

PERANCANGAN APLIKASI ANTREAN MAKANAN BERBASIS CROSS-PLATFORM DENGAN FRAMEWORK FLUTTER (STUDI KASUS RM. PADANG BUARAN)

Rifqi Mufidianto¹, Jefry Sunupurwa Asri², Nizirwan Anwar³, Tri Ismardiko Widyawan⁴
Universitas Esa Unggul^{1,2,3,4}

E-mail: rmufidianto@student.esaunggul.ac.id¹, jefry.sunupurwa@esaunggul.ac.id²,
nizirwan.anwar@esaunggul.ac.id³, ismardiko@esaunggul.ac.id⁴

ABSTRAK

Antrean makanan di restoran sering menjadi masalah bagi pengunjung dan pemilik restoran, terlebih restoran tersebut sangatlah ramai, karena dapat menyebabkan ketidaknyamanan dan gangguan pada proses pemesanan makanan. Untuk mengatasi masalah ini, diperlukannya sebuah inovasi dalam teknologi untuk antrean makanan. Oleh karena itu, diusulkan sebuah solusi berupa aplikasi antrean makanan yang bertujuan untuk memperbaiki proses pendaftaran dan antrean dalam pemesanan makanan di restoran. Aplikasi ini memungkinkan pelanggan mendaftar dan memesan antrean tanpa harus datang langsung ke restoran, sehingga proses pendaftaran dan antrean dapat dilakukan secara cepat dan efisien. Selain itu juga pengunjung akan mendapatkan informasi terkait pesanan yang telah dipesan sebelumnya. Dengan perancangan aplikasi ini, harapannya penyedia pemesanan makanan dapat mengelola pemesanan dengan lebih mudah dan efisien. Aplikasi antrean ini diharapkan untuk dapat meningkatkan proses pemesanan makanan dan mengurangi waktu tunggu pengunjung serta membantu pemilik restoran meningkatkan layanan pelanggan. Penggunaan teknologi yang tepat diharapkan akan membuat proses mengantre makanan di restoran lebih menyenangkan dan lancar.

Kata kunci : *Antrean makanan, aplikasi mobile, flutter*

ABSTRACT

The food queue in restaurants often becomes a problem for both customers and restaurant owners, especially when the restaurant is crowded, as it can cause inconvenience and disruptions in the food ordering process. To address this issue, an innovation in queuing technology is needed. Therefore, a proposed solution comes in the form of a "Food Queueing Application" aimed at improving the registration and queuing process for food orders in restaurants. This application allows customers to register and queue for their orders without having to physically be present at the restaurant, thus making the registration and queuing process fast and efficient. Additionally, customers will receive information about their previously placed orders. With the design of this application, it is hoped that food service providers can manage orders more easily and efficiently. The food queueing application is expected to enhance the food ordering process, reduce waiting times for customers, and assist restaurant owners in improving customer service. Proper use of technology is expected to make the food queuing process in restaurants more enjoyable and smoother.

Keyword : *Food queue, mobile application, flutter.*

1. PENDAHULUAN

Rumah makan adalah tempat atau bangunan yang diselenggarakan secara komersial dan memberikan pelayanan yang baik dalam bentuk makanan atau minuman kepada semua tamu (Marsum, 2005). Rumah Makan Padang Buaran (RM. Padang Buaran) adalah salah satu dari sekian rumah makan yang enak, murah, dan berkualitas. Rumah makan padang buaran ini berlokasi di Jalan Raya Narogong kota Bekasi, tempatnya yang mudah dijangkau karena berada di pinggir jalan tersebut membuat siapa saja yang merasa sudah lapar dapat menuju ke tempat makan tersebut.

