

Dasar Pemrograman Bahasa Dart

Fadjar Fathurrahman

1 Tujuan

- Mengenal dan menggunakan tools yang tersedia pada Dart SDK
- Memahami konsep dasar bahasa pemrograman Dart
- Mampu membuat program sederhana pada console dengan menggunakan Dart

2 Persiapan

Pastikan bahwa program-program atau pengaturan berikut ini berjalan dengan baik pada komputer yang Anda gunakan.

2.1 (Windows only) cmdr

Pastikan bahwa cmdr dengan Git for Windows sudah terpasang dan berjalan dengan baik. Jika belum silakan pasang terlebih dahulu. cmdr dapat diunduh pada tautan berikut:

<https://github.com/cmdrdev/cmdr/releases/download/v1.3.16/cmdr.zip>

2.2 Flutter SDK

Pastikan bahwa Flutter SDK sudah terpasang dengan baik. Kita akan mengasumsikan bahwa yang digunakan adalah Flutter channel beta yang memungkinkan kita untuk membuat aplikasi Web dan juga Android atau iOS.

Flutter SDK (channel beta) dapat diunduh pada tautan berikut ini. Cari Beta Channel dan pilih yang file yang sesuai dengan sistem operasi yang Anda gunakan.

<https://flutter.dev/docs/development/tools/sdk/releases>

Flutter SDK sudah memiliki Dart SDK sehingga kita tidak perlu menginstal Dart SDK secara terpisah.

2.3 Setting PATH

Pastikan bahwa Anda sudah dapat mengakses perintah flutter dan dart pada cmdr. Jika belum, Anda dapat menambahkannya dengan cara mengedit file

```
C:\Users\jendela\cmdr\config\user_profile.cmd
```

Silakan menyesuaikan bagian

```
\txtline{C:\Users\jendela\cmdr}
```

menjadi lokasi di mana cmdr berada.

Pada file user_profile.cmd, silakan tambahkan baris berikut

```
set "PATH=C:\Users\jendela\flutter\bin;%PATH%"
set "PATH=C:\Users\jendela\flutter\bin\cache\dart-sdk\bin"
```

2.4 Visual Studio Code atau text editor lain

Visual Studio Code (VSCode) dapat diunduh pada tautan berikut.

<https://code.visualstudio.com/download>

Pasang ekstensi Flutter pada VSCode. Instruksi terkait dapat ditemukan pada tautan-tautan berikut:

<https://flutter.dev/docs/get-started/editor?tab=vscode>

<https://flutter.dev/docs/development/tools/vs-code>

3 Struktur kode program Dart

Bahasa pemrograman Dart memiliki banyak kesamaan dengan bahasa pemrograman C/C++.

Program Dart memiliki satu entry point, yaitu fungsi `main`. Fungsi ini biasanya diberikan classifier atau tipe `void`.

```
void main() {  
    // kode program  
}
```

Suatu program Dart dapat terdiri dari beberapa fungsi yang dapat didefinisikan pada di satu file yang sama.

Dart mendukung tiga tipe komentar.

```
// Inline comment  
  
/*  
Block comment.  
Bisa mencakup lebih dari satu baris  
*/  
  
/// Untuk Dokumentasi  
///  
/// Berikut ini adalah dokumentasi kode.
```

Tipe data dasar pada Dart mirip dengan tipe data yang ada pada C/C++. Contoh: `int`, `double` dan `String`. Perbedaan adalah semua data tersebut pada Dart merupakan suatu object. Kita akan mempelajari mengenai object-oriented programming pada praktikum selanjutnya.

4 Latihan Program

Program Dart ditulis dalam file dengan ekstensi `.dart`.

Untuk program pertama, silakan buka VSCode dan ketikkan kode berikut ini. Beri nama file tersebut dengan nama `hello.dart` atau nama yang lain. Usahakan nama tersebut tidak mengandung karakter spasi atau karakter tidak umum yang lain. Ingat lokasi di mana Anda menyimpan file ini. Sangat disarankan untuk menyimpan file ini pada satu direktori atau folder tersendiri.

```
void main() {  
    print('Hello Dart');  
}
```

Buka cmd, kemudian pindah ke direktori atau folder di mana Anda menyimpan file `hello.dart` sebelumnya. Gunakan perintah `cd` untuk berpindah dari satu direktori ke direktori lainnya. Jika hal ini sudah dilakukan, ketikkan perintah berikut ini pada cmd. Apa yang Anda amati?

```
dart hello.dart
```

Jalan juga perintah berikut.

```
dart2native hello.dart
```

Perintah ini akan menghasilkan file baru. Bisakah Anda menemukan file tersebut?

