



Mata Kuliah : Pemrograman Mobile
Program Studi : D4 – Sistem Informasi Bisnis
Semester : 5

Kelas : SIB-3E
NIM : 2241760135
Nama : Agung Nugroho
Jobsheet Ke- : 2

Laporan Jobsheet

Tugas Praktikum

Soal 1

1. 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
```

Jawab:



Program:

```
import 'package:pertemuan 2/pertemuan 2.dart' as pertemuan_2;

Run | Debug
void main() {
  for (int i = 20; i > 10; i--) {
    print('Nama saya adalah Agung, sekarang berumur ${i - 2}');
  }
}
```

Output:

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

Exited.
```

Soal 2

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

Jawab: Karena dart adalah bahasa dasar dari framework Flutter, flutter dibangun di atas bahasa pemrograman Dart.

Soal 3

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

Pengertian Dart

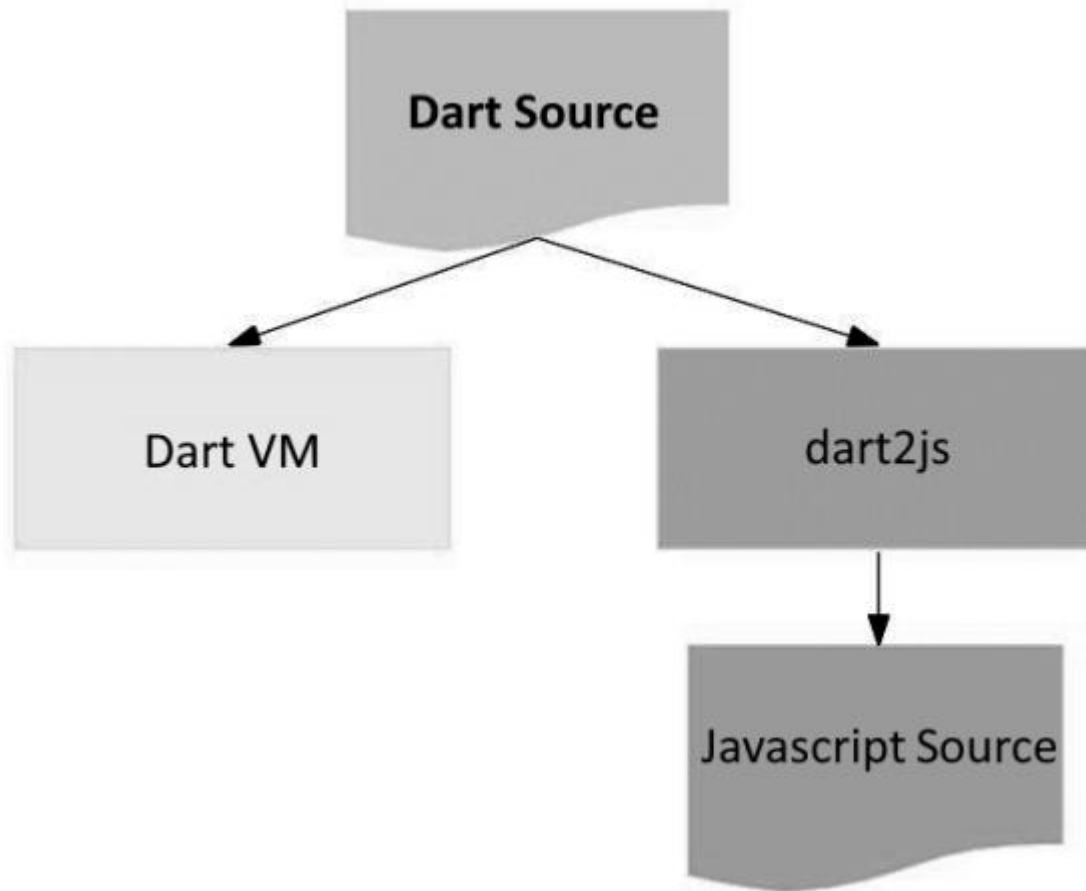


Bahasa Dart adalah inti dari framework Flutter. Kerangka kerja modern seperti Flutter membutuhkan bahasa modern tingkat tinggi agar bisa memberikan pengalaman terbaik kepada pengembang, serta memungkinkan untuk membuat aplikasi seluler yang luar biasa.

Tujuan belajar Dart:

- 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

Bagaimana Dart Bekerja?