Metode pemesanan makanan pada rumah makan ini, sama seperti proses pemesanan makanan di tempat rumah makan lain. Dimulai dengan datangnya pelanggan yang ingin makan, yang selanjutnya pelayan akan menanyakan kepada pelanggan ingin memesan makanan & minuman apa yang ingin dipesan. Pelanggan akan memilih makanan yang ingin dipesan, kemudian pelayan akan mencatat makanan dan minuman apa saja yang dipesan oleh pelanggan tersebut. Setelah semua selesai dicatat, pelayan akan memberikan nomor antrean pesanan dari pelanggan tersebut. Ketika makanan tersebut sudah siap untuk diberikan ke pelanggan maka pelayan akan memanggil nomor antrean tersebut dan memberikannya ke pelanggan.

Ketika waktu jam makan telah tiba, yang terjadi pada rumah makan ini adalah pada proses pemesanan antrean makanannya yang masih kurang efektif, belum lagi jumlah rata-rata pelanggan ketika masuk waktu jam makan yang membuat antrean tersebut menjadi lebih banyak, sehingga dari antrean tersebut menjadi tidak teratur. Dan permasalahan lainnya adalah tidak adanya informasi dari antrean pesanan makanan tersebut yang membuat pelanggan akan bertanya-tanya

mengenai makanan tersebut, apakah sudah diproses ataukah belum.

Dengan banyaknya pelanggan yang sudah menggunakan smartphone, maka dari itu solusi yang ingin ditawarkan dari masalah tersebut adalah perancangan aplikasi *mobile* mengenai antrean pemesanan makanan. Sebuah aplikasi yang akan diberi nama "AntreeOrder" ini dapat mempermudah pelanggan untuk memesan makanan dan antrean dari rumah makan ini, karena pelanggan dapat memesan melalui online maupun di lokasi dengan ponsel yang dimiliki. Sehingga waktu yang dimiliki pelanggan tidak terbuang begitu saja, dan juga dapat terhindar dari masalah antrean.

2. LANDASAN TEORI

Aplikasi

Menurut Pramana (2012), Aplikasi merupakan suatu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas seperti sistem perniagaan, periklanan, atau semua proses yang hampir dilakukan manusia. Menurut Yuhefizar (2012), Aplikasi merupakan sebuah program yang dikembangkan untuk dapat memenuhi kebutuhan pengguna dalam menjalankan pekerjaan tertentu. Dari kedua pendapat tersebut, dapat disimpulkan bahwa aplikasi merupakan sebuah program perangkat lunak untuk dapat mempermudah pekerjaan manusia dalam melakukan tugas-tugasnya.

Antrean

Antrean adalah orang-orang atau barang dalam barisan yang sedang menunggu untuk dilayani (Heizer dan Barry Rander, 2009) Sistem antrean mencakup pelanggan yang datang ke suatu fasilitas layanan dengan laju tetap atau berbagai macam laju untuk mendapatkan layanan (Antono, 2010). Jika pelanggan dapat memasuki fasilitas layanan, pelanggan dapat langsung

dilayani; jika pelanggan harus menunggu untuk dilayani, pelanggan dapat berpartisipasi atau membentuk antrian, dan akan berada dalam antrian hingga pelanggan dapat dilayani.

Cross-Platform

Cross-platform digunakan dalam teknologi informasi untuk menggambarkan perangkat lunak yang dapat berjalan pada berbagai sistem operasi. Salah satu keuntungan dari penggunaan teknologi *cross-platform* adalah yaitu efektivitas biaya *cross-platform* lebih murah daripada membuat aplikasi khusus untuk berbagai *platform* (Lushena, 2016). Sehingga penggunaan *framework* flutter sangatlah mendukung *cross-platform* untuk pembuatan sebuah aplikasi yang nantinya akan diterapkan.

