

**Pemrograman Mobile**  
**Jobsheet 2**



**From:**

DAVIS MAULANA HERMANTO

**Class:**

3 I

**Absence:**

05

**Student Number Identity:**

2241720255

**Department:**

Information Technology

**Study Program:**

Informatics Engineering

## Practicum

1. Modify the code on line 3 in VS Code or your favorite Code Editor below to get the output as requested!

```
1 void main() {  
2     for (int i = 0; i < 10; i++) {  
3         print('hello ${i + 2}');  
4     }  
5 }
```

Requested output:

```
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
```

```
1 void main() {  
2     for (int i = 18; i >= 9; i--) {  
3         print('Nama saya adalah Fulan, sekarang berumur $i');  
4     }  
5 }  
6
```

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

2. Why is it very important to understand the Dart programming language before we use the Flutter framework? Explain!
  - a. Because basically the Dart language is the essential of the Flutter so if we have learned the Dart language at least the basics, we can easily use Flutter to build any apps
3. Summarize the material from this codelab into key points that you can use to help you develop mobile applications using the Flutter framework.
  - a. **Dart language** is the essential of the Flutter
  - b. **Productive Tooling**: Includes code analysis tools, IDE plugins, and a large package ecosystem.

- c. **Garbage Collection:** Manages memory deallocation for unused objects.
- d. **Type Annotations (Optional):** Provides safety and consistency for data control.
- e. **Statically Typed:** Uses type-safe and type inference features for error checking during compilation.
- f. **Portability:** Dart can be compiled to JavaScript for web and to native code for ARM and x86 platforms.
- g. **Flutter Development:** Flutter framework relies on Dart for app code, plugins, and dependency management.
- h. **Execution Environments:** Dart code can be run in environments that support Dart, either through:
  - i. **Dart Virtual Machines (VMs):** For Just-In-Time (JIT) compilation, useful for development with debugging and hot reload.
  - ii. **JavaScript Compilation:** Converts Dart to JavaScript for web environments.
- i. **Compilation Modes:**
  - i. **JIT Compilation:** Compiles code as needed, providing flexibility during development.
  - ii. **AOT Compilation:** Pre-compiles code for faster performance in production, without debugging features.
- j. **Object-Oriented Programming (OOP):** Based on objects that hold data and methods, with principles like encapsulation, inheritance, and polymorphism.
- k. **Operators:** Operators in Dart are methods with special syntax:
  - i. **Arithmetic Operators:** `+`, `-`, `*`, `/`, `~/`, `%`, and negation.
  - ii. **Increment/Decrement Operators:** `++var`, `var++`, `--var`, `var--`.
  - iii. **Equality and Relational Operators:** `==`, `!=`, `>`, `<`, `>=`, `<=`.
  - iv. **Logical Operators:** `!` (negation), `||` (logical OR), `&&` (logical AND).