

Dhiva Kesuma Pertiwi

0806022410025

Proyek Flutter – State Management

Struktur kode saya bagi ke dalam tiga file utama, yaitu `main.dart`, `global_state.dart`, dan `counter_tile.dart`. Pada file `global_state.dart`, saya membuat sebuah kelas `state` yang diturunkan dari `ChangeNotifier`. Kelas ini berfungsi sebagai pusat penyimpanan data, dalam hal ini berupa nilai `counter`. Di dalam kelas tersebut terdapat method untuk menambah nilai `counter` dan memanggil `notifyListeners()` agar setiap widget yang mendengarkan `state` bisa segera ter-update.

File kedua, `counter_tile.dart`, berisi sebuah widget kustom yang saya buat untuk menampilkan nilai `counter` sekaligus menyediakan tombol untuk menambah nilainya. Dengan memisahkan widget ini, kode menjadi lebih modular dan mudah digunakan kembali tanpa perlu menulis ulang logika yang sama. Widget ini juga terhubung langsung dengan `state` global, sehingga setiap interaksi pengguna akan memengaruhi data yang sama.

Sementara itu, file `main.dart` berfungsi sebagai titik masuk aplikasi. Pada bagian ini, saya membungkus aplikasi dengan `ChangeNotifierProvider` agar `global_state.dart` bisa diakses dari mana saja di dalam widget tree. Di dalamnya juga saya memanggil `CounterTile`, sehingga `counter` dapat ditampilkan dan dikontrol oleh pengguna.

Selama proses pengerjaan, ada beberapa tantangan yang saya hadapi. Tantangan pertama adalah bagaimana menghubungkan `state` global dengan widget agar nilainya bisa otomatis diperbarui di layar. Hal ini teratasi setelah saya memahami bahwa `notifyListeners()` harus selalu dipanggil setiap kali ada perubahan data. Tantangan kedua adalah ketika widget tidak merespons perubahan `state`. Setelah ditelusuri, masalah ini muncul karena saya lupa menggunakan `Consumer` atau `Provider.of` dengan `context` yang benar. Tantangan terakhir adalah bagaimana membuat struktur kode tetap rapi, karena pada awalnya semua kode saya letakkan di `main.dart`. Masalah ini saya pecahkan dengan memisahkan kode ke dalam file khusus: `global_state.dart` untuk logic dan `counter_tile.dart` untuk widget kustom.