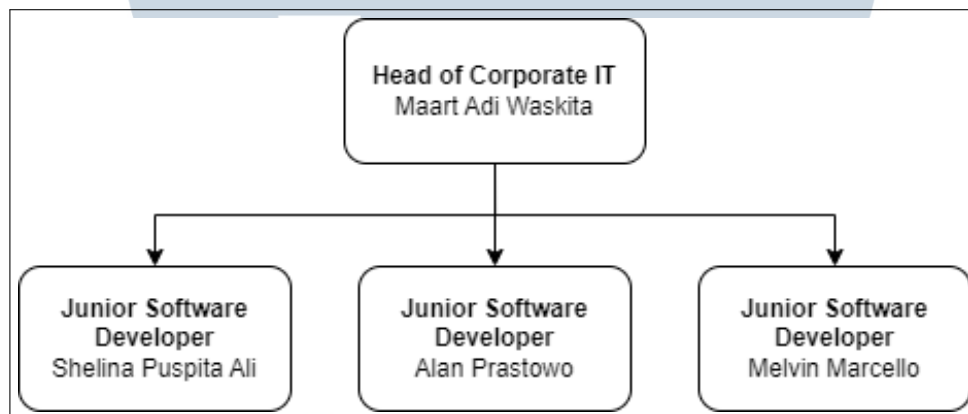


BAB 3 PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Organisasi

Dalam pelaksanaan program magang, peserta magang berperan sebagai *Junior Software Developer* yang bergabung dengan divisi *Corporate IT*, serta juga mendapat bimbingan secara langsung dari Bapak Maart Adi Waskita sebagai *Head of Corporate IT* yang turut serta membantu kegiatan magang ini. Berikut merupakan bagan singkat yang ada pada Gambar 3.1, menggambarkan kedudukan organisasi khususnya yang berkaitan dengan posisi peserta magang.



Gambar 3.1. *Struktur Organisasi peserta magang*

Dalam divisi ini peserta magang sebagai *Junior Software Developer* memiliki peran untuk mengembangkan aplikasi *mobile* Jasa Boga Raya POS App dimana akan digunakan sebagai aplikasi dibagian kasir, tanggung jawab utama peserta magang melibatkan desain, pengembangan, dan pengujian aplikasi *mobile* untuk memastikan bahwa pengalaman pengguna di bagian kasir menjadi lebih baik, lebih lancar, dan lebih efisien. Rekan-rekan sesama *Junior Software Developer* juga memiliki peran krusial dalam proyek ini. Dimana bertanggung jawab untuk mengembangkan bagian web yang akan menjadi antarmuka penting untuk pengguna di berbagai platform. Selain itu, juga terlibat dalam menggarap *back-end* web dari aplikasi POS ini, yang menjadi fondasi dari semua operasional sistem. Selain dari tugas utama sebagai *Junior Software Developer* peserta magang juga bertugas sebagai *help-desk* dimana membantu karyawan lain yang menghadapi permasalahan seputar IT.

3.2 Tugas yang Dilakukan

Selama kegiatan magang sebagai *Junior Software Developer* di perusahaan Bakrieland, tugas utama peserta magang adalah turut berkontribusi dalam melakukan pengembangan dan pengimplementasian produk dari anak perusahaan Bakrieland. Kegiatan tersebut dilakukan menggunakan Visual Studio Code dengan bahasa pemrograman Dart sebagai basisnya dan Flutter sebagai *framework* nya. Berikut merupakan peran peserta magang selama melaksanakan kegiatan magang pada perusahaan Bakrieland:

1. Melakukan riset terhadap aplikasi *Point of Sales* (POS) sebelumnya lalu membuat *mock-up* berdasarkan saran dari pihak klien yaitu Jasa Boga Raya.
2. Melakukan Observasi di lapangan terhadap bagaimana penggunaan aplikasi POS sebelumnya.
3. Mengerjakan aplikasi POS berdasarkan hasil riset dan melakukan *bug fixing* untuk memastikan aplikasi berjalan dengan baik saat digunakan oleh pengguna.
4. Melakukan *meeting* terhadap *progress* pembuatan aplikasi POS dan Mengerjakan revisi yang diajukan mencakup perubahan dan penyesuaian dari klien.
5. Sebagai *help-desk* bertugas melakukan *troubleshooting* terhadap permasalahan yang dialami oleh karyawan divisi lain.

3.3 Uraian Pelaksanaan Magang

Tabel 3.1 mencakup berbagai tugas dan proyek yang telah dikerjakan kepada peserta magang selama masa magang. Informasi yang akan dicantumkan pada tabel akan dibuat berurutan berdasarkan minggu pertama kali peserta magang mendapatkan tugas tersebut. Namun penting untuk diketahui bahwa beberapa dari kegiatan yang dilakukan dapat dikerjakan secara paralel, dimana bisa saja pada dua tugas yang berbeda memiliki interval waktu yang saling tumpang tindih. Hal seperti itu dapat terjadi dikarenakan beberapa faktor seperti terdapat tugas baru dengan prioritas lebih tinggi dari yang sedang dikerjakan, selain itu juga bisa disebabkan karena pada tugas tertentu terdapat beberapa kendala baik secara tim maupun individu yang menyebabkan tugas tersebut mengalami penundaan. Berikut adalah

rincian dari pekerjaan yang telah dilakukan peserta magang selama melaksanakan program magang:

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu Ke -	Pekerjaan yang dilakukan
1	Melakukan <i>set up tools</i> yang diperlukan dalam melakukan pekerjaan, membuat <i>mock-up</i> aplikasi POS(<i>Point of Sales</i>), mempresentasikan hasilnya kepada pihak Jasa Boga Raya.
2	Menerapkan desain UI/UX yang ada di <i>mock-up</i> pada <i>front-end</i> aplikasi POS, melakukan pengujian terhadap API, pergi ke salah satu booth Jasa Boga Raya untuk melakukan observasi terhadap aplikasi POS sebelumnya.
3	Mencoba mengintegrasikan API Login ke aplikasi POS dan membuatnya agar saat masuk menu makanan yang ditampilkan sesuai dengan outlet dari pengguna dan membuat <i>detail screen</i> .
4	Membantu persiapan acara RUPS (Rapat Umum Pemegang Saham) untuk persiapan keperluan <i>streaming</i> menggunakan OBS, menambahkan gambar dummy pada menu makanan dan memperbaiki tampilan login, melakukan meeting untuk <i>update progress</i> aplikasi POS.
5	Memperbaiki <i>bug</i> dimana saat pindah navigasi tampilan <i>gridview</i> hilang, menambahkan <i>log-out</i> , menambahkan cart, memperbaiki <i>bug</i> pada <i>counter</i> yang menghitung jumlah item
6	Menambahkan pilihan tunai dan non-tunai pada <i>payment screen</i> , membuat produk yang tampil di menu sesuai dengan outlet pengguna
7	Mencoba menginstall aplikasi pada <i>physical device</i> , mencoba mengintegrasikan agar bisa menggunakan <i>physical device</i> walau masih di <i>localhost</i>
8	Menambahkan catatan pembeli, memperbaiki <i>bug</i> dicatat pembeli, menggunakan API baru dengan <i>framework</i> laravel
9	Mengintegrasikan library untuk bluetooth print dengan aplikasi agar dapat mencetak struk dengan <i>printer bluetooth</i> , memperbaiki tampilan struk sesuai hasil <i>meeting</i> dengan pihak Jasa Boga Raya

3.3.1 Tools yang digunakan

Selama periode magang, perusahaan telah menyediakan perangkat keras dan lunak yang diperlukan untuk kegiatan kerja. Berikut adalah daftar perangkat keras dan perangkat lunak yang digunakan oleh peserta magang selama magang

1. Perangkat Keras:

- Sistem Operasi: Microsoft Windows 11 Home Single Language
- *Processor*: 12th Gen Intel(R) Core(TM) i7-12650H 2.30 GHz
- Kartu Grafis: NVIDIA GeForce RTX 3050 Laptop GPU GDDR6 @ 4GB (128 bits)
- RAM: 16 GB

2. Perangkat Lunak:

- Github Desktop: digunakan sebagai *version control system* agar perubahan *code* dapat ter-*manage* dengan baik.
- Visual Studio Code: digunakan sebagai Text Editor untuk pengembangan perangkat lunak berbasis *mobile* yang menggunakan *framework flutter*.
- Flutter: digunakan sebagai *framework* untuk membuat aplikasi *mobile* dalam hal ini aplikasi android.
- scrcpy : digunakan untuk melakukan *screen mirroring* pada perangkat Android yang terhubung dengan laptop menggunakan kabel USB.
- SQLyog: digunakan untuk melakukan pengecekan *local DB* pada perangkat yang digunakan untuk menjalankan aplikasi.

