

Dasar-Dasar Pemrograman 2

Lab 03

Arrays



FAKULTAS
ILMU
KOMPUTER

I. Arrays

Apa itu Arrays?

Secara informal, Arrays adalah sebuah kumpulan N buah data dengan tipe data yang sama, di mana N adalah panjang dari Arrays itu sendiri. Sederhana, bukan? Arrays dapat menyimpan objek-objek dengan tipe data primitive maupun tipe data reference/object.

Inisiasi Arrays

Untuk menggunakan struktur data Arrays, tentunya kita perlu menginisiasinya terlebih dahulu. Nah, untuk menginisiasi Arrays, kalian dapat memakai format:

```
tipeData[] namaVariabel = new tipeData[panjangArray];
```

atau, kalian bisa langsung mendeklarasikan Arrays beserta isinya dengan format:

```
tipeData[] namaVariabel = {<Anggota array, dipisahkan "<,<"><"}<};
```

Sebagai contoh, misalkan kalian ingin membuat sebuah Arrays bernama “kelompokAsdos” yang beranggotakan Stev, Lita, Litha, dan Naufal. Untuk menginisiasi Arrays tersebut, kalian dapat melakukannya dengan cara:

```
String[] kelompokAsdos = new String[4];
kelompokAsdos[0] = "Stev";
kelompokAsdos[1] = "Lita";
kelompokAsdos[2] = "Litha";
kelompokAsdos[3] = "Naufal";
```

Atau

```
String[] kelompokAsdos = {"Stev", "Lita", "Litha", "Naufal"};
```

Kalian dapat memilih salah satu dari kedua cara tersebut sesuai dengan kebutuhan.

Pertanyaannya sekarang, bagaimana jika tiba-tiba Stev keluar dari kelompok asdos dan digantikan oleh Ryaas? Kasus ini dapat ditangani dengan melakukan *overwrite* terhadap *value* yang ingin diganti. Kalian bisa mengganti *value* Arrays tersebut dengan cara:

```
kelompokAsdos[0] = "Ryaas";
```

Dengan demikian, isi Arrays kalian berubah menjadi: {"Ryaas", "Lita", "Litha", "Naufal"}

Iterasi Arrays

Bagaimana jika kalian ingin mengeluarkan output dari setiap elemen di Arrays "kelompokAsdos"? Tentu saja kalian dapat melakukannya dengan cara seperti berikut:

```
System.out.println(kelompokAsdos[0]);
System.out.println(kelompokAsdos[1]);
System.out.println(kelompokAsdos[2]);
System.out.println(kelompokAsdos[3]);
```

Namun cara tersebut tentu tidak efektif. Bayangkan kalian memiliki sebuah Arrays dengan size 100 dan kalian harus mengeluarkan output dari setiap elemen di Arrays tersebut. Mengerikan sekali, bukan?

Untuk mempermudah mengakses elemen secara terus menerus, kita dapat melakukan iterasi pada Arrays tersebut. Kalian dapat mengiterasi sebuah Arrays dengan berbagai cara, namun yang paling umum di antaranya adalah:

1. Menggunakan for loop

Kalian pasti sudah tidak asing dengan cara ini setelah mempelajari penggunaan for loop pada minggu sebelumnya. Kalian dapat mengakses setiap indeks pada Arrays dengan:

```
for (int i = 0; i < kelompokAsdos.length; i++) {  
    System.out.println(kelompokAsdos[i]);  
}
```

Output:
Stev
Lita
Litha
Naufal

2. Menggunakan for-each loop

Dibandingkan mengakses setiap indeks, cara ini memungkinkan kalian untuk langsung mengakses setiap *value* dari Arrays sesuai dengan *datatype* ataupun *object* yang dirujuk. Hal ini dapat dilakukan dengan cara:

```
for (String asdos: kelompokAsdos) {  
    System.out.println(asdos);  
}
```

Output:
Stev
Lita
Litha
Naufal

Tentunya kalian dapat menggunakan salah satu dari kedua cara tersebut sesuai dengan kebutuhan kalian.

Mencetak Arrays

Kalian sudah belajar cara untuk melakukan iterasi terhadap Arrays. Jika kita lihat lagi, ternyata saat belajar melakukan iterasi, kita juga belajar untuk mencetak elemen yang dimiliki oleh Arrays. Tetapi bukankah cara itu terlalu biasa? Untuk mencetak Arrays, kita dapat menggunakan static method yang dimiliki oleh Class Arrays. Caranya adalah sebagai berikut:

```
System.out.println(Arrays.toString(kelompokAsdos));
```

Jika kita menjalankannya, kita akan mendapatkan output berikut:

```
[Stev, Lita, Litha, Naufal]
```

Mudah sekali, bukan?

