# PERANCANGAN APLIKASI POINT OF SALES

# BERBASIS WEB PADA TOKO DI SHINESHOP MENGGUNAKAN

# GOLANG DAN REACTJS

**SKRIPSI**

Karya Ilmiah Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Komputer (S.Kom.) dari Universitas Advent Indonesia

Disusun Oleh:

**DANIEL REINHARD CAPAH**

NIM : 2081029

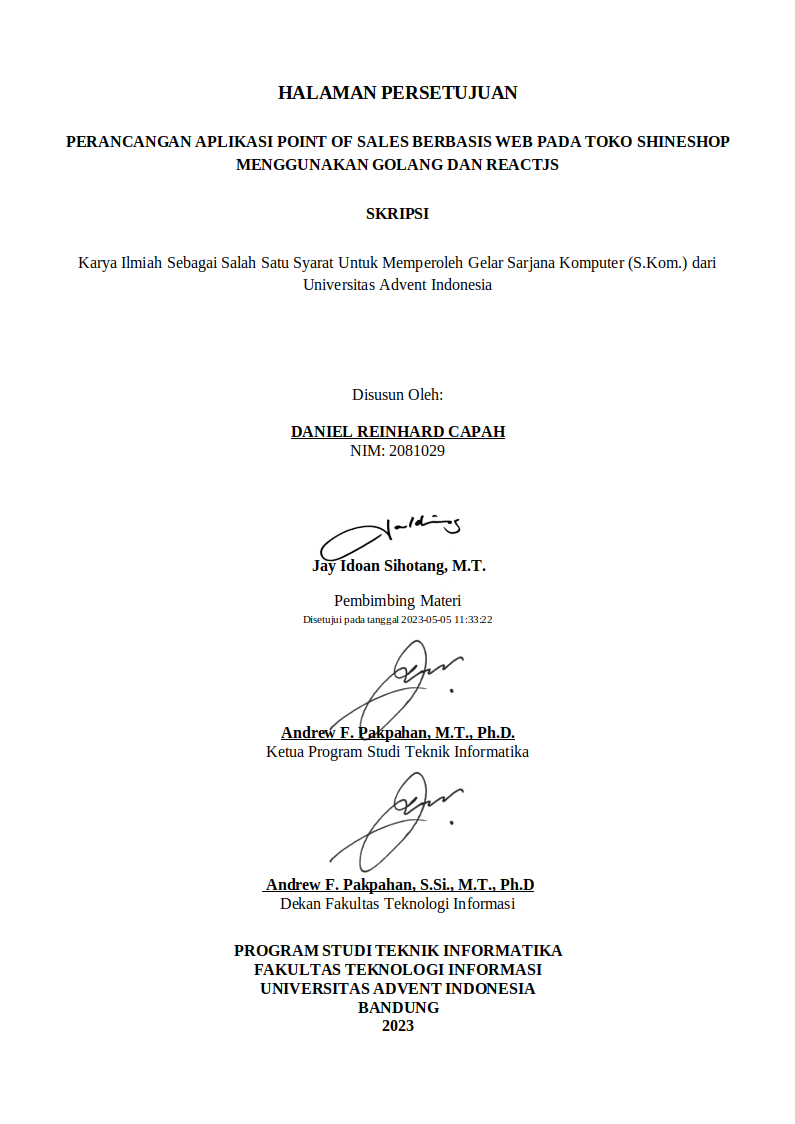
**PROGRAM STUDI TEKNIK INFORMATIKA**

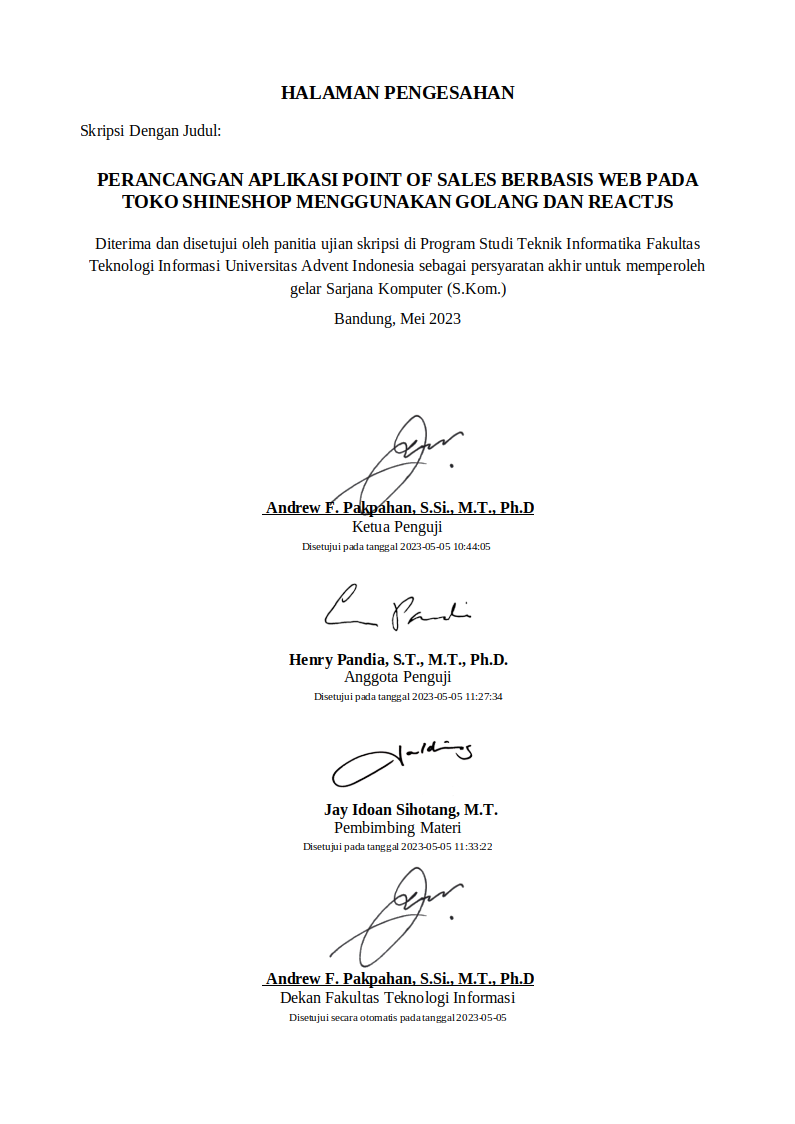
**FAKULTAS TEKNOLOGI INFORMASI**

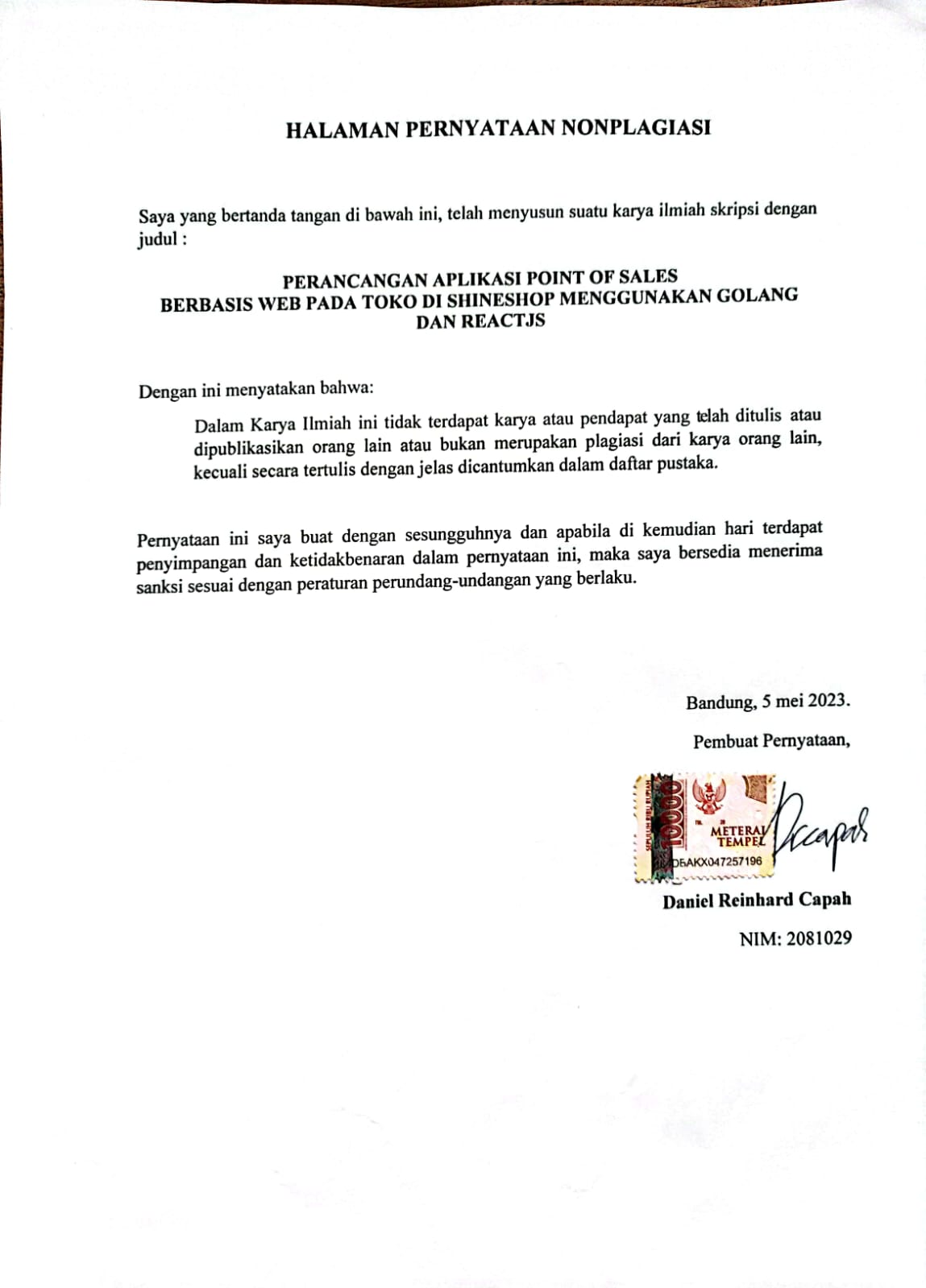
**UNIVERSITAS ADVENT INDONESIA**

**BANDUNG**

**2023**







Perancangan Aplikasi *Point of Sales* Berbasis Web pada Toko di ShineShop Menggunakan Golang dan ReactJS