3.4 Pembaruan aplikasi POS Jasa Boga Raya

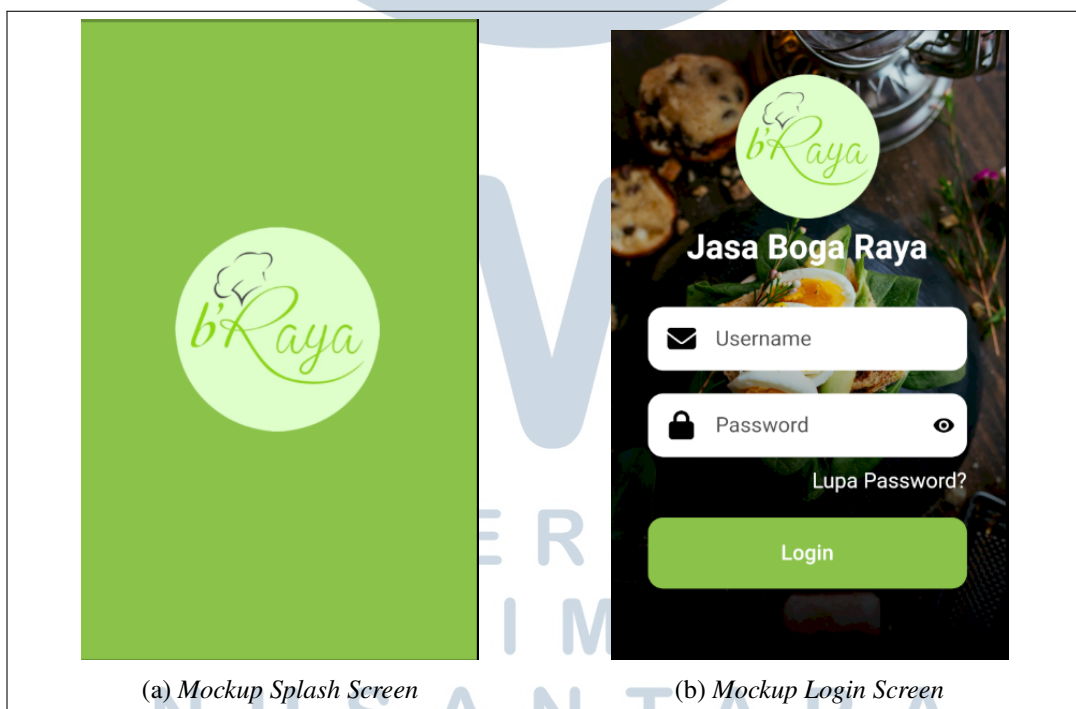
Kegiatan ini merupakan tugas yang diberikan kepada peserta magang oleh Bapak Maart Adi Waskita selaku *Head of Corporate IT*. Proses pengerjaan dimulai dengan pengerjaan *mock-up* untuk aplikasi POS Jasa Boga Raya dan diakhiri dengan peserta magang mempresentasikan hasil dari development aplikasi tersebut.

3.4.1 Pembuatan Mock-up

Pada bagian ini peserta magang akan secara rinci menjelaskan proses pembuatan *mock-up* untuk aplikasi POS Jasa Boga Raya yang telah dikembangkan. Mock-up ini dirancang dengan tujuan untuk disajikan dalam sebuah presentasi kepada pihak Jasa Boga Raya, dengan harapan dapat mendapatkan masukan berharga. Setelah masukan didapat peserta magang akan merangkumnya dan mengimplementasikannya di *front-end* aplikasi.

A. Mock-up Splash Screen & Login Screen

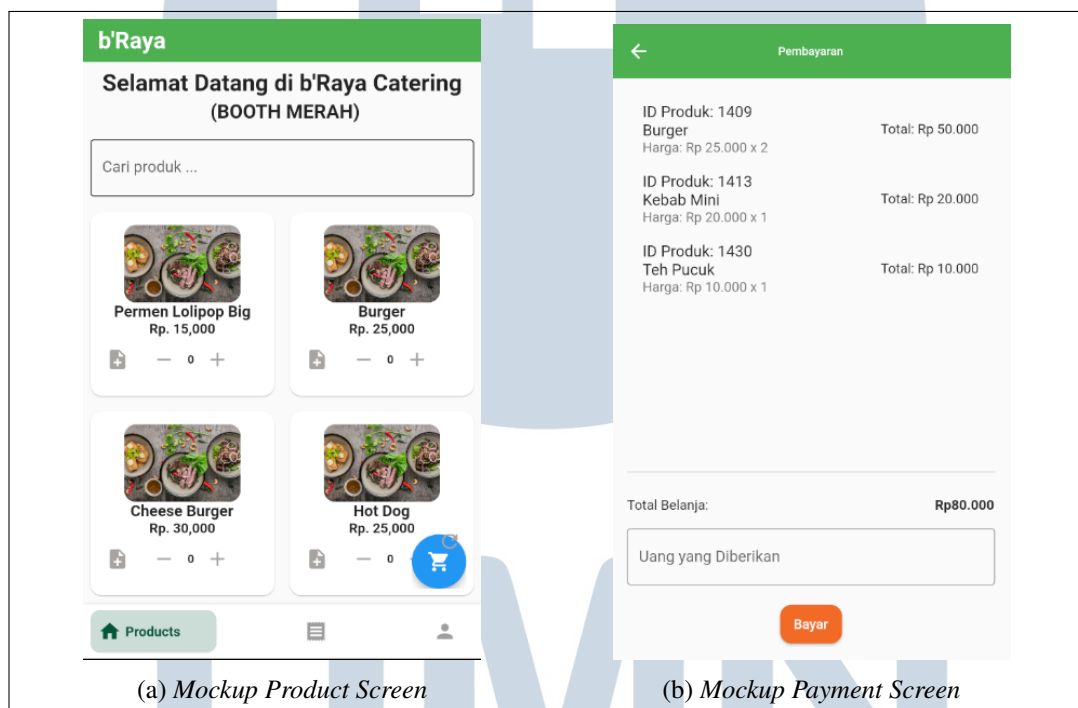
Pada Gambar 3.2 terdiri dari 2 bagian, pertama *splash screen* pada Gambar 3.2a merupakan layar pertama yang muncul ketika pengguna membuka aplikasi, menampilkan logo khas dari aplikasi. Setelah itu pengguna akan diarahkan ke *login screen* yang ada pada Gambar 3.2b . Pada tahap ini, seorang pengguna yang bertugas sebagai kasir dapat masuk ke aplikasi dengan melakukan *login* menggunakan akun kasir yang telah terdaftar pada website *backend*.



