SOAL 1

```
void main() {
  for (int i=18; i >= 9; i--){
    print('Nama saya Fulan, Sekarang Berumur : ${i}');
  }
}
```

```
Nama saya Fulan, Sekarang Berumur : 18
Nama saya Fulan, Sekarang Berumur : 17
Nama saya Fulan, Sekarang Berumur : 16
Nama saya Fulan, Sekarang Berumur : 15
Nama saya Fulan, Sekarang Berumur : 14
Nama saya Fulan, Sekarang Berumur : 13
Nama saya Fulan, Sekarang Berumur : 12
Nama saya Fulan, Sekarang Berumur : 11
Nama saya Fulan, Sekarang Berumur : 10
Nama saya Fulan, Sekarang Berumur : 9
```

SOAL 2

Karena Dart adalah bahasa inti dari framework Flutter. Flutter, sebagai kerangka kerja modern, memerlukan bahasa pemrograman tingkat tinggi yang modern untuk memberikan pengalaman optimal bagi pengembang dan memungkinkan pembuatan aplikasi seluler yang luar biasa.

SOAL 3

Pengertian Bahasa Dart

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, kelebihannya, dan mengapa itu adalah bahasa pemrograman yang dipilih untuk Flutter.

Kelebihan Bahasa Pemograman Dart

Adapun beberapa Keunggulan Bahasa Pemograman dart seperti

- Productive tooling: merupakan fitur kakas (tool) 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 type annotations bersifat opsional, Dart tetap aman karena menggunakan fitur type-safe dan type inference untuk menganalisis types saat runtime. Fitur ini penting untuk menemukan bug selama kompilasi kode.
- Portability: bahasa Dart tidak hanya untuk web (yang dapat diterjemahkan ke JavaScript) tetapi juga dapat dikompilasi secara native ke kode Advanced RISC Machines (ARM) dan x86.

Sejarah Perkembangan Bahasa Pemograman Dart

Dart, diluncurkan pada 2011, awalnya difokuskan pada pengembangan web dengan tujuan menggantikan JavaScript. Setelah rilis stabil pada 2013 dan pembaruan besar di Dart 2.0 pada 2018, Dart kini lebih fokus pada pengembangan aplikasi mobile, termasuk Flutter. Dart berupaya memperbaiki kelemahan JavaScript dengan menawarkan performa tinggi dan alat yang kuat untuk proyek besar. Bahasa ini menggabungkan fleksibilitas dengan ketangguhan melalui fitur OOP dan type annotations opsional. Sebagai bahasa modern yang mendukung lintas platform, Dart terus berkembang, menjadikannya pilihan utama untuk framework Flutter.

Cara Dart Bekerja

Dart dapat dieksekusi melalui dua metode: Dart Virtual Machines (VMs) dan JavaScript compilation. Kode Dart dapat dijalankan di lingkungan yang mendukung bahasa ini dengan memperhatikan fitur-fitur seperti runtime systems, core libraries, dan garbage collectors. Eksekusi kode Dart dapat dilakukan dalam dua mode: Just-In-Time (JIT) compilation, yang mengompilasi kode saat dibutuhkan dan mendukung fitur debugging serta hot reload selama pengembangan; dan Ahead-Of-Time (AOT) compilation, yang mengompilasi kode sebelumnya untuk performa yang lebih tinggi, meskipun fitur seperti debugging dan hot reload tidak tersedia.

Struktur Bahasa Pemograman Dart

Dart adalah bahasa pemrograman yang memiliki banyak kesamaan dengan bahasa lain seperti C dan JavaScript, sehingga sintaksnya mudah dipahami oleh pengembang yang sudah familiar dengan bahasa-bahasa tersebut. Dart dirancang sebagai bahasa object-oriented (OO), dengan fitur seperti encapsulation, inheritance, composition, abstraction, dan polymorphism, mirip dengan bahasa OOP lainnya seperti Java. Dart menyediakan berbagai operator standar, termasuk aritmatika, inkrementasi, dekrementasi, persamaan, relasional, dan logika, yang sebagian besar mirip dengan bahasa lain, tetapi dengan beberapa perbedaan penting, seperti operator == yang membandingkan isi variabel, bukan referensinya. Operator-operator ini dapat disesuaikan sesuai kebutuhan, dan Dart juga mendukung shortcut operator untuk operasi yang lebih efisien. Secara keseluruhan, Dart adalah bahasa yang kuat dan fleksibel, dengan banyak fitur yang mendukung pengembangan aplikasi yang kompleks.

Hands-on with Dart

Flutter sangat dipengaruhi oleh bahasa Dart, sehingga pemahaman Dart penting untuk sukses dalam menggunakan Flutter. Untuk memulai, Anda dapat menggunakan DartPad, sebuah kakas online yang memungkinkan Anda menulis dan menjalankan kode Dart dengan mudah. DartPad mendukung core library Dart kecuali untuk library VM seperti dart:io. Contoh sederhana di DartPad melibatkan menulis fungsi main() yang mencetak "hello" lima kali.

Fungsi main() adalah titik masuk untuk eksekusi kode dalam aplikasi Dart. Dalam Dart, sebuah fungsi atau method menerima data, menjalankan kode, dan mengembalikan hasil. Fungsi main() pada contoh ini tidak mengembalikan data dan tidak menerima parameter, tetapi merupakan dasar yang penting dalam memahami struktur dan sintaksis bahasa Dart.