



SISTEM PEMESANAN DAN PENCARIAN *FOOD COURT* BERDASARKAN LOKASI PENGGUNA BERBASIS ANDROID

Rabbali Afdhal Yasyfiya¹, Sutarman²

^{1,2}Program Studi Informatika, Universitas Teknologi Yogyakarta, Indonesia

Abstrak: Peningkatan urbanisasi yang pesat telah mengubah pola konsumsi masyarakat dalam mencari informasi mengenai tempat makan, khususnya di sektor *Food Court*. Berdasarkan data BPS, tingkat urbanisasi di Indonesia diperkirakan meningkat dari 56,7% pada tahun 2020 menjadi 65,6% pada tahun 2025, bersamaan dengan pertumbuhan industri makanan dan minuman sebesar 8,67% pada 2018. Namun, *Food Court* di Indonesia, yang mencakup 71,65% dari total 11.223 bisnis kuliner, umumnya hanya memberikan informasi dasar tanpa daftar *Tenant* dan menu yang terstruktur. Pemesanan secara *manual* masih umum dilakukan, yang memperlambat pelayanan dan berpotensi menimbulkan kesalahan. Penelitian ini bertujuan untuk mengembangkan aplikasi berbasis *framework* Flutter dengan penggunaan *library* Geolocator dan *database* Firebase untuk memudahkan pemesanan dan pencarian *Food Court*. Pendekatan metode *Waterfall* digunakan dengan pengumpulan data melalui observasi langsung di Sleman, Yogyakarta, serta wawancara dengan *Manager* dan *Tenant*. Pengujian dilakukan pada 23 pengguna, terdiri dari 10 *Customer*, 3 *Manager*, dan 10 *Tenant*, menggunakan pendekatan *Black Box* dan kuesioner untuk menilai kebergunaan fitur aplikasi. Hasil penelitian menunjukkan bahwa 95% pengguna merasa aplikasi ini bermanfaat. Nilai rata-rata pemahaman pengguna meningkat dari 64,7% pada *pre-test*, menjadi 92,4% pada *post-test* setelah penggunaan aplikasi selama satu minggu. Aplikasi ini efektif dalam menyediakan informasi dan pemesanan yang lebih terstruktur dan diharapkan dapat diterapkan lebih luas dengan fitur tambahan seperti pembayaran *digital* melalui *scan QR* khususnya *e-wallet*.

Kata kunci: *food court*, Geolocator, Google Maps, urbanisasi, *Waterfall*

I. PENDAHULUAN

Fenomena urbanisasi telah mengubah pola konsumsi masyarakat, khususnya dalam mengakses informasi tempat makan (Rijal & Tahir, 2022). Data Badan Pusat Statistik (BPS) menunjukkan peningkatan signifikan tingkat urbanisasi di Indonesia dari 56,7% pada tahun 2020 dan diproyeksikan mencapai 65,6% pada tahun 2025. Pertumbuhan ini

berbanding lurus dengan perkembangan industri makanan dan minuman yang mencapai 8,67% pada triwulan II tahun 2018, melampaui pertumbuhan ekonomi nasional sebesar 5,27% (Syahbani, 2020).

Merespons pertumbuhan ini, data BPS (Juni 2022) mencatat dari 11.223 usaha kuliner di Indonesia, sebanyak 71,65% di antaranya merupakan usaha *Food Court*. Namun, tantangan utama dalam industri *Food Court* adalah minimnya integrasi teknologi dalam sistem pencarian dan informasi.

Google Maps saat ini sebagai *platform* yang umum digunakan, hanya menyediakan

¹rabbali.afdhal06@gmail.com, ²sutarman@uty.ac.id

Diterima: 16 Oktober 2024

Direvisi: 6 November 2024

Disetujui: 9 November 2024

DOI: 10.23969/infomatek.v26i2.19139

informasi dasar seperti lokasi, ulasan, dan *rating*, tanpa menyertakan detail penting seperti daftar *Tenant* dan menu yang terorganisir. Disamping itu, menurut (Jamaludin dkk., 2023) proses pemesanan di beberapa *Food Court* masih mencatat tulisan tangan secara *manual*. Hal ini dapat memperlambat proses pelayanan, dan kemungkinan menimbulkan kesalahan pemesanan yang dapat mempengaruhi kepuasan konsumen.

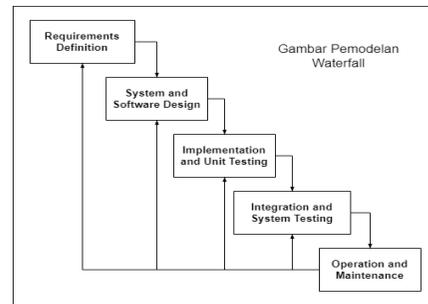
Menjawab tantangan tersebut, penelitian ini menawarkan keunggulan dalam bentuk aplikasi yang mengintegrasikan pemesanan *digital* untuk *Tenant* secara cepat dan efisien. Selain itu, *Manager* dapat memantau pertumbuhan keuangan dan penjualan *Tenant* secara *real-time*, dan konsumen dapat menemukan menu antar *Tenant* dalam satu lokasi *Food Court*, sehingga mengurangi waktu pencarian.

Sistem ini dibangun dengan mengintegrasikan *library Geolocator* dari *framework Flutter* (Ismail & Hadi Zakaria, 2023), dan *database Firebase*. Layanan ini terhubung dengan Google Maps untuk memudahkan pengguna dalam melakukan navigasi dan menemukan tempat secara efisien (Jason dkk., 2020). Sekaligus menawarkan beragam pilihan makanan dan minuman dalam satu lokasi yang dikelola oleh *Manager* (Dhanny, 2019)

II. METODOLOGI

2.1. Metode Pengembangan

Metode *Waterfall* merupakan suatu proses dalam pengembangan perangkat lunak, yang seperti air terjun dengan melewati *fase-fase* atau tahapan, biasanya disebut dengan model *sequential linear* atau *classic cycle*. Berikut metode pengembangan yang digunakan dalam penelitian ini adalah metode *Waterfall* menurut (Pressman, 2008 dalam Wijaya & Astuti, 2019), seperti terlihat pada Gambar 1.