Gambar 3.2. Tampilan *Mockup Splash Screen & Login Screen*

B. Mock-up Product Screen & Payment Screen

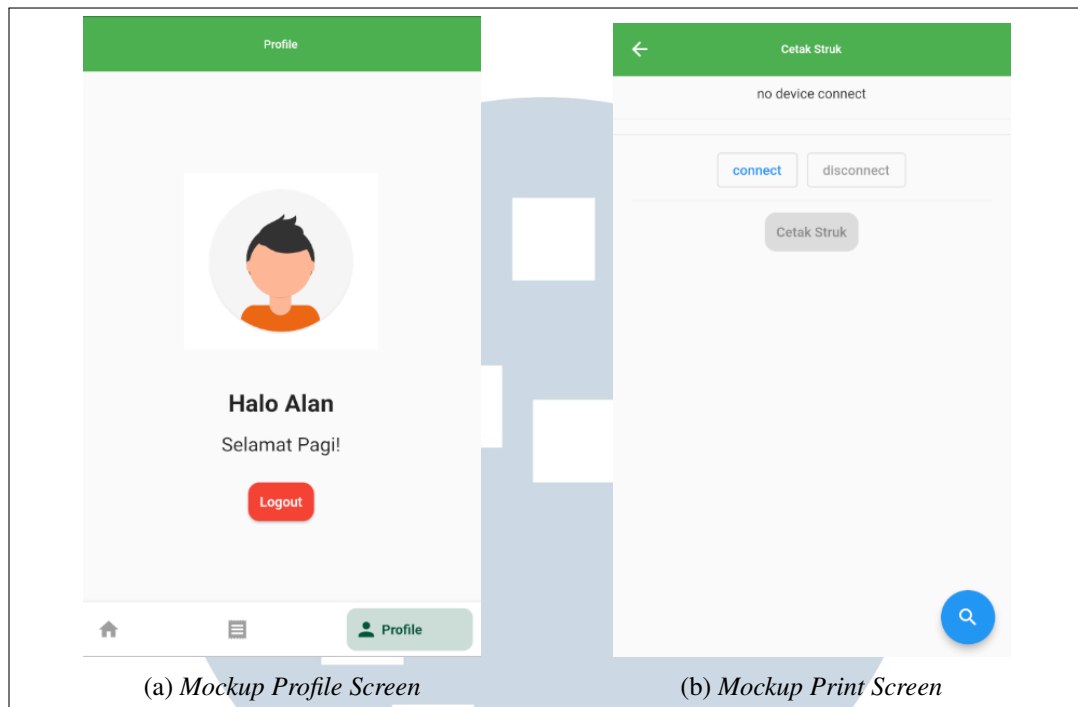
Pada Gambar 3.3 terdiri dari 2 bagian, pertama *product screen* pada Gambar 3.3a merupakan tempat dimana kasir dapat memilih makanan yang dipesan pembeli dan ditampilkannya jumlah item dengan *counter*, kasir juga dapat mencatat *note* dari pembeli. Setelah selesai memilih daftar makanan, kasir akan diarahkan ke halaman keranjang pembelian. *Payment screen* pada Gambar 3.3b merupakan tahap dimana kasir dapat menyelesaikan proses transaksi dimana dimasukan nominal uang yang diberikan pembeli dan dihitung dengan harga keseluruhan *item*.



Gambar 3.3. Tampilan Mockup Product Screen & Payment Screen

C. Mock-up Profile Screen & Print Screen

Pada Gambar 3.4 terdiri dari 2 bagian, pertama *profile screen* pada Gambar 3.4a layar akan menampilkan nama pengguna yang *login* dan ucapan "selamat pagi", "selamat siang", atau "selamat malam" yang disesuaikan dengan waktu lokal perangkat. Ucapan ini ditampilkan di bagian bawah nama pengguna. Di bagian *print screen* pada Gambar 3.4b, akan ditampilkan daftar perangkat *bluetooth* yang terdeteksi oleh perangkat. *Printer* yang dipilih akan digunakan untuk mencetak struk penjualan. Struk penjualan akan menampilkan informasi transaksi, seperti produk yang dibeli, jumlah produk yang dibeli, dan jumlah uang kembalian.



Gambar 3.4. Tampilan *Mockup Profile Screen & Print Screen*

3.4.2 Pembuatan Flowchart Aplikasi POS

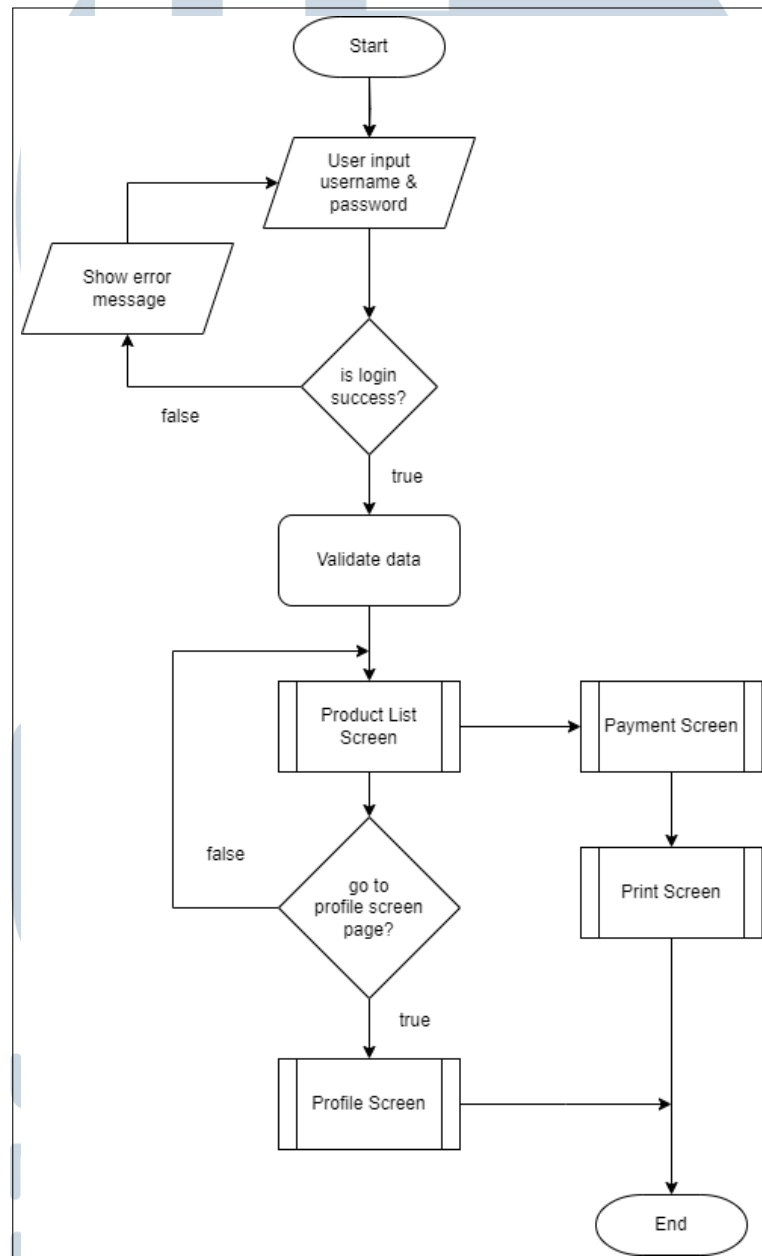
Pada bagian ini akan dijelaskan mengenai *flowchart* yang dibuat berdasarkan hasil *meeting* dan juga pengamatan dari aplikasi POS sebelumnya dibuat *flowchart* untuk aplikasi POS yang baru. Berikut beberapa bagian *flowchart* dari aplikasi POS.

A. Flowchart Aplikasi POS

Gambar 3.5 menunjukkan *flowchart* dari aplikasi POS Jasa Boga Raya. *Flowchart* ini menggambarkan proses penggunaan aplikasi POS Jasa Boga Raya dari awal hingga akhir. Pada awal proses, pengguna melakukan login dengan memasukkan *username* dan *password*. Jika *username* dan *password* yang dimasukkan tidak ada, maka akan ditampilkan pesan error. Jika *username* dan *password* yang dimasukkan ada, maka pengguna akan diarahkan ke *product list screen*. Jika pengguna tidak berpindah ke (*profile screen*) maka akan tetap di *product list screen*.

Pengguna dapat memilih produk yang ingin dibeli dan menambahkannya ke keranjang (*cart*). Setelah selesai memilih produk, pengguna akan diarahkan ke

payment screen. Pada *payment screen*, akan ditampilkan daftar produk yang telah dipilih beserta total harga. Pengguna dapat mengubah jumlah produk yang ingin dibeli atau menghapus produk dari keranjang. Jika pengguna sudah yakin dengan produk yang ingin dibeli, maka dapat menyelesaikan transaksi. Setelah transaksi selesai, pengguna dapat mencetak struk pembelian pada *print screen*.



