Akhmad Fauzi (PC-11)

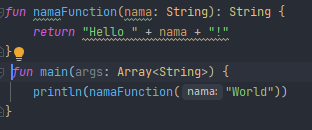
*Mobile Application and Technology*

**LAPORAN MATERI**

**Kotlin - Function**

Function

Function sederhananya adalah sebuah kumpulan baris kode yang dapat kita panggil dengan sederhana (sering kali cukup dipanggil hanya dengan 1 baris saja) tanpa perlu menulis baris kode yang banyak berulang-ulang kali. Lalu bagaimana cara membuat function di Kotlin? Berikut contoh sederhananya:

****

contoh sederhana dari function yang bertugas membuat tulisan “Hello [nama]!”

Bagaimana hasilnya ketika dijalankan?



hasil ketika program di atas dijalankan.

Untuk membuat sebuah function, kita perlu mendefinisikan ‘fun’.

Lalu diikuti dengan nama function. Penulisan nama function harus digabung, gunakan camel case seperti ‘namaFunctionKamuBebas’ untuk memudahkan orang lain dalam membaca kode kamu.

Setelah itu diikuti dengan **parameter**. Parameter adalah sebuah data yang dibutuhkan oleh function kamu untuk nantinya diolah di dalam function. Untuk menuliskan paramter, tulis nama variabel dari parameter terlebih dahulu. Penamaan variable digunakan untuk memanggil variabel di dalam function (seperti saya memanggil variabel nama untuk digabungkan dengan teks lainnya). Setelah itu diikuti dengan tipe datanya, bisa String ataupun menggunakan tipe data yang lain, disesuaikan dengan kebutuhan.

Setelah itu, diikuti dengan **return dari function**. Return dari function ini berguna untuk mendefinisikan apakah function kita memiliki return atau tidak. Jika memiliki return, maka kita perlu mendefinisikan tipe data apa yang dihasilkan oleh function yang kita buat. Contohnya di function namaParamter, saya menggunakan String sebagai return dari function, karena saya ingin mengembalikan nilai String yang telah saya olah.

Setelah menuliskan return dari function, kita buka bracket untuk menentukan isi dari function kita apa. Jika function kita memiliki return, maka pastikan untuk memanggil syntax return di akhir dari functionnya.

Untuk memanggil function sendiri, kita bisa memanggil namanya diikuti dengan buka kurung berisi data yang ingin kita gunakan di dalam function (arguments).

Parameter

Sebelumnya kita telah menyinggung sedikit apa itu parameter. Parameter adalah sebuah data yang dibutuhkan oleh function kamu untuk nantinya diolah di dalam function. Ketika membuat function, kita tidak hanya dapat menggunakan 1 parameter saja. Kita juga bisa menggunakan lebih dari 2 parameter. Kita juga bisa tidak menggunakan paramter sama sekali. Gunakanlah parameter sesuai dengan kebutuhan masing-masing. Berikut contohnya:

Function yang menggunakan 2 parameter, yakni function operasi aritmatika dasar.

*fun tambah(angka1: Int, angka2: Int): Int {}*

Function yang menggunakan 3 parameter, yakni function untuk menentukan angka tertinggi.

*fun angkaTertinggi(angka1: Int, angka2: Int, angka3: Int): Int {}*

Function yang tidak menggunakan parameter, yakni function untuk melakukan pengacakan nomor.

*fun acakNomor(): Int {}*

dan masih banyak lagi contohnya. Selain itu, kita dapat menggunakan berbagai tipe data yang telah dijelaskan sebelumnya dari String, Int, Double, Boolean, hingga Array. Berikut contohnya:

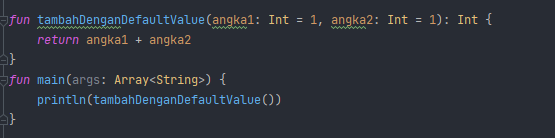
*fun menggunakanBanyakTipeData(angka: Int, huruf: Char, kata: String, Desimal: Double, array: Array<Double>): Int {}*

Parameter dengan Default Value

Di Kotlin, kita juga dapat mengisi default value dari parameter. Maksudnya bagaimana? Kita dapat memberikan opsi pada function kita. Jika saat memanggil function, lalu kita tidak mengirimkan data sesuai jumlah data yang dibutuhkan di parameter, maka akan terjadi error. Untuk menghindari error tersebut, kita dapat menggunakan default value sehingga meskipun data tidak dikirimkan sesuai jumlah parameter, function tetap akan berjalan dengan baik. Contohnya dalam function tambah, kita membutuhkan 2 data untuk ditambahkan. Namun bagaimana jika kita tidak mengirimkan 2 data tersebut? Akan terjadi error, karena tidak ada data ditambah tidak ada data maka tidak dapat melakukan penjumlahan. Berikut cara menggunakan default value untuk menghindari error:

*fun tambahDenganDefaultValue(angka1: Int = 1, angka2: Int = 1): Int {}*

Setelah mendefinisikan tipe data dari parameter, kita dapat mengisi default value dari parameter tersebut. Berikut contoh programnya:



Contoh penggunaan dari default value

Berikut hasilnya:

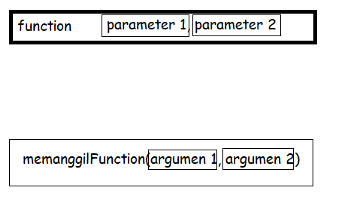


hasil dari contoh penggunaan default value

Jadi meskipun kita tidak mengirimkan data sesuai dengan jumlah parameter, kita tetap dapat menggunakan function tersebut tanpa ada masalah.

