

Nama :SYAVA APRILIA P

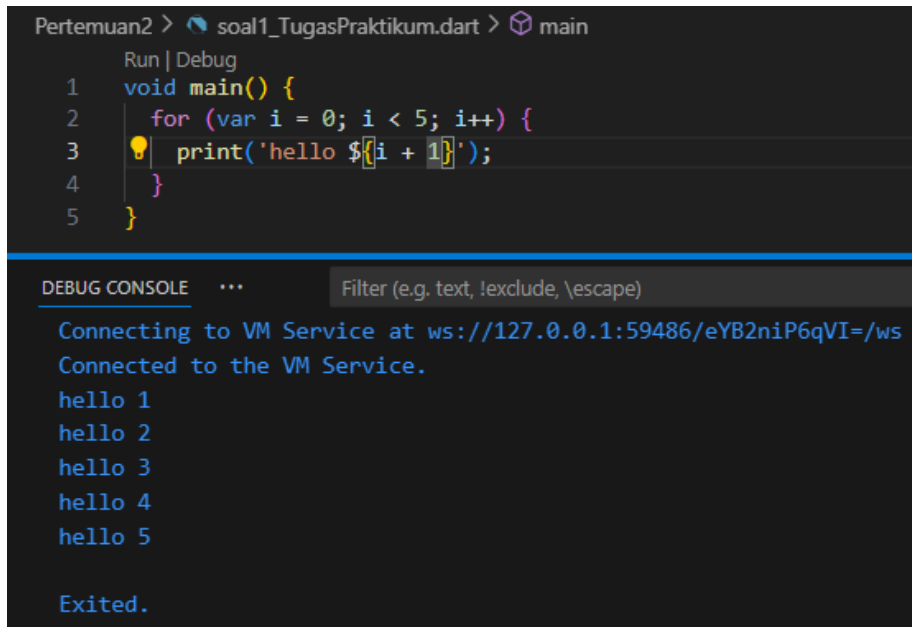
NIM :2241760129

Kelas/absen :SIB 3E

JOBSHEET 2 PERTEMUAN 2

#02 | Pengantar Bahasa Pemrograman Dart - Bagian 1

Hands-on with Dart



```
Pertemuan2 > soal1_TugasPraktikum.dart > main
Run | Debug
1 void main() {
2   for (var i = 0; i < 5; i++) {
3     print('hello ${i + 1}');
4   }
5 }

DEBUG CONSOLE ... Filter (e.g. text, !exclude, \escape)
Connecting to VM Service at ws://127.0.0.1:59486/eYB2niP6qVI=/ws
Connected to the VM Service.
hello 1
hello 2
hello 3
hello 4
hello 5

Exited.
```

Soal 1

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

```
Pertemuan2 > soal1_TugasPraktikum.dart > main
Run | Debug
1 void main() {
2   for (int i = 19; i >= 10; i--) {
3     print('Nama saya adalah Syava, sekarang berumur ${i - 1}');
4   }
5 }
```

DEBUG CONSOLE ... Filter (e.g. text, !exclude, \escape)

```
Connecting to VM Service at ws://127.0.0.1:63917/EXnt0Sd0C-c=/ws
Connected to the VM Service.
Nama saya adalah Syava, sekarang berumur 18
Nama saya adalah Syava, sekarang berumur 17
Nama saya adalah Syava, sekarang berumur 16
Nama saya adalah Syava, sekarang berumur 15
Nama saya adalah Syava, sekarang berumur 14
Nama saya adalah Syava, sekarang berumur 13
Nama saya adalah Syava, sekarang berumur 12
Nama saya adalah Syava, sekarang berumur 11
Nama saya adalah Syava, sekarang berumur 10
Nama saya adalah Syava, sekarang berumur 9

Exited.
```

Soal 2

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

Jawab:

Flutter Dibangun di Atas Dart

Memahami bahasa Dart adalah fondasi penting untuk bekerja dengan Flutter. Tanpa pemahaman yang mendalam tentang Dart, pengembang akan kesulitan dalam menulis kode yang efisien, memahami widget hierarchy, dan menggunakan fitur-fitur Flutter seperti asynchronous programming dan state management. Dart juga membuat pengembangan aplikasi dengan Flutter lebih lancar dan lebih mudah beradaptasi dengan kebutuhan aplikasi mobile.

