

DINI WIJAYAWATI

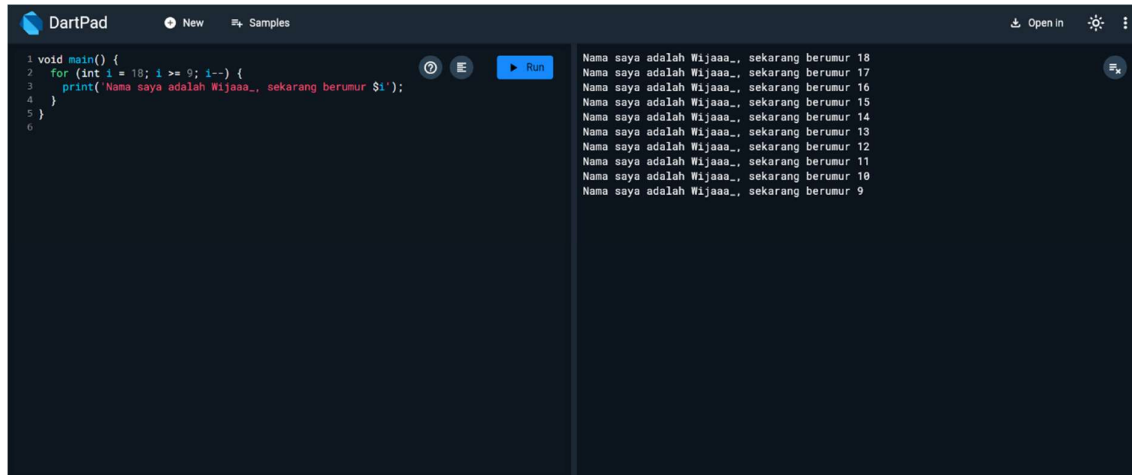
PEMROGRAMAN MOBILE

MATERI 1

SOAL 1

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

JAWABAN :



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 = 10; i >= 0; i--) {  
3     print( 'Nama saya adalah Wijaaa., sekarang berumur $i' );  
4   }  
5 }  
6
```

On the right, the output console displays the result of running the code, showing the name and age for each iteration from 18 down to 9:

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

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

JAWABAN :

Memahami bahasa pemrograman Dart sebelum menggunakan Flutter sangat penting karena beberapa alasan berikut:

1. Flutter Dibangun Menggunakan Dart

Flutter adalah framework yang sepenuhnya menggunakan Dart sebagai bahasa pemrogramannya. Jika tidak memahami dasar-dasar Dart, akan sulit untuk memahami cara kerja Flutter, seperti pengelolaan state, widget, dan interaksi antar komponen.

2. Konsep OOP (Object-Oriented Programming) dalam Dart

Dart adalah bahasa pemrograman berorientasi objek. Memahami konsep class, object, inheritance, polymorphism, dan encapsulation akan sangat membantu dalam membangun aplikasi Flutter yang terstruktur dan mudah dikelola.

3. Asynchronous Programming dengan Future dan Stream

Flutter sering menggunakan Future dan Stream untuk menangani operasi asinkron, seperti mengambil data dari API atau membaca database. Tanpa pemahaman tentang async-await, Future, dan Stream, pengembangan aplikasi Flutter akan lebih sulit.

4. Null Safety dalam Dart

Dart memiliki fitur null safety untuk mencegah error akibat null references. Jika tidak memahami cara kerja null safety (seperti penggunaan `?`, `!`, dan `late`), kita bisa mengalami banyak error yang sulit diatasi dalam pengembangan aplikasi Flutter.

5. Memahami Struktur Data dan Koleksi di Dart

Dalam pengembangan aplikasi Flutter, kita sering menggunakan List, Set, Map, dan berbagai metode manipulasi data. Memahami cara kerja koleksi dalam Dart akan memudahkan pengolahan dan penyajian data dalam aplikasi.

6. Optimasi dan Kinerja Aplikasi

Memahami Dart membantu dalam mengoptimalkan kinerja aplikasi Flutter, seperti penggunaan `const`, `final`, lazy initialization, dan teknik pemrograman yang lebih efisien.

SOAL 3

1. Pengenalan Dart

- Bahasa utama yang digunakan dalam pengembangan Flutter.
- Mendukung garbage collection, type-safe, dan statically typed untuk keamanan dan efisiensi.

2. Cara Kerja Dart

- Dart Virtual Machine (VM) untuk pengembangan cepat dengan Just-In-Time (JIT) Compilation.
- Ahead-Of-Time (AOT) Compilation untuk performa tinggi dalam aplikasi produksi.
- Hot reload membantu mempercepat proses pengembangan dan debugging.

3. Struktur Bahasa Dart

- Sintaks mirip dengan C dan JavaScript, memudahkan adaptasi bagi pengembang.
- Object-Oriented Programming (OOP) dengan konsep encapsulation, inheritance, polymorphism, dan abstraction.

4. Operator dalam Dart

- Aritmatika: `+`, `-`, `*`, `/`, `~/`, `%`
- Increment/Decrement: `++`, `--`
- Perbandingan: `==`, `!=`, `>`, `<`, `>=`, `<=`
- Logika: `!`, `&&`, `||`
- Operator digunakan dalam kontrol alur aplikasi seperti looping dan conditional statements.

5. Eksekusi dan Pengujian Kode Dart

- Menggunakan DartPad untuk eksperimen tanpa perlu instalasi tambahan.
- Menjalankan kode secara lokal dengan perintah `dart nama_file.dart`.
- Flutter SDK menyediakan alat bantu seperti Flutter DevTools untuk debugging.

6. Fungsi dalam Dart

- Function `main()` sebagai titik awal eksekusi program.
- Function vs. Method: Function berdiri sendiri, sedangkan method terkait dengan class.

7. Integrasi dengan Flutter

- Memahami dasar-dasar Dart membantu dalam memahami struktur kode Flutter.
 - Flutter menggunakan widget berbasis Dart, yang membutuhkan pemahaman tentang state management dan asynchronous programming.
-