

Dasar-Dasar Pemrograman 2

Lab 2

Java Fundamentals :
Data Types (Strings and Characters),
Conditionals, Loops, and Methods



UNIVERSITAS
INDONESIA
Veritas, Probitas, Justitia

FAKULTAS
ILMU
KOMPUTER

Kamis, 21 Feb 2019 - 1103 - 16:00 WIB

1. Selection & Conditionals

Conditionals pada Java tidak jauh berbeda dengan Python, hanya perihal sintaks saja. Berikut beberapa statement conditional yang ada di Java:

If - Else If - Else Statement

```
int score = 85;

if (score < 55) {
    System.out.println("You got C.");
} else if (score < 70) {
    System.out.println("You got B.");
} else {
    System.out.println("You got A. Congrats!");
}

// prints "You got A. Congrats!"
```

Switch Case Statement

Di Java, ada keyword **switch** yang serupa dengan rentetan conditional.

```
int age = 20;

switch(age) {
    case 2: System.out.println("You're still a baby.");
           break;
    case 15: System.out.println("You're a cool teen.");
            break;
    case 20: System.out.println("You are old enough.");
            break;
    default: System.out.println("Invalid age.");
}

// Prints "You are old enough."
```

Keyword **default** menyatakan aksi yang dilakukan bila variabel yang dikenakan **switch** tidak terdefinisi pada case-case yang ada.

Ternary Operator

Java juga mendukung operasi ternary yang merupakan bentuk “if-else if-else” yang disimplifikasi. Contoh:

```
Scanner scanner = new Scanner(System.in);
int inputNumber = scanner.nextInt();

boolean luckyNumber = inputNumber % 3 == 1 ? false
                    : (inputNumber % 3 == 2 ? false : true)
```

Assignment pada variable luckyNumber di atas serupa dengan:

```
boolean luckyNumber;
if (inputNumber % 3 == 1) {
    luckyNumber = false;
} else if (inputNumber % 3 == 2) {
    luckyNumber = false;
} else {
    luckyNumber = true;
}
```

Note: Penggunaan operator ternary untuk conditional yang rumit dan memiliki banyak cabang tidak disarankan karena akan mengurangi *readability* dari code. Gunakan dengan bijak.

2. Math Functions

Pada java, penggunaan method pada Math tidak perlu diimport karena telah otomatis terimport secara default. Berikut beberapa function yang disediakan:

TABLE 4.1 Trigonometric Methods in the Math Class

Method	Description
<code>sin(radians)</code>	Returns the trigonometric sine of an angle in radians.
<code>cos(radians)</code>	Returns the trigonometric cosine of an angle in radians.
<code>tan(radians)</code>	Returns the trigonometric tangent of an angle in radians.
<code>toRadians(degree)</code>	Returns the angle in radians for the angle in degree.
<code>toDegree(radians)</code>	Returns the angle in degrees for the angle in radians.
<code>asin(a)</code>	Returns the angle in radians for the inverse of sine.
<code>acos(a)</code>	Returns the angle in radians for the inverse of cosine.
<code>atan(a)</code>	Returns the angle in radians for the inverse of tangent.

(Source: Introduction to Java Programming, Comprehensive Version, 10th Edition. Daniel Y. Lang)

TABLE 4.2 Exponent Methods in the Math Class

Method	Description
<code>exp(x)</code>	Returns e raised to power of x (e^x).
<code>log(x)</code>	Returns the natural logarithm of x ($\ln(x) = \log_e(x)$).
<code>log10(x)</code>	Returns the base 10 logarithm of x ($\log_{10}(x)$).
<code>pow(a, b)</code>	Returns a raised to the power of b (a^b).
<code>sqrt(x)</code>	Returns the square root of x (\sqrt{x}) for $x \geq 0$.

(Source: Introduction to Java Programming, Comprehensive Version, 10th Edition. Daniel Y. Lang)

Pemanggilan method menggunakan prefix Math. Contoh: `Math.sqrt(number)`

Selain itu, terdapat constant seperti **Math.PI** dan **Math.E** serta operasi-operasi yang lebih umum seperti:

```
Math.max(2, 3) returns 3
Math.max(2.5, 3) returns 4.0
Math.min(2.5, 4.6) returns 2.5
Math.abs(-2) returns 2
Math.abs(-2.1) returns 2.1
```

(Source: Introduction to Java Programming, Comprehensive Version, 10th Edition. Daniel Y. Lang)

Selengkapnya dapat dilihat di: <https://docs.oracle.com/javase/10/docs/api/java/lang/Math.html>