Gambar 3.5. Flowchart Aplikasi POS

B. Flowchart Product List Screen

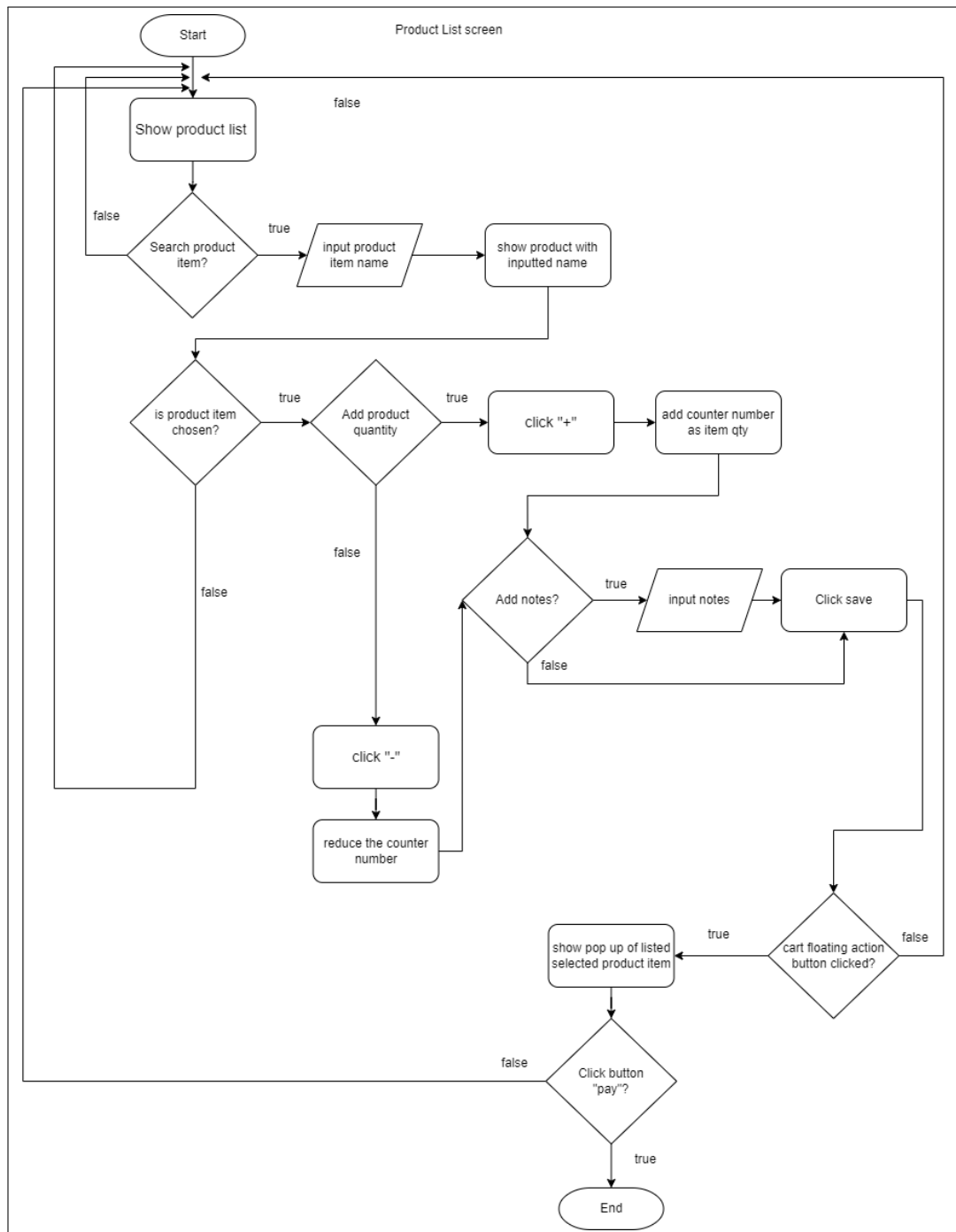
Flowchart pada Gambar 3.6 menunjukkan proses menambahkan produk ke keranjang pada layar *product list screen*. Prosesnya dimulai dengan pengguna yang seorang kasir berada di layar daftar produk. Pengguna dapat melakukan search produk jika produk ada maka tampilkan. Pengguna memilih produk yang ingin ditambahkan ke dalam keranjang. Jika produk tersebut sudah dipilih, *flowchart* akan langsung melanjutkan ke bagian *add notes*.

Pada bagian *add notes* pengguna dapat menambahkan catatan dari pelanggan ke produk misal jika produk makanan bisa ditambahkan catatan seperti "tambahkan sausnya". Jika pengguna ingin menambahkan catatan, dapat dilakukan dengan memasukkan catatan dan mengklik tombol "Simpan". Jika pengguna tidak ingin menambahkan catatan, dapat langsung melanjutkan ke bagian untuk menambahkan jumlah produk atau mengurangi jumlah produk pada keranjang.

Pengguna dapat menambah atau mengurangi jumlah produk yang ingin ditambahkan ke keranjang dengan mengklik tombol "+" atau "-". Pengguna dapat mencari produk dengan memasukkan nama produk pada *field search*. Jika pengguna tidak ingin mencari produk, maka akan diarahkan ke bagian pilih produk lagi.

Jika pengguna mengklik tombol FAB (*Floating Action Button*) maka akan dipratinjau produk item yang sudah masuk ke dalam keranjang. Jika pengguna tidak ingin menambahkan produk ke keranjang belanja, maka akan diarahkan ke bagian pilih produk lagi. Saat tombol "bayar" ditekan maka pengguna akan diarahkan layar *payment screen* dan akhiri proses.

U M N
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.6. *Flowchart* Product List Screen

C. Flowchart Payment Screen

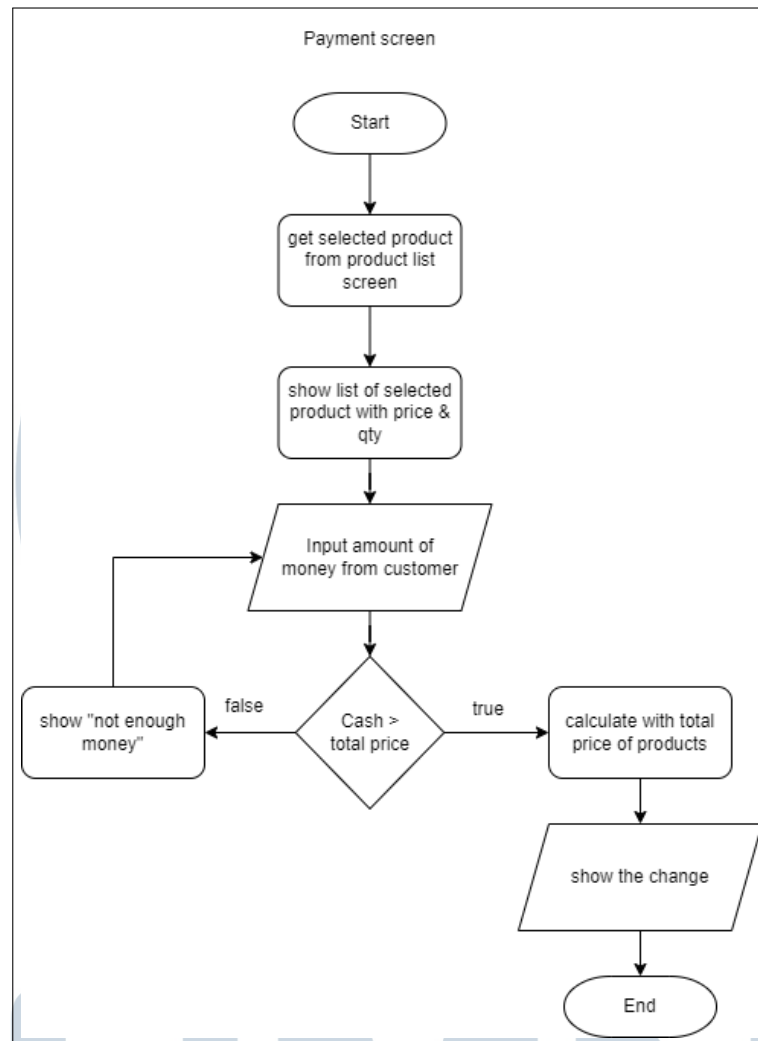
Flowchart pada Gambar 3.7 menggambarkan langkah-langkah dalam proses pembayaran transaksi pada produk yang telah dimasukkan ke dalam keranjang. Proses dimulai dengan mendapatkan daftar produk dari keranjang yang tersedia di

layar *product list screen*. Setelah itu, produk-produk tersebut ditampilkan bersama dengan harga dan jumlahnya di layar *payment screen*.

