

LAPORAN SINGKAT TUGAS 14

Optimasi dan Debugging Aplikasi

Teknik optimasi yang diterapkan yaitu:

1. Optimasi kode, dengan melakukan refactoring dan menghapus kode yang tidak digunakan agar struktur kode lebih rapi dan mudah dipelihara.
2. Optimasi akses data, dengan mengurangi proses pemanggilan data yang tidak diperlukan sehingga performa aplikasi menjadi lebih efisien.

Kasus Error:

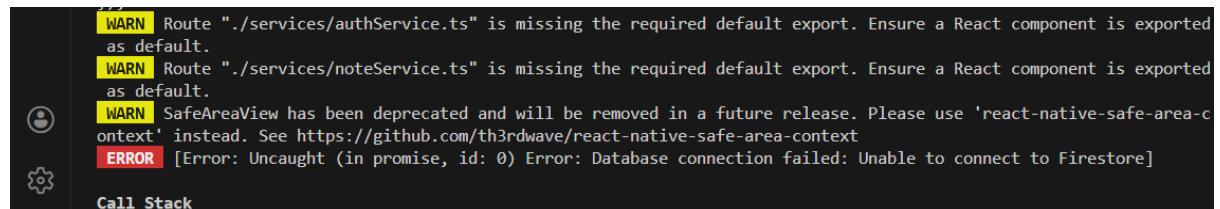
Pada tahap ini dilakukan simulasi error secara sengaja untuk melatih kemampuan analisis dan debugging aplikasi. Error yang disimulasikan berkaitan dengan kegagalan koneksi database Firestore, yang merupakan salah satu komponen penting dalam proses penyimpanan dan pengambilan data aplikasi. Simulasi error dilakukan dengan menambahkan perintah sebagai berikut:

```
throw new Error("Database connection failed: Unable to connect to Firestore");
```

Error tersebut digunakan untuk merepresentasikan kondisi di mana aplikasi tidak dapat terhubung ke database Firestore, yang dapat disebabkan oleh beberapa faktor, seperti:

- Konfigurasi database yang tidak valid
- Kredensial akses Firestore yang salah
- Koneksi jaringan yang tidak stabil
- Layanan database yang sedang tidak tersedia

Dengan adanya error ini, proses eksekusi aplikasi secara otomatis dihentikan dan sistem menampilkan pesan error ke console dan terminal.



```
WARN | Route "./services/authService.ts" is missing the required default export. Ensure a React component is exported as default.  
WARN | Route "./services/noteService.ts" is missing the required default export. Ensure a React component is exported as default.  
WARN | SafeAreaView has been deprecated and will be removed in a future release. Please use 'react-native-safe-area-context' instead. See https://github.com/th3rdwave/react-native-safe-area-context  
ERROR | [Error: Uncaught (in promise, id: 0) Error: Database connection failed: Unable to connect to Firestore]
```

Berdasarkan pesan error yang muncul, dapat diketahui bahwa sumber permasalahan berasal dari lapisan backend atau pada proses inisialisasi atau pemanggilan koneksi database. Pesan error yang jelas dan spesifik membantu pengembang untuk segera mengidentifikasi bahwa permasalahan bukan berasal dari logika aplikasi, melainkan dari koneksi database.

Proses debugging menggunakan devtools

Ketika aplikasi dijalankan, pesan error muncul di console beserta stack trace yang menunjukkan lokasi file dan baris kode tempat error terjadi. Melalui informasi tersebut, kita dapat mengetahui titik eksekusi kode yang menyebabkan error.