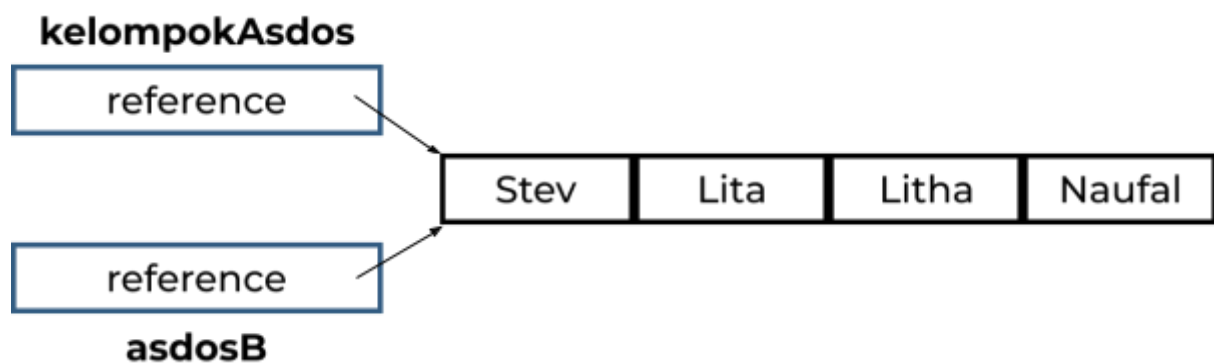
Pass by Reference

Tahukah kalian? Pada bahasa pemrograman Java, Arrays adalah sesuatu yang di-pass by reference. Namun sebenarnya apa yang dimaksud? Agar lebih mudah memahaminya, mari kita lihat contoh berikut ini.

Sebelumnya, kalian sudah memiliki Arrays `kelompokAsdos` yang berisi {"Stev", "Lita", "Litha", "Naufal"}. Karena mereka berempat adalah asdos kelas B, maka kita ingin membuat Arrays baru bernama `asdosB` dengan cara berikut.

```
String[] asdosB = kelompokAsdos;
```

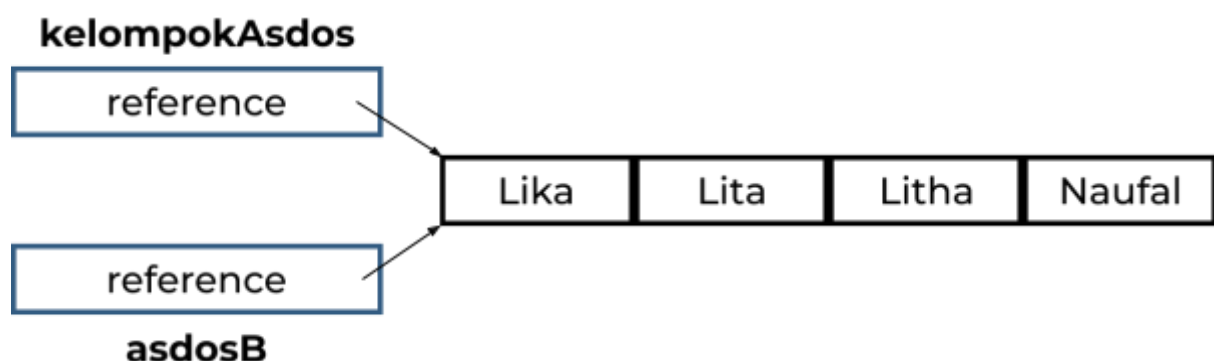
Dengan demikian, Arrays `asdosB` akan memiliki isi yang sama dengan array `kelompokAsdos`, yaitu reference/address dari {"Stev", "Lita", "Litha", "Naufal"}. Hal ini karena Arrays di-pass by reference, sehingga ketika kita melakukan assignment, yang di-pass adalah reference/address nya bukan value nya.



Oleh karena itu, ketika kalian mencoba untuk mengganti **Stev** dengan **Lika** melalui `asdosB` seperti berikut:

```
asdosB[0] = "Lika"
```

Maka `kelompokAsdos[0]` juga akan menjadi **Lika**, bukan **Stev** lagi.



Soal Lab 03

Misi, Pakeeett!

MOMEN KETIKA PAKETNYA SAMPE



Setelah berhasil memborong stok Lay's yang terakhir, Sasha, Niccolo, Jean dan Connie sekarang memiliki pasokan keripik kentang untuk satu tahun. Namun, sekarang mereka dihadapkan dengan masalah baru: bagaimana cara menyimpan makanan yang kemasannya berisi 99% udara?

Berkat jerih payahnya, Connie menemukan sebuah gudang sewaan besar yang lokasinya sangat jauh dari toko Lay's. Untuk memindahkan semua Lay's yang baru dibeli, mereka meminta bantuan Angewomon yang bisa terbang. Kebetulan Angewomon baru memulai bisnis jasa ekspedisi. Hanya saja, Angewomon hanya mau mengirim paket apabila paket pertama yang ia kirimkan hari itu adalah paket paling ringan.

