

LAPORAN PRAKTIKUM 1 PERTEMUAN 2
PEMROGRAMAN MOBILE
SEMESTER 5

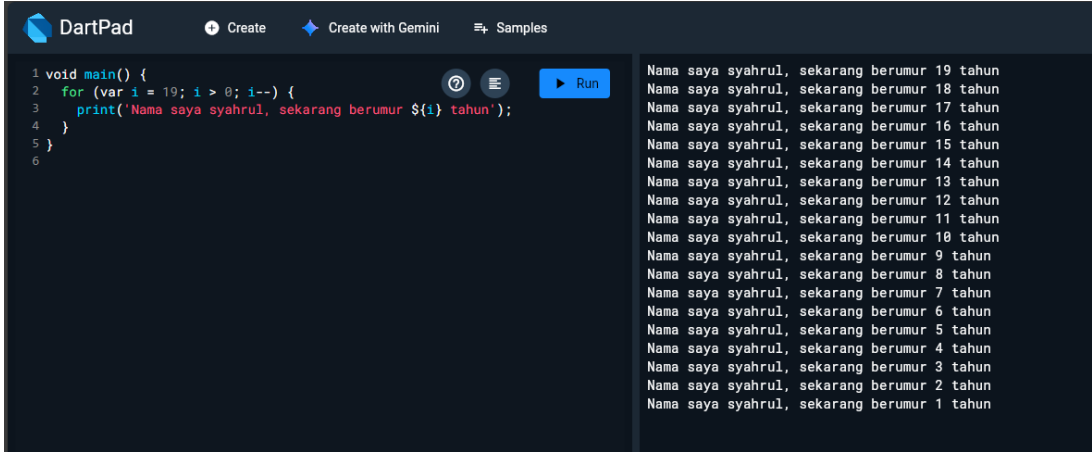
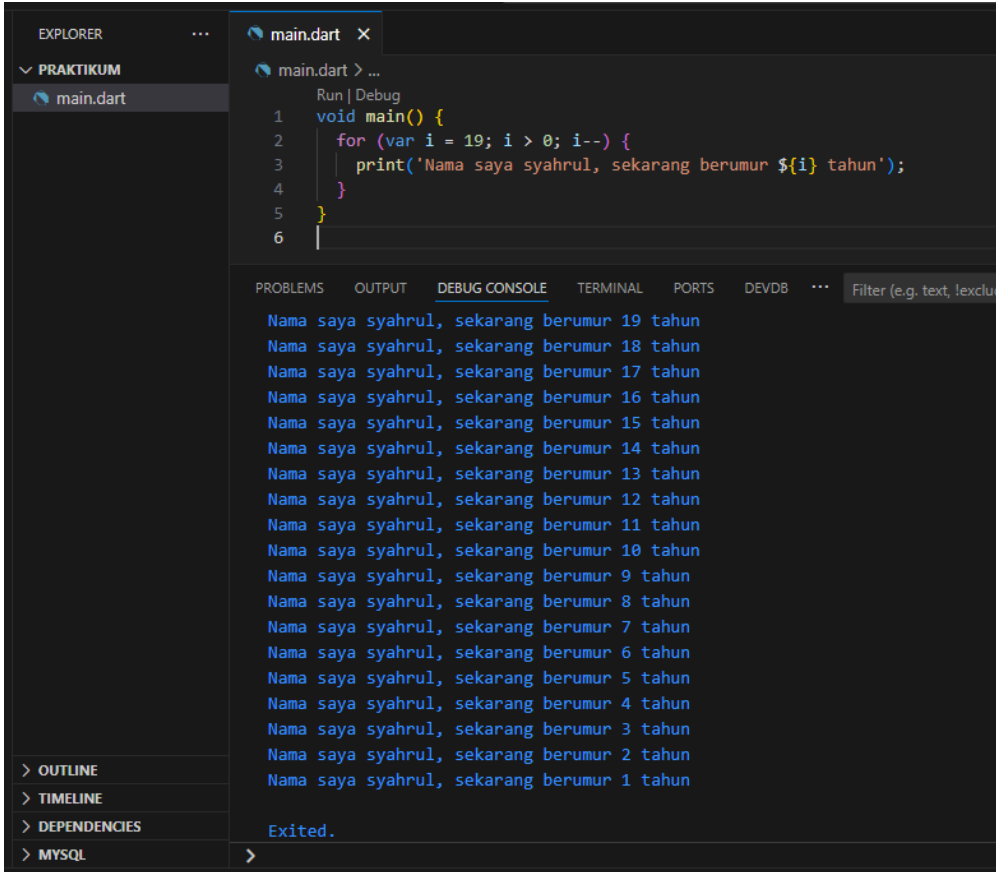


Dosen Pengampu :
Bapak Agung Nugroho Pramudhita, S.T., M.T.

