

Nama : M. Ghaza Al-Ghazali

NIM : 2023573010141

## 1. Membuat Proyek Flutter

Membuat proyek Flutter baru dapat dilakukan dengan mudah melalui tiga pendekatan utama: Terminal/CLI, Android Studio, atau Visual Studio Code. Dengan perintah sederhana seperti `flutter create nama_proyek` di terminal, atau wizard yang disediakan oleh IDE, kerangka aplikasi lintas platform akan langsung terbentuk lengkap dengan folder `lib/`, `android/`, `ios/`, dan berkas standar lainnya.

Sebelum mulai menulis kode, pastikan lingkungan pengembangan siap dengan menjalankan `flutter doctor` untuk memeriksa dependensi. Pahami struktur proyek terutama titik masuk di `lib/main.dart` dan konfigurasi platform di folder `android/` dan `ios/`. Manfaatkan fitur Hot Reload untuk melihat perubahan kode secara instan tanpa memulai ulang aplikasi.

Pilih alur kerja (CLI, Android Studio, atau VS Code) yang paling sesuai dengan kebiasaan Anda, lalu gali lebih dalam dokumentasi resmi Flutter untuk menjelajahi widget, state management, dan praktik terbaik. Dengan fondasi yang benar dan rasa ingin tahu yang tinggi, Anda siap membangun aplikasi lintas platform yang produktif dan menarik.

## 2. Widget

Widget adalah elemen dasar yang membangun antarmuka pengguna (UI) di Flutter. Setiap widget merepresentasikan komponen visual seperti teks, gambar, tombol, atau tata letak, dan disusun dalam hierarki pohon widget untuk membentuk tampilan aplikasi.

Widget terbagi 2, yaitu:

- Stateless Widget

Stateless Widget adalah tipe widget yang bersifat statis dan tidak memiliki data yang dapat berubah selama runtime. Setelah di-build, tampilannya tidak akan diperbarui kembali meski terjadi interaksi pengguna atau perubahan variabel. Stateless widget hanya memanggil metode build satu kali untuk merender UI awalnya, sehingga cocok untuk elemen yang data dan tampilannya konstan, seperti teks sederhana atau ikon.

- Statefull Widget

Stateful Widget adalah widget yang mampu mempertahankan dan memperbarui state internalnya. Ketika state berubah misalnya karena interaksi pengguna widget memanggil `setState()`, yang memicu proses render ulang untuk memperbarui UI. Struktur Stateful Widget terdiri dari dua kelas: kelas widget itu sendiri dan kelas State yang menampung logika serta data yang dapat berubah sepanjang siklus hidup widget.

## 3. Emulator

Emulator adalah perangkat lunak yang mereplikasi lingkungan perangkat seluler di komputer pengembang. Dengan emulator, Anda dapat menjalankan dan menguji aplikasi Flutter seolah-olah berjalan di perangkat Android atau iOS asli tanpa perlu perangkat fisik. Emulator berperan penting sejak tahap pengembangan awal hingga debugging, memungkinkan pengembang melakukan iterasi cepat dengan fitur Hot Reload.

Daftar emulator yang sudah terpasang menggunakan perintah:

- `flutter emulators`

Untuk meluncurkan emulator tertentu, gunakan:

- `flutter emulators --launch <emulatorId>`

Setelah emulator aktif, jalankan aplikasi dengan:

- `flutter run`

Perintah ini secara otomatis mendeteksi emulator aktif dan menerapkan aplikasi ke dalamnya.