**Constants & Final**

Terkadang kita butuh menyimpan sebuah nilai yang tidak akan pernah berubah selama program berjalan. Variabel telah membantu kita untuk menyimpan nilai dan bisa diakses dengan nama yang deskriptif. ***Constants*** adalah hal baru yang akan kita pelajari dan berguna untuk menyimpan nilai yang tidak akan berubah selama program berjalan.

Sesuai pengertian di atas, kita bisa mendefinisikan nilai yang konstan pada program kita. Salah satu contoh paling mudah yang bisa kita ambil adalah nilai pi = 3.14. Untuk mendefinisikan variabel konstan, gunakanlah *keyword* const.

1. const pi = 3.14;

*Type inference* dari Dart akan secara otomatis mendeteksi tipe data dari const pi di atas sebagai double, namun Anda dapat menentukan tipe data secara eksplisit.

1. const num pi = 3.14;

Sehingga pada sebuah aplikasi kalkulator luas lingkaran, kode Anda akan menjadi seperti berikut:

1. const num pi = 3.14;
3. void main() {
4. var radius = 7;
5. print('Luas lingkaran dengan radius $radius = ${calculateCircleArea(radius)}');
6. }
8. num calculateCircleArea(num radius) => pi \* radius \* radius;

Selain const, opsi lain untuk menghindari perubahan nilai variabel setelah diinisialisasi adalah final. Apa bedanya final dan const?

Variabel yang dideklarasikan sebagai const bersifat *compile-time constants*, artinya nilai tersebut harus sudah diketahui sebelum program dijalankan. Sedangkan final, nilainya masih bisa diinisialisasi ketika *runtime* atau ketika aplikasi berjalan. Sebagai contoh kode berikut:

1. final firstName = stdin.readLineSync();
2. final lastName = stdin.readLineSync();
4. //  lastName = 'Dart'; Tidak bisa mengubah nilai
6. print('Hello $firstName $lastName');

Kita masih bisa menerima input dan menyimpannya pada variabel firstName dan lastname, namun nilai variabel tersebut tidak bisa diubah setelah diinisialisasi.

Variabel yang nilainya tidak bisa berubah dikenal dengan***immutable variable***. *Mutability* ini memungkinkan kita terhindar dari *bug* yang tidak terduga karena terjadi perubahan nilai. Maka dari itu jika Anda yakin bahwa variabel Anda nilainya tetap, gunakanlah const atau final.

Jadi kapan kita harus menggunakan const dan kapan final? Kapan pun memungkinkan, selalu gunakan const karena *compile-time constant* memiliki performa yang lebih baik dan menggunakan memori lebih sedikit. Jika tidak memungkinkan untuk menggunakan const, gunakan final untuk melindungi variabel agar tidak berubah.