeluarkan nilai dari variabel yang diinginkan, kita dapat menggunakan perintah sebagai berikut

System.out.println() atau System.out.print()

Berikut ini adalah contoh program,

```
public class OutputVariable{
  public static void main( String[] args ){
    int value = 10;
    char x;
    x = 'A';

    System.out.println( value );
    System.out.println( "The value of x=" + x );
  }
}
```

System.out.println() vs. System.out.print()

Perhatikan pernyataan tersebut,

```
System.out.print("Hello ");
System.out.print("world!");
```

Hasil → Hello world!

Sekarang perthatikan pernyataan berikut,

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

Pernyataan ini akan menghasilkan output sebagai berikut pada layar,

Hasil →Hello

world!

Operator

Ada beberapa tipe operator diantaranya operator aritmatika, operator relasi, operator logika, dan operator kondisi.

Bahasa Pemrograman Java

Operator Aritmatika

Operator	Penggunaan	Keterangan
+	op1 + op2	Menambahkan op1 dengan op2
*	op1 * op2	Mengalikan op1 dengan op2
/	op1 / op2	Membagi op1 dengan op2
%	op1 % op2	Menghitung sisa dari pembagian op1 dengan op2
-	op1 - op2	Mengurangkan op2 dari op1

Contoh penggunaan Operator

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

          //sedikit angka
          int i = 37;
          int j = 42;
          double x = 27.475;
          double y = 7.22;
          System.out.println("Variable values...");
          System.out.println("     i = " + i);
          System.out.println("     j = " + j);
          System.out.println("     x = " + x);
          System.out.println("     y = " + y); //penjumlahan angka
          System.out.println("Adding...");
          System.out.println("     i + j = " + (i + j));
          System.out.println("     x + y = " + (x + y));

          //pengurangan angka
          System.out.println("Subtracting...");
          System.out.println("     i - j = " + (i - j));
          System.out.println("     x - y = " + (x - y));

          //perkalian angka
          System.out.println("Multiplying...");
          System.out.println("     i * j = " + (i * j));
          System.out.println("     x * y = " + (x * y));
         //pembagian angka
          System.out.println("Dividing...");
          System.out.println("     i / j = " + (i / j));
          System.out.println("     x / y = " + (x / y));

          //menghitung hasil modulus dari pembagian
          System.out.println("Computing the remainder...");
          System.out.println("     i % j = " + (i % j));
          System.out.println("     x % y = " + (x % y));

          //tipe penggabungan
          System.out.println("Mixing tipes...");
          System.out.println("     j + y = " + (j + y));
          System.out.println("     i * x = " + (i * x));
     }
}
```

Operator Increment dan Decrement

Operator	Penggunaan	Keterangan
++	op++	Menambahkan nilai 1 pada op; mengevaluasi nilai op sebelum diincrementasi/ditambahkan
++	++op	Menambahkan nilai 1 pada op; mengevaluasi nilai op setelah diincrementasi/ditambahkan
--	op--	Mengurangkan nilai 1 pada op; mengevaluasi nilai op sebelum didecrementasi/dikurangkan
--	--op	Mengurangkan nilai 1 pada op; mengevaluasi nilai op setelah didecrementasi/dikurangkan

Operator increment dan decrement dapat ditempatkan sebelum atau sesudah operand.

Contoh

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

Atau

```
int i = 10,
int j = 3;
int k = 0;

