**Rangkuman Materi CodeLab Dasar Dart**

1. **Bahasa Pemrograman Dart**

Bahasa Dart adalah inti dari framework Flutter. Dart adalah bahasa pemrograman yang dikembangkan oleh Google dengan fokus utama pada pengembangan aplikasi Flutter. Dart digunakan secara luas untuk membangun antarmuka pengguna yang responsif dan indah dalam berbagai platform, seperti aplikasi mobile, web, dan desktop, menggunakan kerangka kerja Flutter.

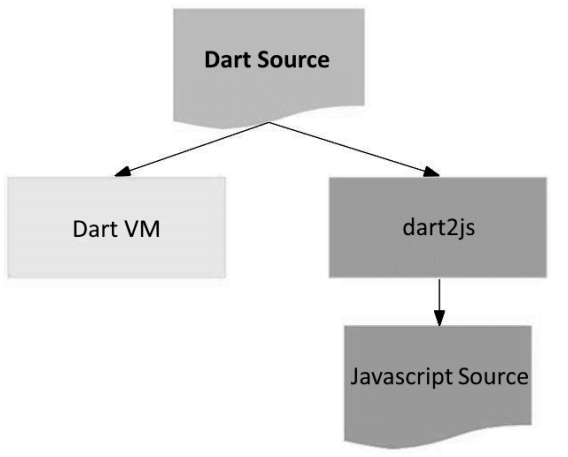
1. **Tujuan Dart**

Dart bertujuan untuk menggabungkan kelebihan-kelebihan dari sebagian besar bahasa tingkat tinggi dengan fitur-fitur bahasa pemrograman terkini, antara lain sebagai berikut :

1. Productive Tooling : fitur kakas (tool) untuk menganalisis kode, plugin IDE, dan ekosistem paket yang besar.
2. Garbage Collection : mengelola atau menangani dealokasi memori (terutama memori yang ditempati oleh objek yang tidak lagi digunakan).
3. Type Annotation : untuk keamanan dan konsistensi dalam mengontrol semua data dalam aplikasi.
4. 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.
5. 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.
6. **Evolution of Dart**

Dart, yang diluncurkan pada tahun 2011 dan merilis versi stabil pada tahun 2013, awalnya ditujukan untuk menggantikan JavaScript dalam pengembangan web, tetapi sekarang berfokus pada pengembangan mobile dengan kerangka kerja Flutter. Tujuannya adalah memecahkan masalah ketahanan JavaScript dan memberikan kinerja terbaik serta alat yang lebih baik untuk proyek berskala besar. Dart tetap kuat dan fleksibel dengan type annotations opsional dan fitur OOP yang ditambahkan. Bahasa ini terus berkembang untuk menjaga relevansinya dan telah dipilih oleh tim Flutter framework untuk pengembangan lintas platform yang modern dan fleksibel.

1. **Bagaimana Dart Bekerja**



Dalam eksekusi kode pada bahasa pemrograman Dart dapat dilakukan dengan dua cara, yaitu Dart VM dan JavaScript Compilation. Eksekusi kode Dart dapat beroperasi dalam dua mode — kompilasi **Just-In-Time (JIT)** atau Kompilasi **Ahead-Of-Time (AOT).**

* Kompilasi JIT adalah tempat kode sumber dikompilasi sesuai kebutuhan—Just in time. Dart VM memuat dan mengkompilasi kode sumber ke kode mesin asli (native). Pendekatan ini digunakan untuk menjalankan kode pada command line atau selama proses pengembangan aplikasi mobile yang dapat memanfaatkan fitur seperti debugging dan hot reload.
* Kompilasi AOT adalah dimana Dart VM dan kode Anda dikompilasi sebelumnya, VM bekerja lebih seperti sistem runtime Dart, yang menyediakan garbage collector dan metode-metode native dari Dart software development kit (SDK) pada aplikasi. Pendekatan ini memiliki keuntungan performa yang sangat besar dibandingkan kompilasi JIT, tetapi fitur lain seperti debugging dan hot reload tidak tersedia.

1. **Struktur Dart**

* **Object Orientation**

Dalam bahasa Dart, pemrograman berorientasi objek mengandalkan konsep objek yang memiliki data (fields) dan kode (methods). Objek-objek ini dibuat berdasarkan cetak biru yang disebut class. Dart mendukung prinsip-prinsip dasar pemrograman berorientasi objek seperti encapsulation, inheritance, composition, abstraction, dan polymorphism.

* **Dart Operator**

Selain itu, dalam Dart, operator adalah method yang didefinisikan dalam class dengan sintaks khusus. Misalnya, operator kesetaraan (==) sebenarnya adalah pemanggilan method x.==(y) untuk membandingkan kesetaraan antara dua objek. Ini berbeda dari bahasa seperti Java yang memiliki tipe data primitif. Dart memiliki operator aritmatika seperti +, -, \*, /, ~/, dan %, serta operator penambahan dan pengurangan seperti ++var dan --var.

* **Arithmetic operators**

Dart juga memiliki operator persamaan dan operator relasional, yang memungkinkan Anda untuk memeriksa kesamaan dan perbandingan antara nilai. Dalam Dart, operator persamaan (==) membandingkan isi variabel, bukan alamat memori, dan Dart memiliki fitur type safety yang menghilangkan kebutuhan untuk operator seperti === yang ada dalam bahasa JavaScript.

* **Increments and Decrements Operators**

**++var** atau **var++** untuk menambah nilai variabel **var** sebesar 1.

**-var** atau **var--** untuk mengurangi nilai variabel **var** sebesar 1.

Operator increment dan decrement dalam Dart berperilaku serupa dengan bahasa pemrograman lainnya. Mereka digunakan untuk operasi perhitungan yang melibatkan peningkatan atau pengurangan nilai variabel, terutama dalam konteks perulangan.

* **Logical Operators**

Dart juga memiliki operator persamaan dan operator relasional, yang memungkinkan Anda untuk memeriksa kesamaan dan perbandingan antara nilai. Dalam Dart, operator persamaan (==) membandingkan isi variabel, bukan alamat memori, dan Dart memiliki fitur type safety yang menghilangkan kebutuhan untuk operator seperti === yang ada dalam bahasa JavaScript.

Terakhir, Dart memiliki operator logika seperti negasi (!), OR (||), dan AND (&&) yang digunakan pada operan boolean untuk menggabungkan dan memanipulasi kondisi logika.