3. Data Types : Characters and Strings

Characters

Character (**char**) merupakan *primitive data type* yang terdiri dari satu karakter saja. **Char diapit oleh single quote (').**

Char memiliki beberapa method:

<code>isDigit(ch)</code>	Returns true if the specified character is a digit.
<code>isLetter(ch)</code>	Returns true if the specified character is a letter.
<code>isLetterOfDigit(ch)</code>	Returns true if the specified character is a letter or digit.
<code>isLowerCase(ch)</code>	Returns true if the specified character is a lowercase letter.
<code>isUpperCase(ch)</code>	Returns true if the specified character is an uppercase letter.
<code>toLowerCase(ch)</code>	Returns the lowercase of the specified character.
<code>toUpperCase(ch)</code>	Returns the uppercase of the specified character.

(Source: Introduction to Java Programming, Comprehensive Version, 10th Edition. Daniel Y. Lang)

Meski char menyimpan 1 karakter, beberapa karakter khusus seperti *escape sequence* disimbolkan dengan lebih dari 1 karakter, seperti:

TABLE 4.5 Escape Sequences

<i>Escape Sequence</i>	<i>Name</i>
<code>\b</code>	Backspace
<code>\t</code>	Tab
<code>\n</code>	Linefeed
<code>\f</code>	Formfeed
<code>\r</code>	Carriage Return
<code>\\</code>	Backslash
<code>\"</code>	Double Quote

(Source: Introduction to Java Programming, Comprehensive Version, 10th Edition. Daniel Y. Lang)

Karena char merupakan *primitive data type*, maka char dapat dibandingkan dengan operasi `>`, `>=`, `<`, `<=`, `==`, dan `!=`. **Yang dibandingkan dari kedua char bukanlah value dari char, melainkan value dari kode ASCII yang bersesuaian dengan tiap char.** Anda telah mempelajarinya di DDP 1.

Strings

String diartikan sebagai kumpulan karakter. Pada Java, String merupakan salah satu *reference data type* (akan dijelaskan lebih lanjut di sesi lab). Berbeda dengan char, String diapit oleh *double quote* (""). Seperti pada Python, String di Java juga bersifat *immutable*. Berikut beberapa method yang dimiliki oleh object String:

TABLE 4.7 Simple Methods for **String** Objects

<i>Method</i>	<i>Description</i>
length()	Returns the number of characters in this string.
charAt(index)	Returns the character at the specified index from this string.
concat(s1)	Returns a new string that concatenates this string with string s1.
toUpperCase()	Returns a new string with all letters in uppercase.
toLowerCase()	Returns a new string with all letters in lowercase

(Source: Introduction to Java Programming, Comprehensive Version, 10th Edition. Daniel Y. Lang)

Tidak seperti di Python, perbandingan antara value dari dua String tidak bisa dilakukan dengan operasi perbandingan matematis (>, >=, <, <=). Berikut method-method yang disediakan:

TABLE 4.8 Comparison Methods for **String** Objects

<i>Method</i>	<i>Description</i>
equals(s1)	Returns true if this string is equal to string s1.
equalsIgnoreCase(s1)	Returns true if this string is equal to string s1; it is case insensitive.
compareTo(s1)	Returns an integer greater than 0, equal to 0, or less than 0 to indicate whether this string is greater than, equal to, or less than s1.
compareToIgnoreCase(s1)	Same as compareTo except that the comparison is case insensitive.
startsWith(prefix)	Returns true if this string starts with the specified prefix.
endsWith(suffix)	Returns true if this string ends with the specified suffix.

(Source: Introduction to Java Programming, Comprehensive Version, 10th Edition. Daniel Y. Lang)

Tahukah kamu String juga merupakan *non-primitive datatype* yang bisa dibuat sebuah objeknya secara eksplisit, Tapi mengapa setiap membuat suatu string baru bisa saja tidak menggunakan *new String()*? Agar kalian lebih tahu cobalah potongkan kode berikut pada jshell

```
String a = "abc";  
String b = "abc";  
  
System.out.println(a == b);  
  
String c = new String("abc");  
String d = new String("abc");  
  
System.out.println(c == d);
```

(Hint: karena non-primitive berarti objek yang dibuat memiliki reference masing-masing!)