Flutter dan Dart

Dalam buku Pemrograman Android dengan Flutter (Raharjo, 2019), Flutter adalah Software Development Kit (SDK) yang dikembangkan oleh Google. Tujuan dari Flutter adalah untuk membuat proses pengembangan aplikasi mobile yang berjalan di Android dan iOS menjadi lebih mudah dan lebih cepat tanpa harus mempelajari pemrograman untuk kedua bahasa tersebut. Dart, bahasa pemrograman yang dikembangkan oleh Google, dirancang oleh Lars Bak dan Kasper Lund, Dalam buku Pemrograman Android dengan Flutter (Raharjo, 2019). Dart tampil untuk pertama kalinya pada 10 Oktober 2011. Proses penerjemahan ke bytecode, atau kode objek, tidak diperlukan sebelum Dart dapat dijalankan secara langsung oleh Dart Virtual Machine (VM).

BLoC

Business Logic Component (BLoC) *Pattern* adalah pola desain yang membantu memisahkan *user interface* dengan *business logic* dan memungkinkan *developer* untuk membagi komponen pada proyek menjadi komponen

presentational, BLoC, dan *backend* (Dila Anastasia & Papatungan, 2022). Tujuan dari *package* ini adalah untuk memudahkan pemisahan antara tampilan (*presentation*) dan logika bisnis, sehingga memfasilitasi pengujian dan penggunaan ulang (reusabilitas).

BLoC juga merupakan salah satu *state management* yang tersedia pada *framework* flutter. BLoC dipilih karena dapat memisahkan antara *event* yang berguna untuk men-trigger perubahan *state*, *state* yang berguna untuk memberikan data dan BLoC yang berguna untuk mengelola *event* dan *state*.

Headless CMS

Headless CMS merupakan sistem manajemen konten yang dapat berfungsi sebagai pengelolaan konten dan memiliki fungsionalitas hanya pada bagian backend. Ini memungkinkan konten dibuat dan didistribusikan ke berbagai aplikasi melalui GraphQL maupun API (Yuliyana, 2023). Dengan demikian, pada aplikasi ini nantinya akan menggunakan Strapi sebagai *Headless CMS*.

API

Application Programming Interface atau biasa juga dikenal sebagai API, adalah sekumpulan protokol dan standar yang digunakan untuk mengintegrasikan aplikasi dengan sistem sehingga pengguna dapat mengakses dan menggunakan fungsi *software* aplikasi (Fitri, 2022). Salah satu manfaat API adalah memungkinkan aplikasi berinteraksi satu sama lain dengan sistem. Oleh karena itu, API sangat penting untuk mempermudah akses dan pengembangan aplikasi di internet.

Black-box Testing

Pengujian kotak hitam adalah pengujian perangkat lunak yang dilakukan hanya berdasarkan spesifikasi fungsional tanpa menguji desain atau kode program (Sukamto dan Shalahuddin, 2014). Tujuan

pengujian ini adalah untuk mengetahui apakah fungsi, masukan, dan keluaran perangkat lunak memenuhi spesifikasi.

3. HASIL DAN PEMBAHASAN

Lokasi Penelitian

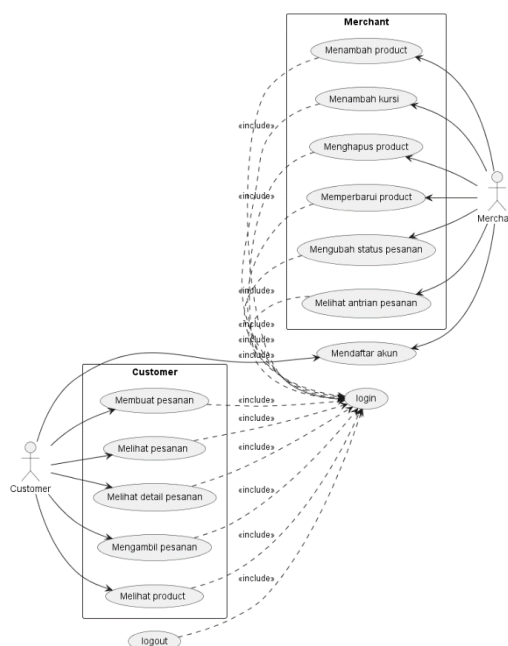
Penelitian ini dilakukan di Rumah Makan Padang Buaran (R.M Padang Buaran) yang berlokasi di Jl. Raya Narogong RT/RW. 002/004 Kel Bojong Menteng, Kec Rawa Lumbu Kota Bekasi.

