

Lab 3

Data Types dan Control Flow

Dasar-Dasar Pemrograman 2

CSGE601021

Semester Genap 2016/2017

Batas waktu pengumpulan:

Senin, 27 Februari 2017 pukul 18.00 Waktu Scele (Lab Senin)

Tujuan dari Lab ini adalah melatih Anda agar menguasai bahan kuliah yang diajarkan di kelas. Mahasiswa diperbolehkan untuk berdiskusi, tetapi Anda tetap harus **menuliskan sendiri** solusi/kode program dari soal yang diberikan tanpa bantuan orang lain. Belajarlah menjadi mahasiswa yang mematuhi integritas akademik. **Sikap Jujur merupakan sebuah sikap yang dimiliki mahasiswa Fasilkom UI.**

Peringatan: Jangan mengumpulkan pekerjaan beberapa menit menjelang batas waktu pengumpulan karena ada kemungkinan pengumpulan gagal dilakukan atau koneksi internet terputus!

Lab 3

DATA TYPE, CONTROL FLOW

Data Type

Tipe data merupakan suatu klasifikasi yang menentukan nilai apa yang dapat di assign ke dalam suatu variabel, serta apa saja operasi matematika, relasi, dan logika yang dapat diaplikasikan terhadap variabel tersebut tanpa menimbulkan error.

Berbeda dengan bahasa python, di Java, saat mendeklarasikan sebuah variabel, kita harus secara eksplisit mendefinisikan tipe data dari variabel tersebut.

Python	Java
<pre>aString = "ini kalimat" anInt = 1 aFloat = 1.0</pre>	<pre>String aString = "ini kalimat"; int anInt = 1; double aDouble = 1.0;</pre>

Umumnya, di setiap bahasa pemrograman memiliki 3 kateogri tipe data. Yaitu

1. Primitive Data Types

Merupakan tipe data yang hanya memiliki maksimum 1 nilai saja. Tipe data ini tidak bisa memiliki lebih dari satu nilai. Ada 8 macam tipe data primitive, yaitu :

Data Types	Default Values	Default Size	Example
boolean	False	1 bit	False / True
char	'\u0000'	2 byte	'a'
byte	0	1 byte	0
short	0	2 byte	0
int	0	4 byte	0
long	0	8 byte	0
float	0.0f	4 byte	0.0f
double	0.0d	8 byte	0.0

Di Java, kita dapat melakukan **type casting**, yaitu assign nilai dari suatu tipe data ke dalam variabel dengan tipe data berbeda. Ada 2 macam **type casting**, yaitu:

1. Widening (implicit)



(<http://www.studytonight.com/java/type-casting-in-java>)

2. Narrowing (explicitly done)



(<http://www.studytonight.com/java/type-casting-in-java>)

Widening	Narrowing
<pre>int anInt = 10; long aLong = anInt; float aFloat = aLong; //jika diprint // anInt = 10 // aLong = 10 // aFloat = 10.0</pre>	<pre>double aDouble = 10.1; long aLong = (long) aDouble; int anInt = (int) aLong; //jika diprint // aDouble = 10.1 // aLong = 10 // anInt = 10</pre>

2. Derived Data Types

Merupakan tipe data yang dapat memiliki lebih dari 1 nilai untuk setiap variabelnya. Contohnya array dan String.

Di Java terdapat teknik **parsing**, yaitu mengubah tipe data String menjadi tipe data primitive, dan juga sebaliknya.

```
String iString = "10";
int i = Integer.parseInt(iString); //i = 10

String bString = "False";
boolean b = Boolean.parseBoolean(bString); //b = False
```

3. User Defined Data Types

Tipe data yang didefinisikan sendiri oleh programmer.

Control Flow

Suatu program pada umumnya dieksekusi secara berurutan dari baris teratas ke baris-baris berikutnya. Control flow merupakan salah satu perintah pada bahasa pemrograman untuk mengubah urutan eksekusi program sesuai kebutuhan. Pada bahasa pemrograman pada umumnya terdapat 3 jenis perintah control flow, yaitu:

1. Decision-making (if-else)

Sama seperti pada bahasa pemrograman python, Java juga memiliki perintah untuk melakukan decision making, yaitu dengan if, else if, dan else. Urutan pengecekan kondisi pada decision making dilakukan secara berurutan mulai dari if lalu ke semua else if berikutnya. Jika seluruh kondisi if dan else if tidak memenuhi, perintah pada else akan dijalankan jika ada, sama seperti python.

Perbedaan decision-making pada Java dan Python hanyalah pada syntax yang digunakan.

Python	Java
<pre>if a > 0: print("a bil positif") elif a < 0: print("a bil negatif") else: print("a bernilai 0")</pre>	<pre>if(a > 0) { System.out.println("a bil positif"); } else if(a < 0) { System.out.println("a bil negatif"); } else { System.out.println("a bernilai 0"); }</pre>

2. Looping (for-while)

Sama seperti pada bahasa pemrograman python, Java juga memiliki perintah untuk melakukan looping atau iterasi.

