

LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
PENGANTAR BAHASA PEMROGRAMAN DART - BAGIAN 1



Oleh:

NECHA SYIFA SYAFITRI

NIM 2341720167

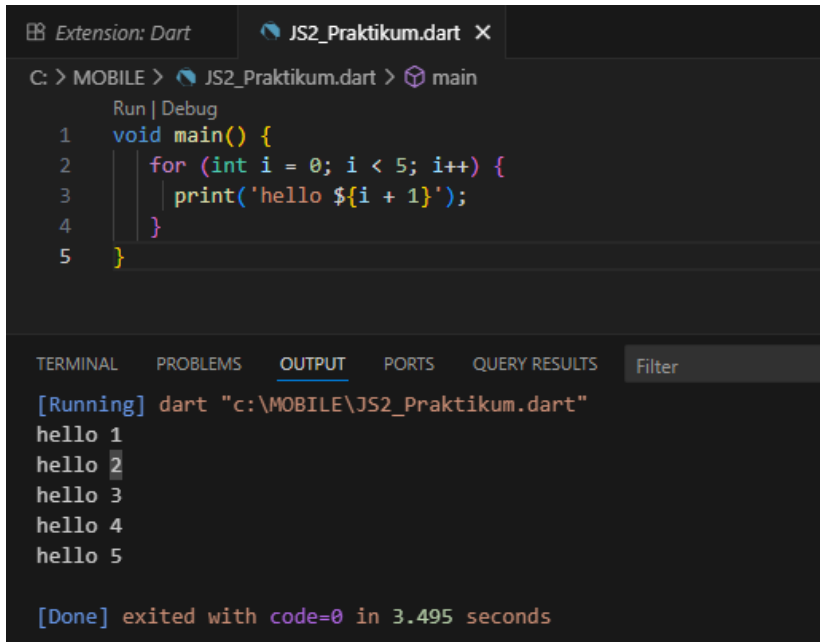
TI-3E / 21

D-IV TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

Praktikum

Berlatih dengan Dart

- VS Code



The screenshot shows the Visual Studio Code editor with a Dart file named `JS2_Praktikum.dart`. The code defines a `main` function that uses a `for` loop to print "hello 1" through "hello 5". The `OUTPUT` panel at the bottom shows the execution results, confirming the output of the program.

```
Extension: Dart JS2_Praktikum.dart X
C: > MOBILE > JS2_Praktikum.dart > main
Run | Debug
1 void main() {
2   for (int i = 0; i < 5; i++) {
3     print('hello ${i + 1}');
4   }
5 }
```

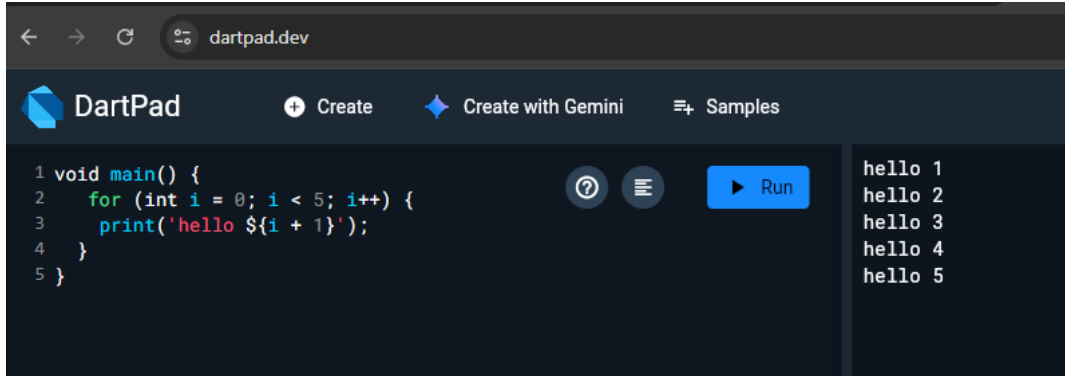
TERMINAL PROBLEMS OUTPUT PORTS QUERY RESULTS Filter

[Running] dart "c:\MOBILE\JS2_Praktikum.dart"

hello 1
hello 2
hello 3
hello 4
hello 5

[Done] exited with code=0 in 3.495 seconds

- DartPad



The screenshot shows the DartPad web interface. It contains the same Dart code as the VS Code screenshot. The `Run` button has been clicked, and the output "hello 1" through "hello 5" is displayed on the right side of the editor.

