

**JOBSHEET 02**  
**PENGANTAR BAHASA PEMROGRAMAN**  
**DART BAGIAN 1**

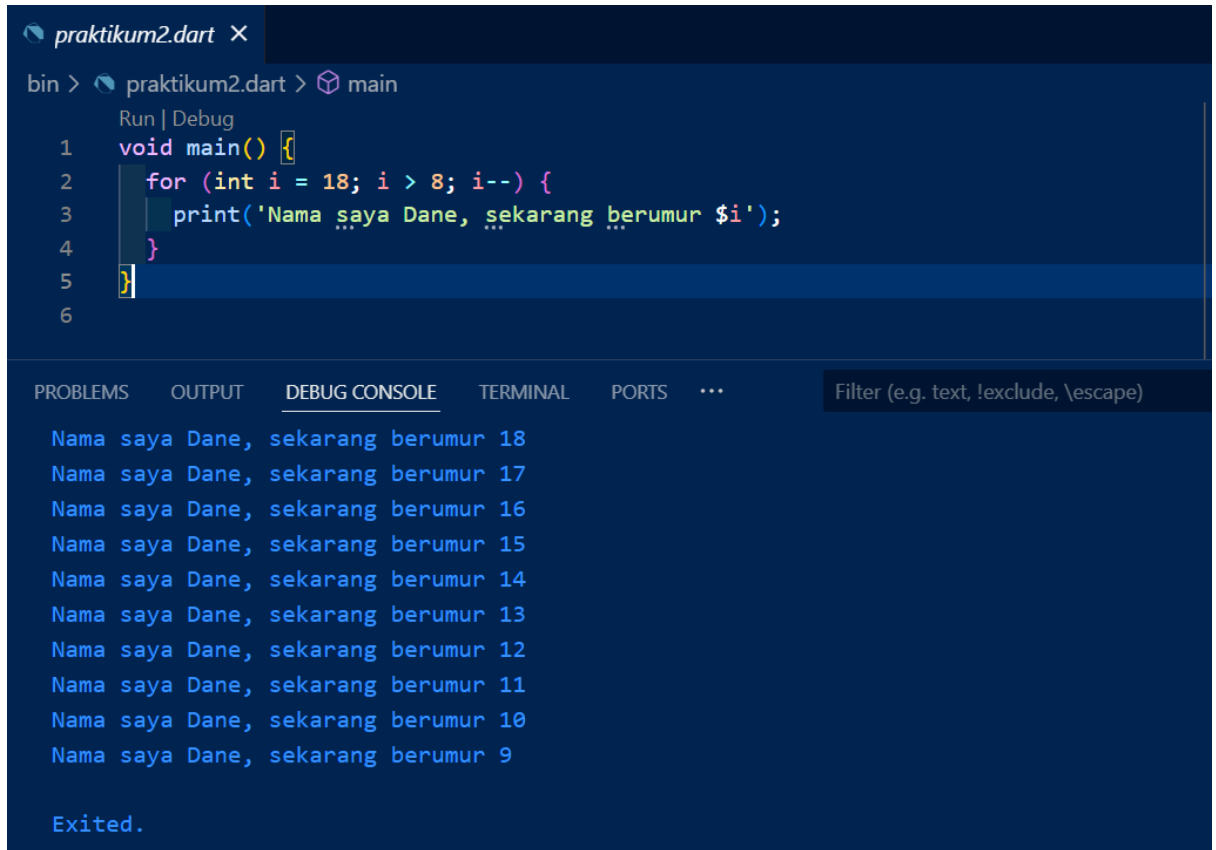


**Oleh:**

**Rega Dane Wijayanta**  
**SIB 3E**  
**2241760113 / 21**

## JAWABAN SOAL PRAKTIKUM

### 1. Hasil modifikasi code



The screenshot shows an IDE window titled 'praktikum2.dart'. The code editor contains the following Dart code:

```
bin > praktikum2.dart > main
Run | Debug
1 void main() {
2   for (int i = 18; i > 8; i--) {
3     print('Nama saya Dane, sekarang berumur $i');
4   }
5 }
6
```

Below the code editor, the 'DEBUG CONSOLE' tab is active, displaying the output of the program:

```
Nama saya Dane, sekarang berumur 18
Nama saya Dane, sekarang berumur 17
Nama saya Dane, sekarang berumur 16
Nama saya Dane, sekarang berumur 15
Nama saya Dane, sekarang berumur 14
Nama saya Dane, sekarang berumur 13
Nama saya Dane, sekarang berumur 12
Nama saya Dane, sekarang berumur 11
Nama saya Dane, sekarang berumur 10
Nama saya Dane, sekarang berumur 9

Exited.
```

### 2. Memahami bahasa pemrograman **Dart** sebelum menggunakan framework **Flutter** sangat penting karena beberapa alasan berikut:

- **Bahasa Utama Flutter:** Flutter menggunakan Dart sebagai bahasa pemrograman utamanya sehingga kita harus memahami dasar-dasar Dart seperti variabel, tipe data, fungsi, objek, dan sebagainya.
- **Pemrograman Asinkron:** Dart memiliki dukungan bawaan untuk pemrograman asinkron melalui konsep **Future**, **async**, dan **await**, yang penting untuk menangani operasi I/O atau tugas yang memakan waktu lama (seperti pengambilan data dari server).
- **Object-Oriented Programming (OOP):** Dart merupakan bahasa yang berorientasi objek, dan Flutter secara ekstensif menggunakan konsep OOP, seperti kelas, objek, pewarisan (inheritance), dan polimorfisme.
- **Optimasi dan Performa:** Dart di-compile secara langsung (AOT - Ahead Of Time) menjadi kode native, yang meningkatkan performa aplikasi Flutter. Dengan memahami Dart, kita dapat menulis kode yang lebih efisien dan dioptimalkan, serta memanfaatkan fitur Dart seperti pengumpulan sampah (garbage collection) dan manajemen memori yang lebih baik.
- **Integrasi dengan Ekosistem Dart:** Dart memiliki banyak pustaka (library) dan paket yang dapat digunakan di Flutter.

### 3. Rangkuman materi:

- **Dart sebagai bahasa dasar Flutter:** Penting untuk memahami Dart karena seluruh pengembangan Flutter menggunakan Dart, termasuk kode aplikasi dan manajemen dependensi.

- **Fitur Dart:** Mendukung Garbage Collection, type-safe, portabilitas, dan productive tooling.
- **Eksekusi kode Dart:** Dapat dijalankan melalui Dart VM atau dikompilasi ke JavaScript.
- **Pemrograman berorientasi objek:** Dart menggunakan konsep OOP yang mirip dengan bahasa lain seperti Java.
- **Operator dan sintaks dasar:** Mirip dengan bahasa pemrograman lainnya, Dart memiliki operator aritmatika, logika, dan relasional.