**Daniel Reinhard Capah1, Jay Idoan Sihotang2**

1,2Fakultas Teknologi Informasi, Universitas Advent Indonesia

e-mail: 12081029@unai.edu, 2jay.sihotang@unai.edu

**Abstrak**

Sandang adalah salah satu kebutuhan primer manusia, oleh karena itu Bisnis dibidang pakaian akan selalu ada .Toko ShineShop adalah salah satu UKM yang bergerak dibidang jual beli batik. Setiap Hari kerja Karyawan di toko ini akan melakukan pencatatan barang masuk dan transaksi dan melaporkannya kepada pemilik toko tersebut dalam bentuk *hardcopy.* Proses pelaporan penjualan dan stok barang yang ada di toko kerap kali tidak akurat dan lama. Tujuan dari penelitian ini adalah untuk membuat sistem informasi *Point of Sales* berbasis web yang dibangun menggunakan bahasa pemrograman Go dibagian *backend* dan ReactJS untuk bagian *frontend* dan dalam pengembangannya menggunakan metode pengembangan *software* *Agile*. Hasil dari penelitian ini adalah sebuah sistem informasi *Point-of-sales* yang dapat membantu pelaporan transaksi dan mendukung proses bisnis di toko tersebut.

**Kata Kunci:** Aplikasi Web, UMKM, *Point Of Sales*

*Development of a Web-Based POS (Point Of Sale)*

# *For ShineShop Using GO Programming Language & ReactJS*

***Abstract***

*Clothing is one of the primary human needs, therefore there will always be a business in the field of clothing. ShineShop is one of the SMEs engaged in buying and selling batik. Every working day the employees at this store will record incoming goods and transactions and report them to the shop owner in hardcopy form. The process of reporting sales and inventory in stores is often inaccurate and long. The results of this research are a Point of Sales Information System that can help report transactions and support business processes at the store.*

***Keywords:*** *Web Applications, SMEs, Point Of Sales*

# Pendahuluan

Dalam era digital yang makin berkembang, teknologi menjadi hal yang sangat penting dalam menjalankan bisnis dan usaha kecil menengah (UKM). Penggunaan teknologi dapat memberikan keuntungan seperti efisiensi, peningkatan produktivitas, dan kemampuan untuk memasarkan produk dan jasa ke pasar yang lebih luas [1]⁠,[2]⁠ . Berikut ini adalah beberapa contoh bagaimana teknologi dapat membantu UKM:

1. Meningkatkan efisiensi proses bisnis dengan penggunaan perangkat lunak yang memudahkan manajemen data dan pengaturan inventaris [3]⁠
2. Meningkatkan pemasaran dengan memanfaatkan media sosial dan platform digital untuk mempromosikan produk dan jasa [1]⁠
3. Meningkatkan aksesibilitas bisnis dengan menyediakan layanan daring seperti *e-commerce*, layanan pengiriman, dan layanan konsumen yang responsif [2]⁠

ShineShop Merupakan salah satu toko retail yang berfokus pada penjualan pakaian-pakaian Batik dalam hal ini ketika karyawan melakukan pencatatan produk masih dilakukan secara manual menggunakan kertas dan pulpen jadi kerap kali ketika pemilik toko melakukan pembelian barang ke *supplier* terjadi redundasi pada barang tertentu dan untuk barang yang lain tidak dibeli serta pemilik toko tidak dapat mengetahui produk mana yang memberi kontribusi terhadap penjualan di toko. Masalah juga terjadi ketika pakaian dengan motif keluaran terbaru masuk, maka perkerja harus membuka barang tersebut dan memajangkannya di patung atau menggantungkannya di *hanger* yang ada di toko tersebut agar menarik perhatian *customer* dan pada akhirnya membeli pakaian tersebut tentunya jika barang yang masuk ada banyak maka pekerja akan kewalahan untuk memajangkan pakaian tersebut dan juga *customer*  harus datang ke toko untuk dapat melihat pakaian tersebut. Permasalah lainnya adalah ketika melakukan pelayanan sistem manual akan memakan waktu yang lama jadi kerap kali pembeli akan pergi ke toko lainnya karena bosan menunggu dan sistem pembayarannya hanya menerima *cash* dan transfer saja, hal tersebut tidak mendukung maraknya penggunaan layanan dompet digital dan QRIS [4]⁠. Terakhir pencatatan transaksi secara manual kerap kali tidak dapat memberikan informasi laba-rugi yang akurat kepada *owner.*

Berdasarkan permasalahan di atas, penulis mengusulkan sistem *Point-of-sale* berbasis web. Adapun kegunakan dari sistem tersebut diharapkan: 1) Memberikan informasi stok barang yang akurat. 2) Membantu owner melakukan *monitoring* terhadap *trend* suatu produk. 3) Membantu mempercepat proses transaksi. 4) Mengelola data customer untuk keperluan promosi produk [1]⁠. 5) Membantu *owner* mengawasi laba-rugi penjualan. 6) Memberikan opsi pembayaran lainnya dengan cara integrasi sistem dengan penyedia *payment gateway* [4]⁠.

Pertanyaan Penelitian:

1.Bagaimana perancangan aplikasi POS dapat memudahkan pengelolaan inventaris dan pengawasan stok barang?

2.Bagaimana perancangan aplikasi POS yang baik dapat meningkatkan efisiensi dan produktivitas bisnis?

3.Bagaimana perancangan aplikasi POS dapat mempertimbangkan kebutuhan bisnis yang semakin kompleks, seperti multi-outlet atau bisnis *e-commerce?*⁠

4.Apa saja fitur-fitur yang dibutuhkan dalam perancangan aplikasi POS yang efektif dan efisien?

5.Apakah penggunaan sistem POS dapat meningkatkan pengalaman pelanggan dan kepuasan pelanggan?

Berikut beberapa penelitian terdahulu yang membantu penulis dalam pembuatan aplikasi point of sale sebagai acuan dalam pemilihan topik. Pertama, penelitian yang dilakukan oleh peneliti [5]⁠ menghasilkan aplikasi Phpdan Mysql dimaksudkan untuk meminimalisir kesalahan dan kehilangan faktur dan memudahkan pemilik untuk untuk melakukan pengecekan stok roti harga produksi, data penjualan, data pembelian dan pengecekan laba perusahaan. Perbedaannya dalam penelitian yang dilakukan menggunakan pendekatan terstruktur sedangkan penulis menggunakan pendekatan Agile*.* Kedua, Penitian yang dilakukan oleh peniliti [6]⁠ mengasilkan Aplikasi yang dapat menyimpan data transaksi yang dapat di unduh oleh pemilik dalam format pdf menggunakan bahasa pemograman Php. perbedaan Penulis menggunakan bahasa Go dan Javascript dalam proses pembangan. Terakhir penelitian yang dilakukan oleh peneliti [7]⁠, menghasilkan aplikasi *Point of sale* berbasis Android menggunakan *framework* Flutternamun belum terintegrasi ke  untuk mencetak *invoice.* Perbedaan penulis membuat aplikasi berbasis web dan dapat dengan mudah untuk mencetak *invoice*.

# Tinjauan Pustaka

***Point of Sale* (POS)**

*Point of Sales* merupakan suatu sistem yang digunakan sebagai transaksi, yang didalamnya terdapat juga menggunakan mesin kasir. Dalam sistem POS, sebuah sistem kasir tidak berdiri sendiri melainkan didalamnya terdapat perangkat lunak lainnya. Sistem POS melakukan lebih dari sekedar transaksi jual beli, namun didalamnya terdapat perhitungan akuntansi, manajemen barang dan stok, modul penggajian karyawan , perhitungan hutang piutang, dan berbagai macam fungsi lainya [8]⁠⁠[9]⁠.