Di layar *payment screen*, kasir memasukkan uang yang dibayarkan oleh pelanggan. Selanjutnya, sistem melakukan kalkulasi dengan mengurangi jumlah uang pelanggan dari total harga produk di keranjang. Jika jumlah uang pelanggan lebih besar dari total harga produk, kasir akan memberikan kembalian kepada pelanggan sesuai dengan selisihnya. Sebaliknya, jika jumlah uang pelanggan yang dimasukkan kurang dari total harga produk, akan muncul pesan bahwa uang yang diberikan kurang.

Proses pembayaran di layar *payment screen* dianggap selesai ketika semua perhitungan telah dilakukan dan transaksi berhasil diselesaikan. Pada titik ini, pelanggan dapat menerima kembalian jika ada, dan kasir dapat menyelesaikan transaksi tersebut. Dengan demikian, proses pada *payment screen* dianggap selesai setelah pembayaran berhasil dilakukan, sesuai dengan langkah-langkah yang tergambar dalam *flowchart* pada Gambar 3.7.





Gambar 3.7. *Flowchart* Payment Screen

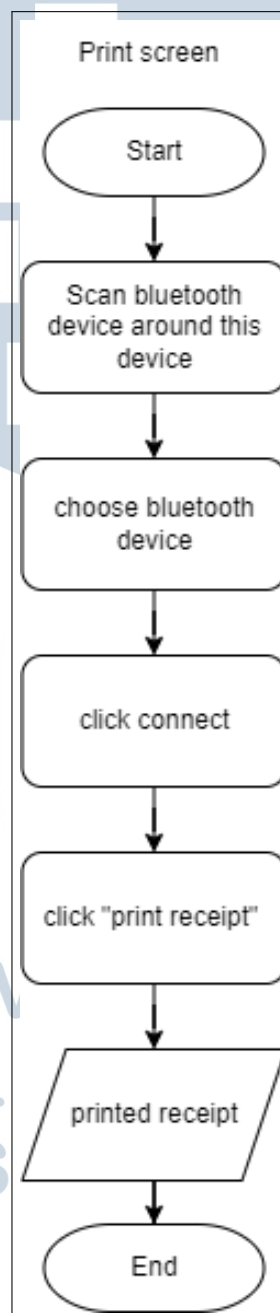
D. **Flowchart Print Screen**

Flowchart pada Gambar 3.8 menggambarkan secara detail proses untuk melakukan pencetakan struk berdasarkan tampilan *print screen*. Proses ini diawali dengan langkah pertama yaitu melakukan pemindaian terhadap perangkat *Bluetooth* yang berada di sekitar perangkat handphone. Kasir kemudian dapat memilih salah satu perangkat yang terdeteksi dan melanjutkan dengan mengklik opsi "connect" untuk membentuk koneksi antara perangkat *Bluetooth* tersebut dan perangkat handphone.

Setelah berhasil terhubung, langkah selanjutnya adalah mengklik tombol cetak struk untuk memulai proses pencetakan. Tahap ini menandai penyelesaian dari seluruh proses *print screen*. Dengan demikian, keseluruhan rangkaian prosedur

tersebut menggambarkan bagaimana struk dapat dicetak melalui koneksi Bluetooth setelah melakukan pemindaian perangkat dan memilih perangkat yang sesuai.

Dengan adanya langkah-langkah tersebut, proses *print screen* menjadi lebih terinci dan memudahkan pengguna untuk mengikuti setiap tahapannya. Keseluruhan prosedur ini dapat diimplementasikan dengan menggunakan gambaran visual yang terdapat pada *Flowchart* Gambar 3.8, memberikan panduan yang jelas bagi pengguna dalam menjalankan fungsi *print screen* pada sistem tersebut.



Gambar 3.8. *Flowchart* Print Screen

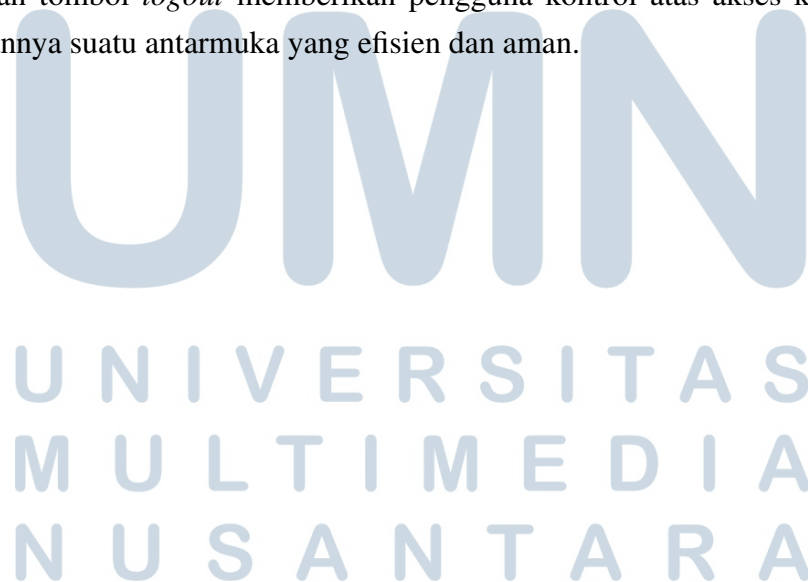
E. Flowchart Profile Screen

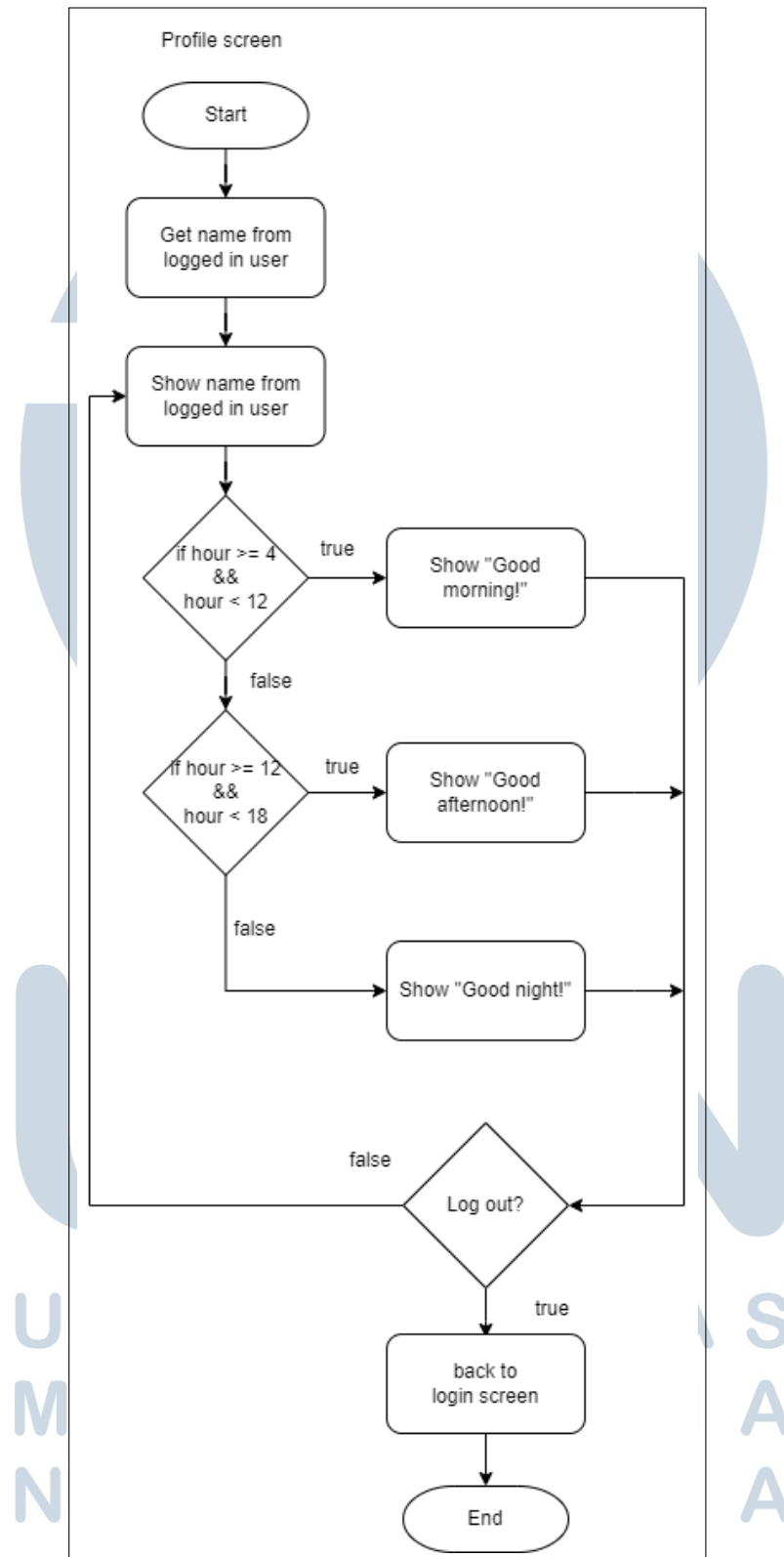
Flowchart pada Gambar 3.9 menggambarkan serangkaian proses yang terjadi pada tampilan *profile screen*. Pertama, informasi nama kasir yang terpampang di *profile screen* diperoleh dari nama akun kasir yang sedang *login*. Ini memastikan bahwa identitas kasir yang terhubung ke sistem adalah yang saat ini aktif.