Mahasiswa:
Muhammad Syahrul Gunawan (19)
2341720002
(2F)

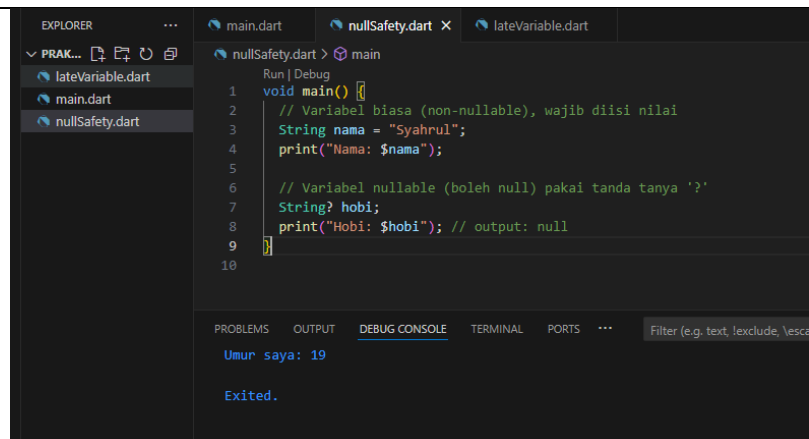
D-IV TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
TAHUN 2025

Tugas Praktikum

Nomor	Soal
1	<p>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 (Gantilah Fulan dengan nama Anda):</p> <p>Jawab:</p> <p>DartPad</p>  <p>VSCode</p> 

2	<p>Mengapa sangat penting untuk memahami Bahasa pemrograman Dart sebelum kita menggunakan framework Flutter? Jelaskan!</p> <p>Jawab:</p> <p>Karena Bahasa Dart adalah inti dari framework Flutter. Kerangka kerja modern seperti Flutter membutuhkan bahasa modern tingkat tinggi agar bisa memberikan pengalaman terbaik kepada pengembang, serta memungkinkan untuk membuat aplikasi seluler yang luar biasa. Memahami Dart adalah dasar untuk bekerja dengan Flutter, pengembang perlu mengetahui asal-usul Bahasa Dart, bagaimana komunitas mengerjakannya, kelebihanannya, dan mengapa itu adalah Bahasa pemrograman yang dipilih untuk Flutter.</p>
3	<p>Rangkumlah materi dari codelab ini menjadi poin-poin penting yang dapat Anda gunakan untuk membantu proses pengembangan aplikasi mobile menggunakan framework Flutter.</p> <p>Jawab:</p> <ol style="list-style-type: none"> 1. Pentingnya memulai untuk memahami Bahasa Pemrograman Dart sebelum menggunakan framework Flutter. <ul style="list-style-type: none"> - Karena Bahasa Dart adalah inti dari framework Flutter. Kerangka kerja modern seperti Flutter membutuhkan bahasa modern tingkat tinggi agar bisa memberikan pengalaman terbaik kepada pengembang, serta memungkinkan untuk membuat aplikasi seluler yang luar biasa. Memahami Dart adalah dasar untuk bekerja dengan Flutter, pengembang perlu mengetahui asal-usul Bahasa Dart, bagaimana komunitas mengerjakannya, kelebihanannya, dan mengapa itu adalah Bahasa pemrograman yang dipilih untuk Flutter. 2. Memahami fitur-fitur Bahasa pemrograman terkini dari penggabungan kelebihan-kelebihan Sebagian besar Bahasa pemrograman tingkat tinggi. <ul style="list-style-type: none"> - Productive tooling: merupakan fitur kakas (<i>tool</i>) untuk menganalisis kode, plugin IDE, dan ekosistem paket yang besar. - Garbage collection: untuk mengelola atau menangani dealokasi memori (terutama memori yang ditempati oleh objek yang tidak lagi digunakan). - Type annotations (opsional): untuk keamanan dan konsistensi dalam mengontrol semua data dalam aplikasi. - Statically typed: Meskipun <i>type annotations</i> bersifat opsional, Dart tetap aman karena menggunakan fitur <i>type-safe</i> dan <i>type inference</i> untuk menganalisis <i>types</i> saat <i>runtime</i>. Fitur ini penting untuk menemukan <i>bug</i> selama kompilasi kode. - Portability: bahasa Dart tidak hanya untuk web (yang dapat diterjemahkan ke JavaScript) tetapi juga dapat dikompilasi secara <i>native</i> ke kode Advanced RISC Machines (ARM) dan x86.

