Dhiva Kesuma Pertiwi
0806022410025

Laporan Proyek Flutter: Advanced State Management

I. Pendahuluan

Proyek ini bertujuan untuk mengembangkan aplikasi counter menggunakan Flutter dengan penerapan manajemen state lanjutan. Aplikasi ini dirancang untuk menunjukkan perbedaan antara local state dan global state, serta bagaimana Provider dapat digunakan untuk mengelola state secara terpusat. Fitur utama aplikasi meliputi kemampuan menambah dan mengurangi nilai counter serta mengubah warna counter sebagai visual feedback saat diklik. Proyek ini juga bertujuan untuk memberikan pemahaman tentang konsep reactive UI di Flutter, di mana perubahan state secara otomatis mempengaruhi tampilan aplikasi.

II. Struktur Kode

Proyek terdiri dari beberapa file utama yang saling berinteraksi. File main.dart berfungsi sebagai entry point aplikasi, menginisialisasi ChangeNotifierProvider untuk global state, dan menampilkan halaman utama (HomePage) yang berisi daftar counter. File global_state.dart berisi class GlobalState yang mewarisi ChangeNotifier dan menyimpan daftar counter dalam bentuk objek Counter. Class ini menyediakan berbagai metode untuk menambah, mengurangi, dan mengubah properti counter, serta memanggil notifyListeners() setiap terjadi perubahan state agar widget yang terkait dapat diperbarui secara otomatis. File counter_tile.dart merupakan widget yang menampilkan masing-masing counter secara individual, mengambil data dari GlobalState, menampilkan nilai dan warna counter, serta menyediakan tombol increment, decrement, dan ubah warna. File counter.dart berfungsi sebagai model untuk menyimpan atribut value dan color setiap counter agar data tersusun secara terstruktur.

III. Alur Program

Saat aplikasi dijalankan, HomePage akan ditampilkan dengan daftar counter menggunakan ListView.builder(). Setiap counter ditampilkan melalui CounterTile, yang mengambil data dari GlobalState. Interaksi pengguna melalui tombol increment dan decrement akan memperbarui nilai counter pada GlobalState, kemudian notifyListeners()

dipanggil sehingga UI secara otomatis ter-refresh. Selain itu, ketika tombol ubah warna diklik, warna counter diperbarui berdasarkan indeks dan waktu saat ini (DateTime.now().second), sehingga memberikan variasi visual yang dinamis. Dengan cara ini, setiap perubahan state global dapat langsung terlihat di UI tanpa perlu reload seluruh aplikasi.

IV. Tantangan dan Solusi

1. Masalah konfigurasi pubspec.yaml dan dependencies package

   Tantangan: Beberapa package belum terpasang atau versinya tidak kompatibel, sehingga aplikasi menampilkan error saat dijalankan meskipun kode terlihat benar.

   Solusi: Memastikan semua dependencies sudah sesuai versi yang direkomendasikan Flutter, menjalankan flutter pub get, dan melakukan clean build (flutter clean) sebelum menjalankan ulang aplikasi.

2. Error saat menambahkan counter baru

   Tantangan: Saat menambahkan counter baru, output console menunjukkan error, meskipun fungsi tetap berjalan.

   Solusi: Mengecek implementasi fungsi penambahan counter di GlobalState, memastikan list counter di-update dengan benar dan notifyListeners() dipanggil setelah perubahan state.

3. Widget tidak merespon perubahan state global

   Tantangan: Beberapa widget tidak melakukan refresh otomatis ketika state global berubah.

   Solusi: Memastikan penggunaan Provider.of<GlobalState>(context, listen: true) atau Consumer<GlobalState> agar widget yang menggunakan state global diperbarui secara otomatis.

4. Variasi warna counter kurang terlihat

   Tantangan: Warna counter terkadang sama karena perhitungan awal yang statis.

Solusi: Menambahkan logika perubahan warna berdasarkan indeks dan waktu saat ini (DateTime.now().second) sehingga setiap klik menghasilkan variasi warna yang berbeda.

5. Increment dan decrement pada StatelessWidget

   Tantangan: StatelessWidget tidak memiliki state internal sehingga sulit menangani perubahan nilai counter secara lokal.

   Solusi: Memanfaatkan global state (GlobalState) untuk menyimpan dan memperbarui nilai counter sehingga widget tetap dapat merespon interaksi pengguna.

V. Kesimpulan

Proyek ini berhasil mengimplementasikan local state dan global state dalam aplikasi Flutter, sekaligus memperkuat pemahaman tentang konsep reactive UI. Penggunaan Provider memungkinkan perubahan state global diperbarui secara otomatis di seluruh widget yang relevan, sementara logika aplikasi tetap terpisah antara UI dan state management. Meskipun beberapa tantangan muncul terkait konfigurasi project, dependencies, dan debugging fungsi counter, proyek ini memberikan pengalaman langsung dalam menyelesaikan masalah nyata pada pengembangan aplikasi Flutter dengan manajemen state yang baik.