Analisa Kebutuhan

Pada aplikasi antrian makanan ini, penulis mengamati proses pemesanan makanan yang terjadi di rumah makan padang buaran, dari hal tersebut, terdapat 2 tipe pengguna yang terjadi, yaitu merchant (pedagang) dan customer (pelanggan).

Use Case Diagram

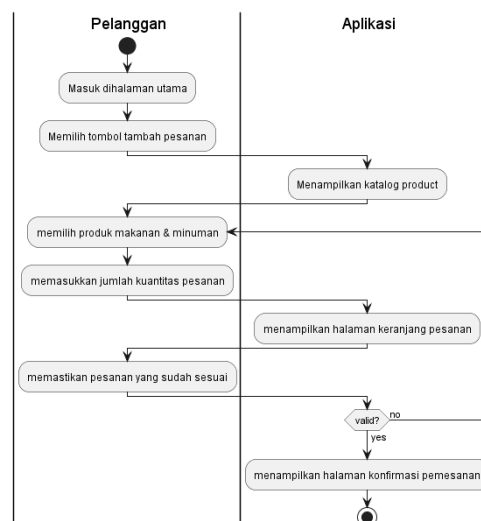
Use case merupakan proses kegiatan dan proses bisnis yang dilakukan oleh aktor yang saling berinteraksi antara aktor dengan use case. Pada tahap ini merupakan sebuah proses perancangan untuk dapat menentukan siapa saja aktor yang berperan serta fungsi yang dapat dilakukan pada aplikasi ini. Pada aplikasi terdapat 2 aktor yaitu pelanggan (*customer*) dan pedagang (*merchant*).



Gambar 2. Use case diagram

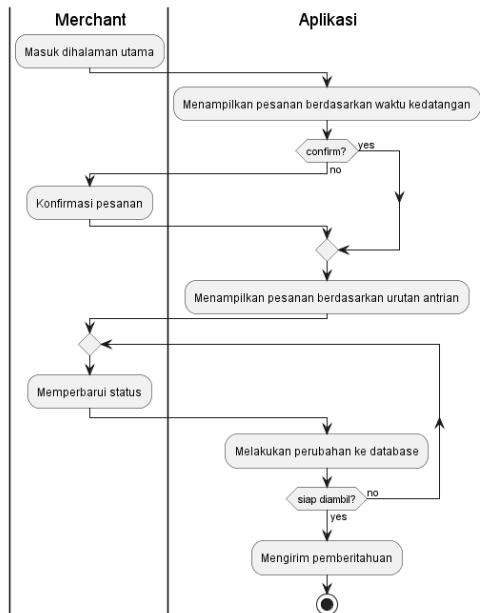
Activity Diagram

Berikut merupakan perancangan dari *activity diagram* pada aplikasi ini.



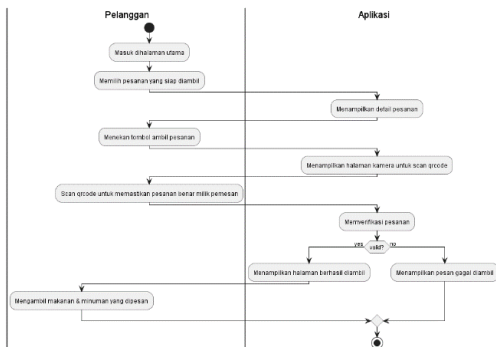
Gambar 1. Activity diagram pemesanan makanan

Gambar dibawah ini merupakan *activity* diagram untuk sebuah proses antrean pemesanan makanan yang terjadi pada sisi pengguna pedagang (*merchant*).