Lebih lengkapnya dapat dilihat di

<https://docs.oracle.com/javase/10/docs/api/java/lang/String.html>

4. Loops

Di Java, terdapat tiga bentuk loop yang umum:

For Loop

Perulangan menggunakan **for** dilakukan ketika kita sudah mengetahui dengan tepat berapa kali perulangan dilakukan. Penulisan perulangan **for** adalah sebagai berikut:

```
for (start; condition; mutation) {  
    // statements  
}
```

Start adalah kondisi pada saat awal perulangan. Biasanya kondisi awal ini digunakan untuk membuat dan memberikan nilai variabel yang digunakan untuk mengontrol perulangan.

Condition adalah kondisi yang harus terpenuhi agar perulangan dapat berjalan. Pengecekan dilakukan SEBELUM menjalankan block statements.

Mutation adalah bagian yang digunakan untuk memproses variabel agar bisa memenuhi kondisi akhir perulangan atau mencegah terjadinya infinite loop.

Contoh:

```
for (int i = 0; i < 5; i++) {
    System.out.print(i);
}

// prints 01234
```

While Loop

Perulangan menggunakan **While** dilakukan ketika kita tidak mengetahui jumlah pasti berapa kali perulangan dilakukan, tetapi kita hanya mengetahui kondisi yang memenuhi perulangan tersebut. Dan pada perulangan **While**, kondisi akan dicek terlebih dahulu sebelum statement dieksekusi. Adapun penulisan sintaksnya adalah:

```
while (condition) {
    // statements
}
```

Condition merupakan kondisi di mana jika semua nilai **true**, maka statements akan dijalankan. Pengecekan terhadap **condition** dilakukan SEBELUM memasuki block statements.

Contoh:

```
int num = 0;

while (num < 10) {
    System.out.print(num);
    num++;
}

// prints 0123456789
```

Do While Loop

Perulangan ini hampir sama dengan perulangan while, namun ketika menggunakan perulangan **Do While**, statement akan dieksekusi terlebih dahulu, baru kemudian di cek apakah kondisi perulangan terpenuhi. Jadi, pada perulangan **Do While**, minimal statement akan dijalankan satu kali. Adapun sintaksnya adalah:

```
do {
    // statements
} while (condition);
```

Condition merupakan kondisi di mana jika menghasilkan nilai **true**, block statements akan dijalankan. Pengecekan dilakukan SETELAH block statements dijalankan.

Contoh:

```
int num = 0;
do {
    System.out.println("I run at least one time!");
} while (num != 0);

// prints "I run at least one time!" one time
```

Untuk ketika bentuk loop di atas, terdapat **loop-exiting command** yang sudah kalian ketahui fungsinya di DDP 1, yaitu **break** (segera keluar dari loop) dan **continue** (*skip* satu iterasi).

5. Methods

Pada DDP 1, kalian telah mengenal konsep *function*, yaitu sebuah *block of code* yang ketika dijalankan/dipanggil, akan mengeksekusi perintah-perintah yang didefinisikan di dalam *function* tersebut. Di bahasa Java, konsep *function* lebih sering dikenal dengan istilah *method*.

Untuk mendeklarasikan sebuah *function/method* pada Python, kalian hanya perlu memanggil *keyword* **def**. Namun pada Java, cara mendeklarasikan sebuah *method* cukup berbeda. Berikut sintaksnya:

```
modifier TipeValueYangDiReturn namaMethod(kumpulan parameter) {
    // Method body, berisi perintah-perintah yang akan dieksekusi
}
```

modifier merupakan pengaturan yang mengatur bagaimana suatu *method* dapat diakses. Salah satu *modifier* yang sudah familiar dengan kalian adalah **public** pada pendefinisian *class* maupun *method* **main**. Ada beberapa jenis *modifier* yang dapat kalian gunakan, namun untuk saat ini, *modifier* yang digunakan adalah **public**.

TipeValueYangDiReturn adalah tipe data yang diharapkan di-*return* oleh *method* ketika *method* tersebut selesai dieksekusi, tipe data yang dapat kalian gunakan adalah tipe data yang

sudah kalian sering temui seperti **int**, **float**, **double**, **long**, **char**, dan **String** maupun tipe data *reference* lainnya (akan kalian pelajari lebih lanjut pada materi OOP).

Note: jika kalian tidak ingin suatu *method* meng-*return* apapun, gunakan *keyword void*.

namaMethod(kumpulan parameter) terdiri dari nama *method* yang diikuti oleh tanda kurung berisi kumpulan parameter yang akan diolah di dalam *method*. Di Python, kalian hanya cukup untuk memberikan nama-nama parameter, namun pada Java, kalian juga harus memberitahu tipe data dari parameter-parameter tersebut.

