

Pengantar Bahasa Pemrograman Dart
Tugas Praktikum

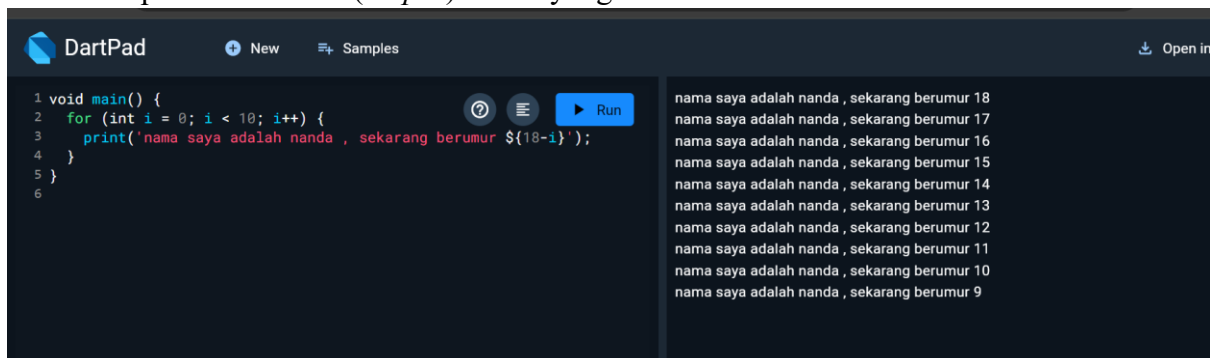


ANANDA AZ HARUDDIN SALIMA
2241720071
2 I

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!



The screenshot shows the DartPad web interface. On the left, the code editor contains the following Dart code:

```
1 void main() {  
2   for (int i = 0; i < 10; i++) {  
3     print('nama saya adalah nanda , sekarang berumur ${18-i}');  
4   }  
5 }  
6
```

On the right, the output console displays the result of running the code, which is ten lines of text, each representing the state of the program at a different iteration of the loop:

```
nama saya adalah nanda , sekarang berumur 18  
nama saya adalah nanda , sekarang berumur 17  
nama saya adalah nanda , sekarang berumur 16  
nama saya adalah nanda , sekarang berumur 15  
nama saya adalah nanda , sekarang berumur 14  
nama saya adalah nanda , sekarang berumur 13  
nama saya adalah nanda , sekarang berumur 12  
nama saya adalah nanda , sekarang berumur 11  
nama saya adalah nanda , sekarang berumur 10  
nama saya adalah nanda , sekarang berumur 9
```

Soal 2

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

Soal 3

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



Dart adalah bahasa pemrograman yang menggabungkan kelebihan berbagai bahasa tingkat tinggi dengan fitur-fitur terkini. Beberapa fitur utama Dart meliputi:

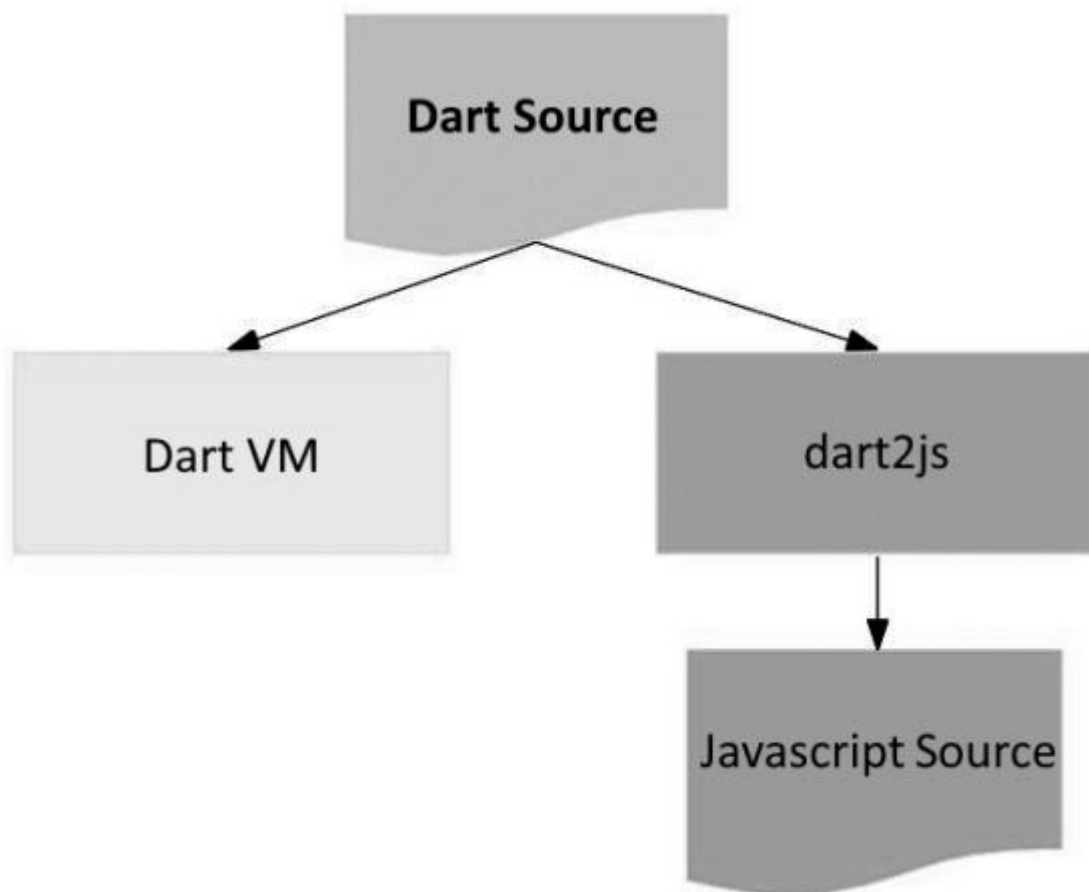
1. **Productive Tooling:** Menyediakan alat analisis kode, plugin untuk IDE, dan ekosistem paket yang besar untuk meningkatkan produktivitas.
2. **Garbage Collection:** Mengelola memori dengan otomatis menghapus objek yang tidak lagi digunakan.
3. **Type Annotations (Optional):** Memberikan keamanan dan konsistensi dalam pengelolaan data.
4. **Statically Typed:** Meskipun anotasi tipe bersifat opsional, Dart tetap aman berkat fitur type-safe dan type inference yang membantu menemukan bug saat kompilasi.
5. **Portability:** Dart dapat digunakan untuk pengembangan web dan juga dikompilasi secara native ke kode ARM dan x86.