Gambar 1. Siklus Pengembangan Metode *Waterfall* (Pressman, 2008 dalam Wijaya & Astuti, 2019)

1. *Requirement Definition*
Proses pencarian kebutuhan diintensifkan dan difokuskan pada *software*. Untuk mengetahui sifat dari program yang akan dibuat, maka para *software engineer* harus mengerti tentang domain informasi dari *software*, misalnya fungsi yang dibutuhkan, *user interface*. Dari 2 aktivitas tersebut (pencarian kebutuhan sistem dan *software*) harus didokumentasikan dan ditunjukkan kepada pelanggan.
2. *System and Software Design*
Proses ini digunakan untuk mengubah kebutuhan kebutuhan di atas menjadi representasi ke dalam bentuk "*blueprint*" *software* sebelum *coding* dimulai. Desain harus dapat mengimplementasikan kebutuhan yang telah disebutkan pada tahap sebelumnya. Seperti 2 aktivitas sebelumnya, maka proses ini juga harus didokumentasikan sebagai konfigurasi dari *software*.
3. *Implementation and Unit Testing*
Untuk dapat dimengerti oleh mesin, dalam hal ini adalah komputer, maka desain tadi harus diubah bentuknya menjadi bentuk yang dapat dimengerti oleh mesin, yaitu ke dalam bahasa pemrograman melalui proses *coding*. Tahap ini merupakan implementasi dari

tahap *design* yang secara teknis nantinya dikerjakan oleh *programmer*.

4. *Integration and System Testing*
 Sesuatu yang dibuat harus diujicobakan. Demikian pula dengan *software*. Semua fungsi-fungsi *software* harus diuji coba, agar *software* bebas dari *error*, dan hasilnya harus benar-benar sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya.
5. *Operation and Maintenance*
 Pemeliharaan suatu *software* dapat diperlukan, termasuk didalamnya berupa pengembangan, karena *software* yang dibuat tidak selamanya hanya seperti itu. Ketika dijalankan mungkin saja masih ada *error* kecil yang tidak ditemukan sebelumnya, atau ada penambahan fitur-fitur yang belum ada pada *software* tersebut. Pengembangan diperlukan ketika adanya perubahan dari eksternal seperti ketika ada pergantian sistem operasi, atau perangkat lainnya (Wijaya & Astuti, 2019).

2.2. Pengumpulan Data

Pengumpulan data menjadi sebuah metode yang dapat digunakan untuk memberikan dukungan dalam penyelesaian penelitian. Beberapa metode yang digunakan dengan melakukan sebagai berikut :

1. Observasi (*Observation*)
 Metode ini dilakukan peneliti dengan mengamati langsung dan mencari sejumlah *Food Court* yang ada di daerah kabupaten Sleman, Yogyakarta pada bulan Desember 2023. Dibawah ini merupakan hasil dari pengamatan langsung yang dilakukan oleh peneliti :
 - a. Sistem aplikasi yang berjalan saat ini tidak menampilkan *Tenant* dan menu di beberapa *Food Court*.
 - b. Sistem pemesanan pada *Tenant* di beberapa *Food Court* masih secara

manual dengan mencatat tulisan tangan.

2. Wawancara (*Interview*)
 Pada wawancara ini dilakukan kepada Manager pemilik *Food Court* beserta *Tenant* yang ada di daerah kabupaten Sleman, Yogyakarta. Beberapa masukan dapat diterima kedalam sistem yang akan dirancang berupa pemesanan menu, laporan penjualan dan pengumpulan data mengenai informasi *Food Court* secara umum seperti nama, alamat lokasi, kontak dan informasi *Tenant* seperti menu yang disajikan.

III. HASIL DAN PEMBAHASAN

Sistem Pembayaran dan Pencarian *Food Court* Berdasarkan Lokasi Pengguna dibuat dengan metode pengembangan *waterfall*, sebagai berikut :

3.1. Requirement Definition

Peneliti melakukan observasi di daerah kabupaten Sleman, Yogyakarta. Menghasilkan 10 data *Food Court* dengan masing-masing data 5 *Tenant* beserta 5-10 menu. Selain melakukan observasi, peneliti juga melakukan beberapa bagian analisis kebutuhan pada sistem, sebagai berikut:

1. Kebutuhan *software*, untuk membangun sistem. Berikut kebutuhan *software* dapat dilihat pada Tabel 1.

Tabel 1. Kebutuhan *Software*

No	Software	Kebutuhan
1	Visual Studio Code	IDE dan <i>Editor Coding</i>
2	<i>Flutter</i>	<i>Framework</i> aplikasi <i>mobile</i>
3	<i>Database</i> Firebase	Pengelolaan <i>Database</i> NoSQL

2. Kebutuhan *hardware*, merupakan proses analisis yang lebih menekankan kepada aspek pemanfaatan perangkat keras untuk membangun sistem.

Berikut kebutuhan *hardware* dapat dilihat pada Tabel 2.

Tabel 2. Kebutuhan *Hardware*

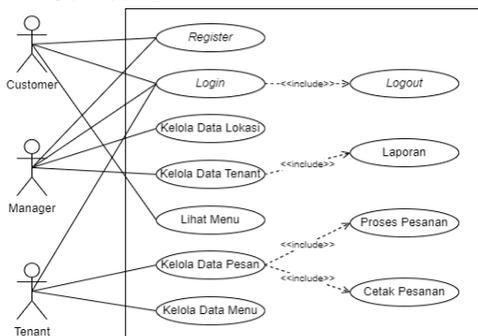
No	Hardware	Spesifikasi
1	Laptop	Asus X415EP
2	Processor	Intel Core i7
3	Harddisk	SSD 256 GB
4	Android	Redmi 10
5	Printer Thermal	EP5859

3.2. System and Software Design

Peneliti merancang desain sistem dengan pemodelan UML (*Unified Modeling Language*). Berikut adalah hasil desain sistem pemodelan.

