**Type Inference**

Seperti yang kita tahu Dart mendukung *type inference*. Dart memiliki *analyzer* yang dapat menentukan menyimpulkan tipe untuk *field*, *method*, variabel lokal, dan beberapa tipe argumen *generic*. Ketika *analyzer* tidak memiliki informasi yang cukup untuk menyimpulkan tipe tertentu, maka tipe dynamic akan digunakan.

Misalnya berikut ini adalah contoh penulisan variabel *map* dengan tipe yang eksplisit:

1. Map<String, dynamic> company = {'name': 'Dicoding', 'yearsFounded': 2015};

Atau, Anda dapat menggunakan var dan Dart akan menentukan tipenya.

1. var company = {'name': 'Dicoding', 'yearsFounded': 2015}; // Map<String, Object>

Type inference menentukan tipe dari entri kemudian menentukan tipe dari variabelnya. Pada contoh di atas, kedua *key* dari *map* adalah String, namun nilainya memiliki tipe yang berbeda, yaitu String dan int, di mana keduanya merupakan turunan dari Object. Sehingga variabel company akan memiliki tipe Map<String, Object>.

Saat menetapkan nilai objek ke dalam objek lain, kita bisa mengganti tipenya dengan tipe yang berbeda tergantung pada apakah objek tersebut adalah *consumer* atau *producer*. Perhatikan *assignment* berikut:

1. Fish fish = Fish();

Fish fish adalah *consumer* dan Fish() adalah *producer*. Pada posisi *consumer*, aman untuk mengganti *consumer* bertipe yang spesifik dengan tipe yang lebih umum. Jadi, aman untuk mengganti Fish fish dengan Animal fish karena Animal adalah *supertype* dari Fish.

1. Animal fish = Fish();

Namun mengganti Fish fish dengan Shark fish melanggar *type safety* karena bisa saja Fish memiliki *subtype* lain dengan perilaku berbeda, misalnya FlyingFish.

1. Shark fish = Fish();  // Error

Pada posisi *producer*, aman untuk mengganti tipe yang umum (*supertype*) dengan tipe yang lebih spesifik (*subtype*).

1. Fish fish = Shark();