

Materi Flutter Development

A. Apa itu Flutter ?

Penjelasan singkatnya, Flutter merupakan sebuah tools (Software Development Kit) yang digunakan untuk membuat beragam aplikasi (Website, Android Mobile, iOS Apps, bahkan Desktop Apps) hanya dengan satu basis pengkodean (Single Code Base). Flutter merupakan project open-source yang dipelihara oleh Google sejak 2018, walaupun sudah ada versi alpha pada tahun 2017. Dan teknologi Flutter ini menggunakan bahasa pemrograman Dart yang digunakan sebagai pengganti JavaScript yang dinilai masih banyak memiliki kekurangan.

B. Dart

Dart merupakan bahasa pemrograman yang digunakan pada pengembangan Flutter. Oleh karena itu, materi Dart ini perlu dipelajari guna menunjang keberlangsungan pembelajaran Flutter kedepannya.

Karakteristik Dart :

- Statically typed

Dart adalah bahasa yang statically typed, kita perlu mendefinisikan variabel sebelum digunakan.

```
var name = "Hello World";  
  
print(name);
```

- Type inference

Karakteristik ini dapat membuat suatu variable akan mengikuti tipe data dari valuenya. Namun, hal tersebut tidak berlaku jika kita sudah memberi tipe data pada variabelnya.

```
// variable name secara otomatis jadi String  
var name = "Hello World";  
  
// tidak akan berubah ke int -> malah jadi error  
String name = 10;  
  
print(name);
```

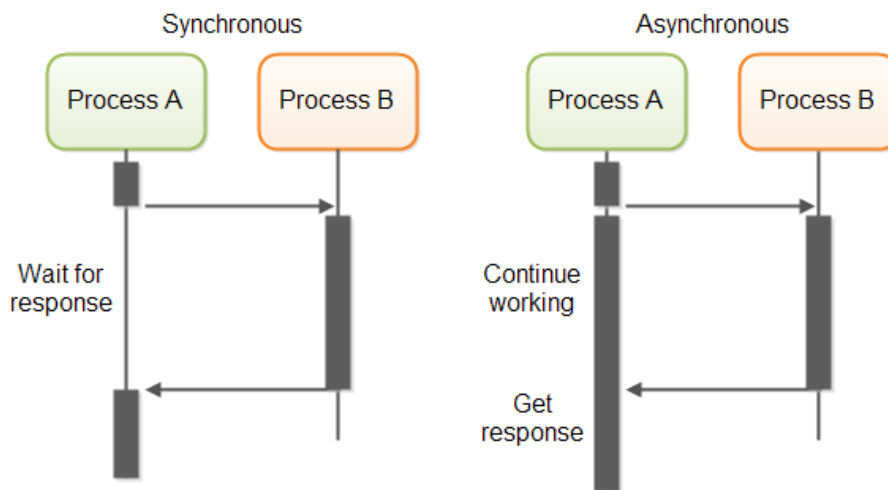
- String expressions
Kita dapat menggunakan tanda \$ untuk menampilkan suatu variable bertipe apapun ke bentuk String. Hal ini dapat kita sebut sebagai String Interpolation.

```
var name = "Hello World";
String language = "Dart";

print("$name – bahasa yang digunakan : $language");
```

- Object Oriented Programming
Dart berfokus pada paradigma OOP yang mana berorientasi pada object. Penggunaannya seperti apa ? kita akan bahas pada materi tentang OOP.

Pada bahasa Dart, terdapat 2 jenis pengekseskusion kode. Kode akan dijalankan secara synchronous (sync) atau asynchronous (async). Berikut merupakan gambaran perbedaan keduanya :



C. Object Oriented Programming

OOP merupakan sebuah paradigma pemrogramman berorientasi object. Hal ini bertujuan agar project kita lebih mudah untuk digunakan dan juga untuk dipelihara. Jika dianalogikan konsep OOP seperti apa, simak gambar berikut ini :



Analogi berjualan tanpa konsep OOP

Jika dilihat dari gambar di atas, gambar tersebut merupakan analogi suatu project yang kita buat tanpa berdasarkan orientasi pada suatu object. Kita tidak memisahkan baju, celana, kaos kaki, kerudung / hijab dll. Baju merupakan sebuah object, celana adalah object, kaos kaki juga object, dan seterusnya. Jika kita menatanya berdasarkan kategori jenisnya (object), maka akan terlihat lebih indah, rapi, dan mudah untuk dicari seperti gambar berikut ini :



Analogi berjualan dengan konsep OOP

D. Widget Dasar

Semua Class pada Flutter bisa kita sebut sebagai widget. Kalau dianalogikan, jika di meja kita sekarang ada secangkir kopi, maka kopi tersebut adalah widget. Namun, perlu diperhatikan dari secangkir kopi tersebut, bahwa itu semua terdiri dari widget Cangkir, widget Kopi, widget Air, dan widget Gula. Oleh karena itu, untuk menjadi Flutter Developer Sejati, kita harus mengetahui beragam widget sebanyak-banyaknya.

- **Material App**
Widget MaterialApp merupakan widget yang paling penting, kenapa ? karena widget ini berfungsi sebagai wadah kita dalam membuat tampilan aplikasi. Kalau dianalogikan, widget ini berfungsi sebagai papan kayu untuk melukis.

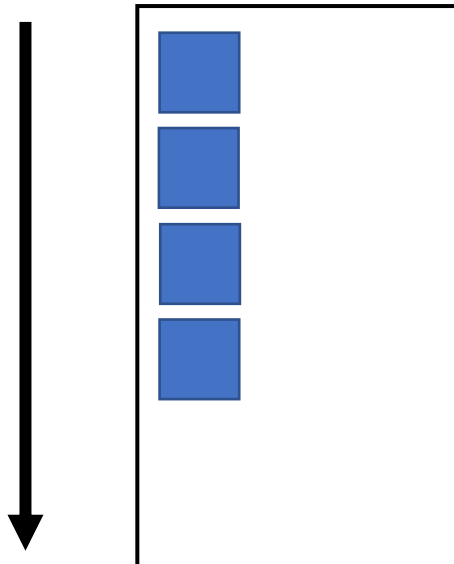


- **Scaffold**
Jika widget MaterialApp merupakan papan kayu untuk melukis, maka widget Scaffold ini bisa diumpamakan sebagai canvas untuk melukisnya.

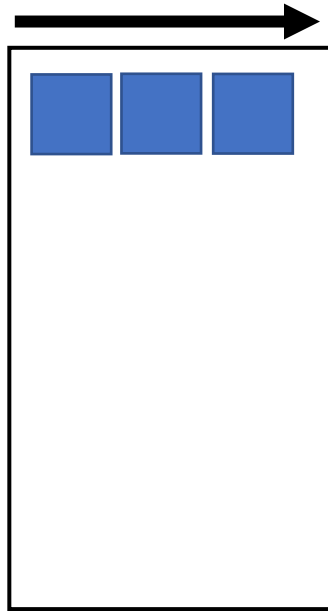


Untuk membuat lukisan yang baru, kita bisa mengganti canvas ini. Sama halnya dengan mengganti halaman (page) pada Flutter. Kita memerlukan widget Scaffold lagi pada page yang baru.

- Column
Untuk menyusun satu atau lebih widget secara vertical, maka widget ini adalah pilihan yang tepat.



- Row
Untuk menyusun satu atau lebih widget secara horizontal, maka widget ini adalah pilihan yang tepat.



- Stack
Untuk menyusun satu atau lebih widget pada sumbu Z (depan – belakang), maka widget ini adalah pilihan yang tepat.