Di bawah ini adalah contoh *method* yang menerima dua buah parameter bilangan bulat dan meng-*return* hasil kali dari keduanya.

```
public static int multiplyTwoNumbers(int a, int b) {  
    return a * b;  
}
```

Jika kalian perhatikan *method* di atas, terdapat *keyword static* yang menandakan bahwa *method* di atas dapat langsung dieksekusi tanpa harus membuat *instance* dari suatu *class* di mana *method* tersebut didefinisikan (lebih lanjut akan dipelajari pada materi OOP).

Potongan kode di bawah menunjukkan bagaimana cara untuk memanggil *method multiplyTwoNumbers*.

```
int a = 2;  
int b = 3;  
int result = multiplyTwoNumbers(a, b);  
System.out.println(result); // print angka 6
```

Perlu dicatat bahwa tipe data dari variabel yang akan menampung nilai dari hasil *return method* di atas harus sesuai dengan tipe data yang didefinisikan pada *method head*. Perlu diketahui juga bahwa *method* yang tidak meng-*return* apapun (*void method*) tidak dapat di-assign ke suatu variabel.

TUGAS LAB

Jalan-Jalan Bersama Dek Depe

Sebagai ucapan terimakasih karena kamu telah membantu Dek Depe menyelesaikan persoalan minggu lalu, Dek Depe pun mengajak kamu jalan-jalan pada hari Minggu. Setelah menunggu 10 menit di Stasiun UI, Dek Depe pun muncul dengan jaket biru dan kaos merah khasnya. “Maaf ya telat” begitu ujarnya sembari mengajak kamu menaiki kereta. Saat di kereta, Dek Depe berkata:

“Hari ini temani aku belanja di toko QuantaMart ya, aku mau beli banyak barang. Habis itu bantu aku menghitung total barang belanjaan ku ya”

Karena kamu memang orang yang baik hati, kamu pun menyanggupinya. Setelah googling selama 10 menit kamu mencari sebenarnya toko apa sih QuantaMart itu. Ternyata di toko QANT ada **3 kategori barang**, yaitu barang kategori “**lucu**”, “**unik**” dan “**langka**”. Untuk melihat suatu barang ada di dalam suatu kategori ternyata ada aturannya. Menurut websitenya, aturan pengkategorian suatu barang adalah sebagai berikut :

- Setiap karakter dalam nama barang tersebut harus diubah kedalam nilai ASCII nya.
- Kalikan nilai tersebut dengan nilai pi (3,14.....)* lalu dibulatkan ke bawah**.
- Jumlahkan semua nilai karakter yang telah dikalikan dengan pi dan dibulatkan ke bawah tersebut.
- Bagi total perhitungan tadi dengan 15 dan ambil sisanya.

Buat langkah-langkah di atas menjadi ke dalam satu method yang bernama **encode**. Method tersebut menerima sebuah **string** dan mengembalikan bilangan bulat.

Berdasarkan perhitungan yang diperoleh dari method **encode**, terdapat ketentuan berikut :

- Jika hasil perhitungan berkisar antara 0-4 (4 inklusif) maka barang tersebut termasuk barang “**Lucu**”

- Jika hasil perhitungan berkisar antara 5-10 (10 inklusif) maka barang tersebut termasuk barang “**Unik**”
- Jika hasil perhitungan berkisar antara 11-14 (14 inklusif) maka barang tersebut termasuk barang “**Langka**”

Tentu kamu tidak ingin kesulitan dalam mengetahui barang belanjaan Dek Depe, jadi selama di kereta kamu segera membuat program java untuk membantu kamu mengetahui kategori semua barang belanjaan Dek Depe. Tapi **Kamu tidak tahu berapa banyak barang yang akan dibeli Dek Depe**. Jadi program kamu akan terus menerima input nama barang belanjaan Dek Depe sampai Dek Depe bilang “**selesai**”. Setiap nama barang yang diinput akan menghasilkan output berupa “[Input] adalah barang [Kategori]” yang dicetak di layar.

*Gunakan Library Math yang menyimpan nilai Pi... kira kira apa ya ?

**Gunakan Library Math yang berfungsi untuk membulatkan ke-bawah (meng-floor) sebuah bilangan... hmm apa ya?

Pengerjaan Soal

Batasan Soal :

Setiap input dijamin berupa string *alphanumeric* satu baris, dapat mengandung karakter spasi dan dapat terdiri dari lebih dari satu kata.

Contoh Masukan 1

Selamat datang ! Silahkan masukkan nama barang :

Sate Ayam
Ketoprak
Bubur
selesai

*pada contoh masukan baris yang di **Bold** adalah inputan dari user

Contoh Keluaran 1

Sate Ayam adalah barang Unik
 Ketoprak adalah barang Lucu
 Bubur adalah barang Lucu
 Terimakasih sudah belanja di QuantaMart!

