

Tugas

Task

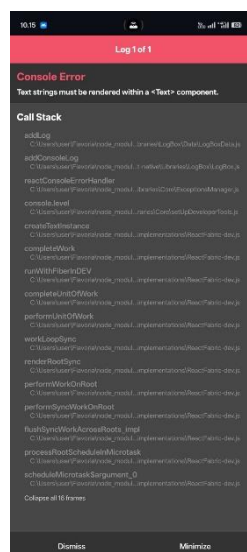
Nama / *Name* : **Salwa Aprilia Santi**
 NIM / *NIM* : **20230040141**
 Kelas / *Class* : **TI23F**
 Mata Kuliah / *Course Name* : **Pemrograman perangkat Mobile**
 Sesi Pertemuan/ *Session Number* : **XIV (Empat Belas)**
 Nama Dosen / *Lecturer* : **Nugraha, M.Kom**
 Jenis Tugas / *Type of task* : **Case Based / ~~Project Based~~**

I. Pendahuluan

Laporan ini disusun sebagai bagian dari evaluasi pada Sesi 14 yang membahas Optimasi dan Debugging aplikasi mobile berbasis React Native (Expo). Aplikasi yang dikembangkan, yaitu Flavoria, digunakan untuk menampilkan dan mengelola informasi resep makanan dari berbagai negara yang diperoleh melalui API eksternal. Kegiatan ini difokuskan pada pengujian kestabilan aplikasi, identifikasi error saat runtime, serta perbaikan kode untuk mencegah terjadinya crash pada perangkat Android.

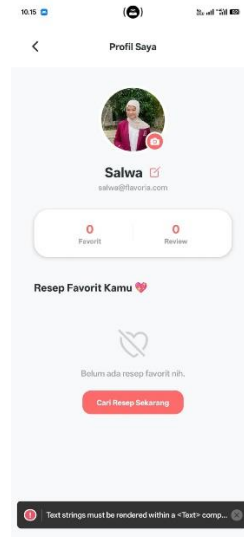
II. Dokumentasi Proses Debugging dan Error

Dalam proses pengujian aplikasi Flavoria, ditemukan error yang menyebabkan aplikasi berhenti saat dijalankan pada perangkat Android. Error ini muncul ketika aplikasi melakukan proses render pada komponen tertentu yang menggunakan logika kondisional.



Gambar 1. Tampilan Error Log pada Aplikasi

Error yang muncul adalah *"Text strings must be rendered within a <Text> component"*. Error ini terjadi karena aplikasi mencoba menampilkan nilai numerik 0 menggunakan operator logika && di luar komponen <Text>. Kondisi tersebut menyebabkan React Native tidak dapat merender tampilan dengan benar sehingga aplikasi mengalami crash.



Gambar 2. Tampilan Error pada Aplikasi

Selain tampilan error pada console log, aplikasi juga menampilkan layar error saat dijalankan. Tampilan ini menunjukkan bahwa aplikasi tidak dapat melanjutkan proses karena kesalahan pada logika render yang digunakan.

III. Perbaikan Penanganan Error

Setelah error berhasil diidentifikasi, langkah selanjutnya adalah melakukan perbaikan pada kode program. Perbaikan difokuskan pada bagian logika kondisional yang sebelumnya menggunakan operator logika &&. Operator tersebut diganti dengan ternary operator agar kondisi yang tidak terpenuhi menghasilkan nilai null, sehingga tidak menimbulkan error saat proses render.

```
1 {currentPage === "home" ? (  
2   <HomeScreen user={user} onSelectMeal={handleSelect} />  
3 ) : null}  
4  
5 {favorites.length === 0 ? (  
6   <View style={styles.emptyContainer}>  
7     <Text style={styles.emptyText}>Belum ada resep favorit.</Text>  
8   </View>  
9 ) : null}
```

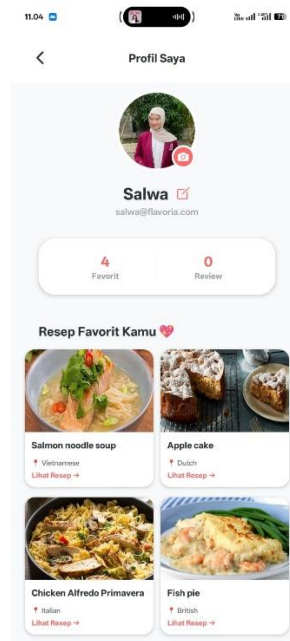
Gambar 3. Code Perbaikan Logika Kondisional

Selain perbaikan pada logika kondisional, dilakukan juga pengecekan terhadap data yang digunakan dalam aplikasi, baik data yang berasal dari API maupun aset gambar lokal.

Setiap data dipastikan memiliki nilai yang valid sebelum ditampilkan pada antarmuka aplikasi. Langkah ini dilakukan untuk mencegah terjadinya error saat aplikasi dijalankan.

IV. Hasil Implementasi Aplikasi

Setelah perbaikan kode dilakukan, aplikasi Flavoria dapat berjalan dengan lebih stabil. Proses navigasi antar halaman, seperti halaman Beranda, Detail Resep, dan Profil Pengguna, dapat digunakan tanpa mengalami kendala atau crash.



Gambar 4. Tampilan pada Aplikasi Setelah Diperbaiki

Selain itu, fitur favorit juga telah berfungsi dengan baik. Resep yang ditandai sebagai favorit dapat langsung ditampilkan pada halaman profil pengguna sesuai dengan data yang tersimpan. Tampilan aplikasi setelah perbaikan menunjukkan bahwa data dapat ditampilkan dengan benar dan aplikasi berjalan dengan stabil saat digunakan.

V. Kesimpulan

Berdasarkan hasil pengujian dan perbaikan yang telah dilakukan, dapat disimpulkan bahwa penggunaan ternary operator lebih aman dibandingkan operator logika && dalam menangani kondisi tertentu pada React Native, khususnya pada platform Android. Kesalahan pada logika render dapat menyebabkan aplikasi mengalami crash apabila tidak ditangani dengan tepat.

Dengan penerapan perbaikan tersebut, aplikasi Flavoria menjadi lebih stabil dan dapat digunakan sesuai dengan tujuan pengembangan pada laporan akhir tugas ini.