

PERANCANGAN DAN IMPLEMENTASI APLIKASI ANDROID MENGGUNAKAN FRAMEWORK FLUTTER PADA PENJUALAN SEBLAK MANDJI

M. Rio Sulhan Abdillah, Rini Mayasari, Ultach Enri

Informatika, Universitas Singaperbangsa Karawang

Jalan HS. Ronggo Waluyo, Karawang, Indonesia

riosulhan022@gmail.com, rini.mayasari@staff.unsika.ac.id, ultach@staff.unsika.ac.id

ABSTRAK

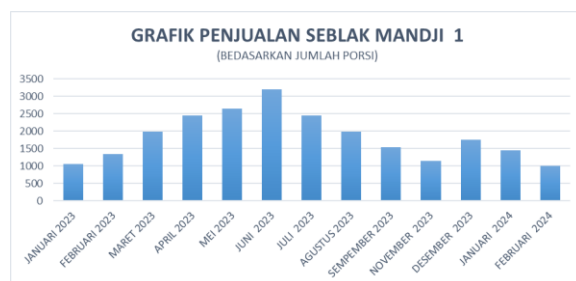
Seblak Mandji menghadapi permasalahan dalam pencatatan penjualan harian yang masih dilakukan secara manual, sehingga menyulitkan untuk mencatat secara rinci jenis dan jumlah menu yang terjual, serta menganalisis performa penjualan dengan akurat. Pemilik hanya bergantung pada total porsi yang terjual sebagai indikator, tanpa memperhitungkan variasi harga dan jenis menu. Akibatnya, pemilik kesulitan dalam membuat keputusan bisnis yang tepat. Aplikasi ini dirancang untuk mengatasi masalah tersebut, dengan menggantikan pencatatan manual dengan sistem yang lebih detail dan akurat. Penelitian ini menggunakan metodologi Software Development Life Cycle (SDLC) dengan model waterfall, yang meliputi analisis kebutuhan, perancangan sistem, implementasi, dan pengujian. Hasil pengujian aplikasi menunjukkan respon positif dari pengguna, dengan tingkat kepuasan mencapai 83,1%. Aplikasi ini berhasil memenuhi kebutuhan bisnis Seblak Mandji, menyederhanakan proses penjualan, dan meningkatkan efisiensi pengelolaan usaha. Dengan berbagai fitur yang ditawarkan, aplikasi ini menjadi solusi digital yang efektif dalam mendukung perkembangan penjualan yang lebih efisien dan modern.

Kata kunci : Perancangan, Aplikasi, Android, Penjualan, Flutter, Waterfall.

1. PENDAHULUAN

Seblak Seblak Mandji, didirikan pada 2020 di Cikarang, menawarkan seblak khas Garut dengan cita rasa autentik dan variasi menu menarik. Bisnis ini berkembang pesat, namun pemilik menghadapi kendala dalam mencatat penjualan harian secara rinci karena masih menggunakan metode manual. Staff hanya mencatat pendapatan kotor, bersih, dan pengeluaran tanpa merinci jenis atau jumlah menu yang terjual, sehingga pemilik mengandalkan perhitungan porsi berdasarkan harga menu terendah.

Dari Januari 2023 hingga Februari 2024, tren penjualan menunjukkan fluktuasi signifikan. Pada Januari 2023, terjual 1.056 porsi, meningkat hingga puncak 3.207 porsi pada Juni 2023. Namun, penjualan menurun dari Juli hingga November 2023, mencapai titik terendah 1.138 porsi. Meski naik sedikit di Desember 2023 (1.761 porsi), tren menurun kembali pada Januari dan Februari 2024, masing-masing 1.457 dan 997 porsi.



Gambar 1. Grafik penjualan seblak mandji berdasarkan jumlah porsi

Pencatatan yang kurang rinci menyulitkan pemilik untuk mengetahui menu apa saja yang terjual, penjualan harian, dan omzet yang seharusnya dicapai.

Hal ini menghambat pengambilan keputusan bisnis yang tepat serta perbaikan strategi pemasaran dan operasional.

Berdasarkan Gambar 1.2, grafik omzet penjualan Seblak Mandji menunjukkan tren *fluktuatif* dari Januari 2023 hingga Februari 2024. Omzet mencapai Rp 13.862.000 pada Januari 2023 dan memuncak di Juni 2023 sebesar Rp 46.369.000. Namun, terjadi penurunan sejak Juli 2023 hingga November 2023, dengan omzet menyusut menjadi Rp 12.316.000. Setelah sedikit meningkat di Desember 2023 ke Rp 20.351.000, penurunan berlanjut pada Januari dan Februari 2024 dengan omzet masing-masing Rp 20.351.000 dan Rp 12.684.000.



Gambar 2. Grafik penjualan seblak mandji berdasarkan omzet

Pada Gambar 1.1, penjualan Seblak Mandji pada Maret 2023 mencapai 1.987 porsi, hanya berbeda 9 porsi dibandingkan Agustus 2023 yang terjual 1.978 porsi. Namun, Gambar 1.2 menunjukkan perbedaan omzet yang signifikan: Rp. 27.842.000 pada Maret 2023 dibandingkan dengan Rp. 22.458.000 pada Agustus 2023, selisih Rp. 5.384.000.

Hal ini mengindikasikan adanya inkonsistensi dalam metode pencatatan penjualan, terutama jika jumlah telur digunakan sebagai indikator utama. Perbedaan besar dalam omset meskipun jumlah porsi hampir sama menunjukkan perlunya sistem pencatatan yang lebih rinci. Sistem ini harus mencatat data secara lengkap, termasuk jenis menu, jumlah terjual, dan harga per menu, sehingga pemilik dapat memantau performa penjualan, mengidentifikasi menu terlaris, dan memahami pengaruh variasi harga terhadap omset. Dengan demikian, keputusan strategis untuk mengembangkan usaha dapat dilakukan lebih efektif.

Oleh karena itu, untuk mengatasi tantangan yang ada pada Seblak Mandji, diperlukan solusi yang lebih modern. Salah satu solusinya adalah dengan menggunakan aplikasi berbasis Android. Menurut data dari StatCounter Global Stats selama periode Desember 2022-2023, sistem operasi Android mendominasi pasar Indonesia dengan pangsa pasar sebesar 50% (StatCounter Global Stats, 2024). Dengan memanfaatkan aplikasi Android, Seblak Mandji dapat meningkatkan efisiensi operasional secara signifikan.

Dengan berbagai macam framework yang tersedia, pengembang aplikasi pada smartphone kini memiliki kemudahan dalam mengembangkan aplikasi mereka. Salah satu framework yang saat ini populer adalah Flutter. Flutter adalah Software Development Kit (SDK) yang dikembangkan oleh Google untuk pembuatan aplikasi mobile yang dapat berjalan di perangkat iOS dan Android. Menggunakan bahasa pemrograman C, C++, Dart, dan Skia, Flutter menjadi salah satu framework yang sangat bermanfaat dan menarik. Salah satu keunggulan utamanya adalah semua kode yang dihasilkan di-compile menjadi kode native, menggunakan Android NDK, LLVM, dan AOT-compiled, tanpa melalui proses interpreter, sehingga proses kompilasi menjadi lebih cepat [9].

Pengujian blackbox dilakukan setelah pengembangan perangkat lunak selesai, dengan fokus pada hasil input dan output tanpa memperhatikan struktur kode program (Nurfauziah & Jamaliyah, 2022). Beta testing dilakukan setelah alpha testing untuk menilai tingkat penerimaan pengguna sebelum aplikasi dirilis [2].

Berdasarkan latar belakang diatas, penelitian akan bertujuan untuk mengembangkan sebuah aplikasi Android untuk Seblak Mandji. Aplikasi akan dirancang untuk memungkinkan pencatatan data penjualan dengan lebih lengkap dan akurat, termasuk detail jenis dan jumlah menu yang terjual, serta harga setiap item. Hal ini akan mengeliminasi kesalahan penulisan yang umum terjadi pada metode konvensional menggunakan lembar kertas.

