**While and do-while**

Metode lain untuk melakukan perulangan adalah dengan while. Sama seperti for, instruksi while mengevaluasi ekspresi *boolean* dan menjalankan kode di dalam blok while ketika bernilai true.

Untuk menampilkan angka 1 sampai 100 dengan while kita bisa menulis kode seperti berikut:

1. var i = 1;
3. while (i <= 100) {
4. print(i);
5. i++;
6. }

Bisa dilihat pada kode di atas bahwa perulangan dengan *while* tidak memiliki ketergantungan dengan variabel index seperti pada *for loop*. Karena itu, meskipun *while* dapat melakukan perulangan yang sama dengan *for*, *while* lebih cocok digunakan pada kasus di mana kita tidak tahu pasti berapa banyak perulangan yang diperlukan.

Bentuk lain dari while adalah perulangan do-while.

1. do {
2. print(i);
3. i++;
4. } while (i <= 100);

Kondisi pada while akan dievaluasi sebelum blok kode di dalamnya dijalankan, sedangkan do-while akan mengevaluasi *boolean expression* setelah blok kodenya dijalankan. Ini artinya kode di dalam *do-while* akan dijalankan setidaknya satu kali.

Salah satu skenario umum dari penggunaan *do-while* adalah pada validasi *user*.

1. String username;
2. bool notValid = false;
4. do {
5. stdout.write('Masukkan nama Anda (min. 6 karakter): ');
6. username = stdin.readLineSync();
8. if (username.length < 6 ) {
9. notValid = true;
10. print('Username Anda tidak valid');
11. }
12. } while (notValid);

Pada contoh di atas jika username yang dimasukkan oleh *user* kurang dari 6 karakter, maka input tersebut tidak valid dan *user* akan diminta lagi untuk memasukkan *username*.

**Infinite loops**

Ketika menerapkan perulangan pada program kita, ada satu kondisi yang perlu kita hindari yaitu ***infinite loop***. *Infinite loop* atau *endless loop* adalah kondisi di mana program kita *stucked* di dalam perulangan. Ia akan berjalan terus hingga menyebabkan *crash* pada aplikasi dan komputer kecuali ada intervensi secara eksternal, seperti mematikan aplikasi.

Kode berikut ini adalah contoh di mana kondisi *infinite loop* dapat terjadi:

1. var i = 1;
3. while (i < 5) {
4. print(i);
5. }

Dapatkah Anda mencari apa yang salah dari dari kode di atas sehingga terjadi *infinite loop*?

Jawabannya adalah karena variabel i selalu bernilai 1. Alhasil kondisi i < 5 akan selalu bernilai true dan akibatnya aplikasi akan terus mencetak 1 ke konsol sehingga mengalami *crash*.

**Challenge**

Sebenarnya program *input username* di atas masih belum lengkap karena memiliki *bug* yang bisa menimbulkan *infinite loop*. Dapatkah Anda menemukan *bug* tersebut dan cara mengatasinya?