Selanjutnya, terdapat elemen "*greetings*" yang menampilkan ucapan sapaan berdasarkan waktu lokal perangkat. Seiring dengan perubahan waktu, antarmuka pengguna akan menampilkan pesan selamat yang sesuai, seperti "Selamat Pagi!", "Selamat Siang!", atau "Selamat Malam!". Hal ini memberikan sentuhan personalisasi dan kenyamanan kepada pengguna sepanjang hari.

Proses berlanjut ke tombol *logout*, yang menjadi elemen penting dalam antarmuka. Jika pengguna memilih untuk keluar dengan menekan tombol tersebut, sistem akan meredirect pengguna ke halaman *login*. Proses ini menandai akhir dari sesi pengguna pada *profile screen*, memastikan keamanan dan privasi akun kasir terjaga.

Dengan demikian, *profile screen* tidak hanya menyajikan informasi dasar seperti nama kasir, tetapi juga menciptakan pengalaman pengguna yang ramah dengan menyertakan ucapan sapaan yang sesuai dengan waktu lokal. Selain itu, ketersediaan tombol *logout* memberikan pengguna kontrol atas akses ke sistem, menjadikannya suatu antarmuka yang efisien dan aman.





Gambar 3.9. Flowchart Profile Screen

3.4.3 Pembuatan Aplikasi POS

Setelah melalui serangkaian *meeting* dan diskusi yang mendalam dengan tim Jasa Boga Raya mengenai desain dan fitur yang diinginkan dalam aplikasi, langkah pertama dalam pengerjaan aplikasi pun dimulai. Visual Studio Code digunakan sebagai *text editor* utama untuk menyusun kode, merancang antarmuka, dan mengelola proyek secara efisien. Keunggulan dari Visual Studio Code, termasuk dukungan penuh untuk Flutter dan banyaknya ekstensi yang mempermudah pengembangan, menjadi faktor utama pemilihannya.

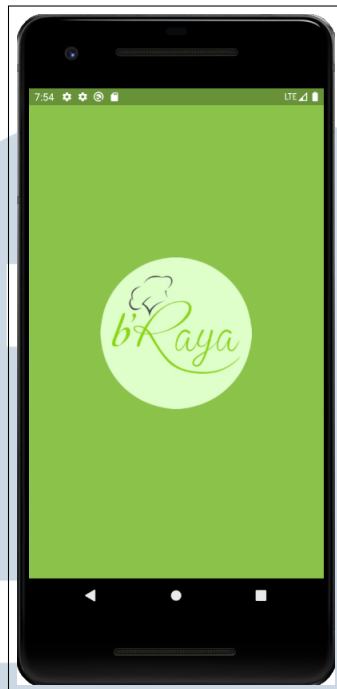
Flutter, sebagai *framework* pengembangan yang digunakan, memberikan keleluasaan dalam merancang antarmuka pengguna yang menarik dan responsif, sambil memastikan keberlanjutan kinerja di berbagai platform, termasuk Android. Keputusan dibuat untuk memberikan pengalaman pengguna yang seragam dan terbaik di berbagai perangkat.

Untuk menguji aplikasi dalam tahap awal, digunakan *emulator* Android. *Emulator* ini memungkinkan untuk mensimulasikan dan melihat sejauh mana aplikasi dapat berinteraksi dengan sistem operasi Android tanpa memerlukan perangkat fisik. Ini menjadi langkah awal yang efektif untuk memastikan bahwa fungsi-fungsi dasar berjalan dengan baik sebelum melakukan pengujian lebih lanjut pada perangkat sebenarnya.

Dengan kombinasi alat dan *framework* yang tepat, tim pengembangan bersemangat untuk memulai perjalanan pengembangan aplikasi POS Jasa Boga Raya, dengan harapan bahwa hasil akhirnya akan mencerminkan visi dan kebutuhan bisnis yang telah dibahas dalam *meeting* dan diskusi sebelumnya.

A. Splash Screen

Saat pengguna membuka aplikasi POS Jasa Boga Raya, pengalaman awal dimulai dengan munculnya *splash screen* yang menampilkan dengan bangga logo khas Jasa Boga Raya, dapat dilihat pada Gambar 3.10. *Splash screen* ini bukan hanya sekadar elemen grafis yang memperlihatkan identitas merek, namun berfungsi sebagai elemen penyambutan yang memberikan kesan pertama kepada pengguna. Selama periode singkat, aplikasi memuat sumber daya dan persiapan awal yang diperlukan untuk memastikan pengalaman pengguna yang lancar dan responsif.



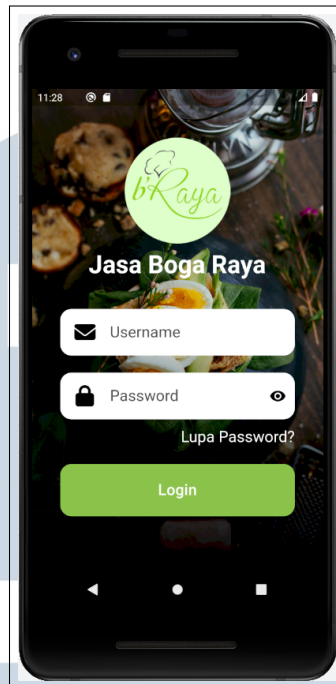
Gambar 3.10. *Splash Screen*

B. Login Screen

Setelah menyelesaikan tampilan *splash screen*, pengguna akan dialihkan ke layar selanjutnya, yaitu *login screen* dapat dilihat pada Gambar 3.11. Pada *login screen* ini, pengguna akan menemui logo Jasa Boga Raya yang kembali memperkuat identitas merek, memberikan kesinambungan visual dari pengalaman awal dengan *splash screen*.

Login screen ini dirancang untuk memberikan pengguna akses ke sistem dengan memasukkan informasi kredensial. Dengan adanya *textfield* khusus, pengguna dapat mengisi nama pengguna dan kata sandi untuk masuk ke dalam aplikasi. Desain yang bersih dan intuitif pada *login screen* bertujuan agar pengguna dapat dengan mudah memahami langkah-langkah yang diperlukan untuk mengakses sistem POS Jasa Boga Raya.

Jika terjadi kegagalan pada proses *login* karena nama pengguna atau kata sandi yang tidak sesuai, *login screen* akan memberikan umpan balik yang jelas dengan menampilkan pesan "Username atau password salah." Hal ini bertujuan untuk memberikan pengguna informasi yang tepat terkait masalahnya dan memfasilitasi upaya pengguna dalam mencapai akses yang sah.



