

PRAKTIKUM PERTEMUAN 3

Disusun untuk memenuhi tugas mata kuliah Pemrograman Berorientasi Objek (Praktek)



Disusun oleh:

Bandyaga Adiansyah Sugandi

NIM 231511037

2B – D3

Jurusan Teknik Komputer dan Informatika

Politeknik Negeri Bandung

2024

1. SOAL 1

Screenshot hasil akhir

```
"C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA
Masukkan sebuah kalimat: I wasn't home
4
I
wasn
t
home
Process finished with exit code 0
```

Pembahasan

Program ini adalah program untuk menginput sebuah kalimat, kemudian memisahkan kata-kata dalam kalimat tersebut berdasarkan karakter yang bukan huruf (misalnya spasi atau tanda baca). Setelah itu, program menghitung berapa banyak kata yang ada di dalam kalimat tersebut dan menampilkannya. Selain itu, setiap kata yang ditemukan juga akan diprint di line berikutnya menggunakan `println`.

Untuk memisahkan kata, program menggunakan metode `split()` dengan ekspresi reguler atau **regex** `[^A-Za-z]+`, yang berarti memisahkan berdasarkan karakter selain huruf (baik huruf besar maupun kecil). Program juga menghitung jumlah kata yang valid (bukan kosong) dengan menghitung setiap token (kata) yang tidak kosong, lalu menampilkan jumlah kata dan seluruh kata-kata tersebut menggunakan `for loop` dengan mengecek kondisi bahwa token (kata) tidak kosong.

2. SOAL 2

Screenshot hasil akhir

```
"C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA
C 82
Python 90
Java 100
=====
C          082
Python    090
Java      100
=====
Process finished with exit code 0
```

Pembahasan

Program ini adalah program untuk menggunakan `printf` sebagai sebuah metode yang juga dapat digunakan untuk menampilkan output pada CLI (Command Line Interface) selain dari `print` dan `println`.

Program meminta untuk meminta input merupakan suatu string dengan maksimum 10 alphabetic characters dan integer dari rentang 0-999. Dengan `printf`, output yang ditampilkan bisa dilakukan perubahan bentuk atau penulisan menggunakan placeholder yang sesuai dengan tipe datanya. Misal input di atas `printf` nya adalah sebagai berikut:

`System.out.printf("%-15s%03d\n", string, integer)`

⇒ `%-15s` adalah left-justified untuk string yang diinputkan

⇒ `%03d` adalah membuat angka menjadi maksimum 3 digit dengan leading 0

⇒ Hasil nya adalah seperti gambar di atas

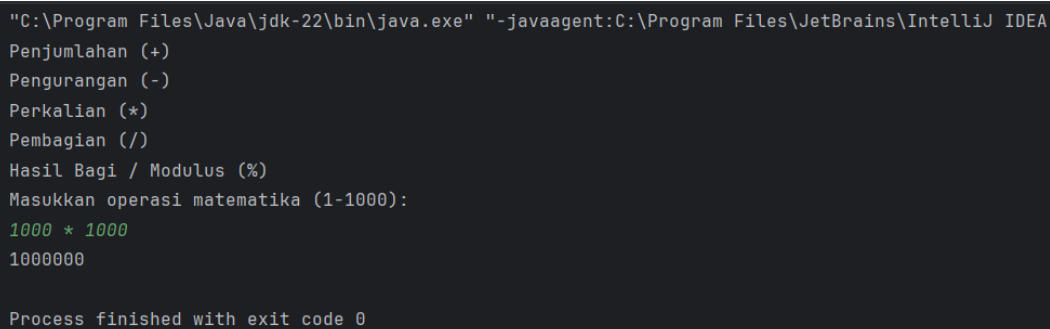
Di program ini saya menginisialisasi loop dengan angka 3 untuk 3 kali perulangan. Kemudian menginputkan untuk 3 kali loop dengan menginputkan string dan integer pada line yang sama. Setiap iterasi string dan integer di masukkan ke dalam array strings[] dan numbers[].

