

Dasar-Dasar Pemrograman 2

Lab 01

Basics of Java



FAKULTAS
ILMU
KOMPUTER

Semester lalu, kita telah mempelajari konsep dasar pemrograman menggunakan bahasa Python. Di DDP 2 ini, kita akan memperdalam pemahaman kita tentang pemrograman melalui konsep-konsep yang menarik seperti pemrograman berorientasi objek dengan menggunakan bahasa pemrograman Java

Tetapi, mungkin di antara kita ada yang penasaran, mengapa kita perlu mempelajari Java?

Jawaban singkatnya, semua aspek yang terdapat di Java adalah Objek. Java merupakan salah satu bahasa pemrograman berbasis objek secara murni. Dengan begitu, Hal ini secara tidak langsung akan mempermudah kita untuk memahami konsep *Object Oriented* yang kita akan pelajari di semester ini. Selain itu, Java juga merupakan salah satu bahasa pemrograman paling populer di dunia. Bahasanya dapat digunakan untuk membuat banyak hal di berbagai *platform*. Patut untuk dipelajari, bukan?

Perbedaan Java dan Python

- Interpreted vs Compiled

Python merupakan salah satu **Interpreted Language**. Saat program dijalankan, komputer akan mengeksekusi satu per satu baris kode yang kita tuliskan. Sehingga, kita bisa langsung saja menjalankan kode tanpa perlu melakukan kompilasi terlebih dahulu.

Sedangkan Java adalah salah satu **Compiled and Interpreted Language**. Artinya, program yang kita buat harus melalui proses kompilasi terlebih dahulu untuk diterjemahkan ke bahasa mesin (*bytecode*), lalu akan di-*interpret* oleh JVM (Java Virtual Machine) sehingga bisa dijalankan oleh komputer.

- Perbedaan *syntax* Java dan Python

Python dan Java memiliki perbedaan dalam hal *syntax*. Tetapi, konsep yang berada di dalamnya adalah sama. Pada dasarnya, setiap bahasa pemrograman bertujuan untuk menjadi penghubung antara manusia yang membuat program dan komputer yang menjalankan program. Jika kita sudah memahami satu bahasa pemrograman, maka akan lebih mudah untuk mempelajari bahasa lainnya.

Berikut adalah perbedaan-perbedaan *syntax* pada Python dan Java:

- **Meng-assign variabel dan deklarasi tipe data**

Variabel di **Python** bersifat *dynamically-typed*, sehingga suatu variabel dapat diisi dengan tipe data apapun, dan dapat diubah juga menjadi tipe data yang berbeda.

Variabel di **Java** bersifat *statically-typed*. Kita perlu mendeklarasikan tipe data dari sebuah variabel secara eksplisit. Variabel tersebut juga hanya bisa diisi nilai dengan tipe data yang sudah kita deklarasikan tadi.

Python

```
# Data Assign
var_int = 3
var_string = "Ngoding Python"
var_bool = True
var_int = "Ini bukan int lagi"
```

Java

```
//Data Assign
int varInt = 3;
String varString = "Ngoding Java";
boolean varBool = true;

//cara assignment variable yang salah
testAssign = 3;
```

Tipe data pada Java antara lain: **int** (bil. bulat), **double** (bil. real), **boolean** (true/false), **String** (kalimat), **char** (satu huruf), dan masih banyak lagi yang akan kita pelajari nanti.

Catatan : Setiap instruksi pada java harus diakhiri titik koma / semicolon (;). Berbeda dengan Python yang setiap instruksinya dipisah oleh baris. (Ya, kita bisa menulis lebih dari satu instruksi pada

baris yang sama -- tetapi tidak dianjurkan karena bisa membuat kode menjadi sulit dibaca oleh manusia.)

Catatan (2) : String di Java wajib menggunakan tanda kutip (" ... "). Tidak seperti Python yang bisa menggunakan tanda kutip maupun apostrof (' ... '). Pada Java, tanda kutip digunakan untuk tipe data *char*.