	<p>3. Memahami cara kerja Dart</p> <ul style="list-style-type: none"> - Kompilasi JIT adalah tempat kode sumber dikompilasi sesuai kebutuhan—<i>Just in time</i>. Dart VM memuat dan mengkompilasi kode sumber ke kode mesin asli (<i>native</i>). Pendekatan ini digunakan untuk menjalankan kode pada <i>command line</i> atau selama proses pengembangan aplikasi mobile yang dapat memanfaatkan fitur seperti debugging dan <i>hot reload</i>. - Kompilasi AOT adalah dimana Dart VM dan kode Anda dikompilasi sebelumnya, VM bekerja lebih seperti sistem runtime Dart, yang menyediakan <i>garbage collector</i> dan metode-metode <i>native</i> dari Dart software development kit (SDK) pada aplikasi. Pendekatan ini memiliki keuntungan performa yang sangat besar dibandingkan kompilasi JIT, tetapi fitur lain seperti debugging dan hot reload tidak tersedia. <p>4. Memahami Struktur Bahasa Dart</p> <ul style="list-style-type: none"> - Object orientation - Dart operators - Arithmetic operators - Operator Increment dan Decrement - Operator Equality dan Relation - Operator Logical
4	<p>Buatlah penjelasan dan contoh eksekusi kode tentang perbedaan Null Safety dan Late variable!</p> <p>Jawab:</p> <p>1. Null Safety</p> <ul style="list-style-type: none"> - Dart dengan null safety berarti suatu variable tidak boleh bernilai null kecuali ditandai khusus. - Jika kita mendeklarasikan variable tanpa tanda tanya ?, maka variable tersebut wajib diisi nilai. - Kalau kita ingin variable bernilai null, maka harus menambahkan ? setelah tipe datanya.



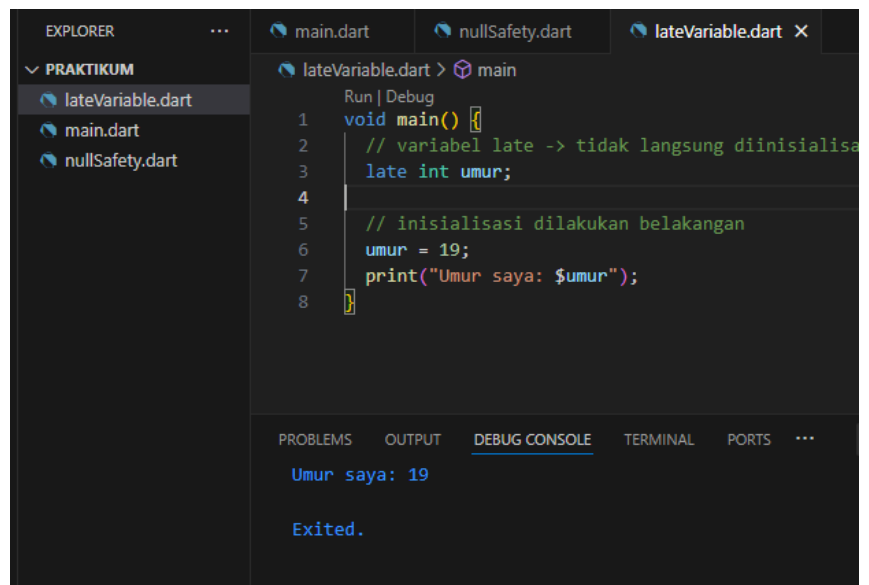
```
1 void main() {  
2   // Variabel biasa (non-nullable), wajib diisi nilai  
3   String nama = "Syahrul";  
4   print("Nama: $nama");  
5  
6   // Variabel nullable (boleh null) pakai tanda tanya '?'  
7   String? hobi;  
8   print("Hobi: $hobi"); // output: null  
9 }  
10
```

Umur saya: 19

Exited.

2. Late Variable

- late digunakan Ketika variable akan diinisialisasi nanti ssebelum digunakan.
- Berguna kalua nilai variable belum bisa ditentukan saat deklarasi.
- Jika late variable dipakai sebelum diisi nilai, maka akan error saat runtime.



```
1 void main() {  
2   // variabel late -> tidak langsung diinisialisasi  
3   late int umur;  
4  
5   // inisialisasi dilakukan belakangan  
6   umur = 19;  
7   print("Umur saya: $umur");  
8 }
```

Umur saya: 19

Exited.