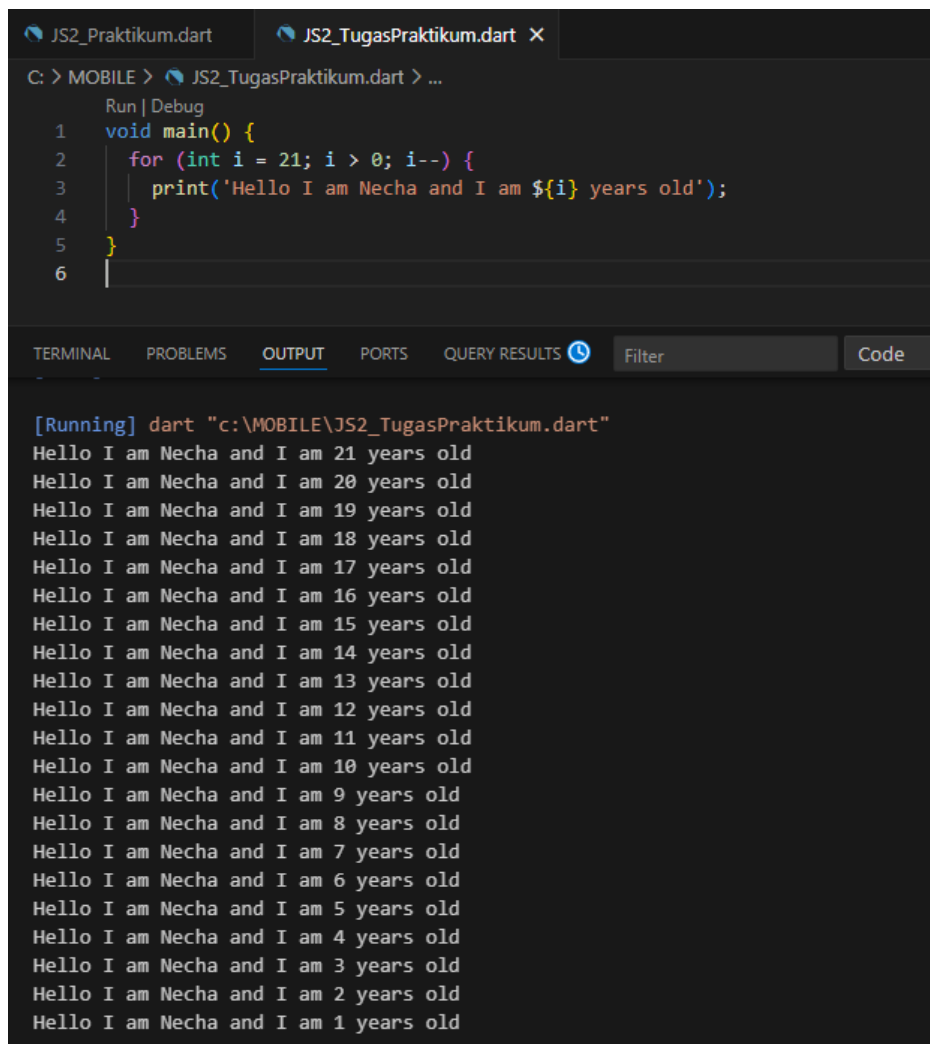
```
DartPad Create Create with Gemini Samples
1 void main() {
2   for (int i = 0; i < 5; i++) {
3     print('hello ${i + 1}');
4   }
5 }
```