Dart adalah inti dari pengembangan Flutter, termasuk penulisan kode aplikasi, plugin, dan manajemen dependensi. Memahami Dart dengan baik akan membuat pengembangan Flutter lebih efisien dan nyaman.

The Evolution of Dart

Dart diluncurkan pada tahun 2011 dan telah mengalami evolusi signifikan. Versi stabil dirilis pada tahun 2013, dengan perubahan besar pada Dart 2.0 pada akhir 2018. Awalnya difokuskan untuk pengembangan web sebagai pengganti JavaScript, Dart kini berfokus pada pengembangan aplikasi mobile, terutama melalui framework Flutter.

Dart menawarkan performa terbaik dan alat modern yang dirancang untuk proyek berskala besar. Dengan fitur seperti type annotations opsional dan dukungan OOP, Dart menyeimbangkan fleksibilitas dan ketangguhan. Sebagai bahasa modern dan lintas platform, Dart terus berkembang, menjadikannya pilihan ideal untuk framework Flutter.



Dart VM dan Kompilasi JavaScript

Kode Dart dapat dieksekusi di lingkungan yang mendukung bahasa Dart, yang mencakup sistem runtime, pustaka inti Dart, dan garbage collector. Dart dapat beroperasi dalam dua mode kompilasi:

1. **Kompilasi Just-In-Time (JIT):** Kode sumber dikompilasi saat diperlukan, cocok untuk pengembangan aplikasi karena mendukung fitur seperti debugging dan hot reload. Dart VM memuat dan mengkompilasi kode menjadi kode mesin asli.
2. **Kompilasi Ahead-Of-Time (AOT):** Kode Dart dikompilasi sebelumnya untuk meningkatkan performa, tetapi mengorbankan fitur seperti debugging dan hot reload. Ini cocok untuk aplikasi yang telah siap untuk produksi.

Introducing the Structure of the Dart Language

Dart memiliki sintaks yang mirip dengan bahasa pemrograman seperti C atau JavaScript, sehingga mudah dipahami jika sudah memiliki pengalaman dengan bahasa-bahasa tersebut. Dart mendukung operasi variabel dengan operator standar, dan konsep flow control serta function-nya mirip dengan bahasa lain.

Object-Oriented Programming (OOP)

Dart adalah bahasa yang berorientasi objek (OO), menggunakan konsep kelas dan objek. Dart mendukung prinsip-prinsip OOP seperti enkapsulasi, pewarisan, komposisi, abstraksi, dan polimorfisme. Bagi yang sudah familiar dengan OOP dalam bahasa lain seperti Java, konsep-konsep OOP di Dart akan terasa familiar.

Dart Operators

Operator di Dart adalah method yang didefinisikan dalam class dengan sintaks khusus. Sebagai contoh, operator `==` berfungsi seperti pemanggilan method `x.==(y)` untuk membandingkan kesetaraan. Dart mendukung berbagai operator aritmatika (+, -, *, /, ~/ , %), operator peningkatan dan penurunan (++ , --), operator kesetaraan (`==`, `!=`), dan operator relasional (>, <, >=, <=).

Logical Operators

Operator logika di Dart diterapkan pada operan bool. Contohnya:

- `!expression`: negasi hasil ekspresi.
- `||`: operasi logika OR.
- `&&`: operasi logika AND.

Tidak seperti Java atau bahasa lain, operator `==` di Dart membandingkan isi variabel, bukan referensi memori, dan tidak memerlukan operator `===` seperti di JavaScript karena Dart sudah memiliki type safety.

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.

https://docs.google.com/presentation/d/16G9lXKCuyVcVhFff_e4rGS2FVL6VPPNc_hrHe09AcRM/edit#slide=id.p