



Mobile Application

Catatan Kuliah #12

Alauddin Maulana Hirzan, M. Kom

0607069401

Pemrograman Dart



Pemrograman Dart

Mengenal Pemrograman Dart

Dart adalah bahasa pemrograman berorientasi objek modern yang dikembangkan oleh Google. Dart dirancang agar efisien, fleksibel, dan dioptimalkan untuk membangun aplikasi web, seluler, dan desktop.

Bahasa ini difokuskan untuk pembuatan aplikasi modern dan dapat dijalankan untuk berbagai macam platform. Dalam konteks perkuliahan ini, Dart digunakan untuk aplikasi mobile.

Pemrograman Dart

Dukungan App Dart + Flutter





Pemrograman Dart

Fitur Dart

- ▶ **Pengetikan yang kuat:** Dart adalah bahasa yang sangat kuat dalam pengetikan, yang berarti programmer harus secara eksplisit mendeklarasikan tipe variabel.
- ▶ **Pengumpulan Sampah:** Dart menggunakan pengumpul sampah untuk mengelola memori secara otomatis, membebaskan pengembang dari manajemen memori manual.
- ▶ **Pemrograman Asinkron:** Dart memiliki dukungan bawaan untuk pemrograman asinkron menggunakan kata kunci **async** dan **await**.



Pemrograman Dart

Fitur Dart

- ▶ **Pemrograman Berorientasi Objek:** Dart adalah bahasa berorientasi objek, mendukung kelas, pewarisan, antarmuka, dan konsep berorientasi objek penting lainnya.
- ▶ Kompilasi **Just In Time (JIT)** dan **Ahead of Time (AOT)**: Dart menawarkan kompilasi JIT dan AOT. Kompilasi JIT memungkinkan siklus pengembangan yang lebih cepat, sementara kompilasi AOT menghasilkan kode asli yang sangat dioptimalkan untuk kinerja yang lebih baik dalam produksi.
- ▶ **Portabilitas:** Ini tidak hanya untuk web (diterjemahkan ke JavaScript), tetapi dapat dikompilasi secara asli ke kode ARM dan x86.



Pemrograman Dart

Cara Kerja Dart

Dalam melakukan kompilasi aplikasi mobile berbasis **Dart**. Dart dapat dikompilasikan dengan menggunakan 3 caraL