1. *Use Case Diagram*

Use case diagram mendefinisikan fitur yang terdapat pada aplikasi yang menunjukkan interaksi antara pengguna dan sistem dalam menggunakan fitur yang ada (Widiastuti & Tamrin, 2020). *Diagram* ini terdiri dari interaksi faktor internal dan external. Pada agen internal dan external ini disebut *actor* (Andriyanto, 2022). Berikut *Use Case Diagram* pada sistem pembayaran dan pencarian *food court* berbasis android, dapat dilihat pada Gambar 2.

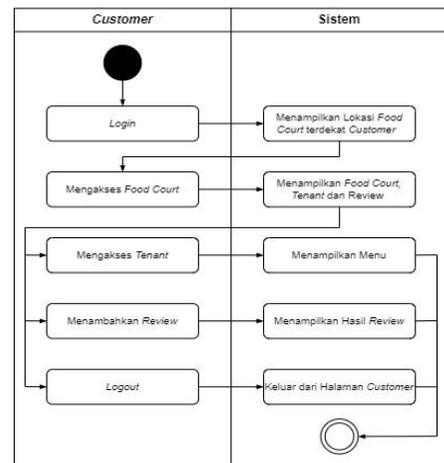


Gambar 2. *Use Case Diagram*

Use Case Diagram ini memiliki 3 aktor yang berperan dalam menjalankan hak akses dan fungsi sistem diantaranya (*Customer, Tenant* dan *Manager*). *Customer* dapat melakukan *Register, Login,* dan *Lihat Menu.* *Manager* memiliki akses untuk mengelola *Data Lokasi, Data Tenant,* dan dapat melihat *Laporan.* Sedangkan *Tenant* dapat mengelola *Data Menu* dan *Data Pesan,* yang terhubung dengan fungsi *Proses Pesanan* dan *Cetak Pesanan* (ditunjukkan dengan relasi `<<include>>`). Semua aktor dapat melakukan *Login* dan terhubung dengan melakukan fungsi *Logout.*

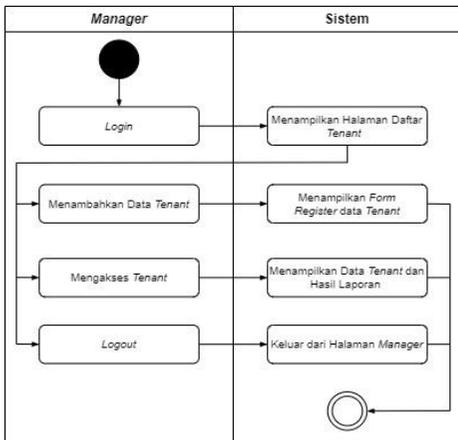
2. *Activity Diagram*

Pemodelan *Activity Diagram* atau sering disebut dengan diagram aktivitas yang digunakan dalam perancangan perangkat lunak. Karena dapat membantu dalam memahami proses sistem secara keseluruhan (Andriyanto, 2022). Pada tahap ini menjelaskan dan menggambarkan aktivitas yang dilakukan oleh 3 aktor dalam penggunaan sistem. *Activity diagram Customer* dapat dilihat pada Gambar 3.



Gambar 3. *Activity Diagram Customer*

Activity diagram Customer ini, menggambarkan alur proses *Customer* dalam sistem. Setelah *login*, *Customer* dapat melihat lokasi *Food Court* terdekat dan mengakses informasi detailnya termasuk daftar *Tenant* dan menu yang tersedia. *Customer* juga memiliki opsi untuk memberikan dan melihat *review*.



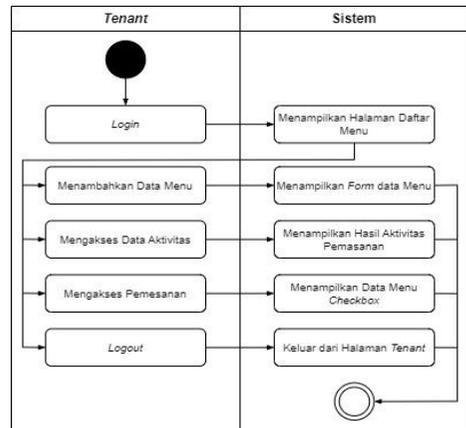
Gambar 4. Activity Diagram Manager

Pada *Activity diagram Manager* pada Gambar 4 ini, menggambarkan alur proses *Manager* dalam sistem. Setelah *login*, *Manager* dapat mengelola data *Tenant* mulai dari menambah *Tenant* baru melalui *form* registrasi hingga mengakses informasi dan laporan penjualan *Tenant* yang sudah terdaftar.

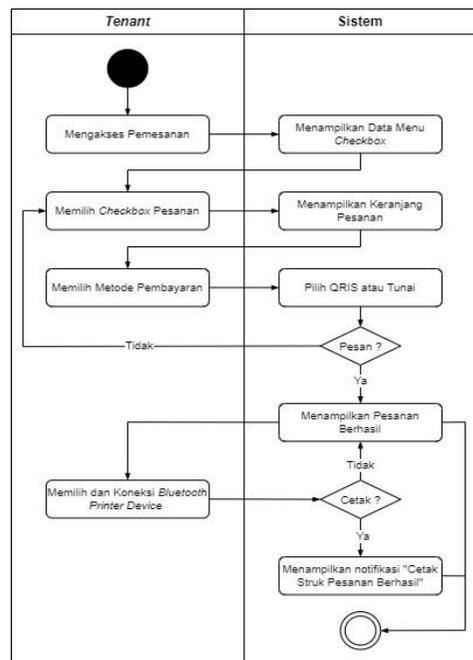
Activity diagram Tenant pada Gambar 5 ini, menggambarkan alur proses *Tenant* dalam sistem. Setelah *login*, *Tenant* dapat mengelola menu dengan menambahkan menu baru serta memantau dan mengakses data pemesanan dengan memilih pesanan yang diinginkan *Customer*.