hello 1
hello 2
hello 3
hello 4
hello 5

Tugas Praktikum

Soal 1

- VS Code

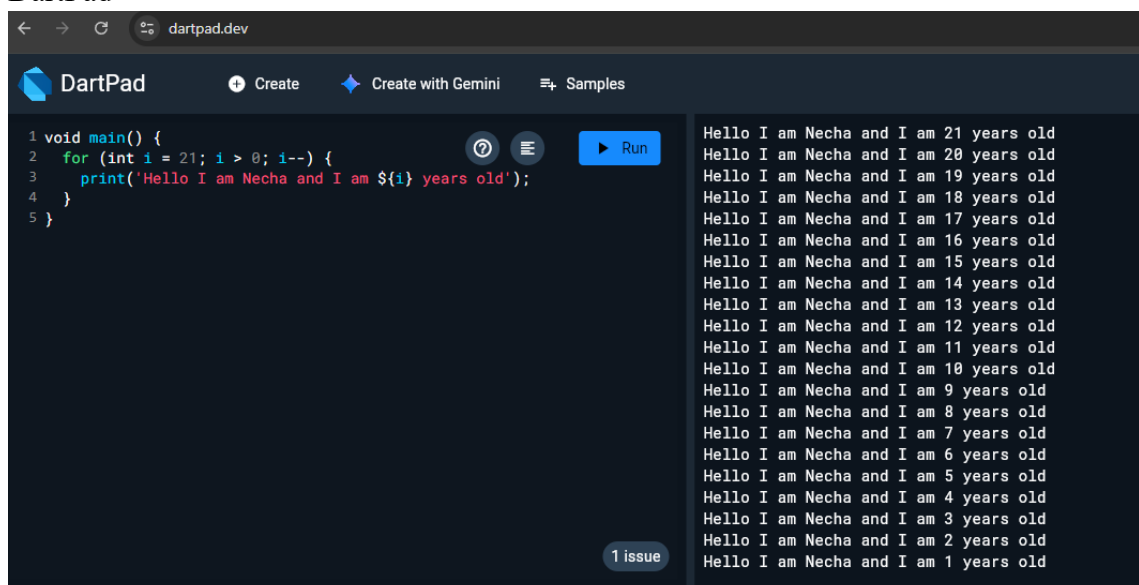


```
JS2_Praktikum.dart JS2_TugasPraktikum.dart X
C: > MOBILE > JS2_TugasPraktikum.dart > ...
Run | Debug
1 void main() {
2   for (int i = 21; i > 0; i--) {
3     print('Hello I am Necha and I am ${i} years old');
4   }
5 }
6

TERMINAL PROBLEMS OUTPUT PORTS QUERY RESULTS Filter Code

[Running] dart "c:\MOBILE\JS2_TugasPraktikum.dart"
Hello I am Necha and I am 21 years old
Hello I am Necha and I am 20 years old
Hello I am Necha and I am 19 years old
Hello I am Necha and I am 18 years old
Hello I am Necha and I am 17 years old
Hello I am Necha and I am 16 years old
Hello I am Necha and I am 15 years old
Hello I am Necha and I am 14 years old
Hello I am Necha and I am 13 years old
Hello I am Necha and I am 12 years old
Hello I am Necha and I am 11 years old
Hello I am Necha and I am 10 years old
Hello I am Necha and I am 9 years old
Hello I am Necha and I am 8 years old
Hello I am Necha and I am 7 years old
Hello I am Necha and I am 6 years old
Hello I am Necha and I am 5 years old
Hello I am Necha and I am 4 years old
Hello I am Necha and I am 3 years old
Hello I am Necha and I am 2 years old
Hello I am Necha and I am 1 years old
```

- DartPad



```
dartpad.dev
DartPad Create Create with Gemini Samples
1 void main() {
2   for (int i = 21; i > 0; i--) {
3     print('Hello I am Necha and I am ${i} years old');
4   }
5 }
Run
Hello I am Necha and I am 21 years old
Hello I am Necha and I am 20 years old
Hello I am Necha and I am 19 years old
Hello I am Necha and I am 18 years old
Hello I am Necha and I am 17 years old
Hello I am Necha and I am 16 years old
Hello I am Necha and I am 15 years old
Hello I am Necha and I am 14 years old
Hello I am Necha and I am 13 years old
Hello I am Necha and I am 12 years old
Hello I am Necha and I am 11 years old
Hello I am Necha and I am 10 years old
Hello I am Necha and I am 9 years old
Hello I am Necha and I am 8 years old
Hello I am Necha and I am 7 years old
Hello I am Necha and I am 6 years old
Hello I am Necha and I am 5 years old
Hello I am Necha and I am 4 years old
Hello I am Necha and I am 3 years old
Hello I am Necha and I am 2 years old
Hello I am Necha and I am 1 years old
1 issue
```

Soal 2

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

- Memahami bahasa pemrograman Dart sebelum menggunakan Flutter sangat penting karena seluruh kode Flutter ditulis menggunakan Dart, mulai dari pembuatan widget, pengelolaan data (state), hingga manajemen dependensi. Jika tidak memahami Dart, maka akan sulit mengerti cara kerja Flutter secara menyeluruh. Dengan memahami dasar-dasar Dart, seperti tipe data, operator, dan konsep OOP, kita dapat menulis kode yang lebih rapi, memanfaatkan fitur Flutter dengan maksimal, serta lebih mudah mencari dan memperbaiki kesalahan (debugging). Oleh karena itu, Dart merupakan fondasi utama yang harus dikuasai agar pengembangan aplikasi dengan Flutter dapat berjalan lancar.

Soal 3

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

Keunggulan Dart:

- Memiliki alat bantu produktif seperti analisis kode, plugin IDE, dan ekosistem paket yang luas.
- Dilengkapi garbage collection otomatis untuk mengatur penggunaan memori.
- Mendukung anotasi tipe opsional, tetap aman karena adanya type-safety dan type inference.
- Bersifat portabel yaitu dapat dikompilasi ke kode native (ARM, x86) atau JavaScript untuk web.

Evolusi Dart

- Awalnya diluncurkan tahun 2011 dengan tujuan menggantikan JavaScript di web.
- Sejak 2018 beralih fokus pada pengembangan mobile melalui Flutter karena performanya yang tinggi dan fiturnya yang lengkap.

Cara Kerja & Mode Eksekusi

- Kompilasi JIT (Just-In-Time): Dipakai saat pengembangan, mendukung debugging dan hot reload.
- Kompilasi AOT (Ahead-Of-Time): Dipakai untuk produksi, menghasilkan kode native dengan performa tinggi.

