Akhmad Fauzi(PC-11)

Mobile Apliaction and Technology

**Dasar dan Cara Membuat Array**

Di bahasa pemrograman lain, array sering digunakan untuk menyimpan kumpulan data yang memiliki tipe data yang sama. Contoh umumnya digunakan untuk menyimpan produk-produk apa saja yang dibeli oleh pelanggan dalam sebuah sistem kasir.

Di Kotlin, kita dapat membuat array dengan 2 cara:

1. Menggunakan Function arrayOf()
2. Menggunakan Lambda Expression

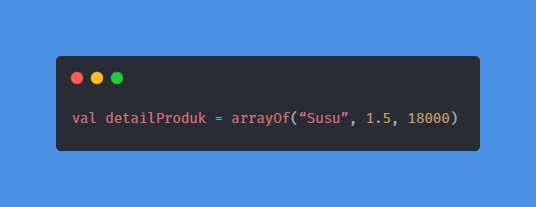
# Membuat Array dengan Function arrayOf()

Untuk cara pembuatan array yang pertama, kita dapat memanfaatkan function arrayOf() sehingga jika kita ingin membuat variabel array yang menyimpan produk-produk yang dibeli oleh pelanggan, menjadi seperti berikut:



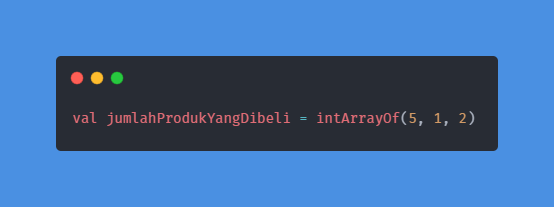
kita membuat variable penampungnya terlebih dahulu, lalu variable tersebut diisi dengan memanggil function arrayOf(). Arguments yang diisi di dalam function arrayOf adalah data yang ingin kita simpan di dalam array.

arrayOf() sendiri merupakan sebuah function dimana kita bisa menyimpan berbagai macam tipe data yang berbeda-beda di dalam suatu array. Contohnya:



Kita bisa memasukkan variable bertipe String, Double, dan Integer di dalam array yang sama dengan menggunakan function arrayOf().

Jika kita ingin spesifik menggunakan 1 tipe data yang sama dalam array, maka kita dapat menggunakan function lain yang sesuai dengan tipe data yang kita inginkan. Contohnya membuat array khusus untuk tipe data Integer, contohnya:

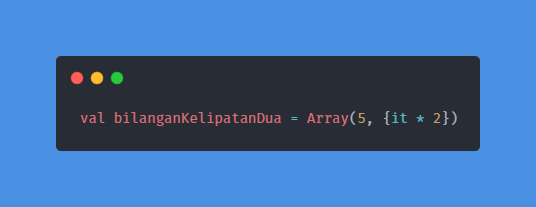


Untuk membuat array yang khusus menyimpan data-data bertipe Integer, kita dapat menggunakan function **intArrayOf()**, yang memang dikhususkan untuk membuat array bertipe Integer. Bagaimana jika ingin membuat array yang khusus menyimpan Boolean saja? Ada banyak function untuk membuat array tergantung dari tipe datanya, berikut beberapa contohnya:

* intArrayOf()
* booleanArrayOf()
* charArrayOf()
* floatArrayOf()
* charArrayOf()
* doubleArrayOf()

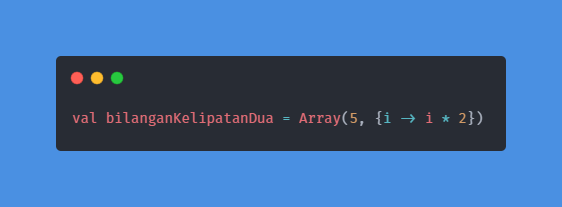
# ****Membuat Array dengan Constructor dan Lambda Expression****

Selain menggunakan function arrayOf(), kita juga dapat membuat array dengan menggunakan Lambda Expression. Lambda Expression merupakan sebuah cara atau teknik di Kotlin yang mirip seperti membuat sebuah function sendiri namun tanpa perlu mendefinisikan fun dan seterusnya. Namun, ketika kita menggunakan lambda expression, kita hanya perlu langsung mengisi isi dari function tersebut. Berikut contoh dari membuat array menggunakan Lambda Expression:



Itu merupakan contoh dari penggunaan Lambda Expression untuk membuat array. Kita harus terlebih dahulu memanggil constructor Array(). Lalu arguments pertama diisi jumlah data, yakni 5. Sedangkan arguments kedua diisi dengan lambda expression. Lambda expression disimbolkan dengan cara berada di dalam sebuah kurung kurawal “{ }”. Lalu di dalamnya terdapat apa yang ingin dilakukan, yakni kalau kasusnya dalam membuat array maka untuk setiap data itu ingin kita isi dengan nilai berapa. **it**merupakan simbol dari urutan data. Maksudnya, ketika kita ingin membuat array yang berisi 5 data, maka untuk setiap urutan data, dikalikan dengan 2. Misalnya, data pertama, memiliki nomor urut atau index bernilai 0, maka nilai dari data index ke-0 adalah 0 dikalikan dengan 2, yakni 0. Lalu data kedua, memiliki nomor index bernilai 1, maka nilai dari data index ke-1 adalah 1 dikalikan dengan 2, yakni 2. Begitu seterusnya sebanyak jumlah data yang didefinisikan di argument pertama.

Sebenarnya, penggunaan **it**adalah cara simple dari penggunaan lambda expression. Seharusnya, lambda expression ditulis seperti ini:

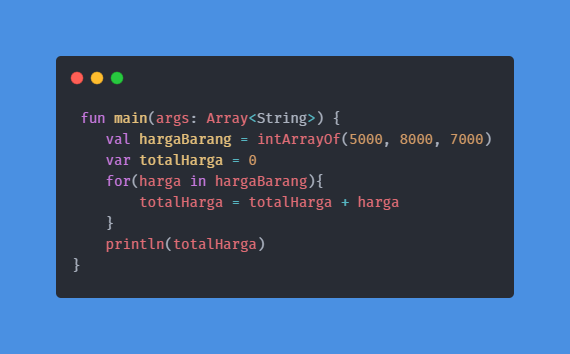


Maksudnya bagaimana? Dalam lambda expression, terdapat sebuah tanda panah ke kanan yang mengartikan bahwa variable di sebelah kiri berarti adalah parameter yang nantinya dapat digunakan di dalam function. Sementara bagian yang di sebelah kanan merupakan syntax yang biasanya berada di dalam function seperti function penjumlahan maka dituliskan syntax penjumlahan seperti **return a + b**.

Namun dalam hal ini, penggunaan lambda expression dapat disingkat jikalau parameter yang berada di sebelah kiri dari tanda panah hanya terdapat 1 saja. Sehingga kita dapat menghapus variable i di sebelah kiri panah, serta menghapus tanda panah dan mengubah variable i yang digunakan di dalam lambda expression menjadi **it**. **it** adalah simbol dari satu-satunya parameter yang dapat digunakan di dalam lambda expression.

# Function forEach()

Array pada Kotlin memiliki Function **forEach()** untuk memudahkan kita dalam melakukan perulangan pada Array. Tanpa adanya forEach(), ketika kita ingin melakukan perulangan pada Array, kita pada umumnya menggunakan perulangan for. Contohnya, ketika kita ingin menghitung total harga pada suatu arrayHarga sebagai berikut:

menghitung total harga menggunakan perulangan for pada array

Lalu bagaimana caranya menggunakan function forEach() untuk melakukan perulangan pada array? Berikut contohnya:



menghitung total harga menggunakan forEach pada Array

Bagaimana? sedikit lebih ringkas dan terlihat keren bukan? Pada saat menggunakan forEach, kita menggunakan versi ringkas dari lambda expression untuk meringkas penggunaan forEach.

Sebenarnya ada function lain, jikalau saat kita melakukan perulangan lalu kita memerulan index dari data yang diulang, maka kita dapat menggunakan function lain, yakni forEachIndexed(). Berikut contohnya:



Function forEach dan forEachIndexed menggunakan Lambda Expression. Karena forEachIndexed memiliki 2 parameter, yakni parameter untuk index dan parameter untuk nilai dari index tersebut, maka penggunaan Lambda Expression pada function forEachIndexed tidak dapat disingkat menggunakan it.

Fungsi Fungsi Type Array

* Get dan Set untuk memanggil dan mengisi sebuah item Array -> Simbol[Index]
* Size untuk memperoleh jumlah Item
* Find untuk memperoleh sebuah item yang memenuhi pernytaan predicate (Lambda expression) yang diberikan
* indexOf untuk memperoleh nomor index dari item yang dicari
* Sort dan sortDecending untuk mengurutkan item
* Random untuk memanggil Sebuah item secara acak
* Average untuk memperoleh nilai rata rata dari IntArray
* Sum untuk memperoleh nilai total dari Item IntArray