Activity diagram Pemesanan pada Gambar 6 menggambarkan alur dalam

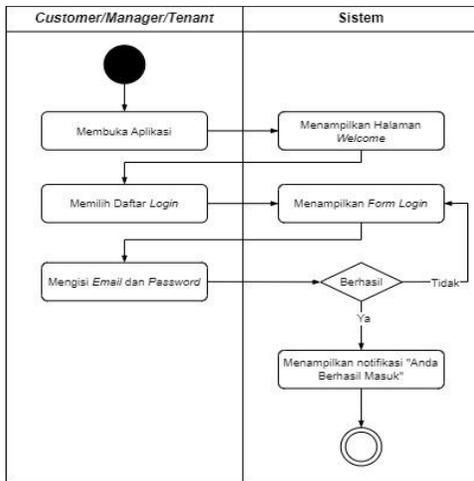
proses pemesanan dari *Tenant* dalam sistem. Dimulai dari *Tenant* mengakses dan memilih pesanan yang akan di-*checkout* dari *Customer* pesan, dilanjutkan dengan memilih metode pembayaran (QRIS / Tunai). Setelah verifikasi pesanan berhasil, *Tenant* dapat mencetak struk pesanan menggunakan *Bluetooth Printer*.



Gambar 5. Activity Diagram Tenant

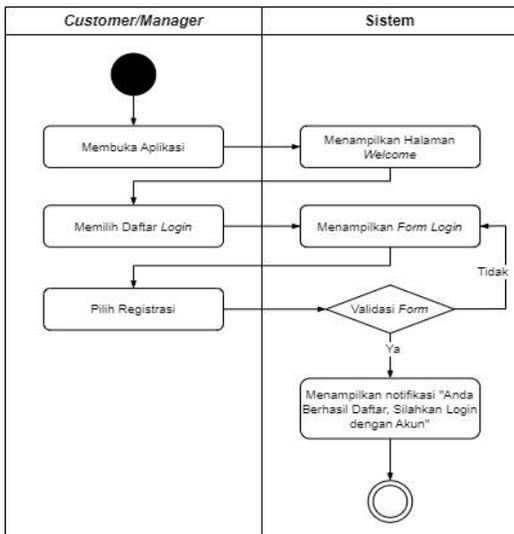


Gambar 6. Activity Diagram Pemesanan



Gambar 7. Activity Diagram Login

Activity diagram Login ini, menggambarkan proses login untuk semua pengguna (Customer, Manager, dan Tenant) dalam sistem. Prosesnya dimulai dari membuka aplikasi dan memilih Daftar Login, dilanjutkan dengan mengisi Email dan Password. Sistem akan memverifikasi data - jika berhasil, pengguna akan masuk ke sistem, jika gagal pengguna harus login kembali.

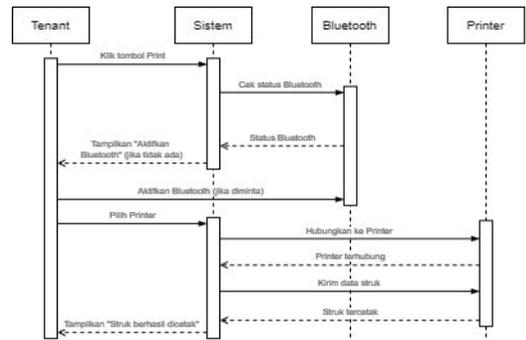


Gambar 8. Activity Diagram Register / Daftar

Activity diagram Register / Daftar ini menggambarkan proses registrasi untuk Customer dan Manager dalam sistem. Prosesnya dimulai dari membuka aplikasi dan memilih Daftar Login, dilanjutkan dengan mengisi form registrasi. Sistem akan memvalidasi data yang dimasukkan - jika berhasil, pengguna akan mendapat notifikasi sukses dan dapat login, jika gagal pengguna harus mengisi ulang form registrasi.

3. Sequence Diagram

Sequence Diagram menggambarkan kelakuan/perilaku objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek (Marbun & Sinaga, 2019).



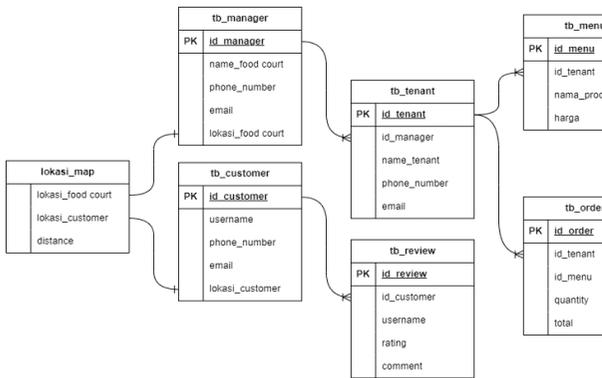
Gambar 8. Sequence Diagram Print Struk

Pada Sequence diagram ini menggambarkan alur pencetakan struk pesanan oleh Tenant dalam sistem. Prosesnya dimulai dari Tenant mengklik tombol Print, dilanjutkan dengan pengecekan dan aktivasi Bluetooth jika belum aktif. Setelah Tenant memilih printer dan terhubung, sistem akan mengirim data struk untuk dicetak. Proses selesai ketika sistem

menampilkan notifikasi bahwa struk berhasil dicetak.

4. *Class Diagram*

Class Diagram atau Diagram kelas diagram adalah menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem (Supiana, 2022).



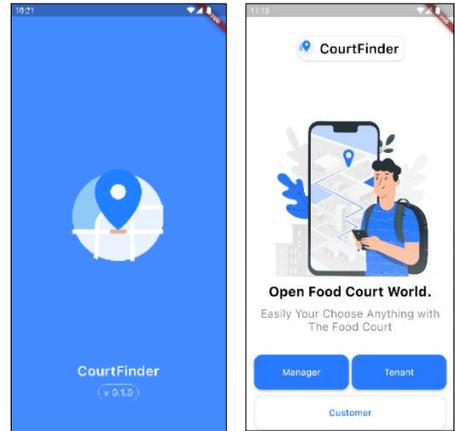
Gambar 9. *Class Diagram*

Pada *Class Diagram* menampilkan berupa tabel-tabel *database* yang berelasi dengan tabel lain. Seperti pada Gambar 5, tabel *Customer* dan tabel *review* memiliki relasi *one-to-many*, yang dapat diartikan setiap *Customer* dapat menambahkan beberapa *review* atau ulasan.