Gambar 3. *Activity* diagram antrean pemesanan makanan

Gambar berikut merupakan *activity* diagram untuk pengambilan sebuah pesanan yang telah dipesan sebelumnya oleh pelanggan.

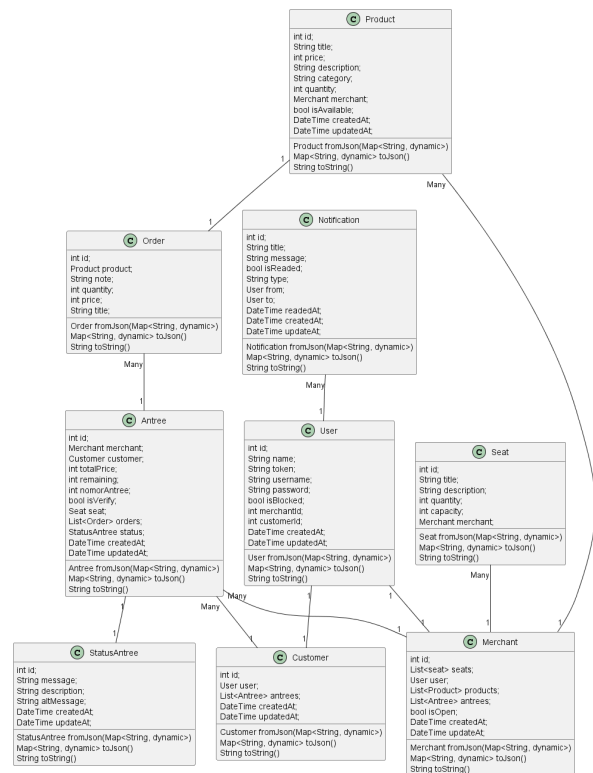


Gambar 4. *Activity* diagram pengambilan pesanan makanan

Class Diagram

Fungsi dari tahap ini adalah menggambarkan struktur dengan

menunjukkan bentuk kelas (*class*), atributnya, metode, dan hubungan antar objek yang terlibat dalam aplikasi tersebut. Terdapat 10 *class* diagram dari aplikasi ini yaitu *order*, *seat*, *merchant*, *customer*, *user*, *notification*, *statusantree*, *antree*, *role*, dan *product*. Dapat dilihat pada gambar yang ada di bawah ini.



Gambar 5. *Class* diagram

Pengelolaan Data

Penggunaan aplikasi strapi ini digunakan untuk pengelolaan data yang dibutuhkan untuk aplikasi ini, agar aplikasi dapat menyimpan dan menerima data-data. *Class* diagram sangat dibutuhkan untuk dapat membuat entitas yang berhubungan antar objek yang terlibat pada aplikasi. Pada strapi ini juga hanya berfokus pada sisi pengelolaan manajemen data saja, sehingga untuk proses logika tetap akan dilakukan dalam aplikasi ini.