Pada python, cara melakukan iterasi adalah dengan perintah for each atau while. Pada java, terdapat perintah tambahan untuk melakukan iterasi, yaitu dengan for loop dan do while. Berikut merupakan contoh penggunaan iterasi pada Java dan python.

Jenis iterasi	Python	Java
for each	<pre>for i in range(4): print(i)</pre>	<pre>int[] arr = {0,1,2,3}; for(int i:arr) { System.out.println(i); }</pre>

while	<pre>while True: print("iterasi")</pre>	<pre>while(true) { System.out.println("iterasi"); }</pre>
for loop	-	<pre>for(int i = 0; i < 4; i++) { System.out.println(i); }</pre>
do while	-	<pre>do { System.out.println("iterasi"); } while(true);</pre>

3. Branching Statements (break - continue - return)

Statement break digunakan untuk keluar dari loop yang sedang dieksekusi, continue digunakan untuk meloncati iterasi, sedangkan return digunakan untuk keluar dari method yang sedang dijalankan. Berikut contoh penggunaan branching statement pada Java dan python.

Jenis Iterasi	Python	Java
break	<pre>for i in range(4): if i == 2: break print(i) # keluaran: # 0 # 1</pre>	<pre>int[] arr = {0,1,2,3}; for(int i:arr) { if(i == 2) { break; } System.out.println(i); } /* Keluaran: 0 1 */</pre>
continue	<pre>for i in range(4): if i == 2: continue print(i) # keluaran: # 0 # 1 # 3</pre>	<pre>int[] arr = {0,1,2,3}; for(int i:arr) { if(i == 2) { continue; } System.out.println(i); } /* Keluaran: 0 1 3 */</pre>
return	<pre>def satu(): return 1 print(satu()) # 1</pre>	<pre>public static int satu() { return 1; } public static void main(String[] args) { System.out.println(satu()); // 1 }</pre>

Soal A

Internship Interview

Agung adalah seorang karyawan HRD (Human Resource and Development) dari perusahaan iSUS. Perusahaan iSUS adalah sebuah perusahaan yang bergerak di bidang game dan teknologi dengan produk unggulannya adalah game online yang sangat sering dimainkan oleh banyak orang, DATA2. Sebagai perusahaan yang sangat besar dan berkembang pesat karena banyaknya peminat orang yang ingin bermain DATA2, maka iSUS membuka lowongan untuk mahasiswa Fasilkom yang “*jago ngoding*” untuk magang di kantornya.

Dalam melakukan interview dengan calon peserta magang, Agung merasa kesulitan karena harus menanyakan mengenai data diri calon pelamar dan itu sangat membosankan! (Bayangkan jika kamu harus menanyakan pertanyaan yang sama terhadap puluhan orang dalam satu hari!). Oleh karena itu, Agung berencana untuk membuat sebuah program sederhana yang akan membantu dia dalam melakukan tugas wawancara ini.

Program sederhana yang akan dibuat oleh Agung akan menerima masukan dari interviewee berupa jawaban dari pertanyaan seputar data diri, namun harus dipastikan bahwa data yang dimasukkan adalah data yang **benar** dan **tepat, sesuai** dengan **tipe datanya**.

Berikut adalah contoh interaksi program (yang digaris bawahi adalah masukan):

Program Wawancara Perusahaan iSUS

Nama : Omprakash

Tempat/Tanggal Lahir : Mumbai/30-11-1997

NIK : 3511113011970003

Tinggi : 164

Berat Badan : 50

Golongan Darah : B

Pengalaman : Menjaga kura-kura

Pendidikan saat ini : Universitas Indonesia

Perkiraan lulus : 2019

Jomblo : True

DATA TERSIMPAN

Omprakash - 19 Tahun - BMI : 18.5901

Asal Mumbai Bergolongan darah B

Lulus dalam 2 tahun dari Universitas Indonesia

Berpengalaman Menjaga kura-kura

Belum mempunyai pasangan hidup :(

Contoh lain dari interaksi yang dapat terjadi adalah:

```
Program Wawancara Perusahaan iSUS
-----
Nama : Sumiyati
Tempat/Tanggal Lahir : Denpasar/21-01-1997
NIK : 3511012111970003
Tinggi : 174
Berat Badan : 62
Golongan Darah : O
Pengalaman : Competitive Eating
Pendidikan saat ini : Institut Teknologi Bandung
Perkiraan lulus : 2018
Jomblo : False

DATA TERSIMPAN
-----
Sumiyati - 20 Tahun - BMI : 20.4783
Asal Denpasar Bergolongan darah O
Lulus dalam 1 tahun dari Institut Teknologi Bandung
Berpengalaman Competitive Eating
Sudah mempunyai pasangan hidup :)
```

Format Output adalah sebagai berikut:

```
DATA TERSIMPAN
-----
[Nama] - [Umur] Tahun - BMI : [BMI]
Asal [Tempat Lahir] Bergolongan darah [Golongan Darah]
Lulus dalam [Perkiraan Lulus] tahun dari [Universitas]
Berpengalaman [Pengalaman]
[Sudah/Belum] mempunyai pasangan hidup :[)]/([)
```

