Jurnal 05: Stack dan Queue

Kerjakanlah soal berikut. Jawaban dinilai berdasarkan dua kriteria: a. Kode sesuai dengan permintaan soal, dan b. Sesi tanya-jawab dengan Asprak. Kerjakan sesuai dengan prinsip Java Coding Style.

a. Stack

Nitori merupakan seorang gadis Kappa, yang terkenal dengan keterampilannya dengan bidang teknologi. Namun, Nitori kesusahan untuk membuat sistem yang bisa mengetahui apakah suatu kurung ditutup dengan baik atau tidak.

Contohnya:

a. [({})] -> cocok, karena semua kurungnya dibuka dan ditutup dengan benar.
b. []{}() -> cocok, semua kurungnya sesuai. Dibuka dan ditutup dengan benar.
c. [{)] -> nggak cocok, karena kurung kurawalnya malah ditutup pake kurung biasa.
d. [{ -> nggak cocok, karena nggak ditutup.

Kita bisa menyelesaikan ini dengan menggunakan stack!

Gimana cara kerjanya?
Setiap kali dia nemu kurung buka, maka masukkan kurung buka tersebut ke dalam stack. Setiap kali dia nemu kurung tutup, maka pop si kurung buka dari dalam stack, terus cocokkan kurung tutup tersebut sama kurung buka dalam stack tadi. Kalau sama, berarti valid! Kalau gak sama, berarti gak valid!

Contoh:

a. [{}]
    i. [-> masukkan ke dalam stack.
    ii. { -> masukkan ke dalam stack.
    iii. } -> pop elemen terakhir dalam stack, didapatkan {. { cocok dengan }, maka valid.
    iv.] -> pop elemen selanjutnya dalam stack, didapatkan [. [cocok dengan], maka valid.
b. [{(})]
    i. [-> masukkan ke dalam stack.
    ii. { -> masukkan ke dalam stack.
    iii. (-> masukkan ke dalam stack.
    iv. } -> pop elemen terakhir. Dia nemu (. (gak cocok sama }, jadi udah pasti gak valid.
c. [{
    i. [-> masukkan ke dalam stack.
    ii. { -> masukkan ke dalam stack.
    iii. Dia gak nemu penutup, berarti gak valid.

Tips Implementasi:

- Kalian hanya butuh satu class, di kelas Main saja.
- Manfaatkan method! Kalian bisa menggunakan method untuk memisahkan pekerjaan-pekerjaan seperti mengecek apakah kurung satu dengan kurung dua itu berpasangan atau tidak.
- Kalian bisa menggunakan for loop yang nanti melakukan looping ke setiap karakter di string kurung kalian.
- Dalam tiap loop, dia masukin kurung buka ke dalam stack, dan kalau nemu kurung tutup, dia melakukan pop dari stack untuk mengambil kurung buka terakhir dan dibandingkan apakah dia sama dengan kurung tutupnya?

b. Queue

Mbak Mystia memiliki usaha angkringan belut. Mbak Mystia ingin tahu berapa orang yang sedang mengantri untuk belutnya, dan berapa orang yang sudah memesan kemudian keluar dari antrian. Bantulah Mbak Mystia untuk membuat sistem manajemen antrian!

Sistem manajemen antrian tersebut harus memiliki:

- Antrian sekarang
- Jumlah orang yang sudah keluar antrian

Persyaratan Implementasi:

- Cukup dua kelas. Kelas manajemen dan kelas Main.
- Bisa menambahkan orang yang antri.
- Bisa mengecek status antrian, yang akan mencetak seluruh orang yang sekarang sedang mengantri, jumlah orang yang sedang mengantri, dan jumlah orang yang sudah keluar antrian.
- Bisa melakukan selesai antri, yang akan mengeluarkan 1 orang yang sedang mengantri dan mencetak nama orang yang sudah keluar dari antrian tersebut, kemudian menambah 1 pada jumlah total orang yang sudah keluar antrian.

```
Antrian saat ini:
Reimu
Marisa
Keine
Mokou
Eirin
Ada 5 orang yang sedang antri

Reimu selesai antri
Marisa selesai antri

Antrian saat ini:
Keine
Mokou
Eirin
Ada 3 orang yang sedang antri
```

Java Coding Style

Ikuti gaya penulisan berikut di kode program agar mendapat nilai maksimal

1. Setiap nama kelas, variabel dan konstanta harus dapat menggambarkan isinya.
   Misal, membuat variabel untuk menyimpan jumlah penumpang di bus.
   Deklarasi:

```
int penumpang = 0;        // Bagus, OK
int p = 0;                // Tidak OK
```

2. Penulisan nama harus menggunakan huruf besar/kecil yang sesuai.
   Nama kelas: UpperCamelCase. Misal: PenumpangBus
   Nama variabel: lowerCamelCase. Misal: penumpangBus
   Nama konstanta: CAPITAL_CASE. Misal: PENUMPANG_BUS
   Nama method: lowerCamelCase. Misal: hitungPenumpangBus
3. Penggunaan tab/spasi ketika berbeda blok program, WAJIB masuk 1 tab atau 4 spasi ke dalam.

```
public class PenumpangBus {
    public static void main(String[] args) {
        int penumpang = 0;
        penumpang = penumpang + 5;
        penumpang = penumpang - 4 + 2;
        penumpang = penumpang - 1;
        penumpang = penumpang - 2 + 3;
        penumpang = penumpang - 2 + 5;
        penumpang = penumpang - 1 + 3;
        System.out.println(penumpang);
    }
}
```

4. Gunakan komentar seperlunya.
   Javadoc comment (yang warna biru) hanya untuk mengomentari kelas, method dan variable

```
/** Jumlah penumpang di bus */
int penumpang = 0;

/* Ada 5 orang penumpang naik ke bus */
penumpang = penumpang + 5;

// Penumpang turun 4 orang, naik 2
penumpang = penumpang - 4 + 2;
```

-