

#02 | Pengantar Bahasa Pemrograman Dart - Bagian 1

Nama : Muhammad Taufiq Abdus Salam (22)

Kelas: SIB 3D

Tugas Praktikum

Soal 1

Modifikasilah kode pada baris 3 di VS Code atau Editor Code favorit Anda berikut ini agar mendapatkan keluaran (*output*) sesuai yang diminta!

```
1 void main() {  
2   for (int i = 0; i < 10; i++) {  
3     print('hello ${i + 2}');  
4   }  
5 }
```

Output yang diminta:

```
Console  
  
Nama saya adalah Fulan, sekarang berumur 18  
Nama saya adalah Fulan, sekarang berumur 17  
Nama saya adalah Fulan, sekarang berumur 16  
Nama saya adalah Fulan, sekarang berumur 15  
Nama saya adalah Fulan, sekarang berumur 14  
Nama saya adalah Fulan, sekarang berumur 13  
Nama saya adalah Fulan, sekarang berumur 12  
Nama saya adalah Fulan, sekarang berumur 11  
Nama saya adalah Fulan, sekarang berumur 10  
Nama saya adalah Fulan, sekarang berumur 9
```

Jawab:

Kode:

```

bin > dart_application_1.dart > ...
1  import 'package:dart_application_1/dart_application_1.dart' as dart_application_1;
2
Run | Debug
3  void main() {
4      for (int i = 18 ; i > 8; i--) {
5          print('Nama saya adalah Taufiq, sekarang berumur ${i+0}');
6      }
7  }
8

```

Out put:

```

Connecting to VM Service at ws://127.0.0.1:49890/ZMQ--rtm5Zs=/ws
Connected to the VM Service.
Nama saya adalah Taufiq, sekarang berumur 18
Nama saya adalah Taufiq, sekarang berumur 17
Nama saya adalah Taufiq, sekarang berumur 16
Nama saya adalah Taufiq, sekarang berumur 15
Nama saya adalah Taufiq, sekarang berumur 14
Nama saya adalah Taufiq, sekarang berumur 13
Nama saya adalah Taufiq, sekarang berumur 12
Nama saya adalah Taufiq, sekarang berumur 11
Nama saya adalah Taufiq, sekarang berumur 10
Nama saya adalah Taufiq, sekarang berumur 9

```

Soal 2

Mengapa sangat penting untuk memahami bahasa pemrograman Dart sebelum kita menggunakan framework Flutter ? Jelaskan!

Jawab:

Memahami bahasa pemrograman Dart sangat penting sebelum menggunakan framework Flutter karena Dart merupakan dasar dari Flutter itu sendiri. Semua komponen, widget, dan logika aplikasi dalam Flutter ditulis menggunakan Dart, sehingga tanpa pemahaman yang kuat terhadap bahasa ini, Anda akan kesulitan untuk mengembangkan aplikasi dengan efektif.

Soal 3

Rangkumlah materi dari codelab ini menjadi poin-poin penting yang dapat Anda gunakan untuk membantu proses pengembangan aplikasi mobile menggunakan framework Flutter.

Jawab:

1. Getting started with Dart

Dart bertujuan untuk menggabungkan kelebihan-kelebihan dari sebagian besar bahasa tingkat tinggi dengan fitur-fitur bahasa pemrograman terkini, antara lain sebagai berikut:

- **Productive tooling:** merupakan fitur kakas (*tool*) untuk menganalisis kode, plugin IDE, dan ekosistem paket yang besar.
- **Garbage collection:** untuk mengelola atau menangani dealokasi memori (terutama memori yang ditempati oleh objek yang tidak lagi digunakan).
- **Type annotations (opsional):** untuk keamanan dan konsistensi dalam mengontrol semua data dalam aplikasi.
- **Statically typed:** Meskipun *type annotations* bersifat opsional, Dart tetap aman karena menggunakan fitur *type-safe* dan *type inference* untuk menganalisis *types* saat *runtime*. Fitur ini penting untuk menemukan *bug* selama kompilasi kode.
- **Portability:** bahasa Dart tidak hanya untuk web (yang dapat diterjemahkan ke JavaScript) tetapi juga dapat dikompilasi secara *native* ke kode **Advanced RISC Machines (ARM)** dan x86.

2. The evolution of Dart

- Awalnya berfokus pada pengembangan web, dengan tujuan utama menggantikan JavaScript, sekarang telah fokus pada mobile development, termasuk framework Flutter.
- **Mencoba memecahkan masalah pada JavaScript:** JavaScript tidak menyediakan ketahanan seperti banyak bahasa pemrograman lainnya, sehingga Dart ingin menjadi penerus daripada JavaScript.
- **Menawarkan performa terbaik dan alat yang lebih baik untuk proyek berskala besar:** Dart memiliki perkakas yang modern dan stabil yang telah disediakan oleh plugin IDE. Hal ini telah dirancang untuk mendapatkan performa terbaik dengan tetap menjaga nuansa bahasa yang dinamis.
- **Dibentuk agar kuat dan fleksibel:** Dengan tetap mempertahankan *type annotations* bersifat opsional dan menambahkan fitur OOP, Dart dapat menyeimbangkan dua fitur utama yaitu fleksibilitas dan ketangguhan.

3. How Dart works

Untuk memahami dari mana fleksibilitas bahasa dart itu berasal, kita perlu tahu bagaimana cara mengeksekusi kode Dart. Ini dapat dilakukan dengan dua cara, seperti yang dijelaskan berikut ini:

- Dart **virtual machines (VMs)**
- JavaScript compilations

4. Dart VM and JavaScript compilation

Kode Dart dapat dieksekusi pada lingkungan yang mendukung bahasa Dart. Lingkungan yang mendukung bahasa Dart perlu memperhatikan fitur-fitur penting seperti berikut:

- *Runtime systems*
- *Dart core libraries*
- *Garbage collectors*

5. Introducing the structure of the Dart language

Jika Anda sudah mengetahui beberapa bahasa pemrograman seperti bahasa C atau memiliki beberapa pengalaman dengan JavaScript, sebagian besar sintaks Dart akan lebih mudah bagi Anda untuk memahaminya. Dart menyediakan sebagian besar operator standar untuk memanipulasi variabel; *built-in types* adalah yang paling umum ditemukan dalam bahasa pemrograman tingkat tinggi. *Control flow* dan *function* sangat mirip dengan bahasa pemrograman lainnya.

- Object orientation
- Dart operators
- Arithmetic operators
- Increment and decrement operators
- Equality and relational operators
- Logical operators