Ada constraint dimana maksimal karakter pada string hanyalah 10 alphabetic characters maka dilakukan loop untuk melakukan substring agar string yang melebihi 10 characters akan menjadi maksimal 10 length characters dan ditampung ke dalam array newStrings[].

Constriant lain adalah untuk integer dari rentang 0-999, maka apabila numbers[i] < 0 membuat nilainya menjadi 0 dan nilai yang melebihi 999 akan menjadi 999 sehingga placeholder %03d bisa menampung integer tersebut menggunakan Math.min(numbers[i], 999).

3. SOAL 3

Screenshot hasil akhir



```
"C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA
Penjumlahan (+)
Pengurangan (-)
Perkalian (*)
Pembagian (/)
Hasil Bagi / Modulus (%)
Masukkan operasi matematika (1-1000):
1000 * 1000
1000000
Process finished with exit code 0
```

Pembahasan

Program di atas adalah program operasi matematika sederhana yang terdiri dari **penjumlahan (+)**, **pengurangan (-)**, **perkalian (*)**, **pembagian (/)**, dan **sisanya bagi / modulus (%)**. Program meminta untuk menginputkan **number1**, operator matematika, dan **number2** pada 1 line yang sama. Hasil tersebut akan ditampilkan pada line berikutnya.

Program ini saya buat menggunakan switch case dari parameter operator (string) matematika tersebut. Jadi ketika menginputkan **1000 * 1000** program akan mengeksekusi case untuk perkalian sehingga nilai **number1** dikalikan dengan **number2**

⇒ 1000 * 1000

⇒ 1000000

Namun karena **constraint** pada soal mengharuskan nilai untuk **number1** dan **number2** di dalam rentang **1 <= number1, number2 <= 1000**. Maka untuk kasus dimana **number2** adalah **0** itu tidak mungkin karena ketika suatu angka dibagi dengan **0** maka hasilnya tidak terdefinisi. Hal ini tidak terjadi karena kondisi di mana program akan berjalan ketika input untuk **number1** dan **number2** di rentang **1-1000**. Kasus lain adalah ketika operator merupakan **pembagian (/)** maka akan dijamin bahwa **number1**

habis dibagi **number2** maka dari itu hasilnya bertipe **integer** ketika **pembagian (/)** di mana **number1 < number2** maka hasilnya akan selalu **0**.

4. SOAL 4

Screenshot hasil akhir

```
"C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA
Masukkan jumlah penjualan item: 35
Gaji Agent = Rp 675000

Process finished with exit code 0
```

Pembahasan

Program di atas adalah program menghitung gaji agent berdasarkan total penjualan item pada 1 bulan. **Gaji pokok agent** adalah **Rp 500.000** dan harga jual setiap item adalah **Rp 50.000**. Kondisi perhitungan gaji agent adalah sebagai berikut:

- **Penjualan > 80**
 - ➔ Mendapat bonus **35%** dari **Penjualan** dikali **harga item**
- **$40 \leq \text{Penjualan} \leq 80$**
 - ➔ Mendapat bonus **25%** dari **Penjualan** dikali **harga item**
- **$15 \leq \text{Penjualan} < 40$**
 - ➔ Mendapat bonus **10%** dari **Penjualan** dikali **harga item**
- **Penjualan < 15**
 - ➔ Mendapat denda sebesar **15%** dari minus Penjualan dari 15 item ($15 - \text{Penjualan}$) dan tidak ada bonus

Jadi jika kita menginputkan **total Penjualan = 35** maka akan masuk ke kondisi ketiga yaitu perhitungan bonus sebesar 10% dengan algoritma sebagai berikut:

- ⇒ $35 \times 50.000 = 1.750.000$
- ⇒ $0.10 \times 1.750.000 = 175.000$
- ⇒ $500.000 + 175.000 = 675.000$
- ⇒ Maka gaji agent setelah menerima bonus adalah Rp 675.000

Program dibuat menggunakan **if else condition**, dengan cara melihat kondisi untuk setiap iterasi **condition** dari **total penjualan** yang diinput.

