**Pertemuan 2**

**Pengantar Bahasa Pemrograman Dart**

A close up of a sign

Description automatically generated

Dibuat Oleh:

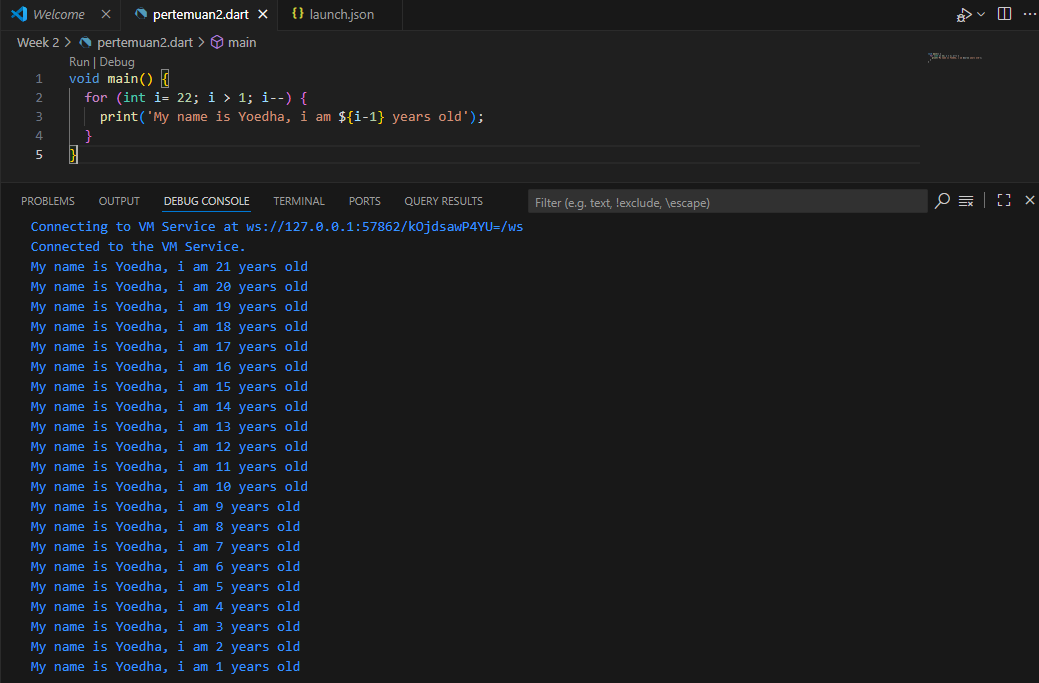
Leon Shan Yoedha Adjie

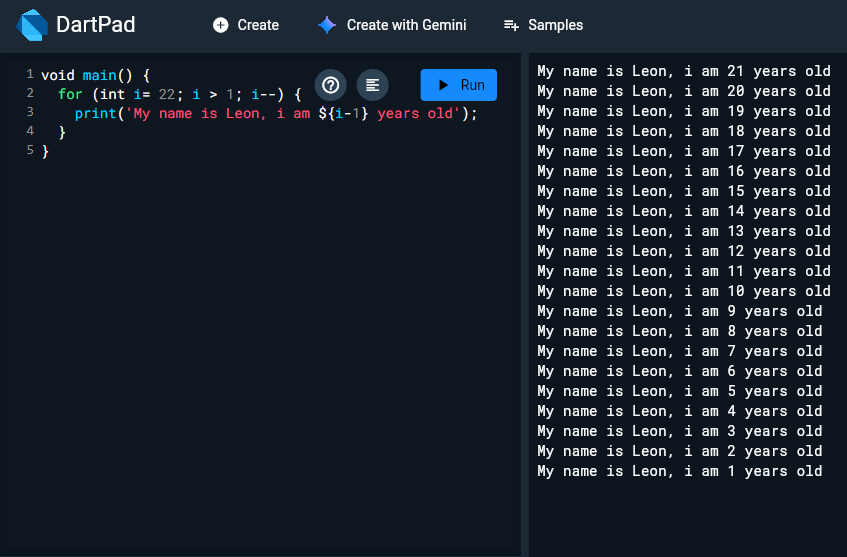
**POLITEKNIK NEGERI MALANG**

**JURUSAN TEKNOLOGI INFORMARSI**

**2025/2026**

# Tugas Praktikum

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

****

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

***Jawab:***

Sangat penting untuk memahami bahasa pemrograman Dart sebelum menggunakan framework Flutter karena Dart adalah inti dan dasar fundamental dari Flutter. Dengan mempelajari Dart, seorang pengembang akan menguasai konsep-konsep inti seperti tipe data, operator, dan pemrograman berorientasi objek (OOP) yang digunakan secara ekstensif dalam pengembangan aplikasi Flutter. Penguasaan ini tidak hanya menjadi fondasi untuk dapat bekerja secara efektif, tetapi juga memberikan kenyamanan dan kepercayaan diri untuk bereksperimen lebih jauh, sehingga memungkinkan terciptanya aplikasi seluler yang luar biasa.