3.3. *Implementation and Unit Testing*

Berikut adalah tampilan implementasi hasil *coding* dari sistem yang telah dirancang. Tampilan ini akan digunakan oleh tiga aktor pengguna (*Customer*, *Manager* dan *Tenant*) dengan antarmuka yang dirancang untuk memfasilitasi interaksi antara pengguna dan sistem. Desain layar dibuat seefisien mungkin guna memudahkan pengguna dalam mengakses berbagai fitur yang tersedia.

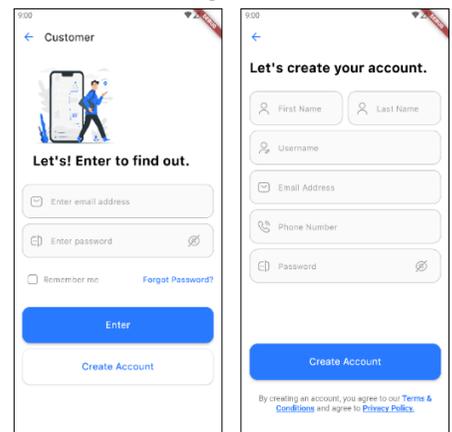
1. Tampilan *Splash Screen* dan *Welcome*



Gambar 10. Tampilan *Splash Screen* dan *Welcome*

Splash Screen adalah layar pertama ketika aplikasi dijalankan, biasanya menampilkan logo dan memberikan waktu singkat untuk bernavigasi menuju halaman selanjutnya. Pada halaman *Welcome*, pengguna dapat memilih sebagai *Manager*, *Tenant* atau *Customer*.

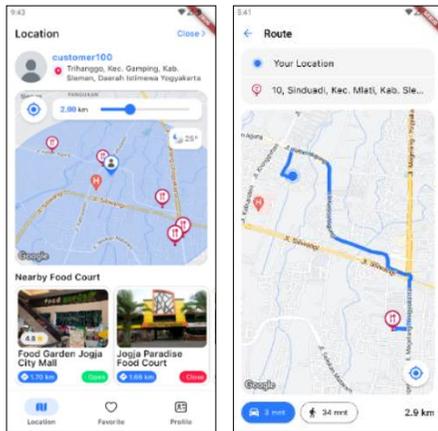
2. Tampilan pada pengguna *Customer*
a. Halaman *Login* dan *Daftar Akun*



Gambar 11. Tampilan *Splash Screen* dan *Welcome*

Pada halaman *Login*, *Customer* dapat masuk menggunakan (alamat *email* dan *password*). Pada halaman *Daftar Akun*, *Customer* dapat membuat akun dan daftar dengan mengisi *form* (*first-name*, *last-name*, *username*, *email*, nomor telepon dan *password*).

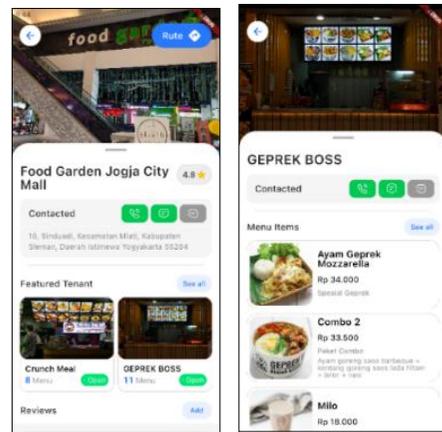
b. Halaman Informasi Lokasi dan Rute *Food Court*



Gambar 12. Halaman Informasi Lokasi dan Rute *Food Court*

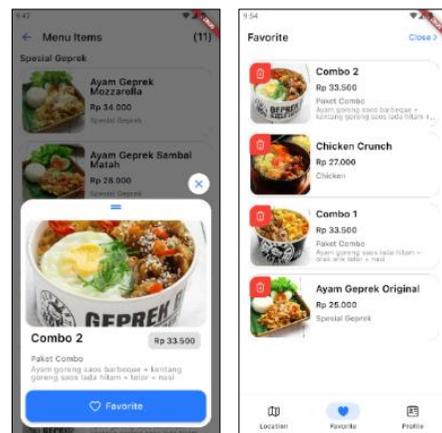
Di halaman Informasi lokasi, terdapat data lokasi *Customer* dan beberapa lokasi *Food Court*. Pengguna dapat mengatur *radius* maksimal hingga 3 km, yang akan menampilkan daftar *Food Court* terdekat. Ketika salah satu daftar tersebut di klik, maka menampilkan halaman *Detail Food Court*, dapat dilihat pada Gambar 13. Selain itu, terdapat informasi cuaca di sekitar lokasi pengguna yang diperoleh dari layanan *open source weather API* dari openweathermap.org

c. Halaman *Detail Food Court*, Daftar *Tenant* dan ulasan



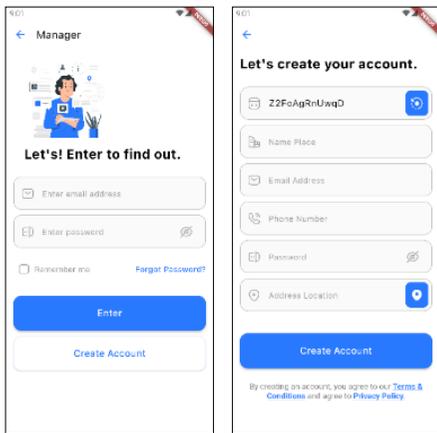
Gambar 13. Halaman *Detail Food Court*, Daftar *Tenant* dan ulasan

Halaman ini menampilkan informasi detail *Food Court* termasuk daftar *Tenant* serta ulasan atau *review*, *rating* dari pengguna lain. Tersedia tombol (*Rute*) yang akan mengarahkan ke halaman *Rute Food Court*, dapat dilihat pada Gambar 12. Setiap dalam daftar *Tenant* di klik, akan menampilkan ke halaman detail *Tenant*. Di halaman *Tenant*, *Customer* dapat menambahkan menu ke dalam daftar *favorite* dengan sekali di klik. Tampilan halaman *favorite* dilihat pada Gambar 14.



