**Data Types**

Pada materi sebelumnya kita telah mempelajari tentang variabel yang dapat menyimpan nilai. Jadi bagaimana komputer membedakan antara variabel yang bernilai angka atau teks? Dan kenapa penting untuk bisa membedakannya?

Dart memiliki banyak tipe data yang mewakili jenis data yang dapat kita gunakan dan bagaimana data tersebut dioperasikan. Dengan tipe data, komputer dapat menghindari operasi yang tidak mungkin serta bisa menghasilkan *bug*, misalnya seperti perkalian karakter alfabet atau mengubah angka menjadi kapital.

Dart adalah bahasa yang mendukung *type inference*. Ketika Anda mendeklarasikan variabel dengan var, Dart akan secara otomatis menentukan tipe datanya. Misalnya :

1. var greetings = 'Hello Dart!';  // String
2. var myAge = 20;               // integers

Komputer akan tahu bahwa variabel greetings memiliki nilai berupa String atau teks dan variabel myAge bernilai angka atau integers meskipun kita tidak mendefinisikannya secara eksplisit.

Anda tetap bisa mendeklarasikan tipe data variabel secara eksplisit untuk menghindari kebingungan dan memudahkan proses *debugging*.

1. String greetings = 'Hello Dart!';
2. int myAge = 20;

Beberapa tipe data yang didukung oleh Dart antara lain:

| **Tipe** | **Deskripsi** | **Contoh** |
| --- | --- | --- |
| int | Integer (bilangan bulat) | 5, -7, 0 |
| double | Bilangan desimal | 3.14, 18.0, -12.12 |
| num | Bilangan bulat dan bilangan desimal | 5, 3.14, -99.00 |
| bool | Boolean | true, false |
| String | Teks yang terdiri dari 0 atau beberapa karakter | ‘Dicoding’, ‘Y’, ‘’ |
| List | Daftar nilai | [1, 2, 3], [‘a’, ‘b’, ‘c’] |
| Map | Pasangan key-value | {“x”: 4, “y”: 10} |
| dynamic | Tipe apa pun |  |

Dart mendukung *type inference*, menariknya ketika kita mendeklarasikan variabel tanpa melakukan inisialisasi, variabel akan memiliki tipe ***dynamic***. Karena sebuah variabel bernilai *dynamic* bisa berisi tipe apa pun, maka tidak ada masalah jika kita mengubah nilai di dalamnya.

1. var x;  // dynamic
2. x = 7;
3. x = 'Dart is great';
4. print(x);

Kode di atas tetap bisa berjalan dan menampilkan pesan ‘*Dart is great*’ tanpa ada masalah. Berbeda jika kita menginisialisasi nilai variabel x secara langsung. Akibatnya, *editor* akan menampilkan eror karena terjadi perubahan tipe data.

1. var x = 7;  // int
2. x = 'Dart is great'; // Kesalahan assignment
3. print(x);

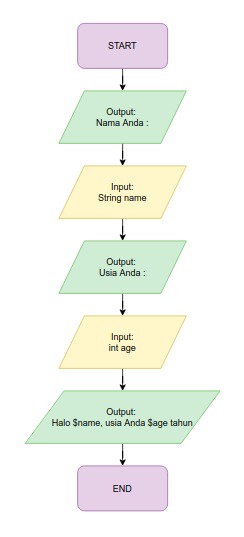
**Menerima input pengguna**

Selain menampilkan pesan ke konsol, kita juga dapat menerima input pengguna untuk selanjutnya diproses dan ditampilkan. Ini memungkinkan kita dapat membuat aplikasi yang interaktif dengan pengguna.

Untuk bisa menerima input, kita akan menggunakan statement stdin.readLineSync(). Fungsi ini merupakan bagian dari *library* dart:io, sehingga kita perlu mengimpor *library* tersebut.

1. import 'dart:io';

Kali ini kita akan membuat aplikasi sederhana yang menerima input nama dan usia dari pengguna lalu menampilkan pesan ke konsol. Sebelum melihat kode solusi di bawah bisakah Anda menerjemahkan *flowchart* berikut menjadi kode aplikasi?

[](https://www.dicoding.com/academies/191/tutorials/7503?from=7500)

**Kode solusi**

Pada persoalan di atas kita perlu menampilkan output, menerima beberapa input, dan menampilkan output lagi sesuai input yang diberikan. Kode yang perlu Anda tulis kurang lebih adalah seperti berikut:

1. import 'dart:io';
3. void main() {
4. stdout.write('Nama Anda : ');
5. String name = stdin.readLineSync();
6. stdout.write('Usia Anda : ');
7. int age = int.parse(stdin.readLineSync());
8. print('Halo $name, usia Anda $age tahun');
9. }

Jika kode Anda berbeda karena menggunakan print() maka tidak masalah. *Statement* print() dan stdout.write() memiliki fungsi yang sama yaitu untuk menampilkan suatu objek ke konsol. Yang membedakan adalah print() akan mencetak baris baru setelah menampilkan sesuatu sehingga selanjutnya Anda perlu memasukkan input pada baris baru. Sementara stdout.write() hanya menampilkan objeknya saja dan ketika ada input atau output baru lagi masih akan ditampilkan di baris yang sama.

Kode baru lain adalah int.parse(). Kita menggunakan kode ini untuk mengkonversi tipe data String menjadi int. Input yang diambil dari stdin.readLineSync() akan memiliki tipe data berupa String. Sehingga ketika ingin menyimpan dan mengoperasikan input dalam tipe data lain kita perlu melakukan konversi terlebih dahulu.