Argument

Argument juga sempat disinggung sebelumnya, jika Parameter digunakan untuk mendefinisikan bahwa sebuah function membutuhkan sebuah data tertentu untuk dikirimkan ke dalam function. Maka, argument adalah sebuah data yang dikirimkan ke function.



gambaran perbedaan antara parameter dengan argument

Di Kotlin, misalkan kita memiki sebuah function yang memiliki 5 parameter yang memiliki default value. Lalu kita ingin memanggil function tersebut, dan menggunakan argument kedua dan kelima saja. Pada bahasa lain, kita harus urut memanggil argumentnya satu-per-satu, seperti berikut:

*fun contohBahasaLain(param1: String = “a”, param2: String = “b”, param3: String = “c”, param4: String = “d”, param5: String = “e”)*

Lalu misalkan kita ingin memanggil function tersebut, namun kita ingin mengirimkan data sehingga parameter ke-3 tidak bernilai “c” lagi tetapi “x”. Maka ketika memanggil function menjadi seperti ini dalam bahasa lain:

*val memanggilFunction = contohBahasaLain(arg1, arg2, arg3)*

Dalam bahasa lain, ketika kita hanya ingin mengirimkan data ke-3, kita harus mengirimkan juga data dari parameter sebelumnya. Sedangkan di dalam Kotlin terdapat **named arguments**. Artinya, ketika nama variabel parameter kita sama dengan nama variabel argument kita, meskipun kita hanya mengubah data parameter ke-3, kita tidak perlu menuliskan data parameter ke-1 dan ke-2. Contohnya berikut masih tetap menggunakan function contohBahasaLain:

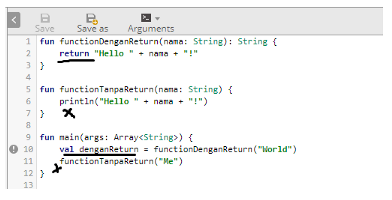
*val memanggilNamedArgument = contohBahasaLain(param3 = “x”)*

Kita cukup menggunakan 1 argument saja, namun nama argument tersebut harus sama dengan nama variabel dari parameter.

Return dari Function

Pada saat membahas function, kita telah berkenalan dengan return dari function. Sederhananya, return dari function membantu Kotlin mendefinisikan apakah Function kita memiliki return value atau tidak. Jika function kita memiliki return value, maka kita perlu mendefinisikan tipe data dari return value tersebut.

Perbedaan sederhana antara function yang memiliki return value dan tidak adalah .., coba perhatikan ilustrasi berikut:



ilustrasi perbedaan antara function dengan return value dan tanpa return value

Perbedaan antara function yang memiliki return value dan tanpa return value adalah, yang pertama, untuk function yang memiliki return value maka di akhir function kita perlu mendefinisikan return dengan nilai apa yang ingin kita kirimkan kembali ke baris yang memanggil function tersebut, yakni baris ke 10. Sedangkan, pada function yang tidak memiliki return value, kita tidak perlu menuliskan return di dalam function tersebut.

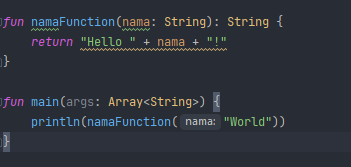
Perbedaan kedua, pada function yang menggunakan return value, kita harus terlebih dahulu menampung atau menyimpan hasil dari pemanggilan function tersebut. Karena function yang memiliki return value akan mengembalikan sebuah nilai atau value hasil dari pengolahan di dalam function. Sementara, pada function yang tidak memiliki return value, kita tidak perlu membuat variabel untuk menyimpan nilai yang dihasilkan dari function tersebut, karena pada dasarnya function yang tidak memiliki return value otomatis tidak mengembalikan nilai apa-apa.

Function yang tidak memiliki return value, sering kali disebut sebagai **void**.

Sementara untuk function yang memiliki return value, kita dapat menggunakan berbagai macam tipe data sebagai return value dari function tersebut. Kita dapat menggunakan baik yang sering kita gunakan seperti String, Int, atau bahkan kita dapat menggunakan Array dan sebagainya.

Single-Expression Function

Selama ini, ketika kita membuat function sederhana, kita memerlukan minimal 3 baris kode. Contohnya sebagai berikut:



contoh function yang simpel

Function namaFunction membutuhkan 3 baris kode, padahal sebenarnya function tersebut sangat simpel. Pada bahasa Kotlin, kita dapat meringkas pembuatan Function yang simpel sehingga menjadi 1 baris, contohnya mari kita ubah function namaFunction menjadi 1 baris:

*fun namaFunction(nama: String): String = “Hello ” + nama + “!”*

nah, kita dapat memperingkas menjadi sebagai berikut. Apa saja yang dapat dihapus? Kurung kurawal, dan return.

Sebenarnya, kita dapat membuatnya menjadi lebih ringkas lagi, caranya dengan cara menghapus tipe data dari return value, menjadi sebagai berikut:

*fun namaFunction(nama: String) = “Hello ” + nama + “!”*

Karena pada bahasa Kotlin, bahasa Kotlin akan secara otomatis menebak tipe data apa yang kita gunakan. Sehingga pada kasus ini kita tidak perlu menuliskan tipe data dari return valuenya.

Sudah memahami penggunaan Function?  
Sekarang saatnya untuk berlatih. Gunakanlah Function pada berbagai hal supaya struktur kode menjadi lebih rapi dan mudah dibaca oleh orang lain. Gunakanlah fitur-fitur yang telah disediakan oleh Kotlin seperti named argument agar mempermudah kamu dalam mengisi parameter tertentu pada function.

Segala saran akan saya terima untuk membuat konten ini menjadi lebih baik sebagai salah satu sarana pembelajaran teman-teman.  
Terima kasih telah membaca. Semangat belajar, selamat berproses!