**LAPORAN PRAKTIKUM PEMROGRAMAN MOBILE**

**Pengantar Bahasa Pemrograman Dart**

Dosen Pengampu**: Ade Ismail, S.Kom., M.TI.**

A logo of a building

Description automatically generated

**Disusun Oleh:**

**Tia Arvivolia**

**2241760031**

**SIB-3E**

**PRODI D-lV SISTEM INFORMASI BISNIS**

**JURUSAN TEKNOLOGI INFORMASI**

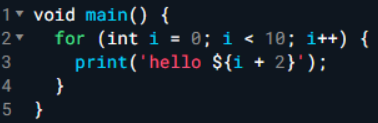
**POLITEKNIK NEGERI MALANG**

**TAHUN 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!



Output yang diminta:

A screen shot of a computer screen

Description automatically generated

**Jawaban:**

**Code:**

A screenshot of a computer

Description automatically generated

**Output:**

A screenshot of a computer

Description automatically generated

**Soal 2:**

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

* **Jawaban**:

Memahami bahasa pemrograman Dart sebelum menggunakan framework Flutter sangat penting karena Dart adalah bahasa utama yang digunakan untuk menulis kode Flutter. Pengetahuan tentang Dart membantu developer memahami sintaksis dan struktur kode yang diperlukan, menulis kode yang lebih efisien, serta mengoptimalkan penggunaan fitur-fitur seperti asynchronous programming. Selain itu, pemahaman yang baik tentang Dart memudahkan dalam debugging, pemeliharaan aplikasi, dan integrasi dengan berbagai library dan package yang tersedia dalam ekosistem Flutter.

**Soal 3:**

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

1. **Pengantar Bahasa Pemrograman Dart**: Dart adalah bahasa pemrograman utama untuk Flutter yang mendukung pemrograman berorientasi objek dan memiliki fitur modern.
2. **Getting Started with Dart**:

* **Productive Tooling**: Memanfaatkan fitur analisis kode, plugin IDE, dan ekosistem paket.
* **Garbage Collection**: Mengelola de-alokasi memori untuk objek yang tidak digunakan.
* **Type Annotations (Opsional)**: Menjamin keamanan dan konsistensi data.
* **Statically Typed**: Menggunakan type-safe dan type inference untuk mengidentifikasi bug selama kompilasi.
* **Portability**: Kompilasi ke JavaScript dan kode native ARM/x86.

1. **The Evolution of Dart**:

* Dari fokus awal pada web, kini lebih berorientasi pada pengembangan mobile dengan Flutter.
* Mengatasi kelemahan JavaScript dengan alat modern dan performa tinggi.
* Menyeimbangkan fleksibilitas dan ketangguhan dengan fitur OOP dan type annotations opsional.

1. **How Dart Works**:

* **Dart Virtual Machines (VMs)** dan **JavaScript Compilation:** Kode Dart dapat dieksekusi dalam mode JIT (Just-In-Time) untuk pengembangan dan AOT (Ahead-Of-Time) untuk performa tinggi.
* **Hot Reload:** Memungkinkan umpan balik cepat terhadap perubahan kode, mendukung pengembangan yang efisien.

1. **Introducing the Structure of Dart Language**:

* **Object Orientation (OOP)**: Dart adalah bahasa berorientasi objek yang menggunakan kelas untuk mendefinisikan objek, dengan fitur seperti enkapsulasi, pewarisan, komposisi, abstraksi, dan polimorfisme.
* **Operator Dart**: Operator di Dart diimplementasikan sebagai metode dalam kelas, memungkinkan penyesuaian logika. Meliputi operator aritmatika (+, -, \*, /, ~/), penambahan/pengurangan (++, --), kesetaraan (==, !=), relasional (>, <, >=, <=), dan logika (!, ||, &&).
* **Perilaku Operator**: Operator aritmatika bekerja pada tipe data numerik dan string, dengan dukungan untuk operasi modulus dan pembagian bilangan bulat. Operator logika bekerja pada tipe bool, memungkinkan kombinasi ekspresi logika.
* **Type Safety**: Operator == membandingkan nilai, bukan referensi, dan Dart tidak memerlukan operator === karena 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)

* **Jawaban:**

Perbedaan Null Safety dan Late Variabel dalam Dart

* 1. **Null Safety**

**Penjelasan:**

* Null Safety adalah fitur di Dart yang mencegah nilai null yang tidak terduga, mengurangi risiko runtime exceptions akibat null dereferencing.
* Dengan Null Safety, variabel yang tidak diinisialisasi secara eksplisit tidak bisa memiliki nilai null kecuali ditandai dengan ?.

**Catatan:**

* Tipe nullable int? dapat memiliki nilai null, sedangkan tipe non-nullable int tidak dapat memiliki nilai null.

A screenshot of a computer program

Description automatically generated

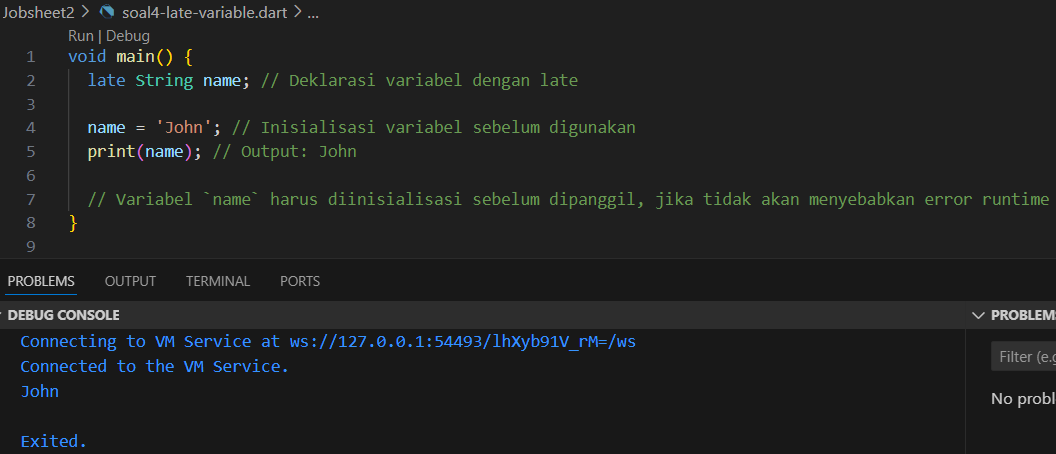
* 1. **Late Variabel**

**Penjelasan:**

* late digunakan untuk menunda inisialisasi variabel hingga variabel tersebut benar-benar diperlukan, tetapi masih tetap menghindari nilai null.
* Variabel yang dideklarasikan dengan late harus diinisialisasi sebelum digunakan.

**Catatan:**

* Variabel late memberikan fleksibilitas untuk menunda inisialisasi sambil memastikan bahwa variabel tersebut akan diinisialisasi sebelum digunakan.



**Ringkasan:**

* **Null Safety**: Mencegah nilai null yang tidak terduga; tipe nullable (int?) vs non-nullable (int).
* **Late Variabel**: Menunda inisialisasi variabel hingga diperlukan; variabel late harus diinisialisasi sebelum digunakan.