MOBILE APPLICATION 2

PERTEMUAN 1



**DESKRIPSI MATAKULIAH**

**Mata Kuliah ini mempelajari konsep bahasa pemrograman Dart, membuat aplikasi mobile dengan flutter menggunakan bahasa Dart, menerapkan widget pada flutter untuk membuat aplikasi mobile**

**CAPAIAN PEMBELAJARAN**

**Mahasiswa dapat memahami konsep dasar bahasa pemrograman Dart**

**DOSEN PENGAMPU**

**Taufikqurrohman, M.Kom**

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER**

**(STMIK) WIDYA PRATAMA**

[Pokok bahasan/bab/pertemuan]

**CAPAIAN PEMBELAJARAN**

**Mahasiswa dapat memahami konsep dasar bahasa pemrograman Dart**

**MATERI PEMBELAJARAN**

**1. Pengenalan bahasa Dart**

**2. Variabel dan Tipe data**

**3. Operator**

**MATERI**

1. **Pengenalan Dart**
2. Pengertian

Dart adalah adalah bahasa berorientasi objek(Object Oriented) dengan  
sintaksis (Syntax) C-style yang dapat diubah secara opsional menjadi JavaScript.  
Mendukung berbagai macam alat bantu pemrograman seperti antarmuka(interface), class, collection, generics, dan opsional typing Kita bisa menggunakan Dart untuk membuat aplikasi Web Android iOS dan juga menjalankan Server. Simple nya dengan menggunakan dart kita dapat membuat UI(User Interface) yang indah dan berkualitas pada setiap device dengan menggunakan:

- Bahasa yang mengoptimalkan client

Dart pertama kali dioptimalkan untuk web apps dan berevolusi untuk  
membantu oengembangan Mobile App. Dart juga dapat kita gunakan untuk  
menjalankan Command Line juga Server-Side.

- Kaya akan Framework

- Tool yang Fleksibel dan menyenangkan

Tool yang dapat digunakan akan bisa digunakan untuk berbagai tujuan. Tool yang lebih direkomendasikan oleh dart adalah Flutter.

1. Mengapa Dart?

Developers yang bekerja di Google dan perusahaan besar lainnya menggunakan  
dart untuk membangun aplikasi Android iOS dan Web yang berkualitas.Dart memberikan  
fitur yang ​*Client Side Development* ​(Pengembangan dari sisi client) yang oleh karena inilah banyak​ ​developer yang memilih menggunakan Dart.

1. Mudah dipelajari

- Dart memiliki banyak kemiripan dengan bahasa pemrograman yang banyak  
digunakan oleh developers(Java, C++, PHP , Java Script...dll).

- Kita bisa jadi tanpa disadari sudah dapat menggunakan dart karena  
kemiripan nya.

- Dart akan lebih mudah dipelajari jika kita sebelum nya sudah memiliki  
pengalaman dalam menggunakan bahasa pemrograman yang ber ​*Object  
Oriented* ​seperti Java ataupun C++.

1. CodeBase​yang sudah di compile​ *Natively*​(bawaan)

- Framework lain memberikan kita sedikit akses untuk menggunakan codingan  
kita pada Platform yang berbeda. Berbeda dengan Dart..

- Dart memberikan kita izin penuh untuk membuat satu aplikasi yang  
codingannya dapat digunakan di berbagai platform. Aplikasi yang kita buat  
akan dapat digunakan pada Android juga iOS.

- Dart tidak hanya dapat kita gunakan untuk mobile develop kita juga dapat  
menggunakan Dart untuk Web Development.

1. Produktif

- Cepat dan mudah dalam Layouting dan menambahkan Feature pada Project.  
- Layout juga dapat kita buat dengan menggunakan Codingan.

