**Higher-Order Functions**

Setelah mempelajari modul sebelumnya, Anda mungkin bertanya apa yang bisa dilakukan dengan *lambda* atau *anonymous function*?

Kita bisa memanfaatkan *lambda* untuk membuat *higher-order function*. *Higher order function* adalah fungsi yang menggunakan fungsi lainnya sebagai parameter, menjadi tipe kembalian, atau keduanya.

Coba perhatikan fungsi berikut:

1. void myHigherOrderFunction(String message, Function myFunction) {
2. print(message);
3. print(myFunction(3, 4));
4. }

Fungsi di atas merupakan *higher order function* karena menerima parameter berupa fungsi lain. Untuk memanggil fungsi di atas, kita bisa langsung memasukkan *lambda* sebagai parameter maupun variabel yang berisi nilai berupa fungsi.

1. // Opsi 1
2. Function sum = (int num1, int num2) => num1 + num2;
3. myHigherOrderFunction('Hello', sum);

6. // Opsi 2
7. myHigherOrderFunction('Hello', (num1, num2) => num1 + num2);

Jika disimulasikan fungsi myHigherOrderFunction akan memanggil fungsi sum yang dijadikan parameter.

1. void myHigherOrderFunction(String message, Function myFunction) {
2. print(message);
3. print(myFunction(3, 4));  // sum(3, 4)    // return 3 + 4
4. }

Namun deklarasi *higher order function* ini bisa menjadi sedikit *tricky*. Misalnya kode di bawah ini tidak akan terdeteksi eror namun ketika dijalankan, aplikasi Anda akan mengalami *crash*. Tahukah kenapa?

1. void myHigherOrderFunction(String message, Function myFunction) {
2. print(message);
3. print(myFunction(4));
4. }

Karena kita tidak menentukan spesifikasi dari fungsi seperti jumlah parameter atau nilai kembaliannya, maka semua jenis fungsi akan bisa dijalankan termasuk pemanggilan myFunction seperti di atas. Untuk mengatasinya kita bisa lebih spesifik menentukan seperti apa fungsi yang *valid* untuk menjadi parameter.

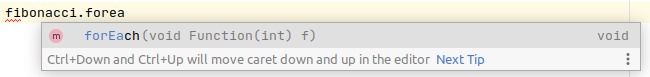
1. void myHigherOrderFunction(String message, int Function(int num1, int num2) myFunction) { }

Pada fungsi di atas kita perlu memasukkan fungsi dengan dua parameter dan nilai kembali berupa int sebagai parameter.

Pada materi *collection* sebenarnya kita telah menggunakan satu fungsi yang merupakan *higher order function* yaitu fungsi forEach(). Sebagai contoh kita punya daftar bilangan *fibonacci* yang disimpan ke sebuah variabel.

1. var fibonacci = [0, 1, 1, 2, 3, 5, 8, 13];

IntelliJ IDEA akan menunjukkan *suggestion* apa saja yang perlu menjadi parameter. Kita bisa melihat bahwa forEach membutuhkan satu parameter berupa fungsi.

[](https://www.dicoding.com/academies/191/tutorials/7650?from=7647)

Sehingga ketika memanggil fungsi ini kita bisa melakukan operasi pada masing-masing item misalnya mencetak ke konsol.

1. fibonacci.forEach((item) {
2. print(item);
3. });