Penjelasan Contoh Keluaran 1**1. Sate Ayam**

S (ASCII 83) * PI di-floor	= 260
a (ASCII 97) * PI di-floor	= 304
t (ASCII 116) * PI di-floor	= 364
e (ASCII 101) * PI di-floor	= 317
spasi (ASCII 32) * PI di-floor	= 100
A (ASCII 65) * PI di-floor	= 204
y (ASCII 121) * PI di-floor	= 380
a (ASCII 97) * PI di-floor	= 304
m (ASCII 109) * PI di-floor	= 342

Total jumlah dari perhitungan adalah 2575 dan di-modulo 15 menghasilkan angka 10. String **Sate Ayam** berada di kategori **Barang unik**.

2. Ketoprak

K (ASCII 75) * PI di-floor	= 235
e (ASCII 101) * PI di-floor	= 317
t (ASCII 116) * PI di-floor	= 364
o (ASCII 111) * PI di-floor	= 348
p (ASCII 112) * PI di-floor	= 351
r (ASCII 114) * PI di-floor	= 358
a (ASCII 97) * PI di-floor	= 304
k (ASCII 107) * PI di-floor	= 336

Total jumlah dari perhitungan adalah 2613 dan di-modulo 15 menghasilkan angka 3. String **Ketoprak** berada di kategori **Barang lucu**.

3. Bubur

B (ASCII 66) * PI di-floor	= 207
u (ASCII 117) * PI di-floor	= 367
b (ASCII 98) * PI di-floor	= 307
u (ASCII 117) * PI di-floor	= 367
r (ASCII 114) * PI di-floor	= 358

Total jumlah dari perhitungan adalah 1606 dan di-modulo 15 menghasilkan angka 1. String **Bubur** berada di kategori **Barang lucu**.

Contoh Masukan 2

Selamat datang ! Silahkan masukkan nama barang :

Figma
SHF

Gunpla
Nendoroid
selesai

Contoh Keluaran 2

Figma adalah barang Lucu
SHF adalah barang Lucu
Gunpla adalah barang Unik
Nendoroid adalah barang Unik
Terimakasih sudah belanja di QuantaMart!

Contoh Masukan 3

Selamat datang ! Silahkan masukkan nama barang :

Buku 1
Buku 2
Buku 3
Buku 4
Buku 5
Buku 6
Buku 7
Buku 8
selesai

Contoh Keluaran 3

Buku 1 adalah barang Lucu
Buku 2 adalah barang Lucu
Buku 3 adalah barang Unik
Buku 4 adalah barang Unik
Buku 5 adalah barang Langka
Buku 6 adalah barang Lucu
Buku 7 adalah barang Lucu
Buku 8 adalah barang Unik
Terimakasih sudah belanja di QuantaMart!

Komponen Penilaian :

Komponen	Penjelasan	Bobot
Implementasi fungsi <i>main</i> .	Mengimplementasikan fungsi <i>main</i> dengan benar	25%

	dalam membaca input, memanggil fungsi lain, dan mengeluarkan output.	
Penggunaan tipe data yang benar.	Menggunakan tipe data yang benar pada variabel-variabel dan juga output tiap fungsi.	10%
Penggunaan <i>library</i> Math.	Menggunakan library Math dengan tepat.	10%
Implementasi fungsi <i>encode</i> .	Mengimplementasikan fungsi <i>encode</i> yang sesuai dengan algoritma pada deskripsi soal.	40%
Penerapan Clean Code.	Penulisan program mengikuti kaidah dan konvensi yang telah diajarkan. Program ditulis dengan rapi, terstruktur, dan mudah dibaca serta dipahami.	15%

Deadline :

Kamis, 21 Feb 2019

Pukul **17:20 WIB**

Format Pengumpulan :

Kumpulkan di slot pengumpulan yang telah disediakan di SCellE dengan format :

KategoriBarang.java yang di-compress ke dalam format **zip** dengan format :

[Kode Asdos]_[Nama]_[Kelas]_[NPM]_Lab2.zip

Contoh :

TES_DemoSuremo_X_17069791412_Lab2.zip

Acknowledged Lecturers :

- Laksmi Rahadiani, S.Kom., M.Sc., Ph.D.
- Fariz Darari, S.Kom, M.Sc., Ph.D.
- Ardhi Putra Pratama Hartono S.Kom, M.Sc

Selamat Bekerja !