- ▶ **Just-in-Time (JIT)**
- ▶ **Ahead-of-Time (AOT)**
- ▶ **Platform-specific**



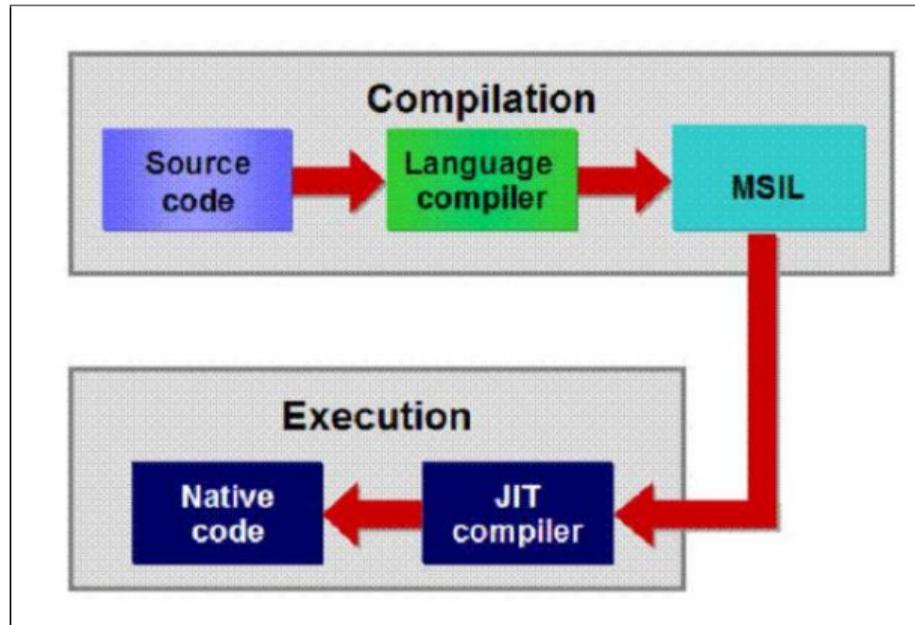
Pemrograman Dart

Cara Kerja Dart - **Just In Time**

- ▶ Selama pengembangan, aplikasi Dart biasanya dikompilasi menggunakan kompilasi JIT. Dengan kompilasi JIT, Dart Virtual Machine (VM) menerjemahkan kode sumber Dart ke dalam bytecode perantara.
- ▶ Bytecode kemudian dieksekusi oleh Dart VM, yang secara dinamis dapat mengoptimalkan kode pada saat runtime untuk kinerja yang lebih baik.
- ▶ Kompilasi JIT memungkinkan fitur-fitur seperti hot reload, di mana perubahan kode dapat diterapkan secara instan tanpa memulai ulang aplikasi.

Pemrograman Dart

Kompilasi Dengan JIT





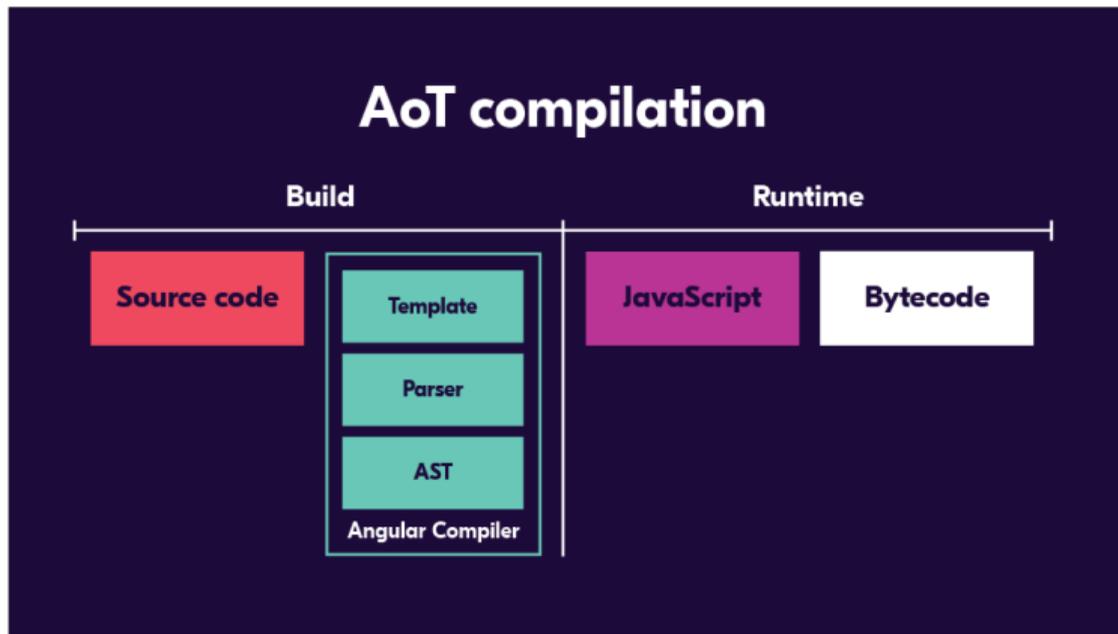
Pemrograman Dart

Cara Kerja Dart - **Ahead of Time**

- ▶ Ketika menerapkan aplikasi Dart untuk produksi, kompilasi Ahead-of-Time (AOT) digunakan.
- ▶ Kompilasi AOT melibatkan penerjemahan kode sumber Dart ke dalam kode mesin sebelumnya, sebelum menjalankan aplikasi.
- ▶ Hal ini menghasilkan file yang dapat dieksekusi mandiri yang dapat dijalankan langsung di platform target, tanpa memerlukan Dart VM.
- ▶ Kompilasi AOT meningkatkan waktu startup, mengurangi penggunaan memori, dan menawarkan kinerja keseluruhan yang lebih baik untuk aplikasi yang digunakan.