2. TINJAUAN PUSTAKA

2.1. Perancangan

Perancangan adalah proses yang menentukan tindakan yang akan diambil dengan memanfaatkan berbagai metode, termasuk menggambarkan struktur,

elemen-elemen yang terlibat, serta kendala yang mungkin muncul selama pelaksanaan [7].

2.2. Implementasi

Implementasi merupakan pelaksanaan dari rencana yang telah dirancang dengan baik dan rinci. Biasanya, implementasi dilakukan setelah perencanaan dianggap sempurna. Implementasi adalah proses mengubah kebijakan menjadi tindakan, menerapkan kegiatan politik ke dalam administrasi. Pengembangan kebijakan dilakukan untuk menyempurnakan suatu program. Selain itu, implementasi melibatkan perluasan aktivitas yang menyesuaikan interaksi antara tujuan dan tindakan untuk mencapainya, serta membutuhkan jaringan pelaksana dan birokrasi yang efektif [5].

2.3. Android

Android adalah sistem operasi untuk perangkat mobile yang berbasis Linux, mencakup sistem operasi, middleware, dan aplikasi. Android menyediakan platform terbuka untuk pengembang agar mereka dapat membuat aplikasi mereka sendiri [1].

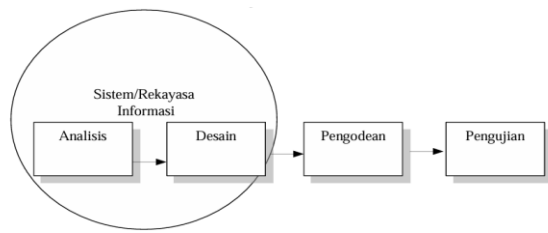
2.4. Unified Modeling Language

Unified Modeling Language (UML) merupakan sebuah bahasa pemodelan yang digunakan untuk menggambarkan sistem atau perangkat lunak dengan paradigma berorientasi objek [10]. UML memiliki berbagai jenis diagram, di antaranya:

- Use Case Diagram* digunakan untuk memodelkan dan menggambarkan interaksi antara aktor-aktor dan aplikasi yang sedang dikembangkan.
- Activity Diagram* mengilustrasikan bagaimana alur kerja atau fungsi suatu proses dari sistem berlangsung.
- Sequence Diagram* menggambarkan interaksi antar objek dalam suatu use case dengan mengilustrasikan urutan pesan yang dikirim dan objek yang menerima pesan.
- Class Diagram* adalah visualisasi yang menyajikan hubungan dan rincian mendalam antar kelas dalam desain sistem. Diagram ini merinci aturan dan tanggung jawab setiap unit, memberikan pemahaman mendalam tentang perilaku keseluruhan sistem.

2.5. Waterfall

Model waterfall, yang juga dikenal sebagai model linier sekuensial atau siklus hidup klasik, adalah pendekatan di mana pengembangan sistem dilakukan secara bertahap, dimulai dengan tahap analisis, desain, pengkodean, pengujian, dan diakhiri dengan tahap pendukung [11].



Gambar 3. Waterfall

2.6. Framework

Framework adalah sebuah komponen pemrograman yang dapat digunakan ulang secara fleksibel, sehingga pengembang tidak perlu membuat skrip yang sama berulang kali untuk tugas yang serupa. Sebagai contoh, jika seorang pengembang ingin membuat halaman web yang memungkinkan pengguna untuk melihat informasi dengan cara menggulir, sebuah framework telah menyediakan fungsi pencarian tersebut. Dalam hal ini, pengembang hanya perlu menggunakan fungsi yang telah disediakan oleh framework saat melakukan pengkodean, namun tetap mengikuti aturan yang telah ditetapkan oleh framework tersebut [6].

2.7. Flutter

Flutter adalah Software Development Kit (SDK) pengembangan aplikasi seluler sumber terbuka yang didukung oleh Google. Digunakan untuk mengembangkan aplikasi untuk platform Android dan iOS, serta menjadi metode utama untuk membuat aplikasi untuk sistem operasi Google Fuchsia. Flutter ditulis menggunakan bahasa pemrograman C, C++, dan Dart, dan menggunakan Skia Graphics Engine. Salah satu fitur utama Flutter adalah ketersediaan serangkaian widget yang dapat disesuaikan sepenuhnya untuk membangun antarmuka pengguna (UI) yang native. Ini termasuk perpustakaan Desain Material yang menarik dan widget Cupertino yang membantu dalam pembangunan UI dengan cepat tanpa kehilangan konsistensi pada berbagai platform, termasuk iOS dan Android, baik pada emulator, simulator, maupun perangkat keras [3].

2.8. Dart

Dart adalah bahasa pemrograman terstruktur open source untuk membuat aplikasi web berbasis browser yang kompleks. Pengguna dapat menjalankan aplikasi yang dibuat di Dart baik dengan menggunakan browser yang secara langsung mendukung kode Dart atau dengan mengkompilasi kode Dart pengguna ke JavaScript. Dart memiliki sintaks yang familiar, dan berbasis kelas, diketik secara opsional, dan single threaded. Ini memiliki model konkurensi yang disebut isolat yang memungkinkan eksekusi paralel. Selain menjalankan kode Dart di browser web dan mengubahnya menjadi JavaScript, Pengguna juga dapat menjalankan kode Dart di baris perintah, di hosting di mesin virtual Dart, memungkinkan klien dan bagian server dari aplikasi pengguna dikodekan

dalam bahasa yang sama. Sintaks bahasanya sangat mirip dengan Java, C#, dan JavaScript. Salah satu tujuan utama Dart adalah agar bahasa itu tampak akrab. Ini adalah skrip Dart kecil, yang terdiri dari satu fungsi yang disebut main [8].

2.9. Blackbox Testing

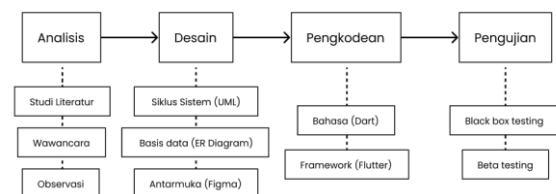
Pengujian blackbox adalah metode pengujian yang dilakukan setelah proses pengembangan perangkat lunak selesai, di mana fokusnya adalah pada pengamatan hasil input dan output tanpa perlu memperhatikan struktur kode programnya. Tujuan dari pengujian ini adalah untuk mengevaluasi kinerja perangkat lunak serta memastikan bahwa fungsi-fungsi yang telah diimplementasikan berjalan sesuai yang diharapkan. Dengan pendekatan ini, pengujian dapat dilakukan dari perspektif pengguna akhir, sehingga memungkinkan untuk mengidentifikasi masalah atau kekurangan dalam fungsionalitas perangkat lunak tanpa harus mengetahui detail teknis dari implementasinya. Ini memungkinkan pengembang untuk memperbaiki dan meningkatkan kualitas perangkat lunak secara keseluruhan [4].

2.10. Beta Testing

Beta testing adalah pengujian yang biasanya dilakukan setelah alpha testing selesai dilakukan. Beta testing merupakan pengujian yang dilakukan dari perspektif pengguna. Pengujian ini dilakukan atas dasar ingin mengetahui seberapa besar tingkat penerimaan pengguna sebelum aplikasi benar-benar dirilis. Hasil perhitungan tingkat penerimaan pengguna tersebut nantinya akan digunakan sebagai masukan untuk melakukan perbaikan aplikasi di masa mendatang [2].

3. METODE PENELITIAN

Penelitian ini akan menggunakan metodologi System Development Life Cycle (SDLC) yang merupakan suatu pendekatan sistematis dalam pengembangan perangkat lunak yang terdiri dari serangkaian tahapan dengan Model waterfall, juga dikenal sebagai model sekuensial linier atau siklus hidup klasik, adalah pendekatan di mana pengembangan sistem dilakukan secara berurutan, dimulai dari tahap analisis, desain, pengkodean, pengujian, dan tahap pendukung [11].