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

Operator Relasi

Operator	Penggunaan	Keterangan
>	op1 > op2	op1 lebih besar dari op2
>=	op1 >= op2	op1 lebih besar dari atau sama dengan op2
<	op1 < op2	op1 kurang dari op2
<=	op1 <= op2	op1 kurang dari atau sama dengan op2
==	op1 == op2	op1 sama dengan op2
!=	op1 != op2	op1 tidak sama dengan op2

```
public class RelasiDemo
 {
     public static void main(String[] args)  {
     //beberapa nilai
          int i = 37;
          int j = 42;
          int k = 42;
          System.out.println("Nilai variabel...");
          System.out.println("     i = " + i);
          System.out.println("     j = " + j);
          System.out.println("     k = " + k);

    //lebih besar dari
          System.out.println("Lebih besar dari...");
          System.out.println("     i > j = " + (i > j)); //false
          System.out.println("     j > i = " + (j > i)); //true
          System.out.println("     k > j = " + (k > j)); //false

    //lebih besar atau sama dengan
          System.out.println("Lebih besar dari atau sama dengan...");
          System.out.println("     i >= j = " + (i >= j)); //false
          System.out.println("     j >= i = " + (j >= i)); //true
          System.out.println("     k >= j = " + (k >= j)); //true

    //lebih kecil dari
          System.out.println("Lebih kecil dari...");
          System.out.println("     i < j = " + (i < j)); //true
          System.out.println("     j < i = " + (j < i)); //false
          System.out.println("     k < j = " + (k < j)); //false
    //lebih kecil atau sama dengan
          System.out.println("Lebih kecil dari atau sama dengan...");
          System.out.println("     i <= j = " + (i <= j)); //true
          System.out.println("     j <= i = " + (j <= i)); //false
          System.out.println("     k <= j = " + (k <= j)); //true

    //sama dengan
          System.out.println("Sama dengan...");
          System.out.println("     i == j = " + (i == j)); //false
          System.out.println("     k == j = " + (k == j)); //true

    //tidak sama dengan
          System.out.println("Tidak sama dengan...");
          System.out.println("     i != j = " + (i != j)); //true
          System.out.println("     k != j = " + (k != j)); //false


     }
 }
```

Operator logika

Operator logika memiliki satu atau lebih operand boolean yang menghasilkan nilai boolean. Terdapat enam operator logika yaitu: && (logika AND), & (boolean logika AND), || (logika OR), | (boolean logika inclusive OR), ^ (boolean logika exclusive OR), dan ! (logika NOT).

Pernyataan dasar untuk operasi logika adalah,

x1 op x2

Dimana x1, x2 dapat menjadi pernyataan boolean. Variabel atau konstanta, dan op adalah salah satu dari operator &&, &, ||, | atau ^.

Contoh

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

            int   i     = 0;
            int   j     = 10;
            boolean test= false;

            //demonstrasi &&
            test = (i > 10) && (j++ > 9);
            System.out.println(i);
            System.out.println(j);
            System.out.println(test);