Pemrograman Dart

Kompilasi Dengan AOT





Pemrograman Dart

Cara Kerja Dart - **Platform-Specific**

- ▶ Dart mendukung kompilasi ke berbagai platform target, termasuk kode mesin asli untuk platform desktop dan seluler, serta JavaScript untuk aplikasi web.
- ▶ Kompiler Dart menghasilkan keluaran khusus platform berdasarkan platform target yang dipilih.
- ▶ Misalnya, ketika menargetkan Android atau iOS, kode Dart dikompilasi ke kode mesin ARM asli. Ketika menargetkan web, kode Dart dikompilasi menjadi kode JavaScript yang efisien.



Pemrograman Dart

Kompilasi Dengan Dart2JS

dart → dart2js → js



Pemrograman Dart

Flutter dengan Dart

Dalam membangun antarmuka mobile, Flutter menggunakan bahasa pemrograman Dart. Sehingga dalam membangun aplikasi, Flutter dapat menghasilkan aplikasi untuk berbagai macam platform.

Kode Dart berinteraksi dengan kerangka kerja Flutter untuk merender komponen UI, menangani input pengguna, mengelola state, dan melakukan tugas-tugas lain yang diperlukan untuk membangun aplikasi.

Kode Dart dikompilasi menjadi kode native atau JavaScript yang efisien, tergantung pada platform target, sehingga memungkinkan aplikasi Flutter mencapai kinerja tinggi dan tampilan dan nuansa asli platform.



Pemrograman Dart

Contoh Kode Flutter dengan Dart

```
class MyApp extends StatelessWidget {  
  const MyApp({super.key});  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      home: MyHomePage(title: 'Flutter Demo Home Page'),  
    );  
  }  
}
```



Pemrograman Dart

Flutter + Dart Rendering

Dalam membangun aplikasi, Dart akan melakukan tampilan visual sesuai dengan Flutter tentukan. Berikut proses rendering yang dilakukan:

- ▶ **Konstruksi Pohon Widget:**
 - ▶ Pengembang menggunakan Dart untuk mendefinisikan struktur pohon widget, yang mewakili antarmuka pengguna aplikasi.
 - ▶ Setiap widget mewakili komponen UI tertentu, seperti tombol, bidang teks, wadah, dll.
 - ▶ Widget disusun secara hierarkis, dengan widget induk yang berisi widget anak untuk membuat tata letak UI yang diinginkan.





dart → dart2js → js



Pemrograman Dart

Flutter + Dart Rendering

- ▶ Rekonsiliasi:
 - ▶ Ketika pohon widget berubah, Flutter membandingkan pohon widget sebelumnya dengan yang baru dan menentukan perbedaannya.
 - ▶ Flutter mengidentifikasi widget mana yang telah ditambahkan, dihapus, atau dimodifikasi, mengoptimalkan pembaruan yang diperlukan untuk mencerminkan perubahan tersebut.
- ▶ Merender Objek:
 - ▶ Objek render bertanggung jawab untuk melukiskan diri mereka sendiri di layar dan menangani interaksi.
 - ▶ Objek render membentuk struktur pohonnya sendiri, yang disebut "pohon objek render", sejajar dengan pohon widget.



Pemrograman Dart

Flutter + Dart Rendering

- ▶ Tata Letak dan Lukisan:
 - ▶ Fase tata letak melibatkan penentuan ukuran dan posisi setiap objek render dalam pohon objek render.
 - ▶ Tata letak dilakukan secara rekursif, mulai dari objek render akar dan menyebar ke bawah pohon.
- ▶ Akselerasi GPU:
 - ▶ Flutter menggunakan mesin grafis Skia dan berinteraksi dengan API grafis platform yang mendasarinya.
 - ▶ Proses rendering Flutter memanfaatkan GPU untuk membebaskan tugas rendering, sehingga menghasilkan rendering grafis yang cepat dan efisien.