Gambar 4. Rancangan penelitian

3.1. Analisis Kebutuhan

Langkah pertama adalah menganalisis kebutuhan pengguna sistem yang akan dibuat dan mengidentifikasi permasalahannya. Sumber data

diperoleh melalui metode studi literatur, wawancara bersama pemilik dan pegawai Seblak Mandji, dan observasi. Data yang dihasilkan kemudian dianalisis dan diolah untuk menginformasikan sistem yang akan dibuat dan spesifikasi kebutuhannya, yang kemudian akan digambarkan dengan menggunakan use case diagram untuk memahami permasalahan yang harus dipecahkan.

3.2. Desain

Langkah pertama adalah menganalisis kebutuhan pengguna sistem yang akan dibuat dan mengidentifikasi permasalahannya. Sumber data diperoleh melalui metode studi literatur, wawancara bersama pemilik dan pegawai Seblak Mandji, dan observasi. Data yang dihasilkan kemudian dianalisis dan diolah untuk menginformasikan sistem yang akan dibuat dan spesifikasi kebutuhannya, yang kemudian akan digambarkan dengan menggunakan use case diagram untuk memahami permasalahan yang harus dipecahkan.

3.3. Pengkodean

Pada langkah ini, sistem akan dikembangkan dengan menggunakan pengkodean menggunakan bahasa pemrograman Dart dengan framework Flutter berdasarkan rancangan desain yang telah dibuat.

3.4. Pengujian

Langkah terakhir dalam proses ini adalah tahap pengujian, di mana peneliti akan menggunakan metode blackbox dan beta testing untuk memastikan bahwa sistem yang telah dibangun memenuhi kebutuhan dan kriteria yang diinginkan. Pengujian ini akan difokuskan pada validasi fungsionalitas dari perspektif pengguna akhir, yang mungkin tidak memiliki pengetahuan tentang bahasa pemrograman tertentu. Tahap ini sangat penting untuk memastikan bahwa sistem yang dikembangkan berfungsi sesuai dengan harapan dan dapat digunakan secara efektif oleh pengguna. Evaluasi hasil pengujian dilakukan menggunakan skala Likert untuk mengukur kepuasan pengguna. Dengan demikian, peneliti dapat memastikan bahwa aplikasi memenuhi analisis kebutuhan yang telah dilakukan sebelumnya.

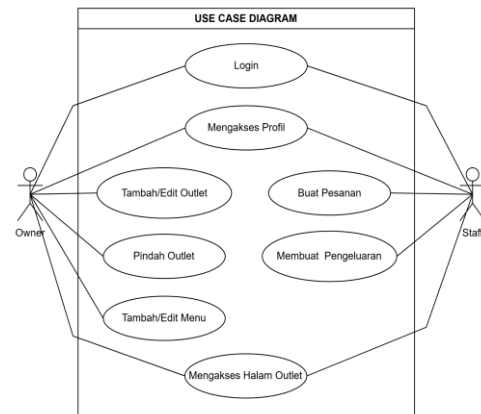
4. HASIL DAN PEMBAHASAN

Penelitian ini menghasilkan perancangan dan implementasi aplikasi *Android* untuk penjualan Seblak Mandji menggunakan *framework Flutter*. Aplikasi ini dikembangkan dengan menggunakan bahasa pemrograman Dart dan mengikuti metode SDLC model *waterfall*. Tahapan pengembangan meliputi analisis kebutuhan, perancangan sistem, implementasi, dan pengujian.

4.1. Use Case Diagram

Oleh karena itu, aplikasi ini akan dirancang dengan dua peran pengguna, yaitu pemilik (owner) dan pegawai (staff), yang digambarkan menggunakan Use

Case diagram seperti yang ditunjukkan pada gambar 5. Diagram ini akan mengilustrasikan berbagai fungsi dan interaksi yang dapat dilakukan oleh masing-masing peran dalam sistem, memberikan gambaran yang jelas tentang tanggung jawab dan hak akses setiap pengguna.

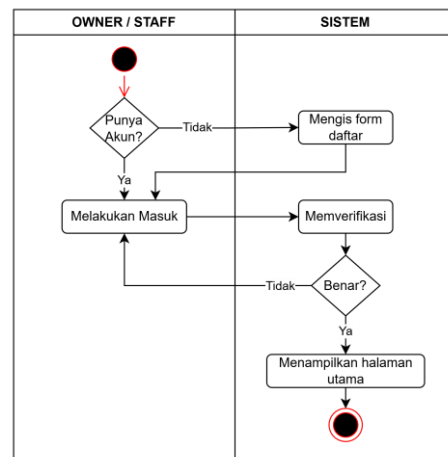


Gambar 5. Use case diagram

4.2. Activity Diagram

Diagram ini membantu memvisualisasikan alur aplikasi, sehingga memudahkan pemahaman tentang proses yang terjadi dalam aplikasi.

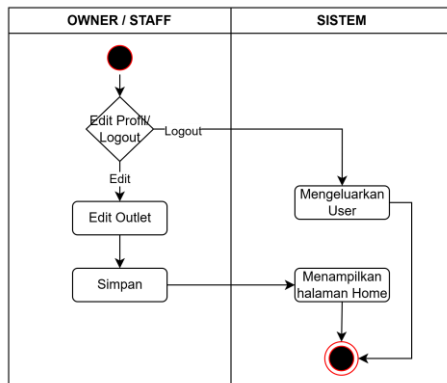
4.2.1. Login



Gambar 6. Activity diagram Login

Saat pengguna melakukan login, mereka akan diarahkan ke halaman login yang meminta untuk memasukkan username dan kata sandi. Sistem kemudian akan memverifikasi informasi yang diberikan. Jika username dan kata sandi valid, sistem akan menentukan peran pengguna, apakah sebagai Owner atau Staff, dan mengarahkan mereka ke halaman Home yang sesuai dengan peran tersebut. Jika data yang dimasukkan tidak valid, sistem akan menampilkan pesan kesalahan. Pengguna yang belum memiliki akun dapat memilih untuk mendaftar dengan mengisi data diri dan memilih peran yang diinginkan, seperti yang ditunjukkan pada ilustrasi Gambar 6.

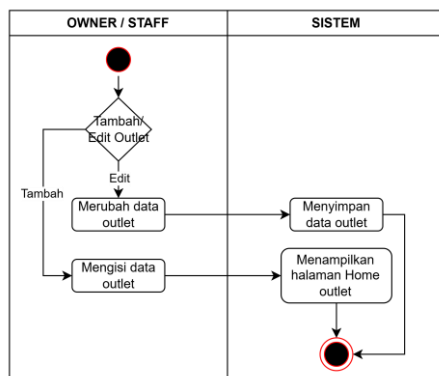
4.2.2. Mengakses Profile



Gambar 7. Activity diagram mengakses profile

Pengguna dapat mengakses halaman akun, yang memungkinkan mereka untuk memperbaiki informasi profil sesuai keinginan. Di halaman ini, mereka juga dapat melakukan proses logout, seperti yang terlihat pada Gambar 7.

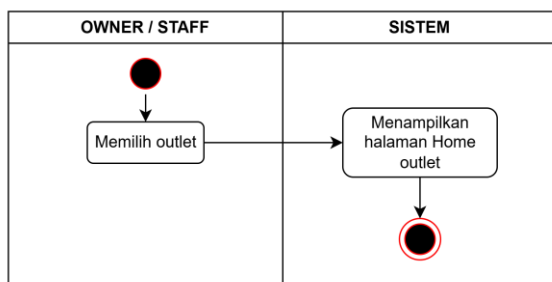
4.2.3. Tambah/Edit Outlet (Owner)



Gambar 8. Activity diagram tambah/edit outlet

Owner dapat menambah outlet baru dengan mengisi informasi seperti nama outlet, alamat lengkap, serta menunjuk staf yang akan bertanggung jawab di outlet tersebut. Selain itu, pemilik juga dapat melakukan perubahan atau pembaruan data outlet yang sudah ada sebagaimana ditunjukkan dalam Gambar 8.