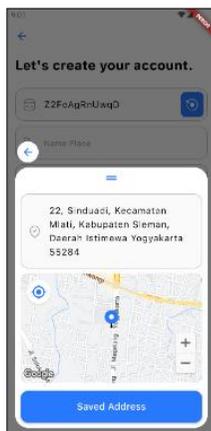
Gambar 14. Halaman *Favorite*

3. Tampilan pada pengguna *Manager*
 a. Halaman *Login* dan Daftar Akun



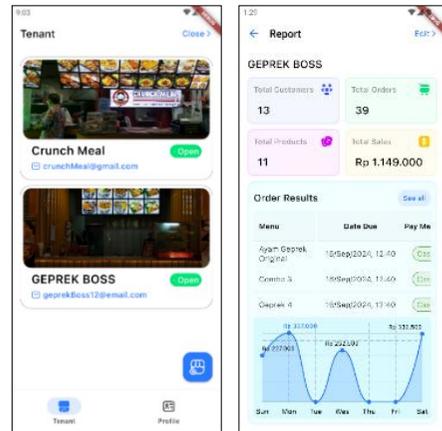
Gambar 15. Halaman *Login* dan Daftar Akun

Pada halaman *Login*, *Manager* dapat masuk dengan (alamat *email* dan *password*) dan dapat membuat akun atau daftar dengan mengisi *form* (nama tempat, alamat *email*, nomor telepon, *password* dan data lokasi dari *Food Court*) pada halaman Daftar Akun. Berikut ini mengambil data lokasi *Food Court*, dilihat pada Gambar 16.



Gambar 16. Mengambil data Lokasi

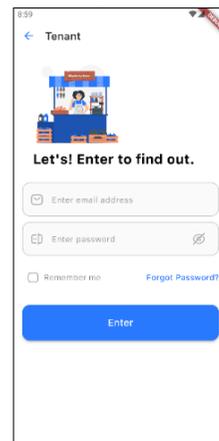
b. Halaman Daftar *Tenant* dan *Laporan*



Gambar 17. Halaman Daftar *Tenant* dan *Laporan*

Pada halaman Daftar *Tenant*, *Manager* dapat menambahkan dan menampilkan akun *Tenant*. Pada salah satu *Tenant* dalam daftar tersebut di klik, maka akan menampilkan ke halaman *Laporan*.

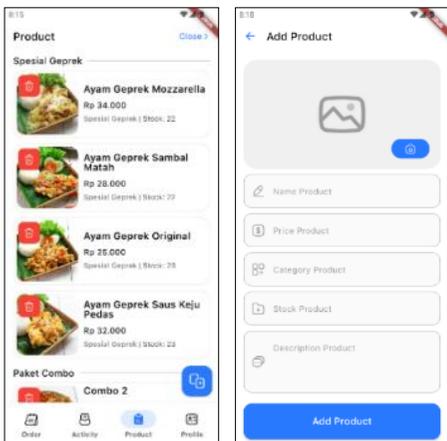
4. Tampilan pada pengguna *Tenant*
 a. Halaman *Login*



Gambar 18. Halaman *Login*

Pada halaman *Login*, *Tenant* hanya perlu masuk menggunakan (alamat *email* dan *password*) yang sudah didaftarkan oleh *Manager*.

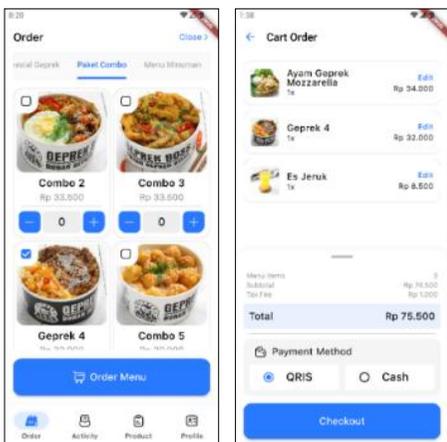
b. Halaman Tambah Daftar Menu



Gambar 19. Halaman Tambah Daftar Menu

Pada halaman ini, *Tenant* dapat memasukkan data menu (gambar, nama, harga, kategori, stok, dan deskripsi menu). Kemudian dapat menampilkan menu tersebut, yang telah dimasukkan.

c. Halaman *Order* dan Keranjang Pesanan



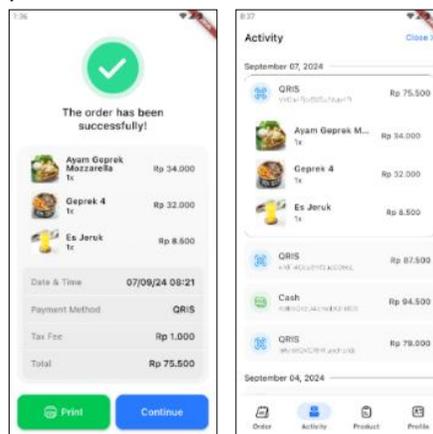
Gambar 20. Halaman *Order* dan Keranjang Pesanan

Pada halaman ini, *Tenant* dapat memilih pesanan secara langsung dari *Customer* melalui *checkbox*. Setelah tombol (*Order Menu*) di klik, data pesanan akan dikirim ke halaman Keranjang Pesanan. Di

halaman tersebut, menampilkan *subtotal* dan memilih metode pembayaran QRIS atau Tunai. Setelah tombol (*Checkout*) di klik, sistem akan mengarahkan ke halaman Pesanan Berhasil.

d. Halaman Pesanan Berhasil dan Aktivitas / *History*

Gambar 21 memperlihatkan halaman pesanan berhasil.



Gambar 21. Halaman Pesanan Berhasil dan Aktivitas / *History*

Kemudian di halaman Aktivitas / *History*, tercatat semua aktivitas dari transaksi pemesanan yang berhasil, termasuk metode pembayaran, *subtotal*, serta rincian menu yang dipesan.