Pemrograman Dart

SKIA Render Engine

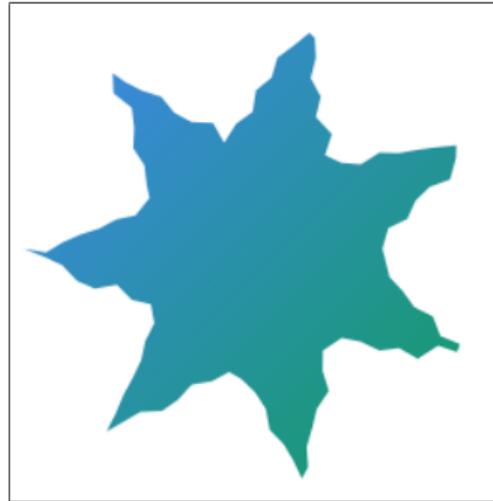
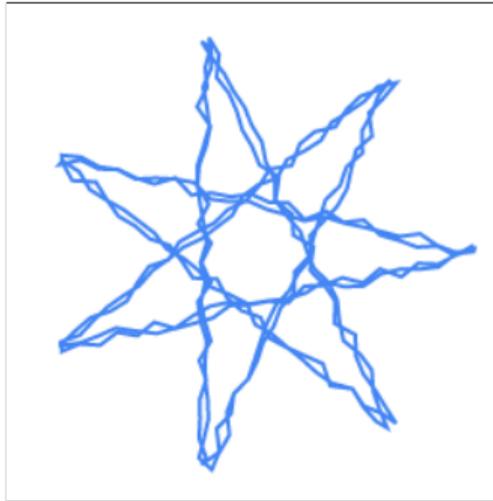
Skia adalah perpustakaan grafis 2D yang dikembangkan oleh Google, dan berfungsi sebagai mesin rendering yang digunakan oleh Flutter. Skia menyediakan seperangkat API dan alat yang komprehensif untuk merender grafis, termasuk jalur rendering, bentuk, teks, dan gambar.

Dengan memanfaatkan Skia sebagai mesin rendering, Flutter dapat menghadirkan antarmuka pengguna yang cepat, mulus, dan menarik secara visual. Dukungan lintas platform, akselerasi perangkat keras, dan kemampuan grafis yang komprehensif dari Skia memainkan peran penting dalam kemampuan Flutter untuk membuat aplikasi yang indah dan berkinerja baik di berbagai platform.

Pemrograman Dart

SKIA Render Engine

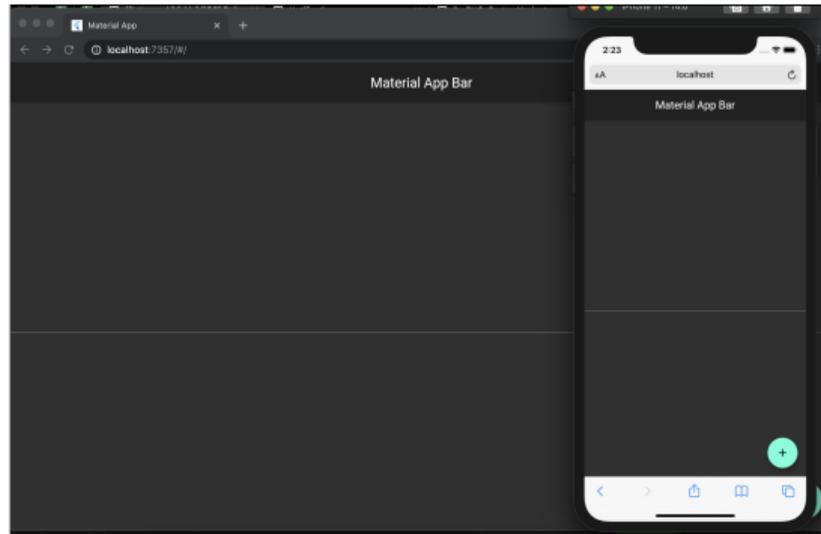
Contoh hasil rendering dengan SKIA:



Pemrograman Dart

SKIA Render Engine

Contoh hasil rendering app dengan SKIA:





THANK

YOU