4.2.4. Pindah Outlet (Owner)

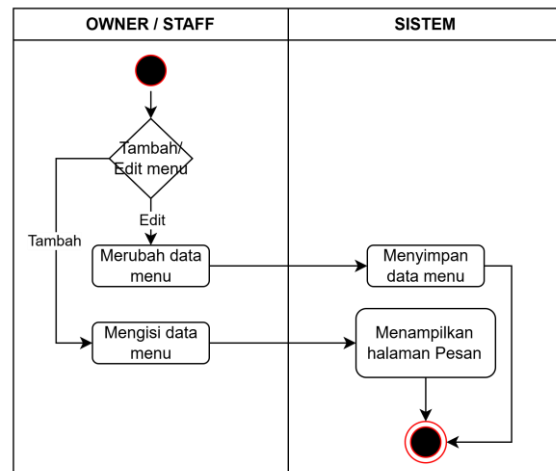


Gambar 9. Activity diagram pindah outlet

Owner dapat beralih ke outlet lain dengan memilih outlet yang tersedia dalam daftar, lalu sistem

akan menampilkan halaman *Home* outlet yang baru di pilih. Seperti yang ditunjukkan pada gambar 9.

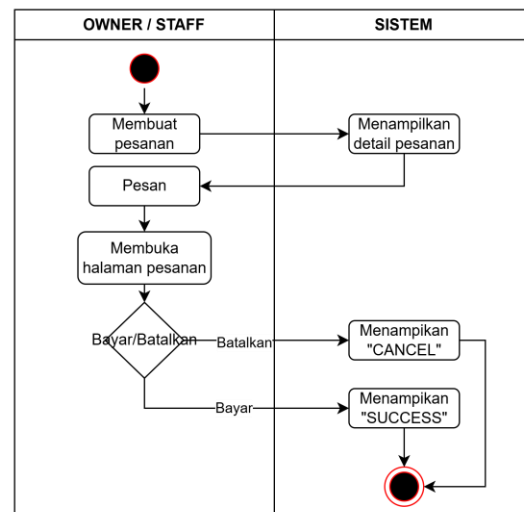
4.2.5. Tambah/Edit Menu (Owner)



Gambar 10. Activity diagram tambah/edit menu

Owner dapat menambah outlet baru dengan cara mengisi informasi lengkap seperti nama outlet, alamat lengkap, serta menentukan staf yang akan bertanggung jawab di outlet tersebut. Selain itu, pemilik juga dapat memperbaiki atau mengubah informasi outlet yang sudah ada sesuai dengan kebutuhan. Seperti yang ditunjukkan pada gambar 10.

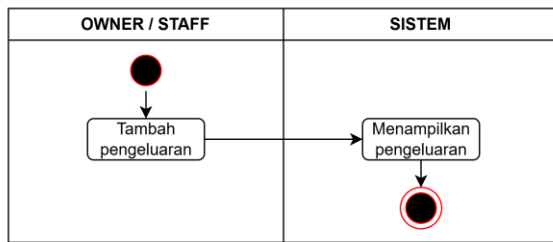
4.2.6. Buat Pesanan (Staff)



Gambar 11. Activity diagram buat pesanan

Staff dapat membuat dan mengonfirmasi pesanan yang di pesan. Pada halaman pesanan, staff dapat membatalkan pesanan jika diperlukan atau melanjutkan ke tahap berikutnya untuk menyelesaikan proses pembayaran. Seperti yang ditunjukkan pada gambar 11.

4.2.7. Membuat Pengeluaran (Staff)

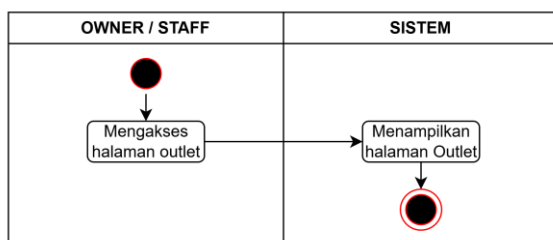


Gambar 12. Activity diagram membuat pengeluaran

Staf dapat mencatat setiap pengeluaran yang terjadi di outlet dengan mengisi informasi yang relevan, seperti nama pengeluaran, serta jumlah biaya yang dikeluarkan. Seperti yang ditunjukkan pada gambar 12.

4.2.8. Mengakses halaman Outlet

Halaman *Outlet* dirancang untuk memberikan informasi lengkap kepada pengguna mengenai outlet yang mereka kelola dalam aplikasi. Pada halaman ini, pengguna dapat dengan mudah mengakses data terkait outlet, seperti yang ditunjukkan pada Gambar 13.

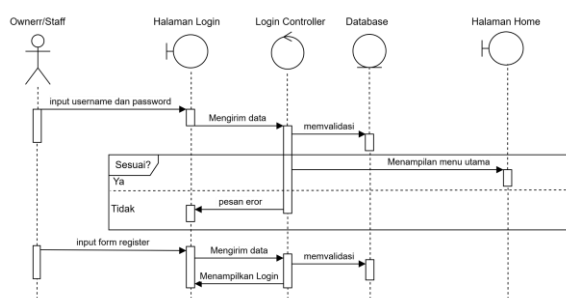


Gambar 13. Activity diagram mengakses halaman outlet

4.3. Sequence Diagram

4.3.1. Login

Sequence diagram ini menggambarkan proses login pengguna ke aplikasi Seblak Mandji, seperti yang terlihat pada gambar 14.



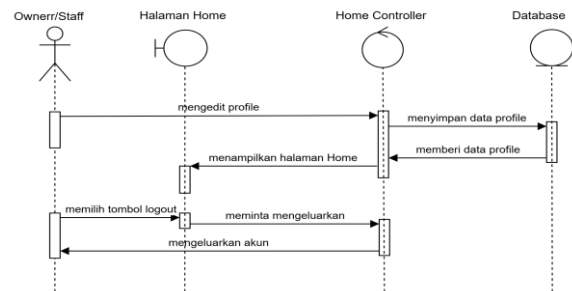
Gambar 14. Sequence diagram Login

Pada tahap awal, pengguna, baik *Owner* maupun *Staff*, membuka aplikasi dan memasukkan username serta kata sandi mereka. Sistem akan mengirimkan data yang dimasukkan ke basis data untuk proses verifikasi. Jika verifikasi berhasil, pengguna akan diarahkan ke halaman utama aplikasi. Namun, jika verifikasi gagal, sistem akan menampilkan pesan

kesalahan. Jika pengguna belum memiliki akun, mereka akan diminta untuk mendaftar dengan mengisi data diri serta memilih peran, apakah sebagai *Owner* atau *Staff*. Setelah itu, pengguna yang telah mendaftar dapat kembali diarahkan ke halaman login untuk mengakses aplikasi.

4.3.2. Mengakses Profile

Sequence diagram ini menggambarkan alur proses yang terjadi saat pengguna mengakses menu Profil dalam aplikasi Seblak Mandji, sebagaimana ditunjukkan pada gambar 15.

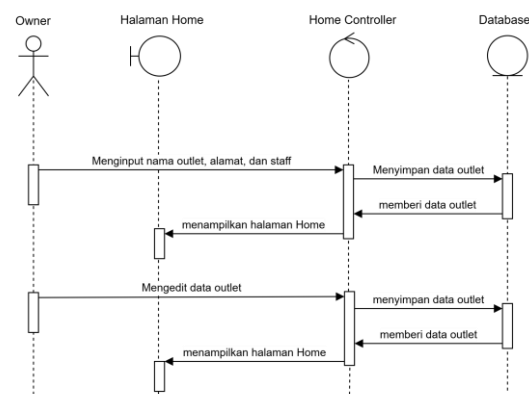


Gambar 15. Sequence diagram Mengakses Profile

Ketika pengguna mengakses halaman akun, sistem pertama-tama menampilkan antarmuka halaman akun kepada pengguna. Pada halaman ini, pengguna memiliki opsi untuk memperbarui informasi profil mereka. Jika pengguna memilih untuk memperbarui informasi profil, permintaan akan dikirimkan ke sistem, yang kemudian memvalidasi data yang dimasukkan sebelum menyimpan pembaruan tersebut ke dalam basis data. Selain itu, pengguna juga memiliki opsi untuk melakukan logout. Jika pengguna memilih opsi logout, sistem akan memproses permintaan tersebut dengan menghapus sesi aktif pengguna dan mengarahkan mereka kembali ke halaman login.

