

TUGAS PRAKTIKUM

Jobsheet 2 : Pengantar Bahasa Pemrograman Dart - Bagian 1

Disusun untuk memenuhi tugas mata kuliah Pemrograman Mobile

Dosen Pengampu :

Samsul Arifin, S.Kom.MMSI



DISUSUN OLEH:

PUTRI AYU ALICIAWATI

2241720132

TI-3C

D-IV TEKNIK INFORMATIKA

TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2023/2024

Soal 3

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

Jawab :

Bahasa Dart adalah inti dari framework Flutter. Dart bertujuan untuk menggabungkan kelebihan-kelebihan dari sebagian besar bahasa tingkat tinggi dengan fitur-fitur bahasa pemrograman terkini, antara lain sebagai berikut:

- Productive tooling: untuk menganalisis kode, plugin IDE, dan ekosistem paket yang besar.
- Garbage collection: untuk mengelola atau menangani dealokasi memori.
- 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.

Dart diluncurkan pada tahun 2011. Dart merilis versi stabilnya pada tahun 2013. 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

Eksekusi kode Dart dapat beroperasi dalam dua mode — kompilasi Just-In-Time (JIT) atau Kompilasi Ahead-Of-Time (AOT).

Object Orientation

- Dart dirancang untuk object-oriented (OO). Secara singkat, Bahasa OOP didasarkan pada konsep objek yang menyimpan kedua data (disebut fields) dan kode (disebut methods).
- Sesuai prinsip OO memastikan bahwa Dart memiliki fitur encapsulation, inheritance, composition, abstraction, dan polymorphism.

Dart Operators

- ketika Anda menggunakan operator seperti $x == y$, seolah-olah Anda sedang memanggil $x.==(y)$ metode untuk melakukan perbandingan kesetaraan.
- operator dapat diganti sesuai logika yang diinginkan.

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 (sisanya 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

Equality and relational operators

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

- !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.