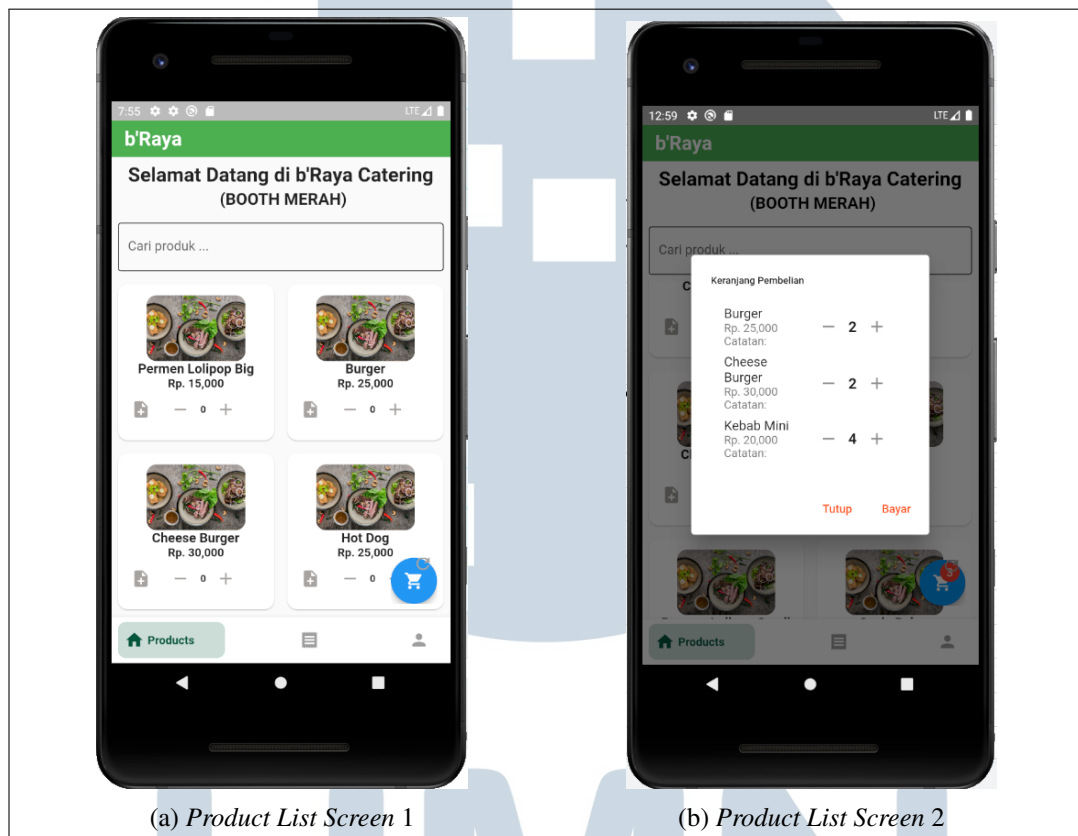
Gambar 3.11. *Login Screen*

C. Product List Screen

Setelah berhasil *login*, pengguna akan diarahkan ke layar *product list screen* yang ada pada Gambar 3.12 yang terdiri dari 2 bagian, bisa dilihat pada Gambar 3.12a, yang menjadi pusat informasi produk pada aplikasi POS Jasa Boga Raya. Sebagai seorang kasir, pengguna akan melihat produk-produk yang tersedia di outlet yang ditugaskan, dan ini diperoleh melalui integrasi dengan RESTful API yang terhubung dengan database. RESTful API (*Representational State Transfer Application Programming Interface*) merupakan cara untuk mengakses data dan fungsionalitas dari suatu sistem melalui internet. RESTful API menggunakan prinsip-prinsip REST, yang membuat API tersebut ringan, skalabel, dan mudah dikelola .

Product list screen dirancang dengan menampilkan produk-produk dalam bentuk gridview dengan dua kolom. Meskipun gambar ilustratif masih bersifat dummy, informasi penting seperti nama menu produk dan harga sudah terintegrasi dengan baik melalui koneksi API. Selain itu, *product list screen* juga mencakup elemen penting lainnya, yaitu indikator jumlah jenis produk yang telah dimasukkan ke dalam keranjang belanja. Fitur ini memudahkan kasir untuk melacak dan mengelola pesanan dengan lebih efisien. Pada bagian Floating Action Button (FAB) bisa dilihat pada Gambar 3.12b, terdapat tampilan ringkas yang menyajikan jumlah

produk yang telah ditambahkan ke dalam keranjang, memberikan pandangan singkat tentang transaksi yang sedang berlangsung. Ketika kasir memilih untuk melanjutkan proses pembayaran, dapat menekan tombol bayar yang akan mengarahkannya ke layar *payment screen*.



Gambar 3.12. Tampilan *Product List Screen*

D. Payment Screen

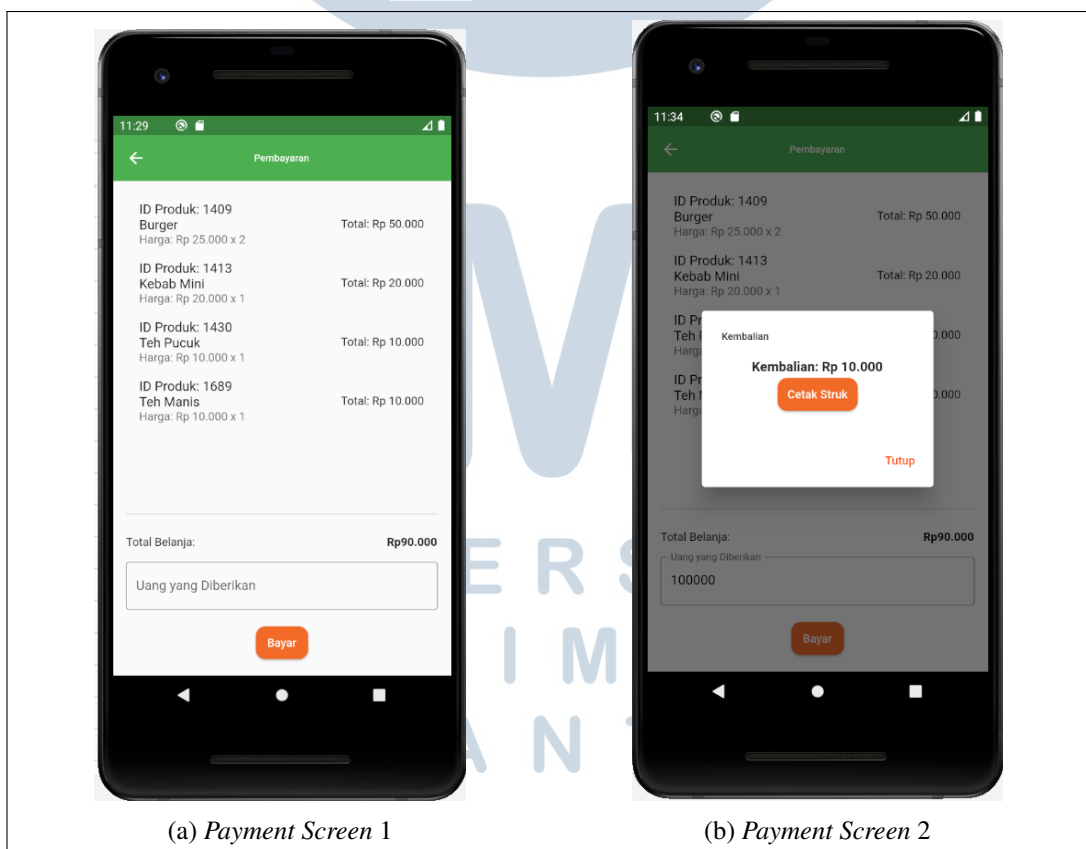
Pada *payment screen* yang diperlihatkan pada Gambar 3.13 terdiri dari 2 bagian pertama, ada pada Gambar 3.13a aplikasi POS Jasa Boga Raya, setelah menampilkan produk-produk yang sebelumnya dimasukkan ke dalam keranjang dari tampilan *product list screen*, pengguna sebagai seorang kasir akan melakukan proses pembayaran. Kasir diminta untuk memasukkan nominal uang yang diterima dari pelanggan. Selanjutnya, sistem akan melakukan perhitungan otomatis untuk mengurangi jumlah uang yang diterima pelanggan dengan total harga produk yang terdaftar dalam keranjang belanja.