Struktur & OOP

- Sintaks mirip C/JavaScript, ada operator aritmatika, logika, kesetaraan, dll.
- Berbasis OOP: mendukung enkapsulasi, pewarisan, abstraksi, dan polimorfisme.
- Semua widget di Flutter sebenarnya adalah class yang ditulis dalam Dart.

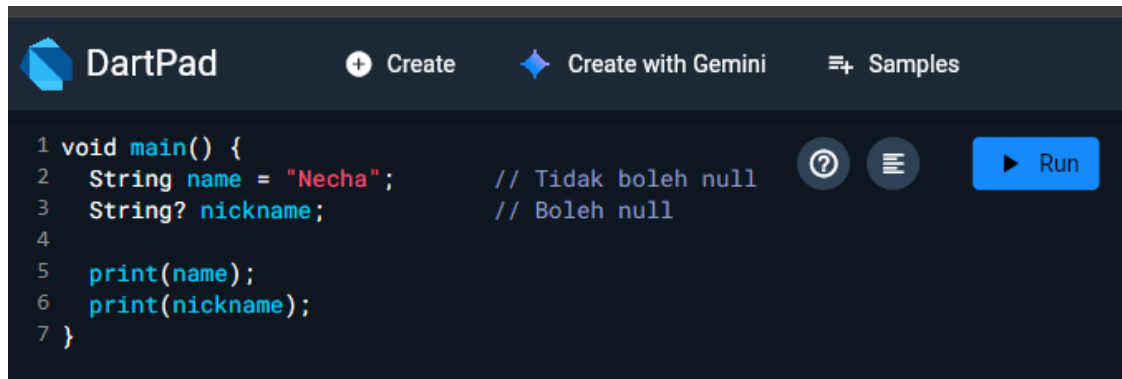
Soal 4

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

- *Null Safety*

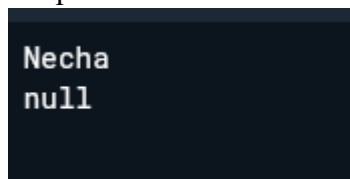
Null safety adalah fitur Dart yang berfungsi mencegah error ketika variabel bernilai kosong (null). Dengan *null safety*, variabel harus ditentukan sejak awal apakah boleh bernilai null atau tidak. Jika variabel diperbolehkan bernilai null, maka harus ditambahkan tanda tanya ? setelah tipe datanya.

Contoh eksekusi kode:



```
1 void main() {  
2   String name = "Necha";           // Tidak boleh null  
3   String? nickname;               // Boleh null  
4  
5   print(name);  
6   print(nickname);  
7 }
```

Output:

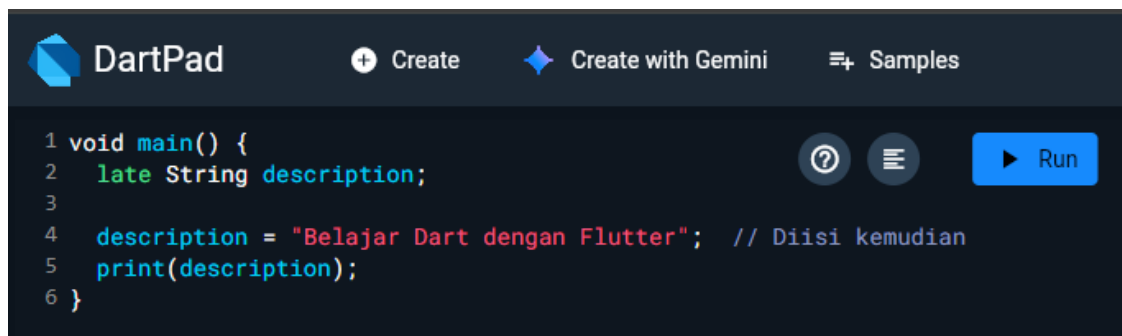


```
Necha  
null
```

- *Late Variabel*

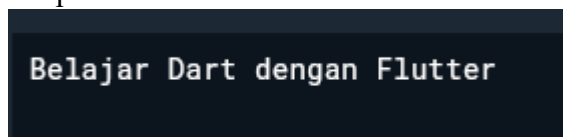
Late digunakan ketika sebuah variabel belum dapat langsung diberi nilai saat dideklarasikan, tetapi dipastikan akan diisi sebelum dipakai. Biasanya dipakai untuk menunda inisialisasi yang bergantung pada proses tertentu.

Contoh eksekusi kode:



```
1 void main() {  
2   late String description;  
3  
4   description = "Belajar Dart dengan Flutter"; // Diisi kemudian  
5   print(description);  
6 }
```

Output:



```
Belajar Dart dengan Flutter
```