Flutter adalah framework yang sepenuhnya menggunakan Dart sebagai bahasa pemrograman utamanya. Semua komponen Flutter, seperti widget, animasi, dan logika aplikasi, ditulis dengan Dart. Tanpa pemahaman yang kuat tentang Dart, sulit untuk menavigasi dan mengembangkan aplikasi Flutter secara 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:

Dart adalah inti dari Flutter, dirancang untuk pengembangan aplikasi modern. Memahami Dart penting untuk Flutter, mencakup tipe data, operator, dan OOP. Dalam codelab ini, Anda akan mempelajari dasar-dasar Dart untuk meningkatkan kenyamanan dalam eksperimen dan pengembangan Flutter.

Dart adalah bahasa yang menggabungkan keunggulan berbagai bahasa tingkat tinggi dengan fitur modern, seperti kaskas produktif, garbage collection, dan type annotations opsional. Meskipun opsional, Dart tetap aman dengan static typing dan type inference untuk menganalisis tipe selama runtime, membantu menemukan bug saat kompilasi. Dart juga portabel, bisa dikompilasi untuk web (dalam JavaScript) atau native (ARM dan x86). Semua pengembangan Flutter menggunakan Dart, termasuk kode aplikasi, plugin, dan manajemen dependensi. Memahami dasar Dart penting untuk produktivitas dalam pengembangan Flutter.

Dart diluncurkan pada tahun 2011 dan terus berkembang. Awalnya difokuskan untuk pengembangan web sebagai pengganti JavaScript, Dart kemudian beralih ke mobile development dengan Flutter. Dart dirancang untuk mengatasi kekurangan JavaScript, menawarkan performa tinggi dan alat modern untuk proyek besar. Dengan type annotations opsional dan fitur OOP, Dart menggabungkan fleksibilitas dan ketangguhan. Sebagai bahasa modern lintas platform, Dart terus meningkatkan fitur-fiturnya, menjadikannya pilihan ideal untuk framework Flutter.

Dart dapat mengeksekusi kode dengan dua cara utama: melalui Dart Virtual Machine (VM) dan kompilasi ke JavaScript. Pada diagram, kode Dart berada di atas dan bisa dijalankan di berbagai platform dengan sedikit perubahan.

Dua Mode Eksekusi Dart:

1. **Kompilasi Just-In-Time (JIT):** Kode dikompilasi saat runtime, digunakan saat pengembangan aplikasi. JIT memungkinkan fitur seperti debugging dan hot reload, yang mempercepat pengembangan dengan memberikan umpan balik langsung atas perubahan kode.
2. **Kompilasi Ahead-Of-Time (AOT):** Kode dikompilasi sebelumnya ke kode mesin asli. AOT menghasilkan performa lebih cepat tetapi tidak mendukung fitur seperti hot reload atau debugging.

Hot Reload:

Dart berperan penting dalam fitur hot reload Flutter, yang memanfaatkan JIT untuk memperbarui aplikasi secara instan saat kode diubah, memungkinkan pengembangan yang lebih cepat dan efisien.

Memahami cara kerja eksekusi Dart penting sebelum mendalami pengembangannya.

Dart memiliki sintaks yang mirip dengan C atau JavaScript dan mendukung operator standar, tipe data umum, serta alur kontrol yang mudah dipahami.

Pemrograman Berorientasi Objek (OOP)

Dart adalah bahasa OOP dengan konsep class, field, dan method. Prinsip seperti inheritance, polymorphism, dan encapsulation diterapkan, mirip dengan Java.

Operator Dart

Operator di Dart adalah method dalam class, seperti `x == y` yang sama dengan `x.==(y)`.

Operator Aritmatika

- Dart mendukung operator +, -, *, /, ~/ , dan %. Shortcut seperti += dan -= juga ada.

Operator Penambahan/Pengurangan

- ++ dan -- digunakan untuk menambah atau mengurangi nilai variabel.

Operator Persamaan/Relasional

- Operator seperti ==, !=, >, < digunakan untuk perbandingan nilai.