            //demonstrasi &
            test = (i > 10) & (j++ > 9);
            System.out.println(i);
            System.out.println(j);
            System.out.println(test);
      }
}
```

Operator Kondisi(?:)

Operatorkondisi **?:** adalah operator ternary. Berarti bahwa operator ini membawa tiga argumen yang membentuk suatu ekspresi bersyarat. Struktur pernyataan yang menggunakan operator kondisi adalah,

exp1?exp2:exp3

Dimana nilai exp1 adalah suatu pernyataan boolean yang memiliki hasil yang salah satunya harus berupa nilai true atau false.

Jika exp1 bernilai true, exp2 merupakan hasil operasi. Jika bernilai false, kemudian exp3 merupakan hasil operasinya.

Contoh

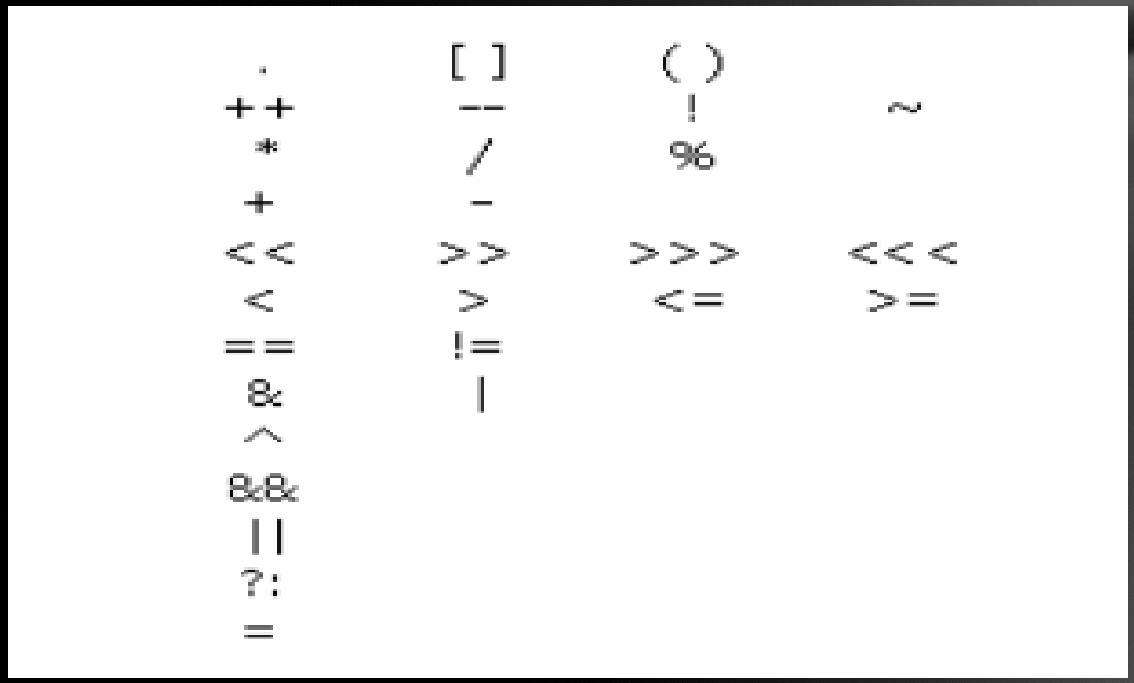
```
public class ConditionalOperator {
public static void main( String[] args ){
   String    status = "";
   int       grade = 80;

   //mendapatkan status pelajar
   status = (grade >= 60)?"Passed":"Fail";

   //print status
   System.out.println( status );
}
}
```

Operator Precedence

Operator precedence didefinisikan sebagai perintah yang dilakukan compiler ketika melakukan evaluasi terhadap operator, untuk mengajukan perintah dengan hasil yang tidak ambigu/ hasil yang jelas.



Diberikan pernyataan yang membingungkan,

```
6%2*5+4/2+88-10
```

Kita dapat menuliskan kembali pernyataan diatas dan menambahkan beberapa tanda kurung terhadap operator precedence,

```
((6%2)*5)+(4/2)+88-10;
```

Latihan

Bahasa Pemrograman Java

Mendeklarasikan dan mencetak variabel

Diberikan tabel dibawah ini, deklarasikan variabel yang terdapat didalamnya dengan tipe data yang sesuai dan berikan nilai inisialisasi. Tampilkan hasil outputnya yaitu nama variabel dan nilainya.

Nama Variabel	*Tipe Data*	*Nilai Awal*
number	integer	10
letter	character	a
result	boolean	true
str	String	hello

- Berikut bentuk tampilannya :

  Number = 10

  letter = a

  result = true

  str = hello

Mendapatkan nilai rata-rata dari tiga angka

Buatlah program yang menghasilkan output nilai rata-rata dari tiga angka. Nilai dari masing-masing tiga angka tersebut adalah 10, 20 dan 45. Tampilan Output yang diharapkan adalah,

number 1 = 10

number 2 = 20

number 3 = 45

Rata-rata = 25

Menampilkan nilai terbesar

Diberikan tiga angka, tuliskan program yang menghasilkan output angka dengan nilai terbesar diantara tiga angka tersebut. Gunakan operator kondisi ?: yang telah kita pelajari sebelumnya (**PETUNJUK**: Anda akan perlu menggunakan dua set operator **?:** untuk memecahkan permasalahan ini). Sebagai contoh, diberikan angka 10, 23 dan 5, Program Anda akan menghasilkan output,

number 1 = 10

number 2 = 23

number 3 = 5

Nilai tertingginya adalah angka = 23