1. Compile AoT dan JiT.

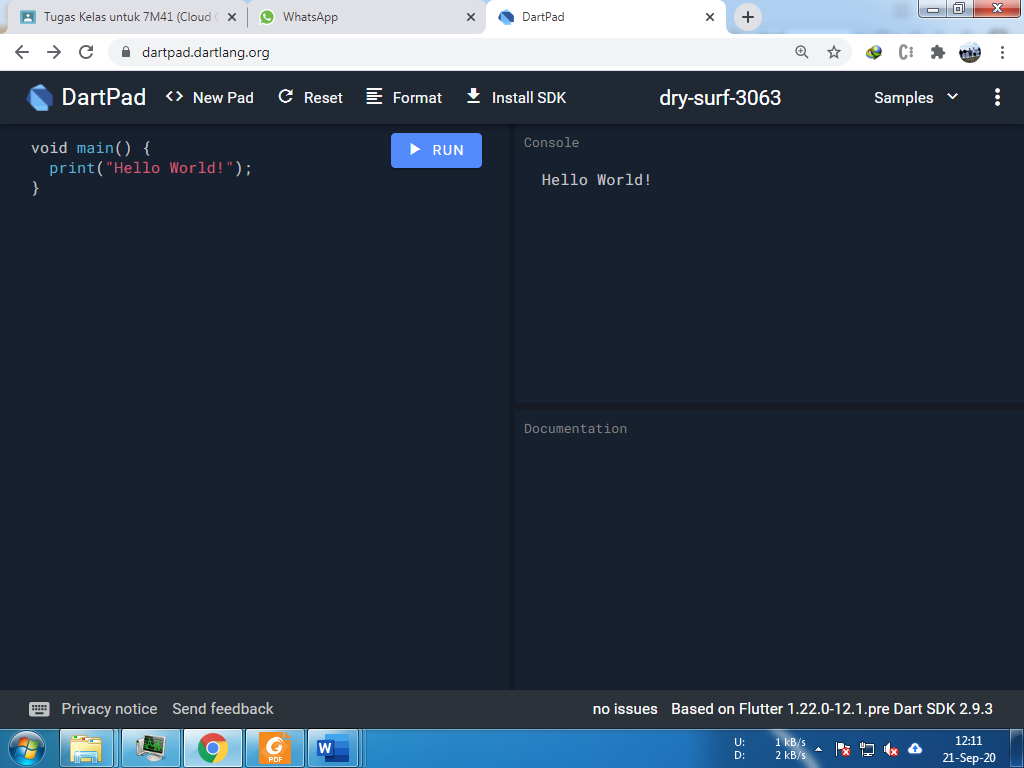
- Perubahan pada project dapat kita lihat secara Instan,pada aplikasi.

- Tidak perlu melakukan Recompile yang memakan banyak waktu. Kita juga tidak perlu untuk menunggu project di load ulang, untuk melihat perubahan.  
- Tinggal Save dan perubahan akan terlihat.

- Ini dikarenakan Framework yang dapat ​*meng compile*​ Ahead of Time(AoT)/Lebih Cepat dan Just in Time(JiT) / Tepat waktu.

1. First Code on Dart

Kita dapat menguji skrip yang kita buat secara online dengan menggunakan editor  
online di ​https://dartpad.dartlang.org/ ​**.** Editor Dart menjalankan skrip dan menampilkan  
HTML serta Console Output.



Cobakan Codingan “Hello World” di Dartpad.

main() {

print("Hello World!");

}

Dartpad akan menampilkan output seperti ini. Fungsi ​**main( )** adalah metode yang telah ditentukan di Dart. Metode ini berfungsi sebagai titik masuk ke aplikasi. Script Dart membutuhkan metode ​**main( )** untuk eksekusi program. ​**print( )** adalah fungsi yang telah ditetapkan yang mencetak string atau nilai tertentu ke output standar.  
Sebelum memasuki pembahasan lebih dalam tentang Dart kita akan membahas  
beberapa info yang perlu diketahui sebelum memulai ​*Coding*​ menggunakan Dart.