Gambar 22. Hasil *Print* Struk Pesanan

Pada halaman pesanan berhasil, *Tenant* dapat mencetak struk pesanan dengan di klik pada tombol hijau (*Print*) setelah mengaktifkan dan mengkoneksikan Bluetooth ke *device printer* yang terhubung pada Android *Tenant*, dapat dilihat pada Gambar 22 hasil dari *Print* struk pesanan.

3.4. Integration and System Testing

Penelitian ini melibatkan 23 pengguna, terdiri dari (10 *Customer*, 3 *Manager*, dan 10 *Tenant*) dan menggunakan pendekatan pengujian *Black Box*, evaluasi kebergunaan fitur melalui kuesioner, serta pengukuran pemahaman pengguna dengan *pre-test* dan *post-test*. Pengujian *Black Box* bertujuan untuk mengevaluasi kinerja sistem secara fungsional, memastikan sistem menampilkan notifikasi *error* jika terjadi kesalahan, sehingga pengujian ini berfokus pada fungsi sistem secara keseluruhan (Maulana dkk., 2024). Kuesioner kebergunaan sistem digunakan untuk menilai efektivitas fitur aplikasi, dengan hasil yang menunjukkan mayoritas pengguna merasa fitur-fitur yang tersedia bermanfaat sesuai kebutuhan pengguna (Magfirah dkk., 2024).

a. Tahap Pengujian *Black Box*

Pengujian dilakukan dengan metode *Black Box* untuk menilai fungsionalitas

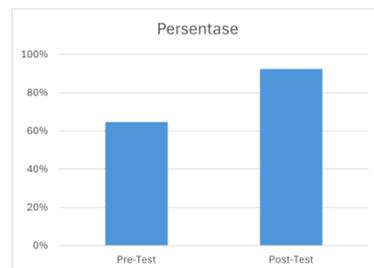
sistem secara rinci. Rincian pengujian sistem dapat dilihat pada Tabel 3.

b. Kuesioner Kebergunaan Sistem

Hasil penggabungan pengujian *Black Box* dan kuesioner menunjukkan bahwa 95% dari 23 pengguna merasa fitur aplikasi berguna. Evaluasi ini memastikan bahwa aplikasi tidak hanya berjalan dengan baik secara teknis, tetapi juga memenuhi kebutuhan pengguna.

c. Hasil Penggunaan Aplikasi

Pada tahap awal, *pre-test* dilakukan untuk mengukur pemahaman pengguna sebelum menggunakan aplikasi, hasil nilai rata-rata 64,7%. Pengguna mencoba akses aplikasi selama 1 minggu, dan hasil *post-test* menunjukkan peningkatan nilai rata-rata hingga 92,4%, menandakan aplikasi tersebut berhasil memenuhi harapan pengguna, seperti ditunjukkan pada Gambar 23.



Gambar 23. Hasil Perbandingan

Tabel 3. Pengujian Sistem *Black Box*

Skenario	Diharapkan	Didapatkan	Hasil
Daftar Akun secara lengkap	Berhasil daftar	Sistem menerima akun baru	Pass
Daftar Akun tidak lengkap	Daftar gagal, peringatan diisi semua atau salah satu kosong	Sistem memunculkan pesan 'mohon isi semua <i>form</i> yang kosong'	Pass
<i>Email</i> dan <i>password</i> diisi dengan benar	Berhasil masuk ke halaman utama	Sistem memunculkan pesan <i>login</i> berhasil dan bisa masuk	Pass
<i>Email</i> dan <i>password</i> tidak diisi dengan benar	Gagal masuk, peringatan <i>login</i> gagal	Sistem memunculkan pesan <i>login</i> gagal dan tidak bisa masuk	Pass
Menghapus data pengguna	Menghapus data pengguna secara permanen	Sistem berhasil menghapus data dan memunculkan pesan	Pass

Skenario	Diharapkan	Didapatkan	Hasil
Menambahkan data menu	Menyimpan data menu yang ditambahkan	Sistem berhasil menyimpan data dan memunculkan pesan	Pass
Mengubah data menu	Mengubah data menu	Sistem berhasil menyimpan perubahan data	Pass
Menghapus data menu	Menghapus data menu	Sistem berhasil menghapus data dan memunculkan pesan	Pass
Mencari data <i>Food Court</i> dengan <i>radius</i>	Muncul data sesuai dengan <i>radius</i> jarak pengguna	Sistem menampilkan data <i>Food Court</i> dengan <i>radius</i> pengguna	Pass
Mengarahkan ke rute lokasi <i>Food Court</i>	Menampilkan rute lokasi <i>Food Court</i>	Sistem menampilkan rute, lokasi pengguna dan <i>Food Court</i>	Pass
Tombol kontak pada <i>Food Court</i>	Terhubung dengan kontak <i>email</i> , nomor telepon <i>Manager Food Court</i>	Sistem terhubung ke aplikasi <i>chat</i> dan <i>email Manager Food Court</i>	Pass
Tombol kontak pada <i>Tenant</i>	Terhubung dengan kontak <i>email</i> , nomor telepon <i>Tenant</i>	Sistem terhubung ke aplikasi <i>chat</i> dan <i>gmail Tenant</i>	Pass
Menambahkan <i>review</i> atau ulasan	Menyimpan gambar, ulasan dan <i>rating</i>	Sistem berhasil menyimpan ulasan dan mengubah <i>rating Food Court</i>	Pass
Menambahkan data menu ke <i>favorite</i>	Menyimpan data menu ke <i>favorite</i>	Sistem berhasil menyimpan data <i>favorite</i> dan memunculkan pesan	Pass
Pemesanan menu	Menampilkan data pemesanan dan metode pembayaran	Sistem menampilkan data pemesanan dengan pembayaran	Pass
Mencetak struk pesanan	Menampilkan <i>device</i> yang terkoneksi <i>printer</i> dari Bluetooth	Sistem berhasil mencetak struk dari printer yang terkoneksi <i>Bluetooth</i> dan memunculkan pesan	Pass

4.5. Operation and Maintenance

Pada tahapan ini, peneliti melakukan pengembangan atau *maintenance*, termasuk dalam memperbaiki kekurangan dengan menambahkan fitur dompet *digital (e-wallet)*. Tujuannya adalah memudahkan *Customer* tetap berada dalam aplikasi saat melakukan pembayaran melalui QRIS. Integrasi ini dapat dilakukan dengan menggunakan pihak ketiga, seperti Midtrans, yang memfasilitasi proses pembayaran berbasis *scan QR*. Dengan demikian, *Customer* hanya perlu koneksikan dengan *provider e-wallet* seperti *Dana*, *GoPay*, *ShopeePay* atau *OVO* dengan menggunakan nomor telepon yang telah terdaftar.