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

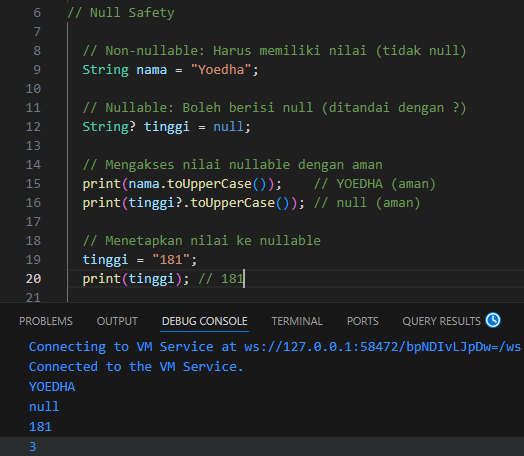
***Jawab:***

* **Dart adalah Fondasi Utama Flutter**: Memahami Dart sangatlah esensial karena Dart merupakan bahasa inti yang digunakan oleh Flutter. Penguasaan Dart akan mempermudah pengembangan dan pemecahan masalah dalam proyek Flutter.
* **Konsep Dasar Pemrograman:** Codelab ini menekankan pentingnya memahami dasar-dasar Dart, termasuk:
  + **Tipe Data dan Operator**: Mengenal tipe data bawaan dan berbagai operator (aritmatika, perbandingan, logika) adalah dasar untuk menulis logika aplikasi.
  + **Pemrograman Berorientasi Objek (OOP)**: Flutter sangat bergantung pada konsep OOP. Memahami cara kerja fungsi dan metode dalam konteks OOP di Dart akan sangat membantu.
* **Fitur-Fitur Unggulan Dart**: Bahasa Dart memiliki beberapa fitur yang mendukung pengembangan yang produktif:
  + **Static Typing**: Membantu menemukan *error* lebih awal saat proses pengembangan, bukan saat aplikasi sudah berjalan.
  + **Garbage Collection**: Manajemen memori otomatis yang membuat developer tidak perlu pusing mengalokasikan dan membebaskan memori secara manual.
  + **Tooling yang Produktif**: Dart dilengkapi dengan berbagai alat bantu yang mempercepat proses pengembangan.
* **Mekanisme Kompilasi JIT dan AOT**: Dart menggunakan dua jenis kompilasi yang sangat berguna untuk Flutter:
  + **Just-In-Time (JIT)**: Digunakan selama pengembangan untuk mengaktifkan fitur *Hot Reload*, yang memungkinkan melihat perubahan kode secara instan tanpa perlu me-restart aplikasi.
  + **Ahead-Of-Time (AOT)**: Digunakan saat aplikasi akan dirilis (*release*) untuk memastikan performa yang cepat dan waktu startup yang singkat di perangkat pengguna.
* **Struktur Program Dasar**: Setiap program Dart dimulai dari fungsi main(). Memahami ini adalah langkah pertama untuk menulis dan menjalankan kode Dart.
* **Mulai Cepat dengan DartPad**: Codelab ini merekomendasikan penggunaan DartPad, sebuah editor kode online.

1. **Buatlah penjelasan dan contoh eksekusi kode tentang perbedaan *Null Safety* dan *Late variabel* !**

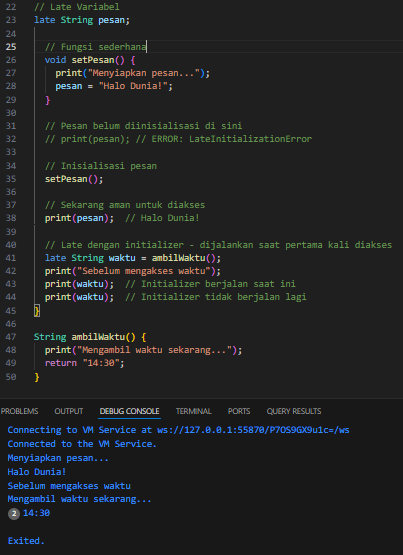
***Jawab:***

**Null safety:** Null Safety adalah fitur keamanan tipe data yang memungkinkan kompiler untuk mencegah error yang terkait dengan nilai null. Dengan null safety, variabel tidak bisa berisi nilai null kecuali secara eksplisit dinyatakan dapat menerima nilai null yang ditandai dengan simbol ‘?’.

**Contoh:** ****

**Late variabel:** adalah fitur yang memungkinkan penundaan inisialisasi variabel non-nullable. Ini berguna ketika variabel tidak bisa diinisialisasi pada saat deklarasi tetapi dijamin akan diinisialisasi sebelum digunakan dan ditandai dengan *keyword ‘*late’.

**Contoh:**

****