4.3.3. Tambah/Edit Outlet (Owner)

Sequence diagram ini menjelaskan alur proses Tambah atau Edit Outlet yang dilakukan melalui aplikasi Seblak Mandji, seperti yang ditampilkan pada Gambar 16.

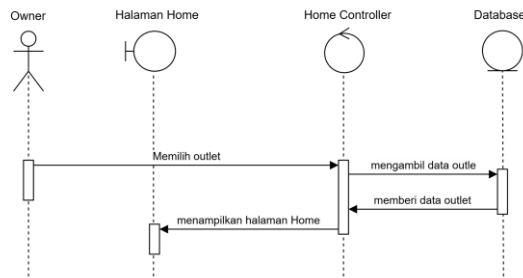


Gambar 16. Sequence diagram Tambah/Edit Outlet

Owner menginputkan data seperti nama outlet, alamat lengkap, serta staf yang ditugaskan untuk outlet

tersebut. Sistem akan menyimpan informasi outlet baru ke dalam database dan memperbarui tampilan halaman Home sesuai dengan outlet yang baru ditambahkan. Untuk memperbarui atau mengedit data outlet yang sudah ada, pemilik dapat memilih outlet dari daftar yang tersedia, lalu mengubah informasi sesuai kebutuhan. Setelah perubahan dilakukan, sistem akan memperbarui data outlet di database dan menyesuaikan tampilan halaman Home dengan outlet yang dipilih.

4.3.4. Pindah Outlet (Owner)



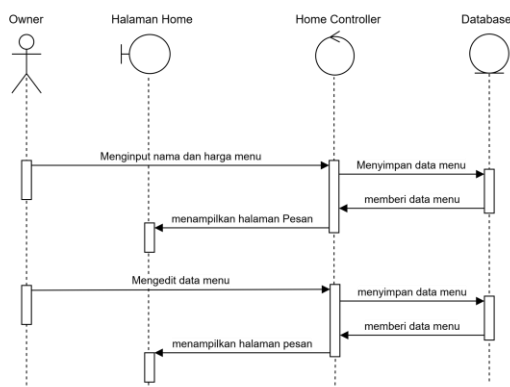
Gambar 17. Sequence diagram Pindah Outlet

Sequence diagram ini menjelaskan alur proses Pindah Outlet yang dilakukan melalui aplikasi Seblak Mandji, seperti yang ditampilkan pada Gambar 17.

Owner memilih outlet dari daftar outlet yang tersedia pada sistem. Setelah pilihan outlet dilakukan, sistem akan menerima permintaan dari Owner dan memproses perubahan outlet. Selanjutnya, sistem akan mengambil data outlet yang dipilih dari database. Setelah data berhasil diambil, sistem akan memperbarui tampilan halaman Home sesuai dengan outlet baru yang dipilih.

4.3.5. Tambah/Edit Menu (Owner)

Sequence diagram ini menjelaskan alur proses Tambah/Edit Menu yang dilakukan melalui aplikasi Seblak Mandji, seperti yang ditampilkan pada Gambar 14.

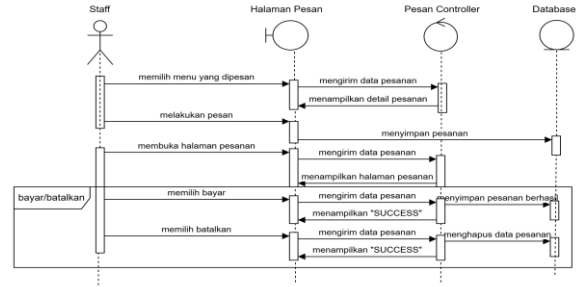


Gambar 18. Sequence diagram Tambah/Edit Menu

Owner menambah outlet baru dengan cara mengisi informasi lengkap seperti nama outlet, alamat lengkap, serta menentukan staf yang akan bertanggung jawab di outlet tersebut. Setelah informasi diisi, sistem menyimpan data tersebut ke dalam database. Sistem

kemudian menampilkan halaman pesan yang menunjukkan menu outlet yang baru saja ditambahkan. Selain itu, jika pemilik ingin memperbarui informasi outlet yang sudah ada, pemilik dapat mengakses data outlet yang sudah ada, mengubah informasi sesuai kebutuhan, dan kemudian mengirimkan data yang telah diperbarui ke sistem. Sistem akan menyimpan perubahan tersebut ke dalam database dan menampilkan halaman pesan yang menunjukkan menu outlet yang telah diperbarui.

4.3.6. Buat Pesanan (Staff)



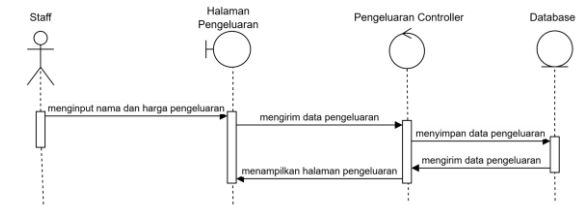
Gambar 19. Sequence diagram Tambah/Edit Menu

Sequence diagram ini menjelaskan alur proses Buat Pesanan yang dilakukan melalui aplikasi Seblak Mandji, seperti yang ditampilkan pada Gambar 19.

Staff memilih menu yang akan dipesan. Setelah itu, sistem mengirimkan data pesanan dan menampilkan detail pesanan kepada staff. Staff kemudian melanjutkan untuk melakukan pemesanan, dan sistem mengirimkan data pesanan ke database. Selanjutnya, staff membuka halaman pesanan dan memilih apakah ingin membayar atau membatalkan pesanan. Jika staff memilih untuk membayar, sistem akan mengirimkan data pesanan ke database dan menampilkan pesan "SUCCESS". Namun, jika staf memilih untuk membatalkan, sistem akan menghapus data pesanan dari database dan menampilkan pesan "CANCEL".

4.3.7. Membuat Pengeluaran (Staff)

Sequence diagram ini menggambarkan proses yang terjadi pada Membuat Pengeluaran pada aplikasi Seblak Mandji, seperti yang terlihat pada gambar 20.



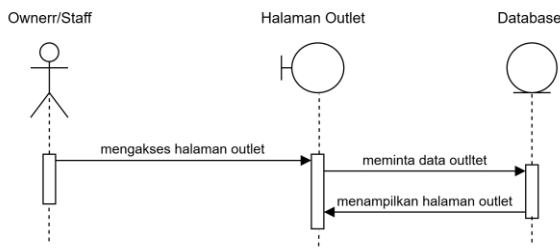
Gambar 20. Sequence diagram Membuat Pengeluaran

Staf memasukkan nama dan harga pengeluaran ke dalam sistem. Setelah itu, sistem akan menerima input tersebut dan memprosesnya untuk menyimpan data pengeluaran ke dalam basis data. Setelah pengeluaran berhasil disimpan, sistem akan menampilkan halaman pengeluaran yang

menunjukkan daftar pengeluaran yang telah dibuat dan disimpan sebelumnya.

4.3.8. Mengakses halaman Outlet

Sequence diagram ini menggambarkan proses yang terjadi pada halaman outlet pada aplikasi Seblak Mandji, seperti yang terlihat pada gambar 21.