5. SOAL 5

Screenshot hasil akhir

Output pada Sample Output (Permasalahan)

```
"C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA
3555 2333 4555 6660
berhenti

Process finished with exit code 0
```

Pada sample output di soal, dicontohkan bahwa jika menginputkan plat nomor untuk setiap mobil (total ada 4) seperti gambar di atas, yaitu:

- ⇒ Mobil1 = 3555
- ⇒ Mobil2 = 2333
- ⇒ Mobil3 = 4555
- ⇒ Mobil4 = 6660

Pada algoritma yang diminta pada soal tersebut, input di atas akan menghasilkan modulus 1 yang artinya mobil dipersilahkan untuk **jalan**. Namun pada sample output justru hasilnya adalah **berhenti**.

Output Seharusnya (Pembahasan dan Solusi)

```
"C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA
3555 2333 4555 6660
jalan
Process finished with exit code 0
```

Pembahasan

Karena saya akan tetap mengikuti apa yang tertera pada soal, output program seharusnya adalah seperti gambar di atas. Input di atas digabungkan kemudian dikurangi dengan **999999** lalu hasil tersebut dibagi dengan **5** maka apabila sisa hasil baginya (**modulus**) adalah **0** maka mobil harus **berhenti**, sedangkan apabila modulusnya bukan **0** maka mobil dipersilahkan untuk **jalan**.

⇒ $(3555233345556660 - 999999) \% 5 = 1$ maka output adalah **jalan**

⇒ $(1223111122224449 - 999999) \% 5 = 0$ maka output adalah **berhenti**

Begitu juga untuk sample input kedua, pada sample output di soal menghasilkan **jalan** yang seharusnya adalah **berhenti**.

6. SOAL 6

Screenshot hasil akhir

```
"C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA
2345
35
2380
82075
Process finished with exit code 0
```

Pembahasan

Program ini adalah program untuk melakukan operasi penjumlahan dan pengurangan ketika input merupakan angka yang sangat besar dan bahkan tidak mampu untuk ditampung oleh tipe data numeric seperti **long** sekalipun. Input merupakan **numberA** dan **numberB** pada line berbeda. Diasumsikan bahwa **number1** dan **number2** adalah angka yang berdigit besar sehingga akan dilakukan konversi ke dalam class **BigInteger** yang diimport dari package `import java.math.BigInteger`.

Jadi import **java.math.BigInteger** adalah sebuah class yang digunakan untuk operasi primitif matematika menggunakan integer yang sangat besar yang dapat melebihi batas tampung tipe data primitif. **BigInteger** ini bisa diinputkan juga apabila parameter di dalamnya merupakan string (**string to BigInteger**).

Setelah itu, **number A** dan **number B** akan dilakukan operasi matematika menggunakan methods yang ada pada **Big Integer** yaitu:

⇒ `numberA.add(numberB)` untuk **penjumlahan**

⇒ `numberA.multiply(numberB)` untuk **perkalian**

Hasil tersebut ditampung ke dalam variabel **hasilTambah** dan **hasilKali** kemudian akan ditampilkan.

Namun, ada constraint di mana nilai dari **number A** dan **number B** saat diinput adalah **non-negatif** integer dan hanya maksimal **200 digit** angka. Kondisi dari input user tersebut dicek melalui validasi berikut:

- ⇒ `input.matches("\\d{1,200}")` untuk maksimum 200 digit
- ⇒ `BigInteger(input).compareTo(BigInteger.ZERO) >= 0` untuk memastikan angka bukan merupakan angka negatif

BigInteger.ZERO adalah sebuah inisialisasi angka **BigInteger** yaitu 0. Ini merupakan konstanta yang lebih efisien dibandingkan harus menggunakan objek baru seperti `new BigInteger("0")`.

Referensi Syntax dan Methods:

W3Schools: <https://www.w3schools.com/java>

Tutorialspoint: https://www.tutorialspoint.com/java/number_parseint.htm

Geeksforgeeks: <https://www.geeksforgeeks.org/biginteger-class-in-java/>

ChatGPT: <https://chatgpt.com/>

Lampiran:

Github: <https://github.com/basganajaah/Pemrograman-Berorientasi-Objek---Praktek>