**Golang**

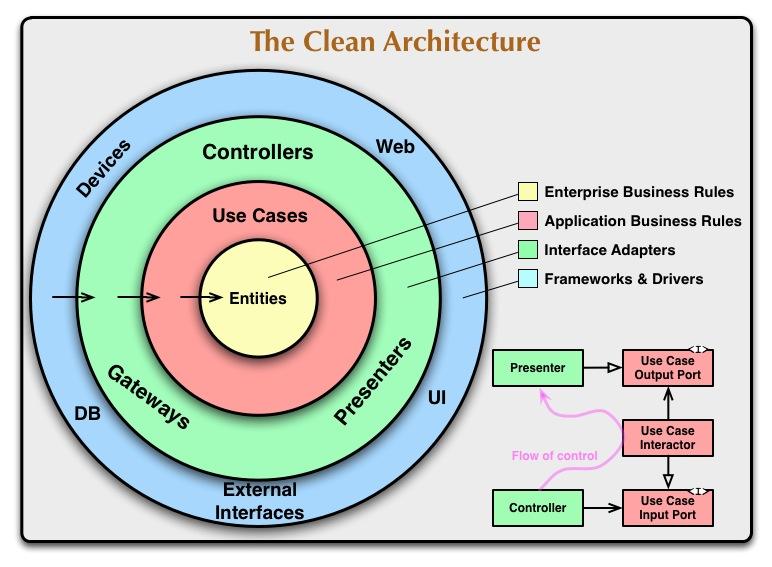
Golang adalah bahasa pemograman yang diciptakan oleh Google bersama dengan Ken Thompson, Robert Griesemer, dan Rob Pike pada tahun 2009. Tujuan dari pengembangannya adalah untuk membangun bahasa yang mempunyai keunggulan dari sisi kecepatan, keandalan, skalabilitas, dan kesederhanaan. Golang juga termasuk dalam bahasa yang dapat diketik secara statis serta menghasilkan kode biner pada mesin yang dapat dikompilasi [10]⁠. Selain itu, Golang juga dihimpun dari bahasa pemrograman C di abad ke–21. Bahasa Go juga dapat digunakan untuk kepentingan pembuatan aplikasi, website, dan software yang lainnya [11]⁠

**ReactJS**

ReactJS adalah f*ront-end library open source* yang dikembangkan oleh Facebook yang berfokus untuk menangani tampilan. ReactJS digunakan sebagai pendukung dari web-framework. ReactJS memiliki beberapa keunggulan diantaranya memberikan kecepatan, *simplicity*, dan *scalability* [12], [13]⁠⁠. ReactJS memungkinkan pengembang dapat membangun sebuah komponen UI yang lebih interaktif, *stateful*, *reusable* dan tak kalah penting dukungan komunitas Javas*cript* terbesar .

***Clean Architecture***

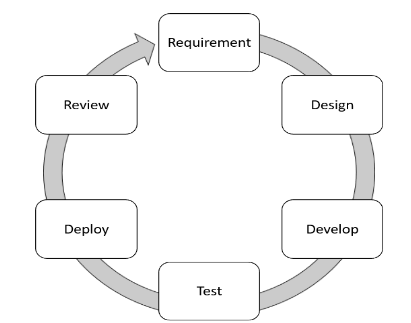
*Clean Architecture* yang ditunjukkan pada gambar 1 merupakan salah satu arsitektur pemrograman ini yang akan membagi sebuah sistem menjadi beberapa komponen, sehingga lebih tertata dan terlihat jelas tugas dari masing-masing komponennya [14]⁠. Penggunaan *Clean Architecture* sendiri dikarenakan setiap komponen bersifat independen, dimana perubahan yang terjadi pada komponen satu tidak akan memberikan dampak pada komponen lainnya. Kunci penting dari arsitektur ini, pengembang sistem harus mengingat betapa pentingnya aturan dependency agar tiap komponen benar mengerjakan tugas mereka masing-masing [14]⁠.

Memang saat mendevelop aplikasi ini penulis memakan waktu yang lebih relatif lebih lama dari biasanya namun, dikarenakan menggunakan arsitektur ini penulis dapat dengan mudah mengisolasi *error* dan menemukan *bug* dari aplikasi sehingga bisa dikatakan kodenya lebih *maintain**able.*

Gambar 1 *Clean Architecture* [14]⁠

# Metode Penelitian

# Konsep *Agile Development* menjadi metode yang digunakan penulis untuk penelitian perancangan sistem ini, inti dari metode ini agar pengerjaan dari suatu sistem dapat ditinjau dan dilakukan revisi terhadap pengembangan sistem informasi [15] 16]⁠.Pendekatan Agile Develop memberikan tingkat keberhasilan pengembangan proyek yang lebih baik dibandingkan dengan metode desain terstruktur dikarenakan akan di lakukan iterasi proses terus menurus terhadapa proses sampai seluruh requierement *client* terpenuhi [17], [18]⁠.Konsep ini menyediakan lima tahapan dalam pengerjaannya (Gambar 2), yaitu sebagai berikut:Gambar 2 Konsep Agile Development⁠

Gambar 2 Konsep Agile Development

Berikut penjelasan tentang konsep *Agile Development* yang digunakan dalam penulisan ini.

1. *Requirement*

Pada Tahap ini penulis mencoba memahami / mengumpulkan informasi bagaimana pengguna sangat membutuhkan sistem *Point of Sale* dengan cara datang ke toko mewawancarai pemilik toko dan mengamati transaksi yang ada di toko tersebut agar mendapat data yang lengkap mengenai kebutuhan pengguna akan sistem yang akan dikembangkan.

1. *Design*

Selanjutnya pada tahapan ini akan merancang suatu desain sesuai dengan data yang telah diperoleh dengan tujuan memberikan gambaran lengkap mengenai apa yang harus dikerjakan.

1. *Development*

Pada tahapan ini penulis menggunakan kode pemrograman untuk menerjemahkan rancangan dan analisis sistem ke dalam bahasa pemrograman tertentu yang dimengerti oleh komputer. Sebuah kode yang dibentuk akan menghasilkan sebuah sistem informasi yang sesuai dengan perintah perancangan serta menguji fungsionalitas sistem tersebut.

1. *Testing*

Setelah sebuah sistem berhasil dikembangkan, maka tahapan selanjutnya adalah melakukan verifikasi dalam bentuk pemeriksaan dan pengujian sistem secara keseluruhan untuk mengidentifikasi kemungkinan adanya kegagalan atau kesalahan.

1. *Deployment*

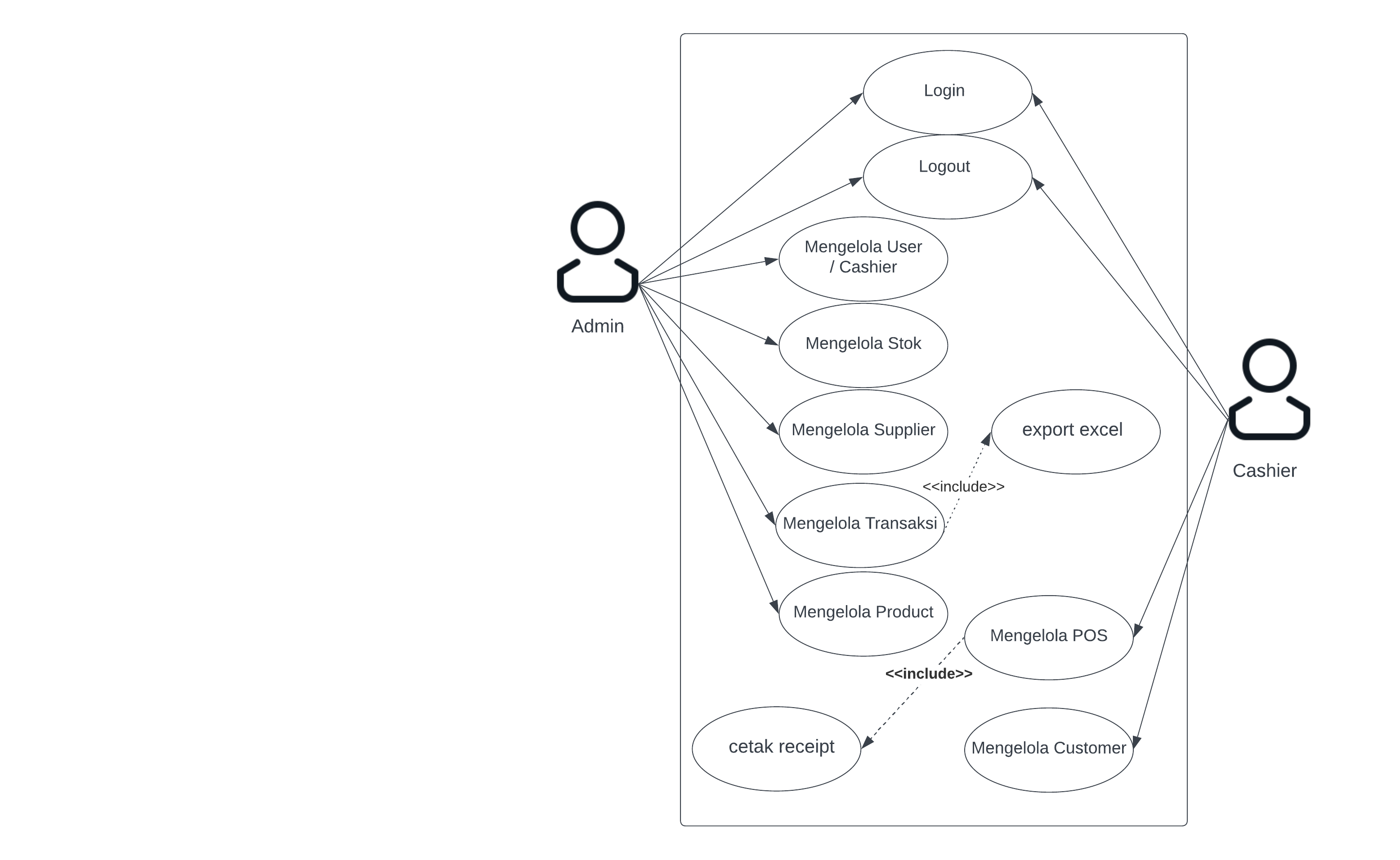
Tahapan ini merupakan tahapan yang bertujuan untuk melakukan penyebaran terhadap aplikasi yang dikerjakan oleh pengembang.