Flutter Package

Salah satu untuk memudahkan pada proses pengkodean ini, digunakannya flutter *package* untuk dapat memudahkan proses pengembangan aplikasi ini. Terdapat 2 tipe flutter *package* yaitu flutter *package* dan flutter *package* dev. Flutter *package* adalah sebuah kumpulan kode-kode yang berisi fungsi khusus maupun kelas, yang telah dibuat sebelumnya dan dapat digunakan kembali untuk dapat mempercepat proses pengembangan aplikasi. Berikut beberapa flutter *package* yang digunakan pada aplikasi ini:

```
cupertino_icons: ^1.0.2
flutter_form_builder: ^9.0.0
shared_preferences: ^2.1.0
flutter_bloc: ^8.1.2
equatable: ^2.0.5
logger: ^2.0.0
dio: ^5.0.2
retrofit: ^4.0.1
json_annotation: ^4.8.0
form_builder_validators: ^9.0.0
get_it: ^7.2.0
bloc: ^8.1.1
injectable: ^2.1.2
infinite_scroll_pagination: ^3.2.0
collection: ^1.18.0
intl: ^0.18.0
drift: ^2.7.0
path: ^1.8.2
path_provider: ^2.0.14
sqlite3_flutter_libs: ^0.5.13
freezed_annotation: ^2.2.0
envied: ^0.3.0+3
ai_barcode: ^3.2.4
permission_handler: ^10.2.0
image_picker: ^1.0.1
cached_network_image: ^3.2.3
firebase_messaging: ^14.6.2
firebase_core: ^2.13.1
firebase_analytics: ^10.4.2
flutter_notification_channel: ^2.0.0
flutter_local_notifications: ^15.1.0+1
webview_flutter: ^4.2.2
uni_links: ^0.5.1
app_links: ^3.4.3
```

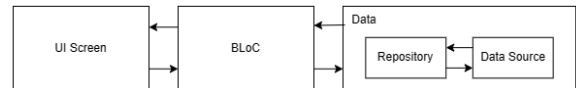
Gambar 6. Flutter *package*

Pola BLoC

Dikarenakan aplikasi ini menggunakan framework Flutter, maka tidak terlepas dengan menggunakan pola ini, yang merupakan salah satu cara untuk manajemen state, supaya memudahkan

aplikasi untuk bisa memisahkan antara UI layar, bisnis logika, dan data.

Dari gambar di bawah ini menunjukkan bagaimana data mengalir dari lapisan data menuju lapisan UI begitu juga sebaliknya.



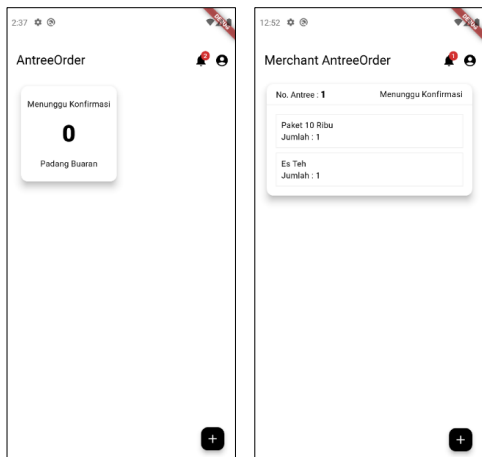
Gambar 7. Pola BLoC

- Terdapat sebuah layar yaitu detail antree, untuk menampilkan data detail dari sebuah antrean, maka dari UI layar tersebut meminta data dengan memberikan sebuah event yang dibutuhkan oleh bloc. Layar detail antree ini cukup menunggu perubahan state yang diberikan oleh DetailAntreeBloc.
- Sebuah event diterima oleh bloc, dilanjutkan untuk meminta data yang telah diolah oleh repository. Ketika repository telah memberikan data yang sudah diolah maka bloc memberikan data yang sudah diolah dirubah menjadi state untuk nantinya dapat diterima oleh DetailAntreeScreen.
- AntreeRepository menerima permintaan dari bloc, maka dari itu repository meminta pihak API Client untuk nanti data mentah tersebut diolah dan diberikan ke bloc.
- API Client meminta permintaan data ke layanan API sesuai dengan apa yang diminta oleh AntreeRepository. Jika layanan API memberikan sebuah jawaban berupa data yang berhasil maupun

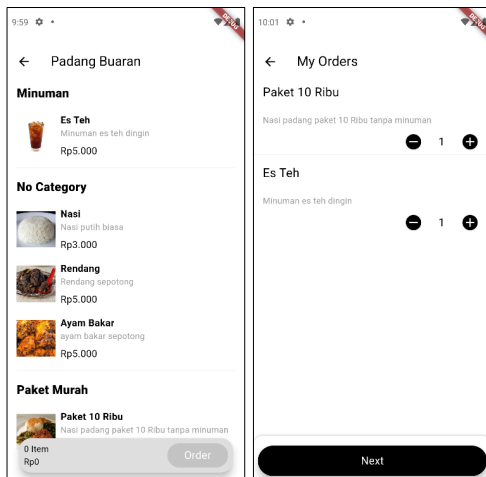
gagal, dilanjutkan kembali data mentah tersebut ke repository.