Perhatikan bawah, pada sistem operasi Windows Anda mungkin perlu menambahkan ekstensi `.exe` sehingga perintah sebelumnya menjadi `dart.exe` dan `dart2native.exe`

4.1 Tugas 1

Perhatikan kode berikut. Ketikkan pada file yang berbeda.

```
void main() {  
    sayHello();  
}  
  
void sayHello() {  
    print('Hello Dart');  
}
```

Dengan menggunakan pengetahuan bahasa C yang sudah Anda pelajari, apakah output dari program tersebut? Gunakan perintah `dart` atau `dart2native`.

4.2 Tugas 2

Perhatikan kode berikut. Ketikkan pada file yang berbeda.

```
void main() {  
    String nama = 'Darto';  
    sayHello(nama);  
}  
  
void sayHello(String name) {  
    print('Hello $name');  
}
```

Apakah keluaran dari program ini?

4.3 Tugas 3

Perhatikan kode berikut. Ketikkan pada file yang berbeda.

```

void main() {
    String Nama = 'Darto';
    String Name = 'Parto';
    sayHello(nama);
    sayHello(name);
}

void sayHello(String nama) {
    print('Hello $name');
}

```

Apakah keluaran dari program ini? Apakah ada error yang dilaporkan? Apakah Anda dapat memperbaiki program ini?

4.4 Tugas 4

Perhatikan kode program berikut.

```

void main() {
    List<String> daftar_nama = [
        'Darto',
        'Parto',
        'Sule',
        'Andre',
        'Nunung'
    ];

    for(var nama in daftar_nama) {
        sayHello(nama);
    }
}

void sayHello(String name) {
    print('Hello $name');
}

```

Apakah keluaran dari program tersebut?

Program ini mengenalkan penggunaan tipe data (atau kelas) List yang mirip dengan array pada C/C++.

4.5 Tugas 5

Perhatikan program berikut.

```

import 'dart:io';

void main() {
    stdout.writeln('Say hello to my bro');
    String input = stdin.readLineSync();
    sayHello(input);
}

void sayHello(String name) {

```

```
print('Hello $name, my bro. Nice to meet you');  
}
```

Apakah keluaran dari program tersebut?

Catatan: Anda perlu menginputkan nama Anda atau teks lain, diikuti dengan tekan tombol Enter pada keyboard.

Bandingkan program tersebut dengan program berikut ini. Apakah ada perbedaan yang Anda amati?

```
import 'dart:io';  
  
void main() {  
  stdout.writeln('Say hello to my bro');  
  stdout.write('Your name bro? ');  
  String input = stdin.readLineSync();  
  sayHello(input);  
}  
  
void sayHello(String name) {  
  print('Hello $name, my bro. Nice to meet you');  
}
```

Dari pengamatan di atas apakah Anda dapat memberikan perbedaan atau persamaan antara:

- **print**
- stdout.write
- stdout.writeln

5 Tugas tambahan 1

Apakah yang dimaksud dengan variabel PATH dan environment variable pada Windows / OSX / Linux ? Bagaimana cara mengubah variabel ini? Bacaan berikut ini dapat dijadikan acuan (salah satunya).

<https://superuser.com/questions/284342/what-are-path-and-other-environment-variables-and-how-can-i-set>

[https://en.wikipedia.org/wiki/PATH_\(variable\)](https://en.wikipedia.org/wiki/PATH_(variable))

<https://www.computerhope.com/issues/ch000549.htm>

<https://superuser.com/questions/502358/easier-way-to-change-environment-variables-in-windows-8>

6 Tugas tambahan 2

Sintaks untuk melakukan percabangan/kondisional dan looping pada Dart juga sangat mirip dengan C/C++. Buatlah masing-masing 4 program sederhana untuk demonstrasi percabangan dan looping pada Dart.

Referensi:

<https://dart.dev/guides/language/language-tour>