Dalam skenario di mana nominal uang yang dimasukkan oleh kasir kurang

dari total belanja, aplikasi akan merespon dengan menampilkan *pop-up* peringatan yang berisi pesan "Uang Kurang". Pesan tersebut memberitahukan kasir bahwa jumlah uang yang dimasukkan kurang dari total belanjaan. Fitur ini bertujuan untuk memberikan peringatan kepada kasir dan memastikan bahwa transaksi selesai dengan benar serta meminimalkan potensi kesalahan.

Sebaliknya, jika jumlah uang yang dimasukkan lebih dari total belanja, *payment screen* akan menampilkan informasi pada Gambar 3.13b mengenai kembalian yang seharusnya diberikan kepada pelanggan. Informasi ini memberikan panduan langsung kepada kasir mengenai jumlah uang kembalian yang harus diberikan, meningkatkan akurasi dan kecepatan dalam menyelesaikan transaksi.

Dengan adanya fitur ini, *payment screen* tidak hanya berfungsi sebagai alat untuk menghitung total belanja, tetapi juga sebagai elemen kunci dalam memastikan transaksi pembayaran dilakukan dengan akurat dan efisien. Fitur peringatan dan informasi kembalian pada layar pembayaran memperkaya pengalaman kasir dan kontribusi terhadap keandalan sistem POS Jasa Boga Raya secara keseluruhan.



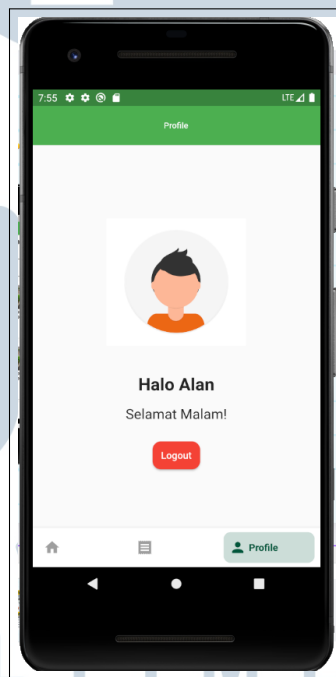
Gambar 3.13. Tampilan *Payment Screen*

E. Profile Screen

Pada *profile screen* yang ada di Gambar 3.14 aplikasi POS Jasa Boga Raya, terdapat fitur informatif yang memuat informasi mengenai nama pengguna yang sedang aktif login. Hal ini memastikan bahwa kasir atau pengguna aplikasi memiliki identifikasi langsung terhadap akun yang sedang digunakan, menciptakan pengalaman pengguna yang personal.

Selain itu, "*greetings*" yang ditampilkan pada layar memberikan sentuhan personalisasi tambahan dengan menyapa pengguna sesuai dengan waktu lokal perangkat. Pesan selamat seperti "Selamat Pagi!", "Selamat Siang!", atau "Selamat Malam!" memberikan nuansa yang ramah dan membuat pengalaman pengguna terasa lebih diperhatikan.

Di bagian bawah layar, terdapat tombol *logout* yang memberikan kemudahan bagi pengguna untuk keluar dari akun yang sedang aktif. Saat tombol ini ditekan, pengguna akan diarahkan kembali ke layar *login*, menandakan berakhirnya sesi pengguna pada aplikasi.



Gambar 3.14. *Profile Screen*

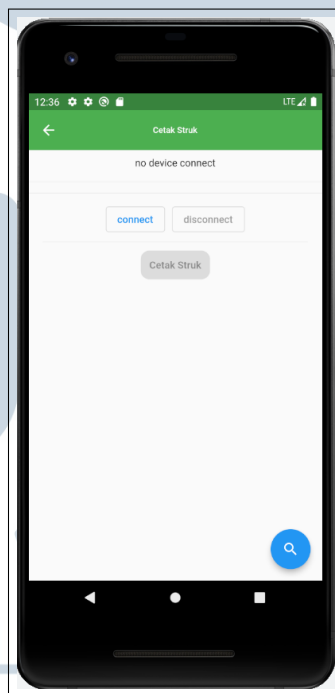
F. Print Screen

Pada bagian *print screen* aplikasi POS Jasa Boga Raya, pengguna diberikan kemampuan untuk melihat perangkat *Bluetooth* yang tersedia di sekitar perangkat

handphone. Langkah ini memungkinkan pengguna untuk memilih perangkat *Bluetooth* yang akan digunakan untuk mencetak struk transaksi.

Setelah memilih perangkat *Bluetooth* yang sesuai, pengguna harus mengklik ”*connect*” lalu dapat dimulai proses pencetakan struk. Struk tersebut dirancang untuk memberikan informasi yang komprehensif terkait transaksi, mencakup total item produk yang dibeli oleh pelanggan beserta jumlahnya, harga masing-masing produk, dan informasi mengenai kembalian yang diberikan.

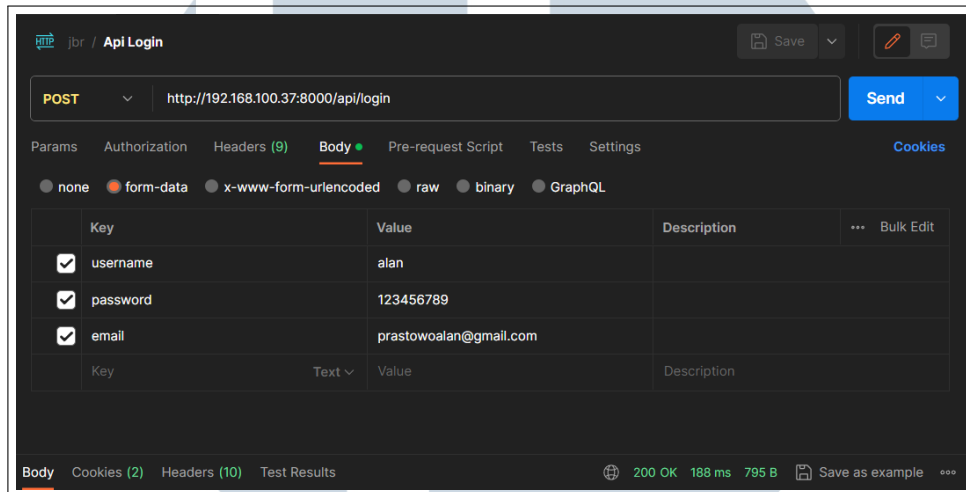
Dengan menggunakan fitur *print screen* ini, Jasa Boga Raya dapat meningkatkan efisiensi dan profesionalitas dalam penanganan transaksi. Kemampuan untuk mencetak struk secara langsung melalui perangkat *Bluetooth* mempercepat proses pembayaran dan memberikan bukti transaksi yang jelas kepada pelanggan. Selain itu, informasi rinci pada struk juga memberikan pelanggan pemahaman yang lebih baik mengenai pembelian, meningkatkan kepercayaan dan kepuasan pelanggan. Implementasi ini menjadikan *print screen* sebagai fitur penting dalam ekosistem aplikasi POS yang memberikan nilai tambah baik untuk pengguna maupun pelanggan Jasa Boga Raya.



Gambar 3.15. *Print Screen*

G. API

Pada Gambar 3.16 merupakan salah satu contoh API yang digunakan pada aplikasi dimana itu merupakan API *Login* dan diperlukan data seperti *username* dan *password* agar bisa sukses saat *login*.



Gambar 3.16. API *Login*

3.5 Kendala dan Solusi yang Ditemukan

Selama peserta magang melaksanakan program magang, terdapat beberapa kendala yang dialami antara lain.

1. Terdapat beberapa fitur yang memerlukan library namun banyak dari library tersebut yang *deprecated*.
2. Kurangnya pemahaman peserta magang terkait proses bisnis dan istilah bisnis pada perusahaan, sehingga terdapat beberapa tugas yang memerlukan waktu untuk dipahami maksud dan tujuannya.

Berdasarkan kendala yang telah disebutkan sebelumnya, berikut merupakan solusi yang diperoleh untuk mengatasi kendala.

1. Harus menguasai lebih lanjut terkait pengembangan aplikasi android dengan menggunakan *framework* flutter, serta melakukan eksplorasi mandiri guna mengetahui library yang akan diperlukan nantinya.
2. Harus aktif bertanya dengan rekan kerja baik dalam satu tim ataupun berbeda tim dalam rangka membantu untuk mengetahui apa yang dibutuhkan oleh perusahaan sehingga solusi yang diberikan dapat bermanfaat dengan baik.