Implementasi

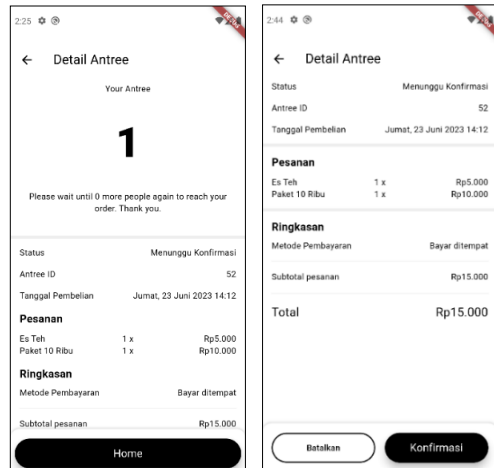
Pada tahap ini merupakan, hasil yang didapat dari perancangan dan pengkodean pada pembahasan sebelumnya. Dan berikut hasil dari implementasi tersebut.



Gambar 8. Tampilan halaman utama



Gambar 9. Tampilan katalog produk dan keranjang



Gambar 10. Tampilan detail pesanan

Pengujian

Pengujian pada penelitian ini menggunakan *Black-box Testing*. Pada penelitian ini objek yang akan diuji adalah fungsionalitas pada aplikasi ini saja. Berikut merupakan tabel pengujian dari aplikasi ini.

Tabel 1. *Black-box testing customer*

No	Pengujian	Hasil	Validasi
1	Login	Pelanggan bisa masuk menggunakan akun	Berhasil
2	Register	Pelanggan bisa mendaftar akun	Berhasil
3	Halaman utama	Pelanggan dapat melihat antrean dan menambah pesanan	Berhasil
4	Katalog Produk	Pelanggan dapat melihat produk	Berhasil
5	Keranjang Produk	Pelanggan dapat menambahkan item pada keranjang	Berhasil
6	Konfirmasi Pesanan	Pelanggan mengkonfirmasi pesanan	Berhasil
7	Detail Pesanan	Pelanggan dapat melihat detail pesanan	Berhasil
8	Verifikasi Pesanan	Pelanggan dapat memverifikasi pesanan	Berhasil
9	Notifikasi	Pelanggan dapat melihat notifikasi mengenai Status pesanan	Berhasil
10	Logout	Pelanggan dapat keluar akun	Berhasil

Tabel 2. *Black-box testing merchant*

No	Pengujian	Hasil	Validasi
1	Login	Pedagang bisa masuk menggunakan akun	Berhasil
2	Register	Pedagang bisa mendaftar akun	Berhasil
3	Halaman utama	Pedagang dapat melihat antrean dan memperbarui status	Berhasil
4	Manajemen Produk	Pedagang dapat memanajemen produk	Berhasil
5	Pengaturan Toko	Pedagang dapat memperbarui pengaturan toko	Berhasil
6	Manajemen Kursi	Pedagang dapat memanajemen kursi	Berhasil
7	Detail Pesanan	Pedagang dapat melihat detail dan memperbarui status pesanan	Berhasil
8	Notifikasi	Pedagang dapat melihat notifikasi mengenai Status pesanan	Berhasil
9	Logout	Pedagang dapat keluar akun	Berhasil

4. KESIMPULAN

Berdasarkan hasil dan pembahasan dengan judul “Perancangan Aplikasi Antrean Makanan Berbasis Cross-Platform Dengan Framework Flutter (Studi Kasus Rumah Makan Padang Buaran)”, menghasilkan beberapa kesimpulan, yaitu:

- a. Aplikasi ini memungkinkan pelanggan untuk melakukan pemesanan tanpa perlu menunggu dalam antrean fisik karena dapat memesannya secara online melalui aplikasi. Selain itu juga, dengan adanya aplikasi ini, dapat membantu mengurangi risiko kecurangan dalam antrean karena pedagang dapat melihat antrean melalui aplikasi.