1. *Review*

Tahapan terakhir konsep *Agile Development*, yaitu tahapan untuk melakukan pengecekan terhadap respon dari *user* yang menggunakan aplikasi.

***Use Case Diagram***

*Use case diagram* merupakan sebuah tahapan pengembangan untuk melihat proses yang terjadi didalam sistem . Model sistem yang digambarkan oleh use case diagram fitur-fitur apa saja yang akan dibuat di sistem tersebut[19]⁠. Pada sistem yang akan di bangun hanya terdapat satu user saja yaitu admin dan kasir



Gambar 3 *Use Case Diagram*

Berdasarkan gambar di atas, penulis akan membuat sebuah aplikasi yang memiliki fitur-fitur sebagai berikut:

1. Sebagai Admin:

a. Fitur *login* untuk bisa mengakses layanan dari aplikasi mewajibkan untuk autentikasi

b. Fitur *user* untuk mengatur seluruh kasir yang ada ditoko yang memiliki akses ke POS

c. Fitur mengelola Stok dari produk

d. Fitur mengelola transaksi untuk dapat memonitor transaksi dari produk dan dapat juga untuk mengekport data tersebut ke bentuk file excel [20]⁠

e. Fitur supplier untuk dapat mengelola data dari pemasok product

f. Admin juga dapat mengelol product.

g. Fitur *Logout* untuk keluar dari aplikasi.

2. Sebagai *Cashier*:

a. Fitur *login* untuk bisa mengakses layanan dari aplikasi mewajibkan untuk autentikasi

b. Fitur mengelola *Point of Sale*  yaitu fitur kasirnya

c. Fitur mengelola *customer*

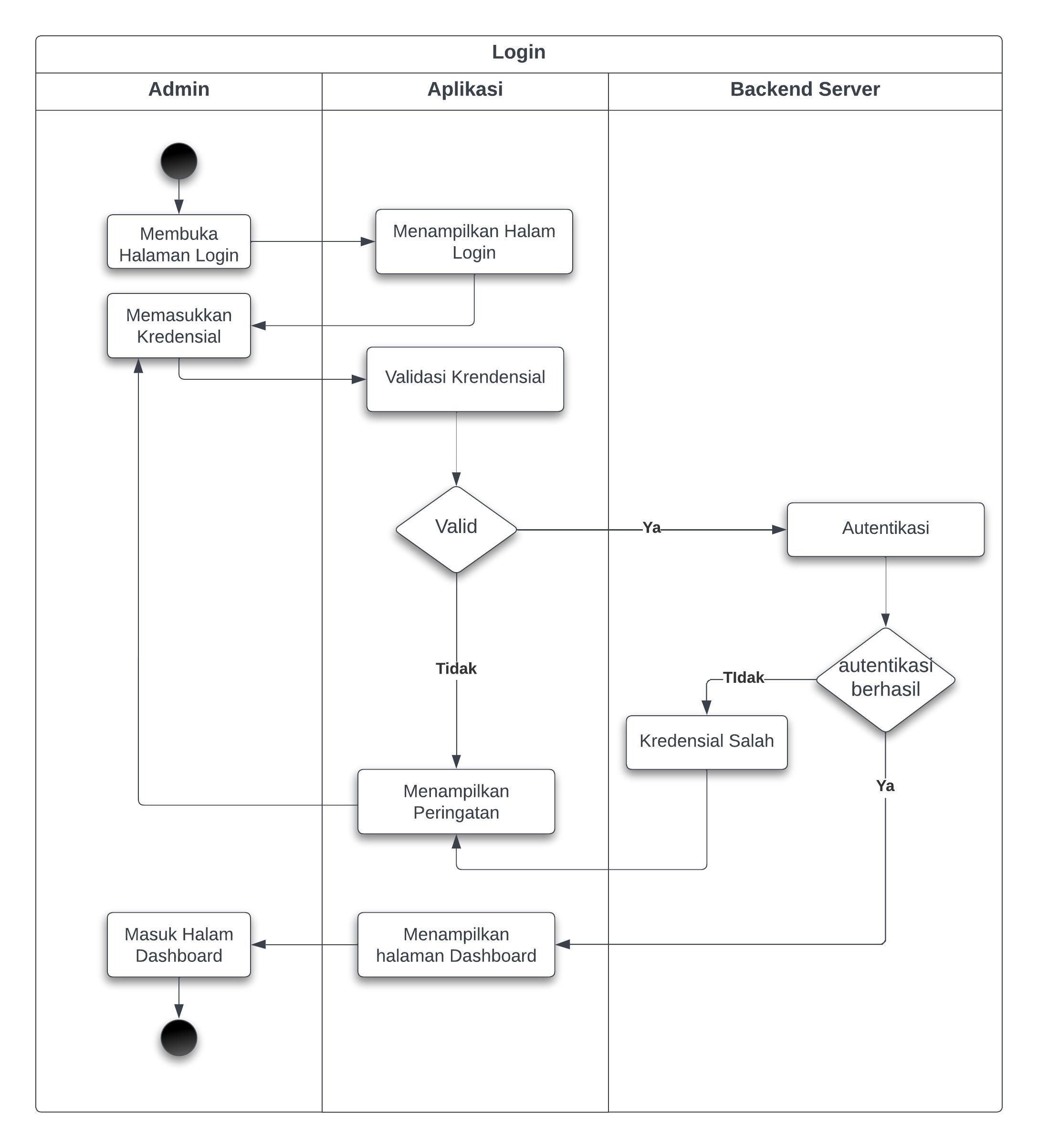
d. Fitur mengelola data *product*

e. Fitur *logout*

***Activity Diagram***

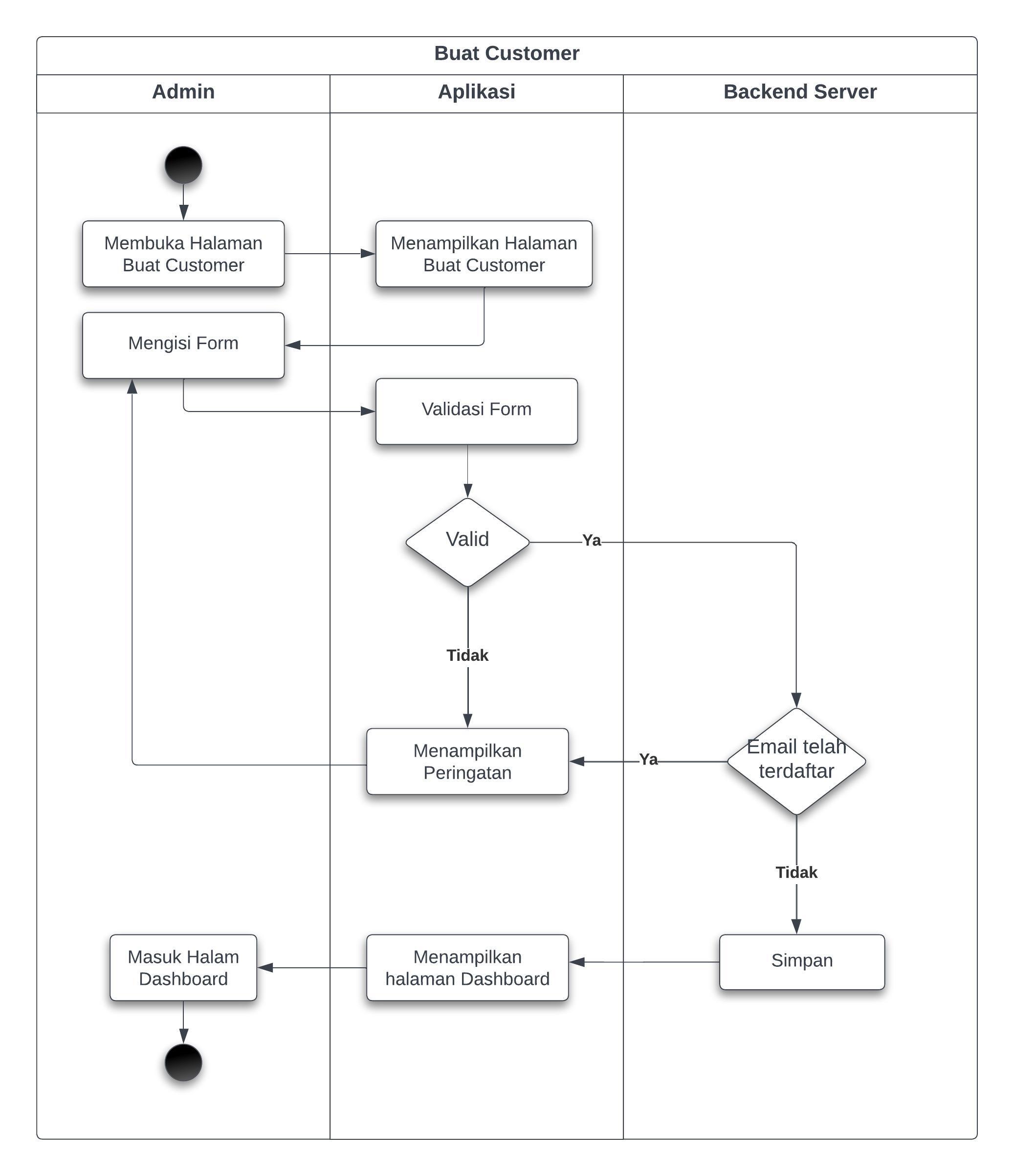
*Activity Diagram* merupakan diagram yang menunjukkan aktivitas sistem berupa bentuk kumpulan aksi-aksi, bagaimana masing-masing aksi tersebut dimulai hingga berakhir. Pada sistem ini terdapat beberapa activity diagram yaitu login, tambah product, tambah transaksi, cetak transaksi

