

LAPORAN PEMROGRAMAN MOBILE
PERTEMUAN 2 PENGANTAR BAHASA PEMROGRAMAN DART



NAMA : Arif Prasajo

NIM : 2241760100

KELAS : SIB 3E

PRODI SISTEM INFORMASI BISNIS
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

Tugas Praktikum

- Modifikasilah kode pada baris 3 di VS Code atau Editor Code favorit Anda berikut ini agar mendapatkan keluaran (*output*) sesuai yang diminta!

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

- Output yang diminta

```
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
```

Jawaban :

```
1 void main() {  
2     for (int i = 18; i > 8; i--) {  
3         print('Nama saya adalah Arif Prasajo, sekarang berumur $i');  
4     }  
5 }
```

```
Connecting to VM Service at ws://127.0.0.1:53180/xy3JMG0VQG0=/ws  
Nama saya adalah Arif Prasajo, sekarang berumur 18  
Nama saya adalah Arif Prasajo, sekarang berumur 17  
Nama saya adalah Arif Prasajo, sekarang berumur 16  
Nama saya adalah Arif Prasajo, sekarang berumur 15  
Nama saya adalah Arif Prasajo, sekarang berumur 14  
Nama saya adalah Arif Prasajo, sekarang berumur 13  
Nama saya adalah Arif Prasajo, sekarang berumur 12  
Nama saya adalah Arif Prasajo, sekarang berumur 11  
Nama saya adalah Arif Prasajo, sekarang berumur 10  
Nama saya adalah Arif Prasajo, sekarang berumur 9  
  
Exited.
```

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

Jawaban :

Pemahaman bahasa Dart penting sebelum menggunakan Flutter karena Dart adalah bahasa resmi Flutter. Dart mendukung fitur-fitur penting Flutter, membantu penulisan kode yang efisien, memungkinkan pengembangan aplikasi lintas platform, dan meningkatkan prospek karir di industri digital.

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

Jawaban :

- Dart adalah bahasa pemrograman inti yang digunakan dalam framework Flutter.
- Memahami Dart sangat penting bagi pengembang Flutter untuk membuat aplikasi mobile yang berkualitas.
- Kelebihan Bahasa Dart:
 - Dart dirancang untuk menggabungkan keunggulan berbagai bahasa pemrograman tingkat tinggi dengan fitur-fitur modern yang dibutuhkan untuk pengembangan aplikasi saat ini.
- Fitur Utama Dart:
 - **Productive Tooling:**
Dart menyediakan alat bantu (tools) produktif seperti analisis kode otomatis, plugin untuk IDE (Integrated Development Environment), dan ekosistem paket yang besar. Ini membantu pengembang menulis, memeriksa, dan memelihara kode dengan lebih efisien.
 - **Garbage Collection:**
Dart memiliki fitur *garbage collection* untuk mengelola dan menangani penggunaan memori secara otomatis. Fitur ini membantu dalam pengelolaan memori dengan menangani dealokasi memori yang ditempati oleh objek yang tidak lagi digunakan, sehingga menghindari kebocoran memori.
 - **Type Annotations (Optional):**
Dart mendukung anotasi tipe yang opsional untuk keamanan dan konsistensi data. Anotasi tipe ini memungkinkan pengembang untuk menambahkan tipe data pada variabel dan fungsi, membantu mencegah bug terkait tipe data selama pengembangan.

Dengan fitur-fitur seperti alat pengembangan yang produktif, pengelolaan memori otomatis, dan anotasi tipe opsional, Dart memungkinkan pengembang Flutter untuk bekerja lebih efisien, menghasilkan kode yang aman dan teroptimasi, serta menjaga performa aplikasi mobile yang baik. Memahami dan memanfaatkan fitur-fitur ini akan sangat membantu dalam proses pengembangan aplikasi menggunakan Flutter.

1. Sejarah dan Evolusi Dart:

- Diluncurkan pada tahun 2011, dengan versi stabil pertama pada tahun 2013.
- Pada tahun 2018, Dart 2.0 dirilis dengan perubahan besar untuk mendukung pengembangan aplikasi mobile, terutama dengan Flutter.
- Awalnya dirancang untuk pengembangan web sebagai pengganti JavaScript, Dart kini berfokus pada pengembangan mobile.

2. Keunggulan Dart:

- **Pemecahan Masalah JavaScript:** Dart mencoba mengatasi masalah yang ada di JavaScript, terutama dalam hal ketahanan dan konsistensi.
- **Performa dan Alat yang Lebih Baik:** Dirancang untuk performa optimal dengan dukungan IDE modern, menjadikannya ideal untuk proyek besar.
- **Kuat dan Fleksibel:** Mendukung anotasi tipe opsional dan fitur OOP, menjaga keseimbangan antara fleksibilitas dan ketangguhan.
- **Bahasa Modern dan Lintas Platform:** Dart terus ditingkatkan dengan fitur-fitur terkini untuk mendukung pengembangan lintas platform.

1. Cara Eksekusi Kode Dart:

- **Dart Virtual Machine (VM):** Eksekusi menggunakan Dart VM memungkinkan kompilasi *Just-In-Time (JIT)*, yang berguna untuk pengembangan karena mendukung fitur seperti debugging dan hot reload.
- **Kompilasi ke JavaScript:** Kode Dart dapat dikompilasi ke JavaScript untuk kompatibilitas dengan web.

2. Mode Eksekusi Dart:

- **Kompilasi Just-In-Time (JIT):** Kode dikompilasi saat dijalankan, memungkinkan pengembangan cepat dan iteratif.
- **Kompilasi Ahead-Of-Time (AOT):** Kode dikompilasi sebelumnya untuk performa optimal, biasanya digunakan untuk rilis produk.

1. Kemudahan Penggunaan:

- Bagi pengembang yang terbiasa dengan C atau JavaScript, sintaks Dart akan terasa familiar.
- Menyediakan tipe data bawaan dan operator standar yang umum ditemukan di bahasa pemrograman lain.

2. Pemrograman Berorientasi Objek (OOP):

- Dart adalah bahasa yang dirancang untuk OOP, mendukung konsep seperti enkapsulasi, pewarisan, komposisi, abstraksi, dan polimorfisme.
- Mirip dengan bahasa seperti Java, sehingga mudah dipelajari bagi mereka yang sudah mengenal OOP.

3. Operator di Dart:

- **Operator Aritmatika:** +, -, *, /, ~/ (pembagian bilangan bulat), %, dan negasi -.
- **Operator Penambahan dan Pengurangan:** ++ dan -- yang bekerja mirip dengan bahasa pemrograman lain.
- **Operator Persamaan dan Relasional:** ==, !=, >, <, >=, <=. Operator == membandingkan isi variabel, bukan referensi memori.
- **Operator Logika:** ! (negasi), || (OR), && (AND).