- b. Pengkodean aplikasi menggunakan *framework* flutter dapat membantu dan memudahkan karena cukup sekali pengkodean untuk bisa dapat berjalan diberbagai sistem operasi.
- c. Aplikasi dapat menerima 2 tipe pengguna yaitu pelanggan (*customer*) dan pedagang (*merchant*) dalam satu aplikasi.
- d. Setelah dilakukannya pengujian, hasil dari pengujian menunjukkan bahwa aplikasi sudah sesuai dan dapat berjalan dengan baik.

DAFTAR PUSTAKA

- Anastasia, F. D., & Papatungan, I. V. (2022). Implementasi Bloc Pattern Pada Pengembangan Frontend Fitur Top Up Linkaja Aplikasi M-Banking Agen46 Dengan Teknologi Flutter (Studi Kasus: Pt. Bank Negara Indonesia Tbk). *Jurnal Sains, Nalar, dan Aplikasi Teknologi Informasi*, 2(1).
- Antono, S. D (2010). Penerapan model simulasi antrian di bagian pengobatan puskesmas prambon kabupaten jeruk nganjuk. *Jurnal penelitian kesehatan flores*.
- Heizer, Jay; Render, Barry; Indra Almahdy; Dwianoegrahwati Setyoningsih (2006). *manajemen operasi / Jay Heizer, Barry Render; penerjemah, Dwianoegrahwati Setyoningsih, Indra Almahdy. Jakarta: Salemba Empat.*
- Lushena, R. A. M, 2016, *Pembangunan Aplikasi Ekspor Barang Berbasis Cross-Platform Mobile Application Studi Kasus*

- Direktorat Jenderal Bea dan Cukai Bandung, Bandung: Jurnal Ilmiah Komputer dan Informatika.
- Marsum, W.A. 2005. Restoran dan Segala Permasalahannya. Yogyakarta : Andi Offset. Media Komputindo.
- Pramana, Henky W. 2012. Aplikasi Inventory Berbasis Access 2003. Jakarta:PT. Elex
- Raharjo, B (2019). Pemrograman Android dengan Flutter. Bandung: Informatika Bandung.
- Septianl, A. N. (2021). IMPLEMENTASI FRAMEWORK FLUTTER UNTUK PENGADUAN MAHASISWA UNIVERSITAS XYZ IMPLEMENTATION OF THE FLUTTER FRAMEWORK FOR COMPLAINTS XYZ UNIVERSITY STUDENTS.
- Shalahuddin, M (2014). Rekayasa Perangkat Lunak: Terstruktur dan Berorientasi Objek. Informatika Bandung.
- W., Fitri. (2022). Apa Itu Restful Api? Pengertian, Cara Kerja, Dan Manfaatnya. Didapat dari <https://www.hostinger.co.id/tutorial/apa-itu-restful-api>. Diakses pada 19 Maret 2023.
- W., Yuliyana. (2023). APA ITU headless CMS? Pengertian, Kelebihan, Dan Fungsinya. Hostinger Tutorial. <https://www.hostinger.co.id/tutorial/headless-cms>. Diakses pada 19 Maret 2023.
- Yuhefizar; Mooduto, HA; Rahmat Hidayat (2009). Cara mudah membangun website interaktif menggunakan content management system joomla (CMS0 / Yuhefizar, HA Mooduto, Rahmat Hidayat. Jakarta: Elex Media Komputindo.