Gambar 4 menjelaskan hal yang harus dilakukan pertama-tama sebelum pegawai atau admin hendak masuk ke dalam sistem yaitu login. Untuk login ke dalam sistem, user diminta untuk memasukkan kredensial yakni email dan password pada input form yang telah disediakan. Setelah user memasukkan kredensial, sistem akan melakukan validasi apakah kredensial yang telah diinput sudah valid. Apakah kredensial sudah valid, sistem akan mengirimkannya kepada back-end. Back-end akan melakukan autentikasi terhadap kredensial user. Apabila autentikasi berhasil maka sistem akan menampilkan halaman dashboard user. Apabila autentikasi tidak berhasil maka sistem akan menampilkan pesan error dimana terdapat 2 kemungkinan pesan yaitu user memasukkan kredensial yang salah atau user masuk pada halaman yang salah.



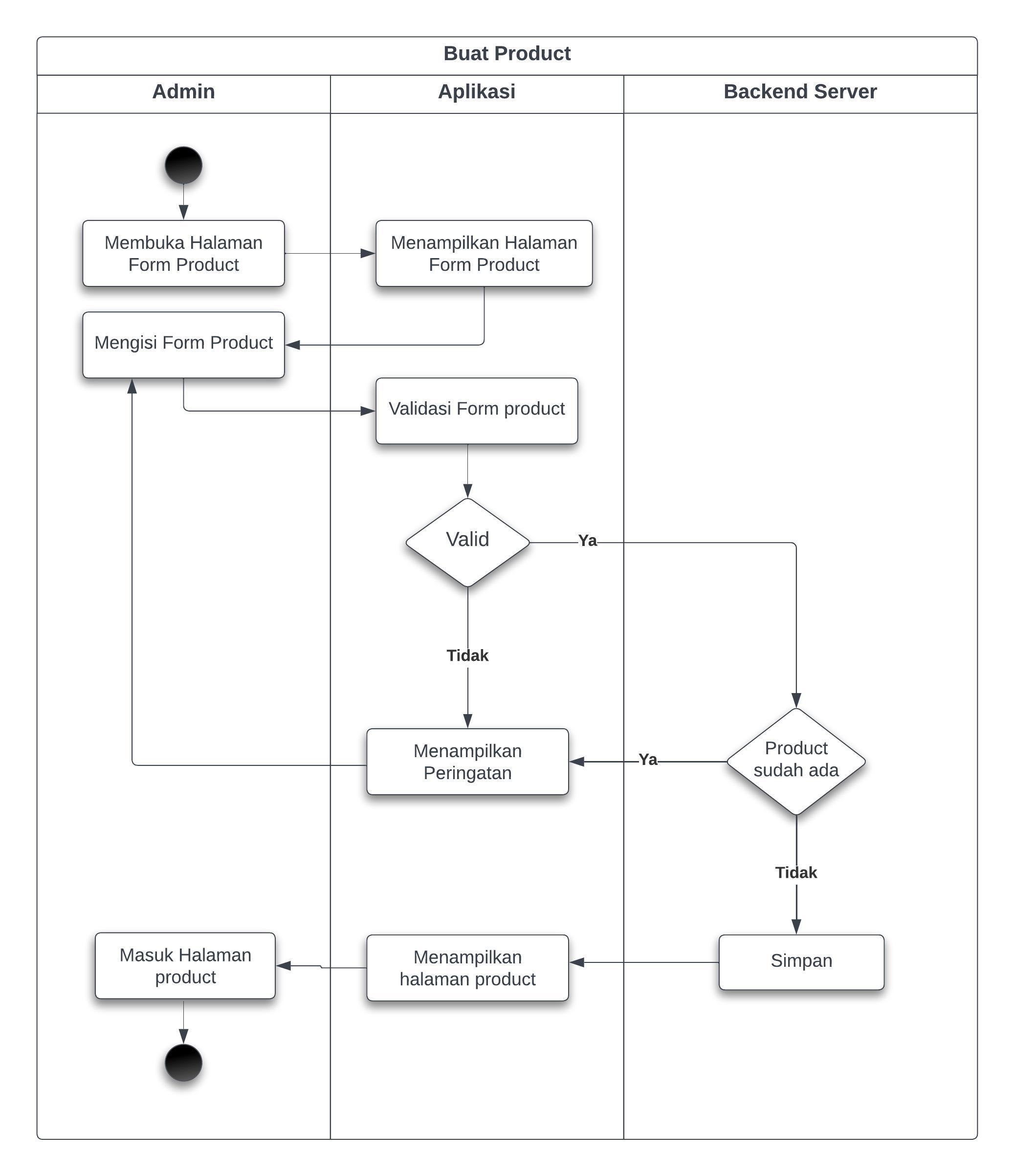
Gambar 4 *Activity Diagram login*

Gambar 5 menjelaskan gambaran admin saat membuat *customer* baru. Untuk membuat *customer* pertama-tama admin akan membuka halaman buat *customer*. Setelah aplikasi menampilkan halaman buat customer, admin akan mengisi kredensial pada form di halaman tersebut. Kemudian form tersebut akan divalidasi oleh aplikasi. Apabila form tidak valid, maka aplikasi meminta admin untuk memperbaiki kredensial yang telah dimasukkan. Namun apabila form valid, aplikasi akan mengirimkannya kepada back-end. Sebelum backend menyimpan kredensial tersebut sebagai customer baru, backend memeriksa apakah email yang ada pada kredensial sudah terdaftar atau belum. Apabila sudah terdaftar, back-end akan menolak untuk membuat customer baru. Namun apabila email belum terdaftar,maka back-end akan menyimpan kredensial tersebut. Kemudian aplikasi akan membawa admin ke halaman customer.



Gambar 5 *Activity Diagram*  *Buat Customer*

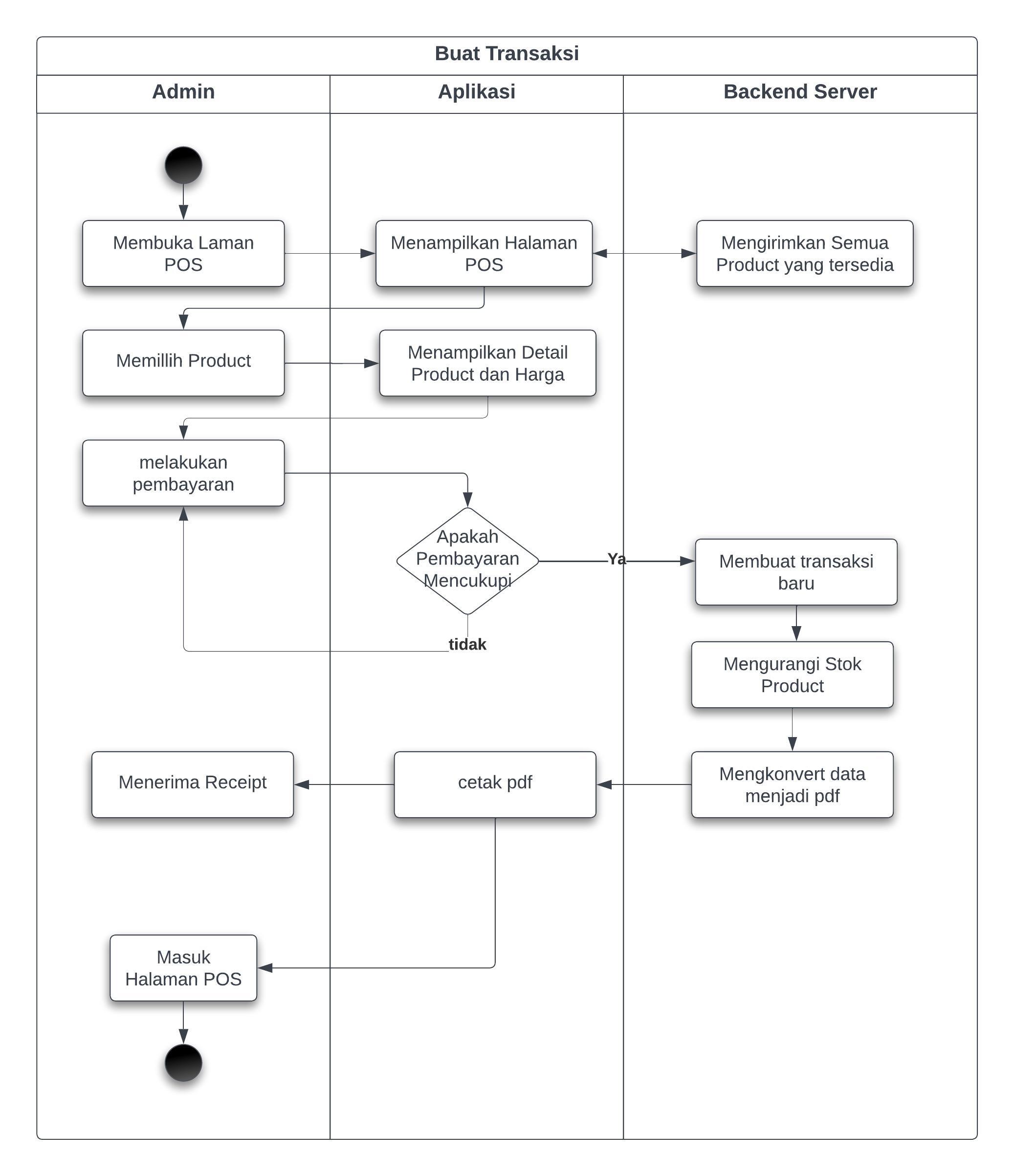
Gambar 6 menjelaskan gambaran admin saat membuat *product* baru. Untuk membuat *product* baru pertama-tama admin akan membuka halaman buat Product. Setelah aplikasi menampilkan halaman buat product, admin akan mengisi keterang product pada form di halaman tersebut. Kemudian form tersebut akan divalidasi oleh aplikasi. Apabila form tidak valid, maka aplikasi meminta admin untuk memperbaiki isian form yang telah dimasukkan. Namun apabila form valid, aplikasi akan mengirimkannya kepada back-end. Sebelum backend menyimpan data tersebut sebagai product baru, backend memeriksa apakah product sudah pernah dibuat atau belum. Apabila sudah pernah dibuat, backend akan menolak untuk membuat product baru. Namun apabila product belum terdaftar,maka back-end akan menyimpan product tersebut. Kemudian aplikasi akan membawa admin ke halaman product.