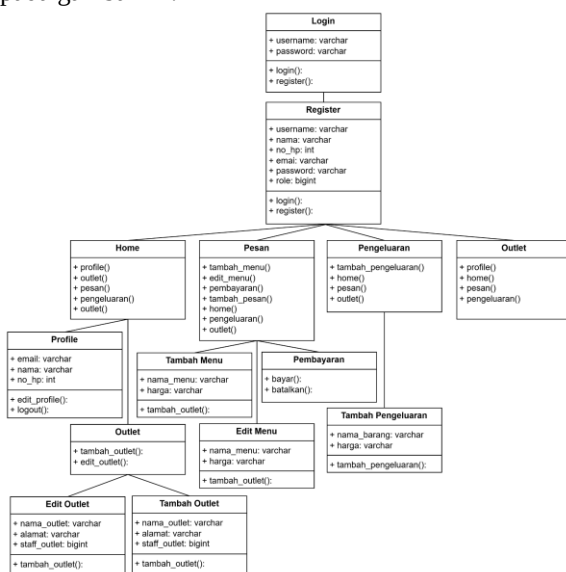


Gambar 21. *Sequence diagram* outlet

Di halaman ini, baik Owner maupun Staff dapat mengakses berbagai data yang berkaitan dengan outlet, termasuk informasi penjualan.

4.4. Class Diagram

Diagram kelas ini menggambarkan struktur sistem aplikasi Seblak Mandji. Seperti yang terlihat pada gambar 22.



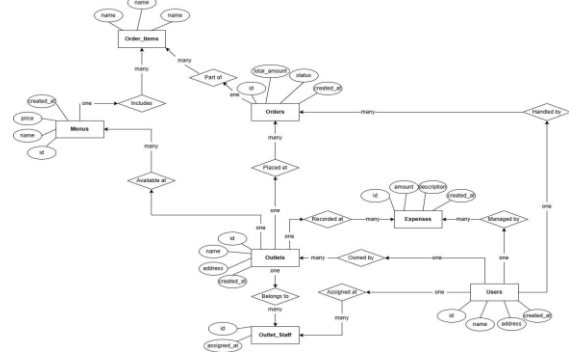
Gambar 22. *Class Diagram*

Diagram ini menggambarkan kelas-kelas utama dalam sistem, termasuk kelas-kelas seperti Login, Register, Home, Pesan, Pengeluaran, Outlet, Profile, Tambah Menu, Pembayaran, dan lainnya. Setiap kelas memiliki atribut yang merepresentasikan informasi terkait entitas tersebut, setiap kelas juga dilengkapi dengan metode yang mendefinisikan fungsi atau perilaku yang dapat dijalankan oleh entitas dalam sistem, seperti proses login, pendaftaran akun, pengelolaan menu, pembayaran, serta interaksi pengguna dengan outlet dan data lainnya. Dengan struktur ini, sistem dapat berjalan secara terorganisir,

dengan setiap entitas berfungsi sesuai perannya dalam aplikasi.

4.5. Entity relationship diagram

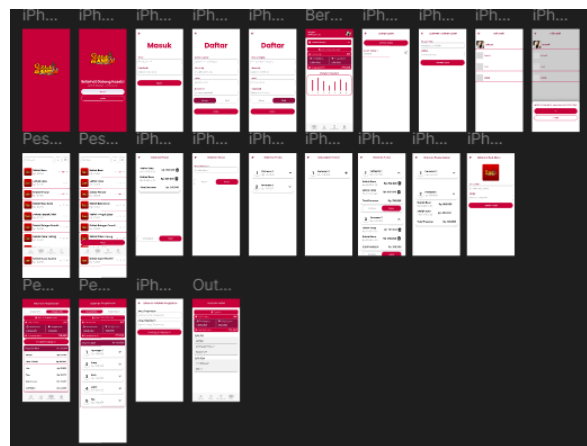
Penulis akan menggunakan *Entity Relationship Diagram* (ERD) untuk menggambarkan hubungan antara berbagai entitas dalam sistem. Diagram ini akan membantu menunjukkan bagaimana data akan disimpan dan dihubungkan satu sama lain sebagai basis data, seperti yang ditunjukkan pada gambar 23.



Gambar 23. *Entity Relationship Diagram*

4.6. Antarmuka

Pada tahap ini, akan dilakukan perancangan dan pembuatan desain antarmuka. Langkah ini mencakup penyusunan elemen visual serta tata letak yang akan digunakan dalam aplikasi atau sistem, dengan fokus pada kenyamanan pengguna, estetika, dan fungsionalitas. Desain antarmuka dirancang untuk memastikan pengalaman pengguna yang intuitif dan menarik.

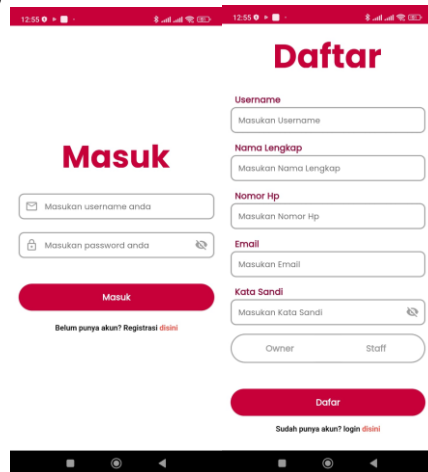


Gambar 24. Desain antarmuka

4.7. Pengkodean

Proses ini melibatkan implementasi desain yang telah dirancang menjadi kode pemrograman bahasa Dart dengan menggunakan *framework* Flutter, dan logika yang mendukung fungsionalitas aplikasi.

4.7.1. Login

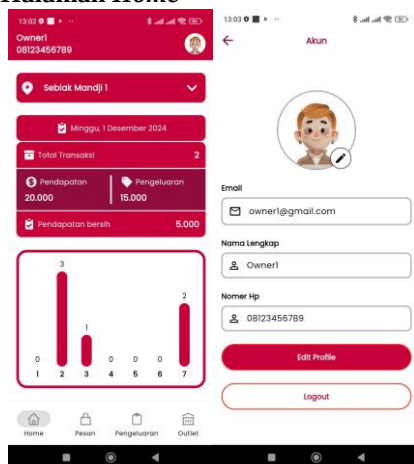


Gambar 25. Halaman login

Pada halaman login, pengguna diminta untuk mengisi username dan kata sandi. Sistem akan memverifikasi keabsahan data tersebut. Jika valid, sistem akan mengenali peran pengguna, apakah sebagai Owner atau Staff, lalu mengarahkan mereka ke halaman utama yang sesuai. Jika data tidak valid, pesan kesalahan akan ditampilkan.

Bagi pengguna yang belum memiliki akun, tersedia opsi pendaftaran dengan mengisi data diri dan memilih peran, seperti ditunjukkan pada Gambar 25. Proses ini memastikan akses sesuai dengan peran pengguna.

4.7.2. Halaman Home



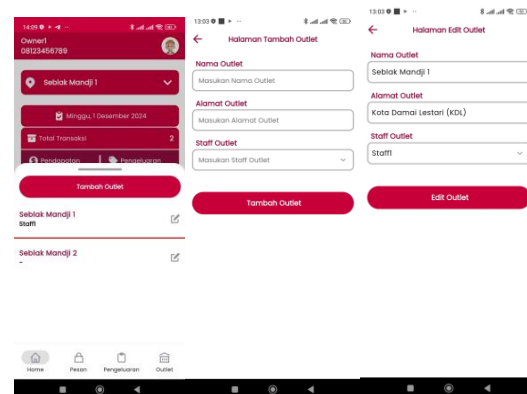
Gambar 26. Halaman home

Halaman ini menampilkan informasi terbaru mengenai outlet yang sedang aktif, termasuk data penjualan yang relevan. Dengan begitu, pengguna dapat melihat performa outlet secara real-time dan memperoleh gambaran tentang aktivitas penjualan pada periode tersebut.

Selain itu, baik Owner maupun Staff memiliki akses untuk melihat dan mengedit profil pribadi mereka, yang memungkinkan mereka memperbarui informasi seperti email, nama, dan nomer hp. Jika pengguna ingin mengakhiri sesi aplikasi, mereka dapat

melakukan logout untuk keluar dengan aman. Seperti Gambar 26.