IV. KESIMPULAN

Penelitian ini menghasilkan Sistem Pemesanan dan Pencarian *Food Court*

berdasarkan lokasi pengguna. Sistem dirancang menggunakan UML (*Use Case, Activity, Sequence, dan Class Diagram*) dengan *Firestore* sebagai *database* dan *Flutter* sebagai bahasa pemrograman, dengan implementasi *library Geolocator*. Sistem ini memiliki tiga fungsi utama bagi aktor :

- a. *Customer* dapat melihat lokasi, ulasan dan informasi menu *Tenant* di *Food Court*,
- b. *Manager* dapat mengakses laporan penjualan, mengelola akun *Tenant* dan lokasi *Food Court*,
- c. *Tenant* dapat mengelola menu, melakukan pesanan *digital* dan mencetak struk pesanan.

Berdasarkan pengujian aplikasi selama 1 minggu, sistem berhasil berfungsi sesuai spesifikasi dengan tingkat keberhasilan 92,4%

pada *post-test*. Pengujian *Black Box* memastikan setiap fitur berjalan dengan baik dan kuesioner menunjukkan 95% dari 23 pengguna merasa fitur pada aplikasi berguna. Secara keseluruhan, sistem memenuhi fungsinya dan kebutuhan pengguna.

DAFTAR PUSTAKA

- Andriyanto, S. (2022). *Pemodelan Perangkat Lunak*. Politeknik Manufaktur Negeri Bangka Belitung (M. Ayu (ed.)).
- Ismail, & Hadi Zakaria. (2023). Rancang Bangun Sistem Presensi Karyawan Berbasis Android Menggunakan Framework Flutter Dan Library Geolocator: (Studi Kasus: PT. KAIA Anugerah Internasional). *LOGIC: Jurnal Ilmu Komputer dan Pendidikan*, 1(5), 1355–1369.
- Jamaludin, R. B., Ramayasa, I. P., & Purwanto, A. (2023). Sistem Informasi Pemesanan Pada Food Court Denpasar Mum Berbasis Web. *Institut Teknologi dan Bisnis STIKOM Bali*, 1(1), 220–225. <https://spinter.stikom-bali.ac.id/index.php/spinter/article/view/62>
- Jason, W., Budiman, E., Setyadi, H. J., Taruk, M., Bambang, M., & Prafanto, A. (2020). Integrasi Location Based Service Informasi Bencana Alam Kota Samarinda. *Jurnal Rekayasa Teknologi Informasi (JURTI)*, 4(2), 182.
- Magfirah, M., Hayati, L. N., & Darwis, H. (2024). Evaluasi Kebergunaan Platform Pembelajaran Digital Sekolah AI-Fityan Menggunakan Metode System Usability Scale. *IDEALIS: InDonEsiA journal Information System*, 7(2), 146–153. <https://doi.org/10.36080/idealis.v7i2.3151>
- Marbun, M., & Sinaga, B. (2019). Buku Ajar Sistem Pendukung Keputusan Penilaian Hasil Belajar. In *Rudang Mayang Publisher* (Vol. 1, Nomor April).
- Maulana, M., Nurhaeni, N., Zulfadhilah, M., & Nugraha, B. (2024). Sistem Pencarian Kos Berbasis Web di Wilayah Kota Banjarmasin. *Infomatek: Jurnal Informatika, Manajemen dan Teknologi*, 26(1), 91–102. <https://doi.org/10.23969/infomatek.v26i1.10309>
- Rijal, S., & Tahir, T. (2022). Analisis Faktor Pendorong Terjadinya Urbanisasi di Wilayah Perkotaan (Studi Kasus Wilayah Kota Makassar). *Journal of Economic Education and Entrepreneurship Studies*, 3(1), 262–276. <https://doi.org/https://doi.org/10.26858/je3s.v3i1.103>
- Supiana, N. (2022). Pengembangan Aplikasi Geolocation Untuk Monitoring Lokasi Mahasiswa Selama Pandemi Berbasis Android Menggunakan Metode Waterfall (Studi Kasus: STMIK Insan Pembangunan). *Jurnal Khatulistiwa Informatika*, 10(1), 74–80. <https://ejournal.bsi.ac.id/ejurnal/index.php/khatulistiwa/article/view/11741/5477>
- Syahbani, A. (2020). Pengaruh Harga, Produk, Fasilitas, Dan Pelayanan Terhadap Kepuasan Food Court Plaza UNY. *Jurnal Pendidikan dan Ekonomi*, 9, 24–43. <https://journal.student.uny.ac.id/index.php/ekonomi/article/view/15383/14889>
- Widiastuti, N. A., & Tamrin, T. (2020). Penerapan Aplikasi Mobile Location Based Service Untuk Persebaran Usaha Mikro Kecil Menengah Dikabupaten Jepara. *Simetris: Jurnal Teknik Mesin, Elektro dan Ilmu Komputer*, 11(1), 271–278. <https://doi.org/10.24176/simet.v11i1.4015>

Wijaya, Y. D., & Astuti, M. W. (2019). Sistem Informasi Penjualan Tiket Wisata Berbasis Web Menggunakan Metode

Waterfall. *Seminar Nasional Teknologi Informasi dan Komunikasi*, 274.