- **Operasi And-Or**

Pada Python, kita menggunakan `and` untuk operasi and dan `or` untuk operasi or. Pada Java, kita dapat menggunakan `&&` untuk operasi and dan `||` untuk operasi or jika yang dioperasikan adalah boolean (*logical operations*). Mengapa memerlukan dua buah simbol `&` dan bukan satu saja? Hal ini karena simbol `&` dan `|` adalah simbol lain untuk operasi *bitwise* (seperti di Python), di mana operasi ini dapat mengoperasikan boolean (bekerja seperti *logical*) dan int (melakukan operasi satu per satu terhadap representasi biner dari int). Perbedaan `&` dan `|` saat digunakan sebagai boolean operator dengan `&&` dan `||` dapat dilihat pada tabel berikut.

exprA [operator] exprB			
		&	&&
Mengevaluasi exprA dan exprB, kemudian melakukan operasi OR	Jika exprA bernilai true, maka exprB tidak perlu dievaluasi (akan mengembalikan nilai true). Operasi OR akan mengembalikan true jika salah satu operand bernilai true	Mengevaluasi exprA dan exprB, kemudian melakukan operasi AND	Jika exprA bernilai false, maka exprB tidak perlu dievaluasi (akan mengembalikan nilai false). Operasi AND akan mengembalikan false jika salah satu operand bernilai false

Contoh:

```
jshell> int i=5;
i ==> 5

jshell> int j=4;
j ==> 4

jshell> i<10 || j++<10
$7 ==> true

jshell>

jshell> j
j ==> 4

jshell> i<10 | j++<10
$9 ==> true

jshell> j
j ==> 5
```

Terlihat saat menggunakan `|` sebagai operator OR, nilai `j` bertambah karena ekspresi `j++` ikut terevaluasi.

Python

```
# Penggunaan Operasi and, or pada Python
a = True
b = True
if a and b :
    print("True AND True")
a = True
b = False
if a or b :
    print("True OR False")

print(10 & 5)
print(10 | 5)
```

Java

```
//Penggunaan Operasi and, or pada Java
boolean a = true;
boolean b = true;
if (a && b) {
    System.out.println("true AND true");
}
a = true;
b = false;
if (a || b) {
    System.out.println("true OR false");
}

//Operasi bitwise juga ada di Java
System.out.println(10 & 5);
System.out.println(10 | 5);
```

- **Equals**

Sama seperti Python, jika kita ingin mengetahui apakah suatu nilai bertipe data primitif sama atau tidak sama dengan nilai bertipe data primitif lainnya (misal: `int` dengan `int`), kita dapat menggunakan `==`. Namun, jika kita ingin membandingkan suatu string dengan string yang lain, kita harus menggunakan method `.equals(...)` di Java. Hal ini karena String bukan tipe data primitif di Java. Menggunakan `==` pada variabel bertipe non-primitif akan membandingkan *reference*-nya (sama seperti `is` pada Python).

Python

```
var_1 = 2
var2 = 3
if var_1 == var_2 :
    print("Kedua nilai sama")

test_1 = "DDP2"
test_2 = "DDP2"
if test_1 == test_2 :
    print("Kedua string sama")
```

Java

```
//Penggunaan Operasi "==" dan .equals
int var1 = 2;
int var2 = 3;
if (var1 == var2){
    System.out.println("Kedua nilai sama");
}

String test1 = "DDP2";
String test2 = "DDP2";
if (test1.equals(test2)){
    System.out.println("Kedua string sama");
}
```

- **If-Else Conditions**

Struktur kode If-Else pada Python dan Java kurang lebih sama. Perbedaannya adalah Java menggunakan **else if** sedangkan Python menggunakan **elif**. Selain itu, Java memiliki **Switch-case statement** yang berguna untuk membandingkan suatu nilai / variabel dengan case tertentu ([referensi](#)).

Python

```
# Sistem Grading SMA
nilai = 77
grade = ""
if nilai >= 90:
    grade = "A"
elif nilai >= 80:
    grade = "B"
elif nilai >= 75:
    grade = "C"
else:
    grade = "Belajar lagi ya :)"
print(grade)

if (grade == 'A'):
    print('Tetap semangat')
elif (grade == 'B'):
    print('Terus berusaha')
```

```
elif (grade == 'C'):  
    print('Jangan menyerah')  
else:  
    print('Kamu pasti bisa')
```

Java

```
//Sistem Grading SMA  
int nilai = 77;  
String grade;  
if (nilai >= 90){  
    grade = "A";  
} else if (nilai >= 80){  
    grade = "B";  
} else if (nilai >= 75){  
    grade = "C";  
} else {  
    grade = "Belajar lagi ya :)";  
}  
System.out.println(grade) ;  
  
// Java memiliki switch case statement  
switch (grade) {  
    case "A":  
        System.out.println("Tetap semangat");  
        break;  
    case "B":  
        System.out.println("Terus berusaha");  
        break;  
    case "C":  
        System.out.println("Jangan menyerah");  
        break;  
    default:  
        System.out.println("Kamu pasti bisa");  
        break;  
}
```

