

LAPORAN PRAKTIKUM

JOBSHEET 2 : Pengantar Bahasa Pemrograman Dart - Bagian 1

Disusun untuk memenuhi tugas

Mata Kuliah : Pemrograman Mobile

Dosen Pengampu : Samsul Arifin



Disusun Oleh :

Nama : Lintang Aprillya Sari

Kelas : 3C

Absen : 13

NIM : 2241720231

PROGRAM STUDI D-IV TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2024/2025

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 :

Rangkuman Pengantar Bahasa Pemrograman Dart

- Tujuan dari dart adalah untuk menggabungkan kelebihan – kelebihan dari Sebagian besar Bahasa Tingkat tinggi dengan fitur – fitur Bahasa pemrograman terkini, antara lain :
 - Productive tooling
 - Garbage collection
 - Type annotations (opsional)
 - Statically typed
 - Portability
- Di launch tahun 2011 & versi stabilnya tahun 2013.
 - Awalnya berfokus pada pengembangan web, dengan tujuan utama menggantikan JavaScript (skrg berfokus pada mobile development, termasuk Flutter).
 - Mencoba memecahkan masalah pada JavaScript.
 - Menawarkan performa terbaik dan alat yang lebih baik untuk proyek berskala besar.
 - Dibentuk agar kuat dan fleksibel.
- Eksekusi kode Dart dapat beroperasi dalam dua mode — kompilasi Just-In-Time (tempat kode sumber dikompilasi sesuai kebutuhan) atau Kompilasi Ahead-Of-Time (VM bekerja lebih seperti sistem runtime Dart).
- Sesuai prinsip object oriented memastikan bahwa Dart memiliki fitur encapsulation, inheritance, composition, abstraction, dan polymorphism.
- Dart hadir dengan banyak operator *typical* yang bekerja seperti banyak bahasa pemrograman lainnya. Dart juga menyediakan *shortcut* operator untuk menggabungkan variabel setelah operasi lainnya. Operator aritmatika atau *shortcut* operator adalah +=, -=, *=, /=, dan ~/=.
- Increment dan decrement
 - ++var atau var++ untuk menambah nilai variable var sebesar 1
 - --var atau var-- untuk mengurangi nilai variable var sebesar 1
- Equality dan operasi relasional
 - == untuk memeriksa apakah operan sama
 - += untuk memeriksa apakah operan berbeda
 - Operasi pengujian relasional
 1. > memeriksa apakah operan kiri lebih besar dari operan kanan
 2. < memeriksa apakah operan kiri lebih kecil dari operan kanan
 3. >= memeriksa apakah operan kiri lebih besar dari atau sama dengan operan kanan
 4. <= memeriksa apakah operan kiri kurang dari atau sama dengan operan kanan
- Logical operators
 - !expression negasi atau kebalikan hasil ekspresi (true menjadi false, dan sebaliknya)
 - || menerapkan operasi logika or antara 2 ekspresi
 - && menerapkan operasi logika and