1. Identifier pada Dart

Pengidentifikasi(identifier) adalah nama yang diberikan kepada elemen dalam  
program seperti variabel, fungsi, dll. Rules(Aturan) untuk identifier adalah -  
Identifier dapat menyertakan karakter, dan digit(angka). Namun, identifier tidak  
dapat dimulai dengan digit.

- Identifier tidak dapat berisikan simbol-simbol spesial selain underscore ( ​**\_**​ )  
dan tanda dolar ( ​**$**​ ).

- Identifier tidak dapat berisikan ​*Keyword*​ \*

- Identifier harus ​***unik***

- Identifier tidak dapat berisikan spasi(space)  
Berikut beberapa contoh identifier yang valid(dapat digunakan) dan  
invalid identifier(yang tidak dapat digunakan).

|  |  |
| --- | --- |
| **valid** | **Invalid** |
| firstName | Var |
| first\_name | first name |
| num1 | first-name |
| $result | 1number |

1. Keywords in Dart

Keywords memiliki makna spesial dalam context bahasa pemrograman.  
Dibawah ini adalah list dari Keyword yang ada pada Dart.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| abstract 1 | continue | false | new | this |
| as 1 | default | final | null | throw |
| assert | deferred 1 | finally | operator 1 | true |
| async 2 | do | for | part 1 | try |
| async\* 2 | dynamic 1 | get 1 | rethrow | typedef 1 |
| await 2 | else | if | return | var |
| break | enum | implements 1 | set 1 | void |
| case | export 1 | import 1 | static 1 | while |
| catch | external 1 | in | super | with |
| class | extends | is | switch | yield 2 |
| const | factory 1 | library 1 | sync\* 2 | yield\* 2 |

1. Whitespace dan Line Breaks

Dart mengabaikan spasi, tab, dan baris baru yang muncul dalam program. Kita  
dapat menggunakan spasi, tab, dan baris baru secara bebas dalam program kita dan kita  
bebas untuk memformat dan memasukkan program kita dengan cara yang rapi dan  
konsisten yang membuat kode mudah dibaca dan dipahami.

1. Dart adalah case-sensitive

Dart adalah case-sensitive. Ini berarti bahwa Dart membedakan antara huruf besar dan huruf kecil.kesalahan penggunaan huruf besar dan huruf kecil sering terjadi pada pembuatan identifier dan class. Untuk itu harap lebih diperhatikan lagi dalam penamaan pada Dart!

1. Pernyataan (statement) diakhiri dengan Semicolon

Setiap baris instruksi disebut pernyataan. Setiap pernyataan Dart harus diakhiri  
dengan titik koma (;). Satu baris dapat berisi beberapa pernyataan. Namun, pernyataan ini harus dipisahkan dengan titik koma.

1. Comment di Dart

Komentar (Comment) adalah cara untuk meningkatkan keterbacaan suatu program. Comment dapat digunakan untuk memasukkan informasi tambahan tentang program seperti penulis kode, petunjuk tentang fungsi / konstruksi dll. Comment akan diabaikan oleh kompilator. Dart dapat berisikan jenis komentar berikut :

* Single Line Comment ( // )

Teks apapun yang berada diantara " // " dan akhir baris diperlakukan sebagai komentar

* Multi Line Comments ( /\* \*/ )

Komentar ini dapat mencakup banyak baris.

|  |  |
| --- | --- |
| Single Line Comments | Multi Line Comments |
| main() {  //Single Line Comment  print("Hello World")  } | main() { /\*Multi Line Comment \*/ print("Hello World") } |

1. Syntax pada Dart

Syntax mendefinisikan seperangkat aturan untuk menulis program. Setiap spesifikasi bahasa mendefinisikan sintaksnya sendiri. Sintaks Dart terdiri dari:

- Variable

- Operators

- Class

- Function

- Loop

- Comments

- Library

- Package

- Typedefs

- Decision making

- Expression dan Programming Construct (konstruksi pemrograman)

- Data Structure yang direpresentasikan sebagai Generics/Collection