4.7.3. Halaman Pindah, Tambah, dan Edit Outlet

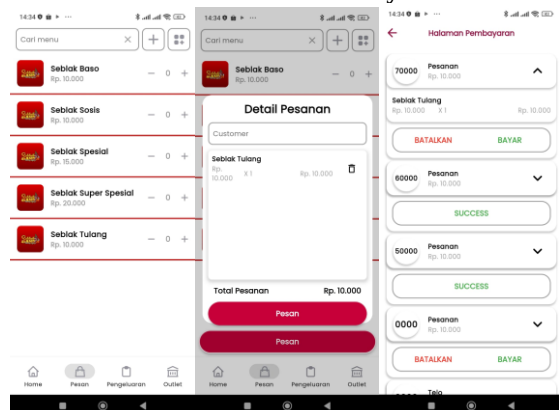


Gambar 27. Halaman pindah, tambah, dan edit outlet

Halaman ini memberikan pengguna kemampuan untuk mengelola outlet dengan mudah. Pengguna dapat memilih outlet tempat mereka ingin pindah, menambah outlet baru, atau mengedit outlet yang sudah ada.

Untuk menambah outlet baru, pengguna diminta untuk mengisi data seperti nama outlet, alamat, serta memilih staf yang akan bertugas di outlet tersebut. Selain itu, pengguna memiliki opsi untuk melakukan perubahan pada data outlet yang sudah ada, seperti memperbarui informasi outlet atau mengalihkan staf ke outlet lain sesuai kebutuhan. Seperti yang terlihat pada Gambar 27.

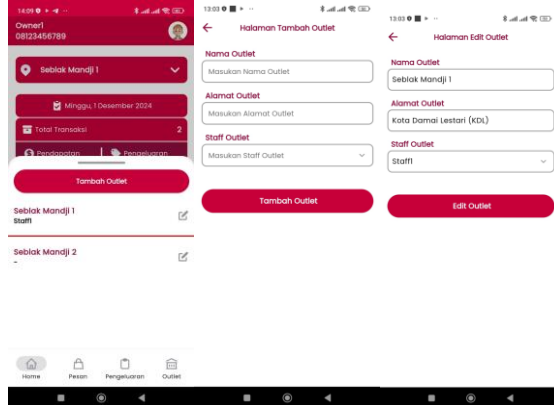
4.7.4. Halaman Pesan dan Pembayaran



Gambar 28. Halaman pesan dan pembayaran

Pada halaman ini, pengguna dapat melakukan aktivitas pemesanan menu yang dilakukan oleh staff. Setelah menu dipilih dan dipesan, sistem akan mengarahkan pengguna ke halaman pembayaran untuk menyelesaikan proses pemesanan tersebut. Di halaman pembayaran, pengguna dapat melihat rincian pesanan, dan mengonfirmasi pesanan apakah di batalkan atau di bayar.

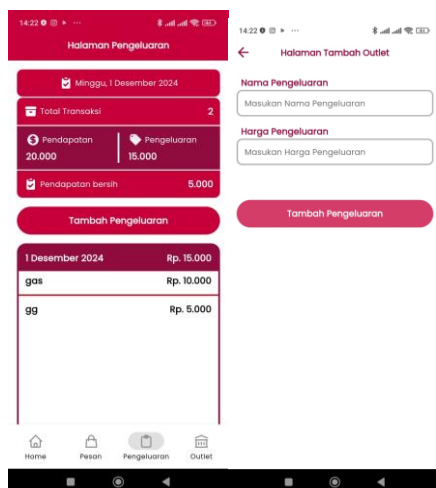
4.7.5. Halaman Tambah dan Edit Menu



Gambar 29. Halaman pindah, tambah, dan edit outlet

Pada halaman ini, pengguna dapat menambahkan menu baru dengan mengisi nama menu dan harga yang diinginkan. Selain itu, sistem juga menyediakan fitur untuk mengedit menu yang sudah ada, memungkinkan pengguna untuk menyesuaikan informasi menu sesuai dengan kebutuhan atau perubahan harga.

4.7.6. Halaman Pengeluaran

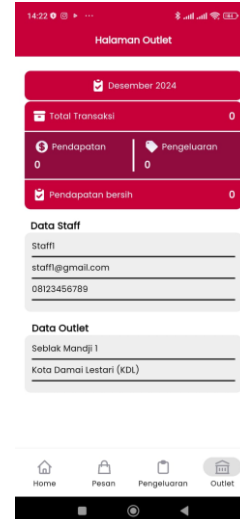


Gambar 30. Halaman pengeluaran

Pada halaman pengeluaran, baik *Owner* maupun *Staff* dapat mengakses dan melihat laporan pengeluaran. Namun, hanya *Staff* yang dapat melakukan pencatatan pengeluaran. Untuk mencatat

pengeluaran, *Staff* akan mengklik tombol "Tambah Pengeluaran". Dalam fitur ini, *Staff* akan diminta untuk mengisi informasi terkait nama barang yang dikeluarkan serta harga barang tersebut. Lalu akan di tampilkan di tabel pada halaman pengeluaran.

4.7.7. alaman Outlet



Gambar 31. Halaman outlet

Di halaman ini, baik *Owner* maupun *Staff* dapat mengakses berbagai data yang berkaitan dengan outlet, termasuk informasi penjualan.

4.8. Pengujian

Pada tahap ini, pengujian sistem dilakukan untuk menguji sejauh mana fungsi fungsi sistem dapat berjalan sesuai dengan tujuan dari perancangan sistem. Terdapat dua pengujian yang dilakukan pada penelitian ini, yaitu *black box testing* dan *beta testing*.

4.8.1. Black Box Testing

Black box testing digunakan untuk menguji kualitas sistem yang menitikberatkan pada evaluasi fungsionalitas sistem. Pengujian dilakukan seperti apakah input diterima dengan benar dan output yang dihasilkan telah sesuai dengan yang diharapkan.

Tabel 1. Pegujian Black Box

No	Item Uji	Skenario Uji	Hasil yang diharapkan	Pengujian
1	Halaman Login	Klik tombol "Masuk" dengan username dan password yang benar	Menampilkan halaman utama	Berhasil
		Klik tombol "Masuk" dengan username dan password yang salah	Menampilkan alert "Username atau password salah"	Berhasil
2	Halaman Register	Klik tombol "DAFTAR" dengan menginput data diri	Menampilkan halaman utama	Berhasil
3	Halaman Home	(Owner) Klik icon "V" lalu klik "Tambah Outlet"	Membuat outlet baru	Berhasil
		(Owner) Klik icon "V" lalu klik icon "Pencil"	Melakukan edit outlet	Berhasil
		(Owner) Klik icon "V" lalu klik "Nama Outlet"	Melakukan pindah outlet	Berhasil
		(Klik "Foto" lalu klik "Edit Profile"	Melakuakn edit profile	Berhasil

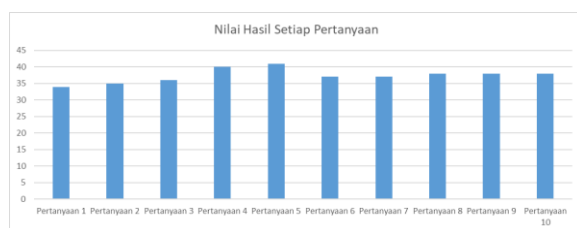
		Klik “Foto” lalu klik “Logout”		Berhasil
4	Halaman Pesan	(Owner) Klik icon “+” yang ada di bagian atas, lalu klik “Tambah Menu”	Menambah Menu	Berhasil
		(Owner) Klik icon “Pencil” pada menu, lalu klik “Edit Menu”	Melakukan Edit Menu	Berhasil
		(Staff) Klik “+” pada menu yang akan di pesan lalu klik “Pesan” Pada pembayaran klik icon “V” lalu klik “bayar” untuk menyelesaikan pesanan	Melakukan pesan dan menyelesaikan pesanan	Berhasil
		(Staff) Klik “+” pada menu yang akan di pesan lalu klik “Pesan” Pada pembayaran klik icon “V” lalu klik “bayar” untuk membatalkan pesanan	Melakukan pesan dan membatalkan pesanan	Berhasil
5	Halaman Pengeluaran	(Staff) Klik “Tambah Pengeluaran” lalu isi nama dan harga barang	Menambah pengeluaran	Berhasil
6	Halman Outlet	Klik icon “Outlet”	Menampilkan halaman outlet	Berhasil

Berdasarkan hasil pengujian black box pada tabel di atas, dapat disimpulkan bahwa seluruh komponen telah berhasil diuji dengan baik. Skenario pengujian telah dijalankan dengan benar dan menghasilkan hasil yang diharapkan pada setiap skenario. Hal ini menunjukkan bahwa sistem dapat berjalan dengan baik tanpa masalah.