- **Input**

Dapat dilihat pada kode di bawah, cara pemanggilan input pada Java memiliki bentuk yang jauh lebih kompleks dibandingkan Python. Pada Java, kita perlu menggunakan objek **Scanner** untuk membaca input. Untuk dapat menggunakan **Scanner**, yang harus dilakukan pertama kali adalah meng-*import* `java.util.Scanner`. Kemudian, kita perlu membuat object **Scanner** untuk membaca input yang dan disimpan ke dalam variabel. Objek **Scanner** sendiri, memiliki berbagai method untuk membaca input berdasarkan tipe data. Berikut method yang sering digunakan:

Method	Penjelasan
<code>(*var).nextInt()</code>	Membaca input bertipe int
<code>(*var).nextLong()</code>	Membaca input bertipe long
<code>(*var).nextFloat()</code>	Membaca input bertipe float
<code>(*var).nextDouble()</code>	Membaca input bertipe double
<code>(*var).nextBoolean()</code>	Membaca input bertipe boolean
<code>(*var).nextLine()</code>	Membaca input satu baris dalam bentuk string

Tabel di atas merupakan method yang sering digunakan. Untuk melihat method lebih lengkap, silahkan buka referensi berikut: [Scanner Method](#).

Python

```
# INPUT
angka = input("Masukkan angka : ")
print(angka)
```

Java

```
//INPUT
import java.util.Scanner;
Scanner input = new Scanner(System.in);

System.out.print("Masukkan Angka : ");

int angka = input.nextInt();
System.out.println(angka);
```


- **For-Loop**

Struktur *syntax* for-loop pada Java dan Python sedikit berbeda. Tidak seperti Python yang cara *looping* menggunakan *range*, di Java kita dapat dan harus memberikan perintah yang lebih spesifik dalam menginisiasi *loop* itu sendiri seperti langkah *looping*-nya bagaimana, seberapa banyak *looping* yang akan dilakukan, serta apa yang akan dilakukan setiap kali selesai iterasi.

Python

```
# print angka 0-9
for i in range(0,10):
    print(i)
```

Java

```
//Print angka 0-9
for (int i = 0; i < 10; i++){
    System.out.println(i);
}
```

Note : *For loop pada Java dan Python sangatlah mirip, hanya penulisan untuk mengakses setiap element sajalah yang berbeda di mana Python menggunakan “in” dan Java menggunakan titik dua (:).*

- **While-Loop**

Struktur *While-Loop* pada Python dan Java sangatlah identik. Perbedaannya hanyalah pada penulisan kondisinya. pada Java, penulisan kondisi harus selalu ditutup oleh tanda kurung **while(kondisi)**, sedangkan Python tidak diharuskan.

Python

```
# Print Angka Genap di range 0-10
angka = 0
while angka <= 10:
    print(angka)
    angka += 2
```

Java

```
//Print Angka Genap di range 0-10
int angka = 0;
while (angka <= 10) {
    System.out.println(angka);
    angka += 2;
}
```

Membuat Program Java

Berikut adalah contoh program “Hello world” menggunakan Java:

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

File program java ini bernama `HelloWorld.java`. Kita perlu menyimpannya dengan ekstensi `.java` karena ekstensi ini memberitahu komputer bahwa ini adalah sebuah program yang dapat di-*compile* oleh Java Compiler.

Lalu, kita membuat sebuah public class. Perhatikan bahwa **nama dari public class yang berada dalam file tersebut harus sama dengan nama filenya** ([referensi](#)). Di dalam class tersebut kita membuat sebuah *method* bernama “main”. Method main berfungsi sebagai *entry-point* ketika program java dijalankan.

Setelah itu, kita menggunakan `System.out.println()` untuk mencetak output ke layar. Hal ini sama seperti penggunaan `print()` di Python. Jangan lupa untuk menggunakan titik koma / *semicolon* (`;`) di setiap akhir statement atau kita akan mendapatkan *compile error*.

Untuk menjalankan program (merujuk ke “Lab 00 - DDP-2 Starter Pack”), pertama save file terlebih dahulu, lalu buka Command Prompt / Terminal di lokasi peletakan file java yang ingin kita jalankan. Kemudian, jalankan perintah **`javac HelloWorld.java`** untuk melakukan *compilation* terhadap file java tersebut (trivia: `javac` merupakan singkatan dari “Java Compiler”).