Gambar 6 *Activity Diagram* Buat *product*

Gambar 7 menjelaskan gambaran admin saat membuat Transaksi baru. Untuk membuat transaksi baru pertama-tama admin akan membuka halaman POS. Setelah aplikasi menampilkan POS aplikasi akan meminta ke backend list product-product yang tersedia lalu aplikasi akan menampilkannya di halaman POS, Admin akan memilih produk , dan kuantitas dari product setelah itu aplikasi akan menampilkan detail product dan nominal harga dari belanjaan tersebut. Selanjutnya pada bagian pembayaran .Apabila nominal saldo tidak mencukupi, maka aplikasi kembali ke pembayaran. Namun apabila saldo mencukupi, aplikasi akan mengirimkannya kepada back-end.

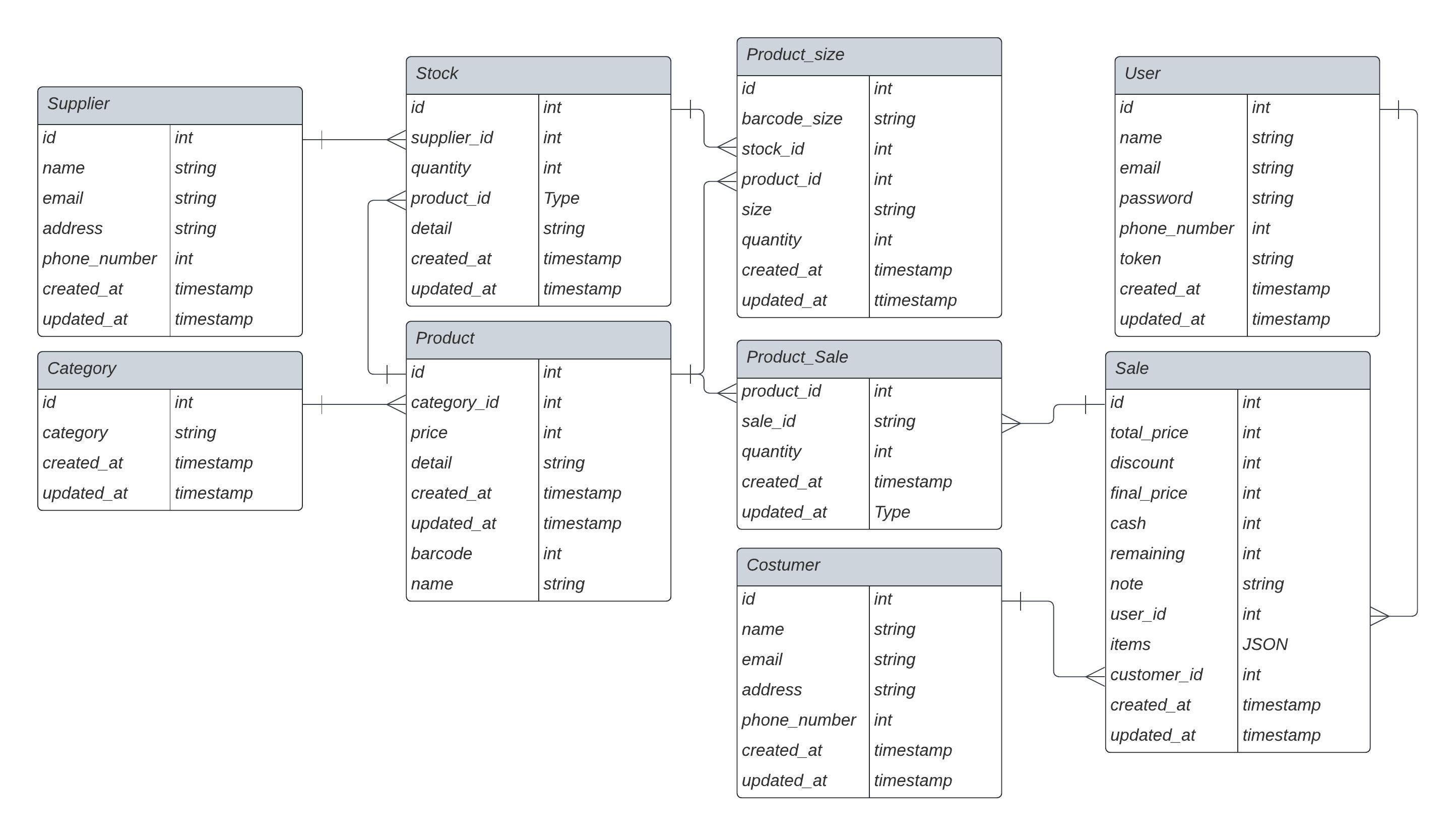
Backend akan dilakukan *query* untuk membuat transaksi baru lalu mengurangi stok dari *product-product* yang dibeli oleh user dan kemudian data tadi akan di *convert*  menjadi data pdf dan aplikasi akan melakukan pencetak *receipt*  dari transaksi tersebut . Terakhir aplikasi akan mengalihkan kembali ke halaman POS.



Gambar 7 *Activity Diagram* Buat Transaksi

***Class Diagram***

*Class diagram* adalah rancangan dasar untuk membuat *database*. Pada gambar 8 dapat dilihat *class diagram* yang digunakan dalam sistem *Point of sale*. Masing-masing *class* merupakan model yang digunakan untuk melambangkan tabel yang digunakan dalam database dan mempunyai data yang berkaitan dengan nama kelasnya