Sasha mencoba menyusun urutan paket Lay's yang harus dikirim sesuai dengan syarat yang diberikan Angewomon. Tetapi tanpa pengetahuan Sasha, Niccolo secara iseng menggeser-geser urutannya. Sekarang Sasha tidak yakin apakah susunannya akan disepakati oleh Angewomon atau tidak. Sasha meminta bantuanmu untuk membuat program yang bisa mengubah urutan pengiriman sekaligus mendeteksi apakah paket pertamanya adalah yang paling ringan.

Bantulah Sasha dan yang lain mengirim paket mereka!

Requirements

Program kamu memproses input berupa array berukuran **N** dan menggeser posisi setiap elemen dalam array tersebut ke kiri maupun ke kanan sesuai urutan. Output yang diharapkan adalah **array yang sudah digeser** dan **boolean** yang menandakan apakah elemen pertama array tergeser adalah elemen terkecil.

Kamu diharuskan mencetak langsung array yang sudah digeser, beserta tanda kurung sikunya.

Input:

- Masukan pertama adalah sebuah bilangan **N** sebagai panjang array.
- Baris berikutnya berisi **N integer** yang dipisahkan oleh spasi.
- Masukan terakhir berupa String yang terdiri dari huruf L (*Left*) dan R (*Right*), setiap hurufnya menunjukkan arah pergeseran yang harus diterapkan. (Lihat contoh input/output)

Output:

- Array dengan posisi yang sudah digeser
- Nilai **true** bila **elemen pertama** array yang sudah digeser adalah **elemen terkecil**, atau nilai **false** bila sebaliknya.
 - Apabila array kosong, maka kembalikan nilai **false**.

Keterangan:

Geser Kiri

[1, 4, 3, 2, 5]	→	[4, 3, 2, 5, 1]
[1]	→	[1]
[]	→	[]

Geser Kanan

[3, 2, 1, 4, 5]	→	[5, 3, 2, 1, 4]
[1]	→	[1]
[]	→	[]

Contoh Input & Output

Keterangan: Teks berwarna **biru** adalah input yang dimasukkan oleh *user*.

Contoh masukan 1

Contoh keluaran 1

Panjang: 5 Barisan: 1 2 3 4 5 Pergeseran: RR	Output: [4, 5, 1, 2, 3] false
---	--

Penjelasan Contoh 1

- Menjalankan perintah **R** pertama, akan menggeser barisan input menjadi:
[1, 2, 3, 4, 5] → [5, 1, 2, 3, 4]
- Menjalankan perintah **R** kedua, akan menggeser barisan input menjadi:
[5, 1, 2, 3, 4] → [4, 5, 1, 2, 3]
- Elemen pertama = 4. Karena 4 bukan merupakan elemen terkecil pada array tersebut, maka dikembalikan nilai **false**. Angewomon tidak mau mengirim paket-paket ini.

Contoh masukan 2

Contoh keluaran 2

Panjang: 6 Barisan: 1 2 3 4 5 6 Pergeseran: RLLR	Output: [1, 2, 3, 4, 5, 6] true
---	--

Penjelasan Contoh 2

- Menjalankan perintah **R**, akan menggeser barisan input menjadi:
[1, 2, 3, 4, 5, 6] → [6, 1, 2, 3, 4, 5]
- Menjalankan perintah **L** pertama, akan menggeser barisan input menjadi:
[6, 1, 2, 3, 4, 5] → [1, 2, 3, 4, 5, 6]
- Menjalankan perintah **L** kedua, akan menggeser barisan input menjadi:
[1, 2, 3, 4, 5, 6] → [2, 3, 4, 5, 6, 1]
- Menjalankan perintah **R** terakhir, akan menggeser barisan input menjadi:
[2, 3, 4, 5, 6, 1] → [1, 2, 3, 4, 5, 6]

- Elemen pertama = 1. Karena 1 adalah elemen terkecil dalam array tersebut, maka dikembalikan nilai **true**. Angewomon akan mengirimkan paket-paket ini dengan selamat sampai tujuan.

Contoh masukan 3

Contoh keluaran 3

Panjang: 0 Barisan: Pergeseran: RRRR	Output: [] false
--	----------------------------

Penjelasan Contoh 3

- Barisan input kosong, sehingga meski dijalankan perintah **R** sebanyak empat kali, kondisi array tidak berubah.
[] → []
- Array kosong. Karena array kosong, maka dikembalikan nilai **false**. Tidak ada paket untuk dikirimkan oleh Angewomon.

Batasan

- Panjang array dipastikan integer nonnegatif.
- Setiap elemen array dipastikan integer positif.
- Urutan rotasi dipastikan huruf kapital dan hanya berisi huruf L atau R.

Komponen Penilaian

- **45%** Kebenaran pergeseran elemen array
- **45%** Kebenaran deteksi elemen terkecil
- **10%** Dokumentasi dan kerapian kode

Catatan:

- [Telah disediakan template yang membantu menangani input](#)
- Kamu **diperbolehkan menambahkan method baru** untuk mempermudah pengerjaan.

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

Lab03_[Kelas]_[KodeAsdos]_[NPM]_[NamaLengkap].zip

Contoh:

Lab03_A_LN_1234567890_DekDepe.zip