Java Compiler akan melakukan kompilasi dan menghasilkan *bytecode* dengan ekstensi `.class` yang dapat kita jalankan dengan menggunakan perintah **java HelloWorld**. Perhatikan bahwa tidak ada ekstensi `.java` saat menjalankan program.

Selamat, kamu sudah berhasil menjalankan program “Hello World” di Java!

(See the next page to do the task...)

Soal Lab 01

1. Sydney Sudah Besar!



Sydney adalah seekor anjing Shih Tzu yang memiliki banyak teman. Tahun ini, Sydney akan merayakan ulang tahunnya yang ke-6. Namun, karena pandemi, Sydney terpaksa merayakannya secara daring. Sebagai teman yang baik, Dek Depe ingin memberikan hadiah yang spektakuler untuk Sydney. Mengingat Dek Depe adalah *programmer* yang handal, ia ingin memberikan hadiah berupa program di mana teman-teman Sydney bisa memberikan ucapan selamat ulang tahun untuknya. Berikut adalah spesifikasi program yang telah dibuat oleh Dek Depe:

- Program menerima tiga input yaitu **tanggal dan bulan ketika pesan disampaikan serta ucapan ulang tahun**.
- Ucapan yang diterima adalah **“happy birthday”**, ucapan lain tidak akan diterima.
- Ucapan hanya akan diterima Sydney jika disampaikan di hari ulang tahunnya yaitu **10 juli**. Program juga dapat mengatasi apabila tanggal dan bulan yang dimasukkan salah.

Sayangnya, Sydney hanya memahami bahasa pemrograman Java dan Dek Depe hanya mahir dalam bahasa Python. Karena kamu sedang mempelajari DDP2, bantulah Dek Depe untuk **mengkonversi program yang telah ia buat ke dalam Java!**

Program ucapan.py

```
day = int(input("Masukkan tanggal: "))
month = input("Masukkan bulan: ")
message = input("Masukkan pesan: ")

correct_message = False
correct_date = False

if day == 10 and month == "juli":
    correct_date = True

if message == "happy birthday":
    correct_message = True

if correct_date and correct_message:
    print("Terima kasih")
elif month == "juli" and correct_message:
    print("Bulannya benar, tanggalnya salah")
elif day == 10 and correct_message:
    print("Tanggalnya benar, bulannya salah")
elif correct_message:
    print("Terima kasih tapi ini bukan ulang tahun aku :D")
else:
    print("Biasalah :v")
```

Contoh Masukan 1

13
juli
happy birthday

Contoh Keluaran 1

Bulannya benar, tanggalnya salah

Contoh Masukan 2

10
mei
sayasenangddp2

Contoh Keluaran 2

Biasalah :v

Contoh Masukan 3

1
desember
happy birthday

Contoh Keluaran 3

Terima kasih tapi ini bukan ulang tahun aku :D

2. Sydney Pizza Place

Setelah merayakan ulang tahun Sydney yang ke-6, Sydney sadar bahwa ternyata ia sudah cukup dewasa (1 tahun umur anjing sama dengan 7 kali lipat umur manusia). Ia merasa harus mulai mencari nafkah untuk menghidupi dirinya sendiri. Dari kesadarannya, ia memutuskan untuk membuka bisnis Pizza.

Harga satuan pizza yang dijual Sydney adalah **Rp 30.000**. Untuk menarik perhatian para pembeli dengan bisnis barunya, ia pun memberi diskon besar-besaran serta minuman gratis sebagai menu pendamping untuk pizza yang telah dibeli. Sistem yang diinginkan Sydney adalah:

- Untuk setiap pembelian **n pizza**, pembeli akan mendapatkan diskon sebesar **$(n + n-1 + n-2 + \dots + 1)\%$** dengan **$n > 0$** .
- Menampilkan harga pizza yang harus dibayar setelah dikenakan diskon.
- Maksimal diskon yang bisa diberikan pada pembeli adalah sebesar **70%**.
- Minuman gratis tertentu akan diberikan jika jumlah diskon mod 7 menghasilkan 0, 2, 4, 5, atau 6.
 - Jika jumlah diskon mod 7 = 0, pelanggan mendapatkan Cola
 - Jika jumlah diskon mod 7 = 2, pelanggan mendapatkan *lemon tea*
 - Jika jumlah diskon mod 7 = 4, pelanggan mendapatkan es teh manis
 - Jika jumlah diskon mod 7 = 5, pelanggan mendapatkan teh tawar hangat
 - Jika jumlah diskon mod 7 = 6, pelanggan mendapatkan air matang

Contoh:

- a. Dek Depe membeli **3** pizza, berarti Dek Depe akan mendapatkan diskon sebesar **$3+2+1 = 6\%$** .

$$6 \bmod 7 = 6$$

Maka, Dek Depe akan mendapatkan **air matang**.