Operator Logika

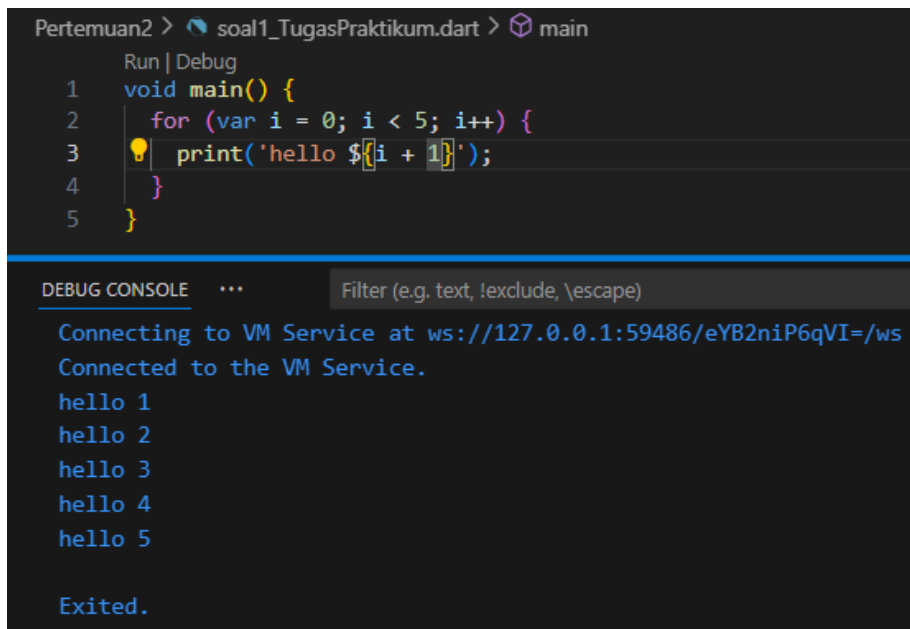
- Dart mendukung !, ||, dan && untuk operasi boolean.

Ini adalah dasar-dasar Dart yang dapat memudahkan dalam memahami dan mengembangkan aplikasi Dart.

Hands-on with Dart

Flutter sangat dipengaruhi oleh bahasa Dart, jadi memahami Dart adalah kunci keberhasilan dalam menggunakan Flutter. Untuk memulai, Anda bisa menggunakan **DartPad**, sebuah kakas online yang memungkinkan Anda menjalankan dan bereksperimen dengan kode Dart langsung di browser tanpa perlu pengaturan khusus.

Contoh Kode di DartPad:



The screenshot shows the DartPad interface. At the top, the file name is 'soal1_TugasPraktikum.dart' and the main function is selected. The code in the editor is as follows:

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

Below the code editor is the 'DEBUG CONSOLE' tab. It shows the output of the program:

```
Connecting to VM Service at ws://127.0.0.1:59486/eYB2niP6qVI=/ws  
Connected to the VM Service.  
hello 1  
hello 2  
hello 3  
hello 4  
hello 5  
  
Exited.
```

Struktur Kode Dart

Contoh kode di atas mencakup beberapa elemen dasar Dart:

- **Fungsi Utama (main):** Fungsi utama ini dipanggil pertama kali oleh Dart VM untuk memulai eksekusi program.

- **Tipe Data:** void digunakan untuk menunjukkan bahwa fungsi tidak mengembalikan nilai.
- **Perulangan For:** Struktur kontrol seperti for sangat mirip dengan bahasa pemrograman lain seperti C atau JavaScript.
- **String Interpolation:** `print('hello ${i + 1}')` menunjukkan cara menggabungkan variabel ke dalam string di Dart.

Menjalankan Kode Dart secara Lokal

Selain DartPad, Anda dapat menjalankan kode Dart di terminal. Simpan kode ke dalam file bernama `hello_world.dart` dan eksekusi dengan perintah:

`bash`

Copy code

`dart hello_world.dart`

DartPad dan pengembangan lokal sama-sama penting untuk memahami dan menguji fitur-fitur bahasa Dart.

Soal 4

Buatlah slide yang berisi penjelasan dan contoh eksekusi kode tentang perbedaan Null Safety dan Late variabel ! **(Khusus soal ini kelompok berupa link google slide)**

Kumpulkan jawaban Anda di spreadsheet pada tautan yang telah disediakan di grup telegram. Untuk soal nomor 1 sampai 3 push repo GitHub Anda.