Adapun catatan yang harus diperhatikan adalah sebagai berikut:

- Buatlah **SESUAI** dengan contoh format yang disediakan.
- Nilai dari :
 - Berat badan antara 30 hingga 80 (kg)
 - Tinggi antara 150 hingga 200 (cm)
 - Dijamin umur tidak akan melebihi 100 tahun.
- Format untuk tanggal lahir adalah DD-MM-YYYY
- Golongan darah adalah A, B, AB, dan O
- Rumus menghitung BMI adalah $\frac{\text{berat badan (kg)}}{(\text{tinggi badan} * \text{tinggi badan})(m^2)}$

- Dalam menghitung umur, asumsikan tanggal yang digunakan adalah tanggal hari ini (hint: kalian dapat menggunakan bantuan kelas Date¹)
- Pastikan anda menggunakan **tipe data yang tepat** untuk menyimpan nilai masukan yang diberikan oleh pengguna. (hint: perhatikan perbedaan byte, word, int, float, double, dsb.)
- Pelajari contoh program berikut untuk memahami pembacaan masukan dan pencetakan keluaran.

```
import java.util.Scanner;

public class Lab3a {
    public static void main (String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Siapa namamu? ");
        String nama = in.nextLine();
        System.out.println("Hai " + nama + ", umurmu berapa?");
        int umur = in.nextInt();
        System.out.println("Umur : " + umur + " tahun");
    }
}
```

¹ <https://docs.oracle.com/javase/8/docs/api/java/sql/Date.html>

Soal B

Interview Result

Setelah melakukan interview beberapa kali, Agung berencana membuat program yang dia miliki menjadi lebih “*pintar*” sehingga dapat melakukan prediksi apakah seorang interviewee memenuhi syarat yang diperlukan perusahaan iSUS untuk dapat bergabung atau tidak. Hal ini didasari oleh keinginan CEO dari iSUS yang lumayan cerewet dalam menerima karyawan. Adapula kebutuhan dari seorang karyawan iSUS adalah harus memenuhi syarat 3 dari 5 diantara tabel berikut:

Syarat	Keterangan
Nama	Nama pelamar haruslah memiliki jumlah karakter ganjil
BMI	Pelamar dengan BMI normal (18-20) lah yang akan diterima
Golongan Darah	Golongan darah pelamar harus AB
Perkiraan Lulus	Pelamar harus berada dalam tahun terakhir di bangku perkuliahan untuk dapat magang
Jomblo	Pelamar yang memiliki pasangan akan diterima
Nilai iSUS	Nilai iSUS = $(\text{umur}^2 \times \text{berat} \times \text{tinggi}^2)$ Nb: tinggi dalam cm, berat dalam kg

Berikut adalah contoh interaksi program (yang digaris bawahhi adalah masukan pengguna):

```
Program Wawancara Perusahaan iSUS
-----
Nama : Omprakash
Tempat/Tanggal Lahir : Mumbai/30-11-1997
NIK : 3511113011970003
Tinggi : 164
Berat Badan : 50
Golongan Darah : B
Pengalaman : Menjaga kura-kura
Pendidikan saat ini : Universitas Indonesia
Perkiraan lulus : 2019
Jomblo : True

DATA TERSIMPAN
-----
Omprakash - 19 Tahun - BMI : 18.5901
```

Asal Mumbai Bergolongan darah B
Lulus dalam 2 tahun dari Universitas Indonesia
Berpengalaman Menjaga kura-kura
Belum mempunyai pasangan hidup :(

PREDIKSI

Nilai iSUS 485472800
Memenuhi 2 dari 5 syarat
Omprakash ditolak

Contoh lain dari interaksi yang dapat terjadi adalah:

Program Wawancara Perusahaan iSUS

Nama : Sumiyati
Tempat/Tanggal Lahir : Denpasar/21-01-1997
NIK : 3511012111970003
Tinggi : 174
Berat Badan : 62
Golongan Darah : 0
Pengalaman : Competitive Eating
Pendidikan saat ini : Institut Teknologi Bandung
Perkiraan lulus : 2018
Jomblo : False

DATA TERSIMPAN

Sumiyati - 20 Tahun - BMI : 20.4783
Asal Denpasar Bergolongan darah 0
Lulus dalam 1 tahun dari Institut Teknologi Bandung
Berpengalaman Competitive Eating
Sudah mempunyai pasangan hidup :)

PREDIKSI

Nilai iSUS 750844800
Memenuhi 4 dari 5 syarat
Sumiyati diterima

Format Output adalah sebagai berikut:

```
PREDIKSI
```

```
-----
```

```
Nilai iSUS [Nilai iSUS]
```

```
Memenuhi [Jumlah Terpenuhi] dari 5 syarat
```

```
[Nama] [diterima/ditolak]
```

Format Pengumpulan

- KELAS_NPM_TUTORIAL3.zip contoh: B_1606123456_TUTORIAL2.zip
- Isi file KELAS_NPM_TUTORIAL3.zip:
 - Internship.java
 - Result.java