- b. Dek Depe membeli 12 pizza, berarti Dek Depe akan mendapatkan diskon sebesar **$12+11+10+9+8+7+6+5+4+3+2+1 = 78\%$** . Oleh karena 78 sudah **lebih besar** dari 70, maka Dek Depe hanya akan mendapatkan diskon sebesar **70%**.

$$70 \bmod 7 = 0$$

Maka, Dek Depe akan mendapatkan **Cola**.

Seperti yang telah kita ketahui bahwa Sydney hanya memahami bahasa pemrograman Java, **bantulah Dek Depe menerjemahkan program yang sudah ia buat dalam Python ke dalam bentuk Java!**

Program pizza.py

```
jumlah = int(input("Masukkan jumlah pizza yang ingin dibeli: "))

harga_pizza = 30000
jumlah_diskon = 0

for i in range(1, jumlah+1):
    jumlah_diskon += i

    if jumlah_diskon >= 70:
        jumlah_diskon = 70
        break

potongan_diskon = (harga_pizza*jumlah)*jumlah_diskon/100
harga_setelah_diskon = (harga_pizza*jumlah) - potongan_diskon

print("Anda mendapatkan diskon " + str(jumlah_diskon) + "%")
print("Yang harus kamu bayar sebesar: Rp " + str(harga_setelah_diskon))

if jumlah_diskon % 7 == 0:
    print("Cola satu liter gratis untuk kamu!")
elif jumlah_diskon % 7 == 2:
    print("Lemon tea dua gelas gratis untuk kamu!")
elif jumlah_diskon % 7 == 4:
    print("Es teh manis gratis untuk kamu!")
elif jumlah_diskon % 7 == 5:
    print("Teh tawar hangat gratis untuk kamu!")
elif jumlah_diskon % 7 == 6:
    print("Air matang gratis untuk kamu!")
else:
    print("Maaf kamu terpaksa seret..")
```

Notes: Kamu boleh menggunakan **switch statements** untuk masalah di atas.

Contoh Masukan 1

10

Contoh Keluaran 1

Anda mendapatkan diskon 55%
Yang harus kamu bayar sebesar: Rp 135000
Air matang gratis untuk kamu!

Contoh Masukan 2

100

Contoh Keluaran 2

Anda mendapatkan diskon 70%
Yang harus kamu bayar sebesar: Rp 900000
Cola satu liter gratis untuk kamu!

Catatan:

- Untuk tutorial kali ini, asumsikan tipe data input sudah sesuai sehingga Anda **tidak perlu memvalidasi input** dari *user*. Misalkan tipe data input yang diminta adalah integer, maka diasumsikan *user* tidak akan memasukan input bertipe data lainnya.
- Anda sangat disarankan untuk **menamakan variabel sesuai konvensi penamaan dalam Java**. Pada Java, konvensi penamaan variabel menggunakan aturan **camelCase**. Untuk penamaan *class*, Java menggunakan aturan **PascalCase**.

Komponen Penilaian

- **15%** Gaya penulisan sesuai Java *code style* (penamaan *class*, *variable*, dan sebagainya)
- **45%** Translasi *syntax* Python ke Java
- **30%** Kebenaran program (*output* sesuai dengan ekspektasi berdasarkan *input*)
- **10%** Kerapian kode

Keterangan Revisi

a. Revisi 1

- ... 10 **Juli** → ...10 **juli**
- ... membeli **6** pizza → ... membeli **3** pizza

Kumpulkan berkas .java yang telah di-zip dengan format penamaan seperti berikut.

Lab01_[Kelas]_[KodeAsdos]_[NPM]_[NamaLengkap].zip

Contoh:

Lab01_A_LN_1234567890_DekDepe.zip

*Notes:

- Submisi akan dibuka pada tanggal 2 Maret 2021 pukul 16.00
- Jika anda belum memiliki kode asdos, maka tulis kode asdos dengan **XX**