Dart VM and JavaScript compilation

Kode Dart dapat dieksekusi pada lingkungan yang mendukung bahasa Dart. Lingkungan yang mendukung bahasa Dart perlu memperhatikan fitur-fitur penting seperti berikut:

- *Runtime systems*
- *Dart core libraries*
- *Garbage collectors*

Dart banyak berkontribusi pada fitur Flutter yang paling terkenal, yaitu hot reload, yang didasarkan pada kompiler Dart JIT. Hal ini memungkinkan pengembang untuk mendapatkan umpan balik yang sangat cepat terhadap perubahan kode yang dilakukan, sehingga memungkinkan untuk beralih lebih cepat. Dapat mendukung pengembangan perangkat lunak yang lebih cepat dan berkualitas tinggi.

Perkenalan struktur bahasa Dart

Object orientation



Seperti kebanyakan bahasa modern, Dart dirancang untuk **object-oriented (OO)**. Secara singkat, Bahasa OOP didasarkan pada konsep **objek** yang menyimpan kedua data (disebut **fields**) dan kode (disebut **methods**). Objek-objek ini dibuat dari cetak biru yang disebut **class** yang mendefinisikan *field* dan *method* yang akan dimiliki oleh sebuah objek.

Sesuai prinsip OO memastikan bahwa Dart memiliki fitur encapsulation, inheritance, composition, abstraction, dan polymorphism. Kita akan mempelajari kelas Dart lebih banyak lagi di pertemuan dengan topik *Class Dart* dan *Construct*, namun sudah cukup jika Anda telah belajar konsep OO dalam bahasa lain seperti Java, maka sebagian besar desain OO pada Dart akan sangat mirip.

Dart operators

Di Dart, operator tidak lebih dari *method* yang didefinisikan dalam *class* dengan sintaks khusus.

Jadi, ketika Anda menggunakan operator seperti `x == y`, seolah-olah Anda sedang memanggil `x.==(y)` metode untuk melakukan perbandingan kesetaraan.

Seperti yang mungkin telah Anda catat, kita menggunakan *method* pada `x`. Untuk semua tipe data, tidak seperti bahasa Java yang memiliki data primitif, `x` selalu berupa turunan dari kelas yang memiliki *method*. Ini berarti bahwa operator dapat diganti sesuai logika yang Anda inginkan.

Arithmetic operators

Dart hadir dengan banyak operator *typical* yang bekerja seperti banyak bahasa pemrograman lainnya; yaitu sebagai berikut:

- `+` untuk tambahan.
- `-` untuk pengurangan.
- `*` untuk perkalian.
- `/` untuk pembagian.
- `~/` untuk pembagian bilangan bulat. Di Dart, setiap pembagian sederhana dengan `/` menghasilkan nilai *double*. Untuk mendapatkan nilai bilangan bulat, Anda perlu membuat semacam transformasi (yaitu, *typecast*) dalam bahasa pemrograman lain; namun Dart sudah mendukung untuk operasi ini.
- `%` untuk operasi modulus (sisa bagi dari bilangan bulat).



- -expression untuk negasi (yang membalikkan suatu nilai).

Increment and decrement operators

Operator penambahan dan pengurangan juga merupakan operator umum dan diimplementasikan pada angka, sebagai berikut:

- ++var atau var++ untuk menambah nilai variabel var sebesar 1
- --var atau var-- untuk mengurangi nilai variabel var sebesar 1

Operator Dart increment dan decrement berperilaku mirip dengan bahasa lain. Penerapan operator increment dan decrement sangat baik untuk operasi perhitungan pada perulangan.

Equality and relational operators

Persamaan operator Dart dijelaskan sebagai berikut:

- == untuk memeriksa apakah operan sama
- != untuk memeriksa apakah operan berbeda

Untuk melakukan pengujian relasional, maka gunakan operator sebagai berikut:

- > memeriksa apakah operan kiri lebih besar dari operan kanan
- < memeriksa apakah operan kiri lebih kecil dari operan kanan
- >= memeriksa apakah operan kiri lebih besar dari atau sama dengan operan kanan
- <= memeriksa apakah operan kiri kurang dari atau sama dengan operan kanan

Logical operators

Operator logika di Dart adalah operator yang diterapkan pada operan bool; bisa berupa variabel, ekspresi, atau kondisi. Selain itu, dapat dikombinasikan dengan ekspresi kompleks dengan menggabungkan nilai ekspresi yang dievaluasi. Operator logika yang disediakan adalah sebagai berikut:

- !expression negasi atau kebalikan hasil ekspresi—
yaitu, true menjadi false dan false menjadi true.
- || menerapkan operasi logika OR antara dua ekspresi.



- && menerapkan operasi logika AND antara dua ekspresi.

Sekarang kita telah mengetahui dasar-dasar bahasa pemrograman Dart, mari kita lihat beberapa contoh kode Dart!