4.8.2. Beta Testing

Setelah Setelah pengujian *blackbox*, tahap berikutnya adalah beta testing dengan evaluasi pengguna. Aplikasi diuji langsung oleh pemilik dan staf Seblak Mandji untuk menilai kesesuaian dengan kebutuhan pengguna nyata. Pengujian ini melibatkan survei menggunakan *Google Form* dengan 10 pertanyaan berbasis skala Likert (1–5) untuk mengevaluasi kualitas sistem, layanan, dan kepuasan pengguna.

Sebanyak sembilan pengguna (dua pemilik dan tujuh staf) terlibat, dengan skor maksimal 45 per pertanyaan dan total 450 jika semua responden memberikan nilai tertinggi pada setiap pertanyaan. Hasil survei ini divisualisasikan dalam grafik batang, memberikan wawasan untuk perbaikan dan pengembangan lebih lanjut, guna memastikan aplikasi memenuhi harapan pengguna secara optimal.



Gambar 32. Hasil setiap pertanyaan

1. Persentase hasil dari pertanyaan pertama $\frac{34}{45} \times 100\% = 75,5\%$
2. Persentase hasil dari pertanyaan kedua $\frac{35}{45} \times 100\% = 77,7\%$
3. Persentase hasil dari pertanyaan ketiga $\frac{36}{45} \times 100\% = 80\%$

4. Persentase hasil dari pertanyaan keempat $\frac{40}{45} \times 100\% = 88,8\%$
5. Persentase hasil dari pertanyaan kelima $\frac{41}{45} \times 100\% = 91,1\%$
6. Persentase hasil dari pertanyaan keenam $\frac{37}{45} \times 100\% = 82,2\%$
7. Persentase hasil dari pertanyaan ketujuh $\frac{37}{45} \times 100\% = 82,2\%$
8. Persentase hasil dari pertanyaan kedelapan $\frac{38}{45} \times 100\% = 84,4\%$
9. Persentase hasil dari pertanyaan kesembilan $\frac{38}{45} \times 100\% = 84,4\%$
10. Persentase hasil dari pertanyaan kesepuluh $\frac{38}{45} \times 100\% = 84,4\%$

Setelah memperoleh hasil sebesar 374, langkah berikutnya adalah menghitungnya menggunakan rumus yang telah dijelaskan sebelumnya. Hasil perhitungannya adalah sebagai berikut:

$$\text{presentase kepuasan pengguna} = \frac{374}{450} \times 100\% = 83,1\%$$

Dengan persentase 83,1%, dapat disimpulkan bahwa aplikasi ini mendapatkan respons yang Sangat Baik dari pengguna, yaitu owner dan staf Seblak Mandji. Hasil ini menunjukkan bahwa aplikasi telah berhasil memenuhi kebutuhan pemilik dan staf Seblak Mandji.

5. KESIMPULAN DAN SARAN

Setelah Setelah melalui serangkaian tahapan penelitian dan pengembangan, aplikasi Android untuk penjualan Seblak Mandji menggunakan framework Flutter berhasil diselesaikan dan diuji. Aplikasi ini dirancang untuk meningkatkan efisiensi pengalaman pengguna, mempermudah proses penjualan, serta mendukung pengelolaan usaha secara praktis. Proses pengembangan mencakup analisis kebutuhan pengguna, pengembangan fitur relevan, dan desain antarmuka yang mudah dipahami. Uji coba yang dilakukan dengan pemilik dan staf Seblak Mandji

menunjukkan tanggapan positif dengan tingkat kepuasan mencapai 83,1%.

Hasil pengujian menunjukkan aplikasi ini berhasil memenuhi kebutuhan operasional bisnis dan memberikan solusi digital yang efektif untuk mendukung penjualan yang lebih efisien dan modern. Berdasarkan umpan balik pengguna, beberapa saran pengembangan termasuk integrasi dengan sistem pembayaran yang lebih variatif, peningkatan sistem pengelolaan stok, perbaikan desain antarmuka agar lebih dinamis dan responsif, serta penambahan fitur ekspor ke format Excel untuk mempermudah pembuatan laporan keuangan dan evaluasi performa bisnis.

DAFTAR PUSTAKA

- [1] Azis, N., Pribadi, G., & Nurcahya, M. S. (2020). Analisa dan perancangan aplikasi pembelajaran bahasa Inggris dasar berbasis Android. *Jurnal IKRA-ITH Informatika*, 4(3), 1-5.
- [2] Menora, T., Primasari, C. H., Wibisono, Y. P., Sidhi, T. A. P., Setyohadi, D. B., & Cininta, M. (2023). Implementasi Pengujian Alpha dan Beta Testing pada Aplikasi Gamelan Virtual Reality. *KONSTELASI: Konvergensi Teknologi dan Sistem Informasi*, 3(1), 48-63.
- [3] Muslim, R., Sari, R. P., & Rahmayuda, S. (2022). Implementasi Framework Flutter pada Sistem Informasi Perpustakaan Masjid (Studi Kasus: Masjid di Kota Pontianak). *Coding: Jurnal Komputer dan Aplikasi*, 10(01), 46-59.
- [4] Nurfauziah, H., & Jamaliyah, I. (2022). Perbandingan metode testing antara blackbox dengan whitebox pada sebuah sistem informasi. *VISUALIKA*, 8(2), 105-113.
- [5] Putri, D. A., Ceicillia, S., Rizky, G. A., & Farida, S. N. (2022). Implementasi analisis SWOT (strength, weakness, opportunities, and threat) dalam strategi pemasaran produk pada PT Global Food Supplies Surabaya. *Jurnal Bisnis Indonesia*, 13(1), 78-104.
- [6] Sari, D. P., & Wijanarko, R. (2019). Implementasi framework Laravel pada sistem informasi penyewaan kamera (studi kasus di Rumah Kamera Semarang). *INFORMATIKA DAN RPL*, 2(1), 32-36.
- [7] Setiyanto, R., Nurmaesah, N., & Rahayu, N. S. A. (2019). Perancangan sistem informasi persediaan barang studi kasus di Vahncollections. *Jurnal Sisfotek Global*, 9(1), 137-142.
- [8] Sofi, N., & Dharmawan, R. (2022). PERANCANGAN APLIKASI BENGKEL CSM BERBASIS ANDROID MENGGUNAKAN FRAMEWORK FLUTTER (BAHASA DART). *Jurnal Teknik dan Science*, 1(2), 53-64.
- [9] Suhendro, J. M., Sudarma, M., & Khrisne, D. C. (2021). Rancang bangun aplikasi seluler penyedia jasa perawatan dan kecantikan menggunakan framework flutter. *Jurnal SPEKTRUM*, 8(2), 68-82.
- [10] Wahyutama, M. F., & Natasyah, N. (2020). Perancangan Sistem Informasi Platform Pencarian Kerja Pada PT. Wira Karya Indonesia. *ADI Bisnis Digital Interdisiplin Jurnal*, 1(2), 46-59.
- [11] Yoki Firmansyah, & Udi. (2024). Penerapan Metode SDLC Waterfall Dalam Pembuatan Sistem Informasi Akademik Berbasis Web Studi Kasus Pondok Pesantren Al-Habi Sholeh Kabupaten Kubu Raya, Kalimantan Barat, 4(1), 184-191.