****

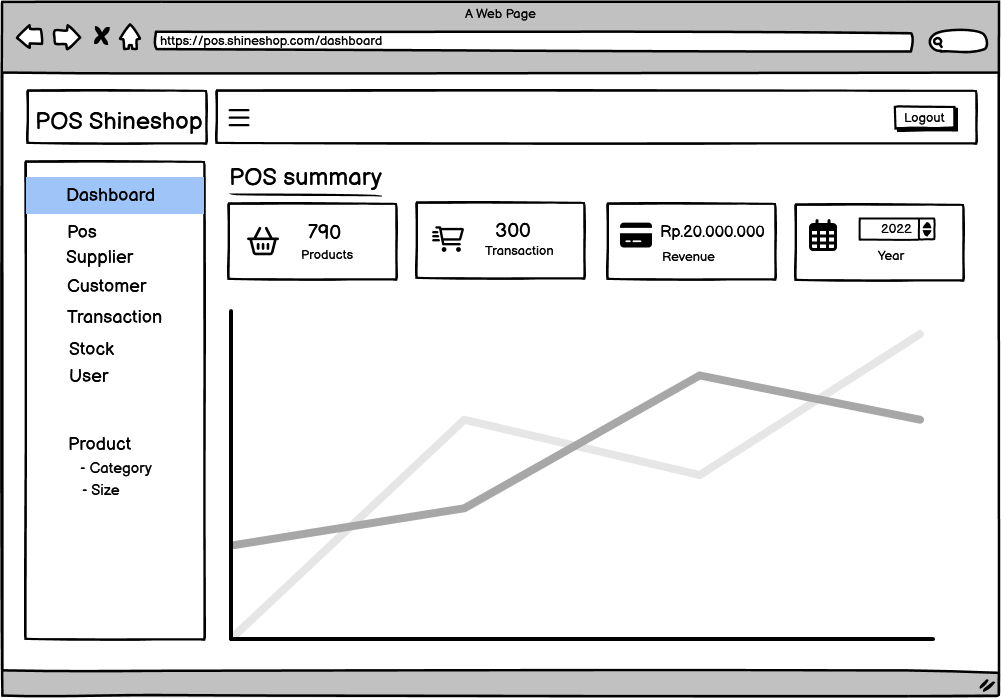
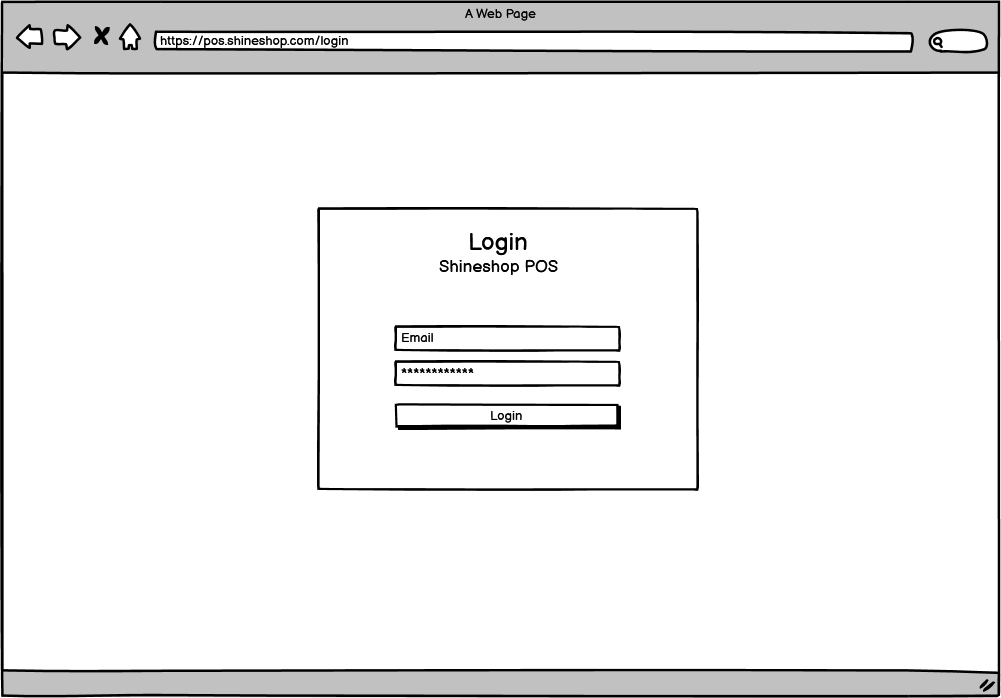
Gambar 8 *Class Diagram*

***Mockup Design***

Perancangan desain ini dibuat untuk memudahkan serta sebagai acuan dalam melaksanakanpengembangan desain *interface* sistem. Perancangan *interface* sistem ini menggunakan aplikasi Balsamiq Wireframes. Berikut dibawah ini merupakan hasil rancangan desain *interface* sistem *Point of Sale*.

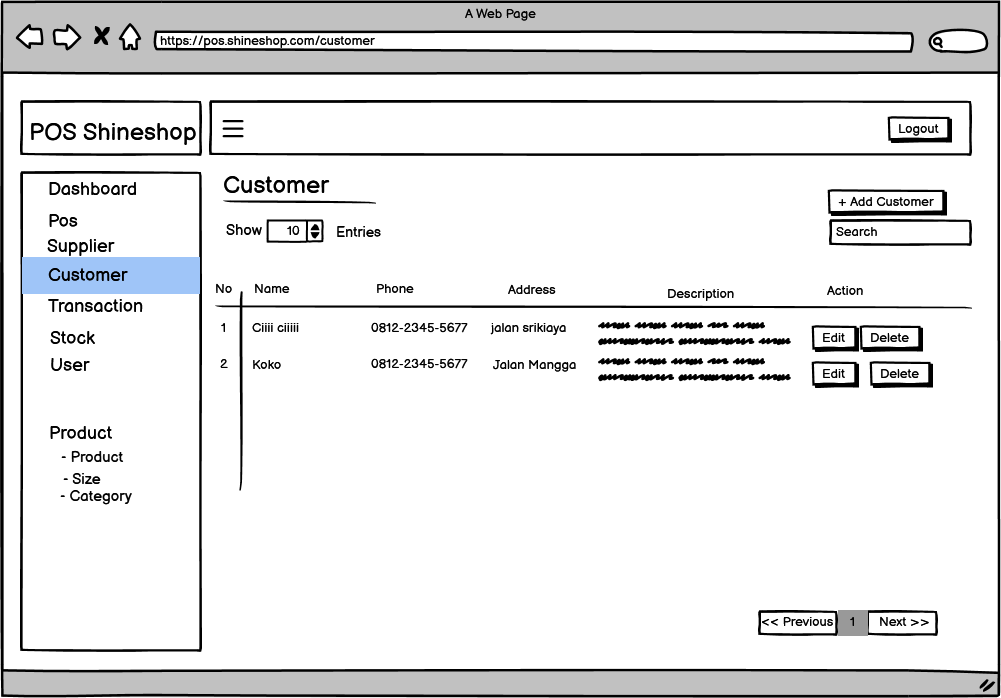
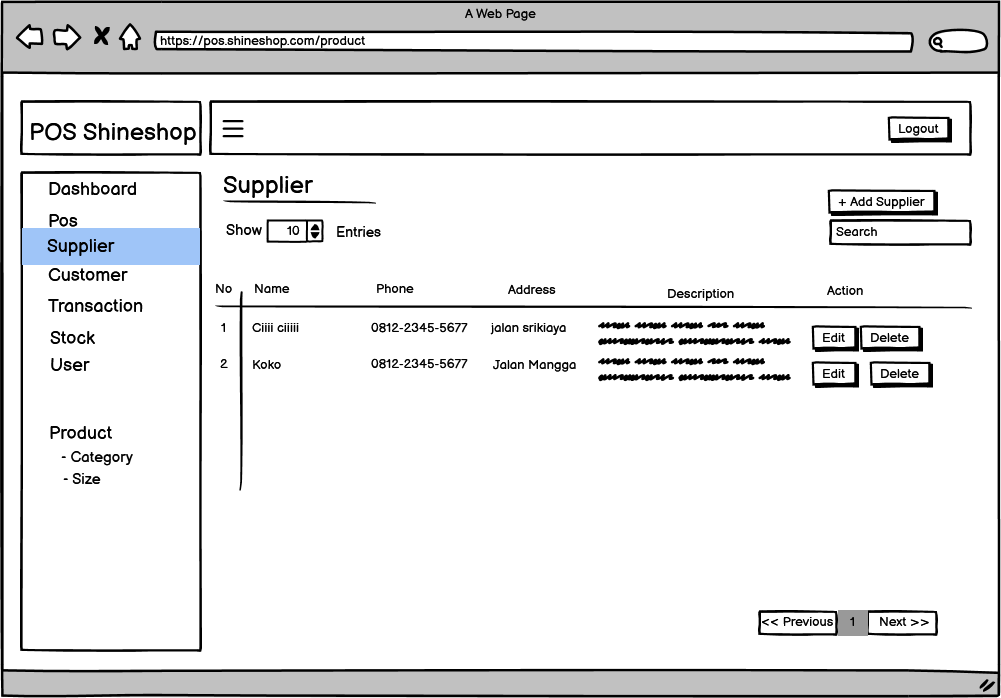
Gambar 9 menunjukkan hasil rancangan antarmuka pengguna untuk halaman login pegawai. Pada halaman ini terdapat 2 input diantaranya adalah input untuk memasukkan email dan input untuk memasukkan password. Selain itu terdapat tombol yang berfungsi untuk melakukan proses autentikasi.

Gambar 10 menunjukkan hasil rancangan antarmuka pengguna untuk halaman *dashboard* pegawai. Pada rancangan antar muka tersebut terdapat ringkasan dari 1) Total jumlah *product* yang ada di dalam aplikasi *Point of Sale* tersebut 2) Total transaksi yang telah berlangsung 3) Total uang masuk yang didapat dalam penggunaan aplikasi tersebut dan 4) pegawai / admin dapat meninjau hasil rekap transaksi, *product* dan *revenue* dalam periode 1 tahun



