

**02 | PENGANTAR BAHASA PEMROGRAMAN DART -
BAGIAN 1
PEMROGRAMAN MOBILE**



OLEH:

Nama : Agung Rizky S

NIM : 2241720187

Kelas : TI – 3C

**PROGRAM STUDI D-IV TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG**

2024

Tugas Praktikum

Soal 1

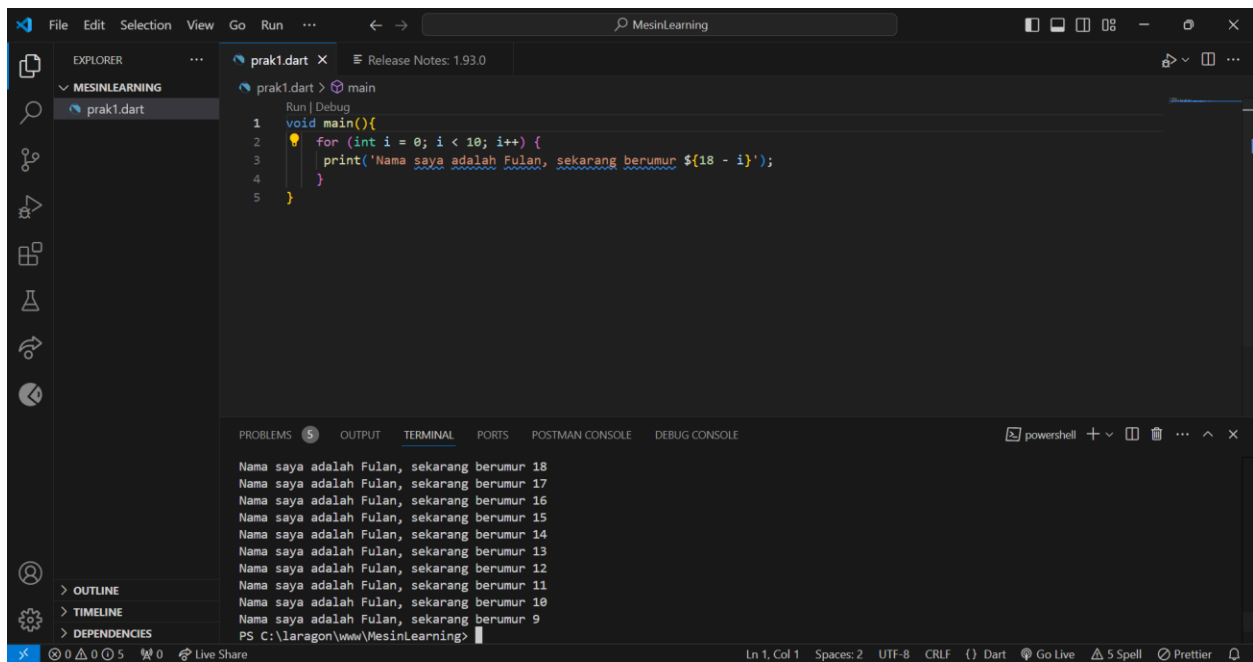
Modifikasilah kode pada baris 3 di VS Code atau Editor Code favorit Anda berikut ini agar mendapatkan keluaran (*output*) sesuai yang diminta!

```
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
```

Jawaban :



Soal 2

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

Jawaban :

Flutter sepenuhnya dibangun menggunakan bahasa Dart. Dengan memahami Dart secara mendalam, pengembangan aplikasi menggunakan Flutter akan menjadi jauh lebih mudah. Pemahaman ini akan membantu dalam menguasai konsep pemrograman asinkron, pengelolaan state, serta pembuatan UI yang efisien dan dinamis. Selain itu, Dart juga memperkuat dasar-dasar penting yang sering digunakan dalam pengembangan aplikasi mobile menggunakan Flutter. Tanpa pemahaman yang baik tentang Dart, membangun aplikasi Flutter yang kompleks akan menjadi tantangan besar.

Soal 3

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

Jawaban :

2. Getting Started with Dart

Dart adalah bahasa pemrograman modern yang memiliki berbagai kelebihan, seperti:

- **Productive tooling:** menyediakan tools untuk analisis kode, plugin IDE, dan ekosistem paket besar.
- **Garbage collection:** untuk manajemen memori secara otomatis.
- **Type annotations:** opsional, tetapi membantu menjaga keamanan dan konsistensi tipe data.
- **Statically typed:** meski opsional, Dart menggunakan *type inference* untuk memeriksa tipe data.
- **Portability:** Dart bisa dikompilasi untuk web, mobile, hingga kode native ARM dan x86.

Dart sangat penting dalam pengembangan Flutter karena framework ini sepenuhnya berbasis Dart.

3. The Evolution of Dart

Dart diluncurkan pada 2011 dengan fokus awal pada pengembangan web, tetapi kini lebih berfokus pada pengembangan mobile, terutama dengan Flutter. Dart diharapkan menjadi bahasa penerus JavaScript, menawarkan performa yang lebih baik dan alat pengembangan yang lebih modern. Versi stabilnya dirilis pada 2013, dengan peningkatan besar di Dart 2.0 pada akhir 2018.

4. How Dart Works

Eksekusi kode Dart dapat dilakukan dengan dua cara:

- **Dart Virtual Machine (VM):** Menggunakan kompilasi *Just-In-Time (JIT)* selama pengembangan, yang memungkinkan debugging dan *hot reload*.
- **JavaScript Compilation:** Dart juga dapat dikompilasi ke JavaScript untuk berjalan di browser. Dalam produksi, kompilasi *Ahead-Of-Time (AOT)* dilakukan untuk meningkatkan performa.

5. Introducing the Structure of the Dart Language

Dart dirancang sebagai bahasa **object-oriented (OO)**, mirip dengan bahasa seperti Java. Dart mendukung fitur OOP seperti **encapsulation**, **inheritance**, dan **polymorphism**.

- **Operators** di Dart adalah metode dalam class. Contoh, `x == y` adalah pemanggilan metode `x.==(y)`. Operator juga dapat di-*override* di Dart.
- **Arithmetic operators** termasuk `+`, `-`, `*`, `/`, `~/`, dan `%`. Ada juga operator tambahan seperti `+=`, `-=`, `++`, dan `--`.
- **Equality and Relational operators:** Dart menggunakan `==`, `!=`, `>`, `<`, `>=`, dan `<=`, namun membandingkan isi variabel, bukan referensi memori seperti di Java.
- **Logical operators:** `!` (negasi), `||` (OR), `&&` (AND) digunakan untuk operasi logika pada tipe `bool`.

6. Hands-on with Dart

Dart menggunakan fungsi untuk mengeksekusi kode. Fungsi utama dalam aplikasi Dart adalah `main()`, yang didefinisikan sebagai `void main() {}` dan menjadi titik awal eksekusi kode Dart. Dart mendukung banyak fitur tingkat lanjut yang akan memudahkan pengembangan aplikasi, terutama dengan Flutter.