Gambar 9 Halaman *Login* Gambar 10 *Halaman Dashboard*

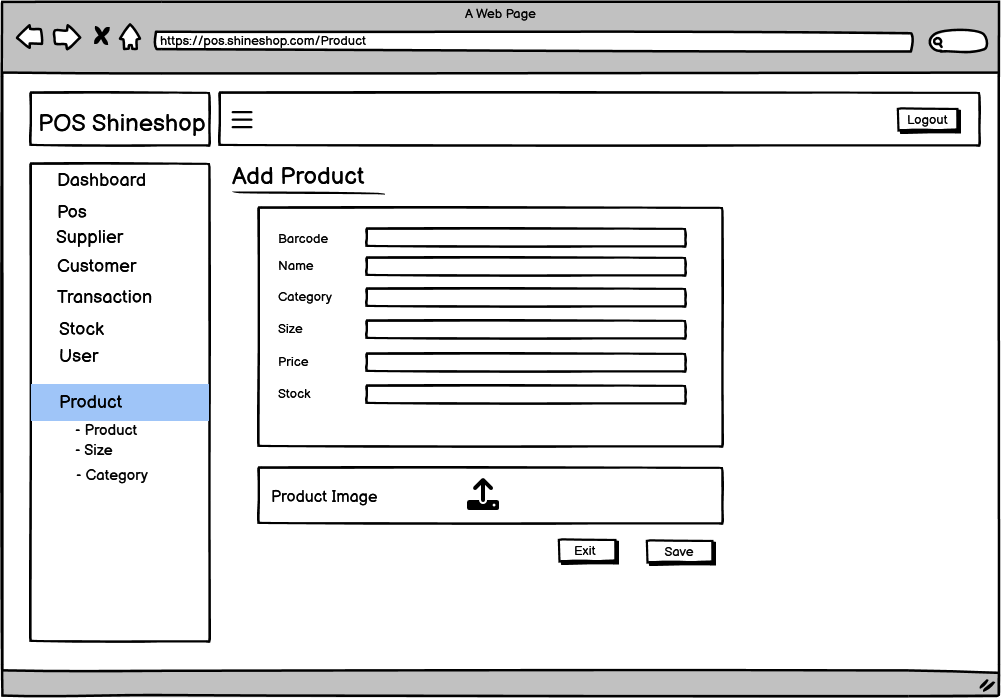
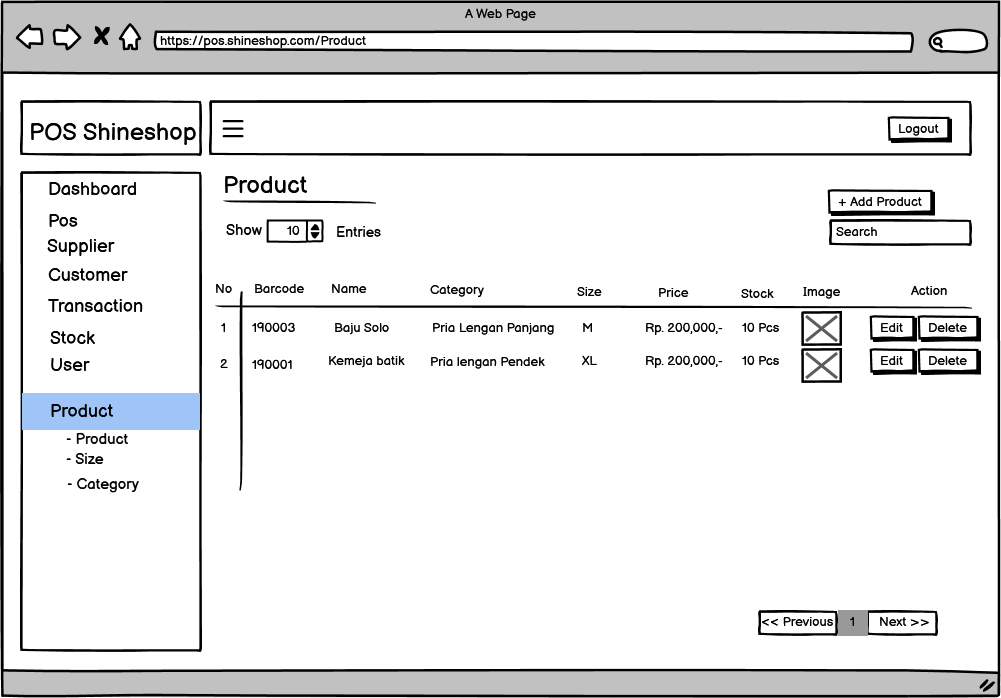
Gambar 11 menunjukkan hasil rancangan antarmuka pengguna untuk halaman *dashboard supplier*. Pada rancangan antar muka tersebut terdapat tabel yang memuat tentang informasi supplier yang menyuplai stok pakaian yang ada di toko tersebut, terdapat beberapa fitur yaitu fitur, pencarian , fitur *pagination* , fitur filter dan fitur-fitur CRUD pada umumnya.

Gambar 12 menunjukkan hasil rancangan antarmuka pengguna untuk halaman *dashboard customer*. Pada rancangan antar muka tersebut terdapat tabel yang memuat tentang informasi *customer* yang membeli pakaian pada toko tersebut. Tujuan utama dibuatnya fitur yang berhubungan dengan customer ini adalah agar si pemilik toko dapat mempertimbangkan fitur berlangganan nantinya yang mengumpulkan poin dan mendapatkan promo-promo tertentu.

Gambar 11 *Halaman Supplier* Gambar 12 *Halaman Customer*

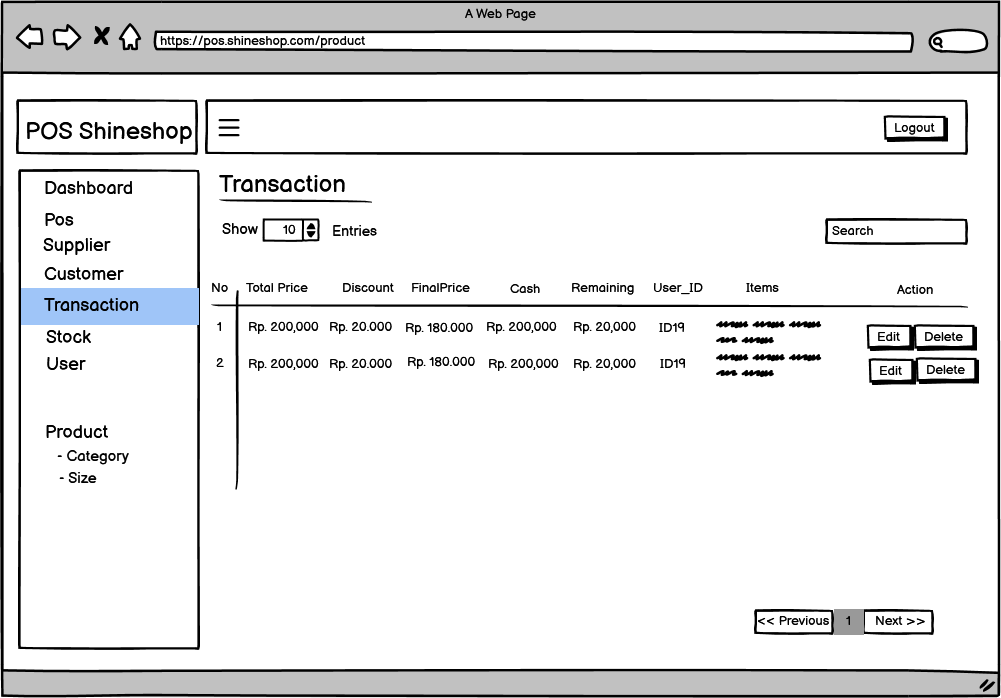
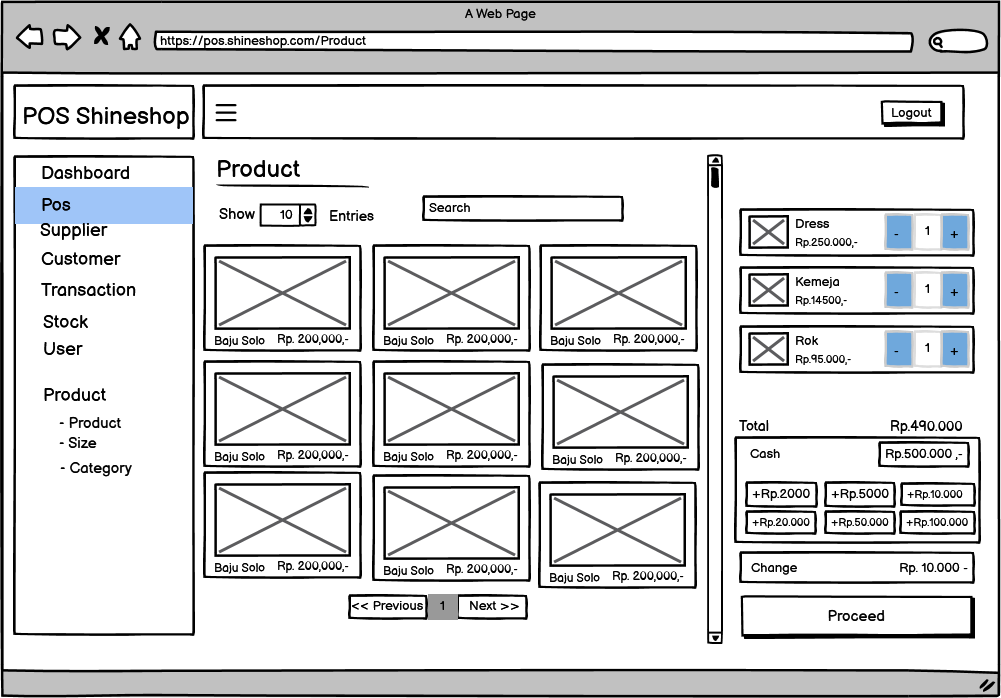
Gambar 13 menunjukkan hasil rancangan antarmuka pengguna untuk halaman *dashboard product*. Pada rancangan antar muka tersebut terdapat tabel yang memuat tentang informasi detail dari *product-product* yang tersedia di toko tersebut Memang informasi tentang barang nantinya juga akan ditampilkan di halaman POS namun di halaman tersebut hanya akan memberikan informasi tentang nama dan harga barang saja.

Gambar 14 menunjukkan hasil rancangan antarmuka pengguna untuk halaman *dashboard add Product*. Pada rancangan antar muka tersebut terdapat *field-field*  yang wajib diisi oleh user yang memuat informasi dari produk tersebut, mulai *barcode* produk, nama produk, *category, size,* harga, *stock* dan yang tidak kalah penting user harus mengupload gambar dari produk tersebut.

Gambar 13 Halaman *Product* Gambar 14.Halaman *Add* *Product*

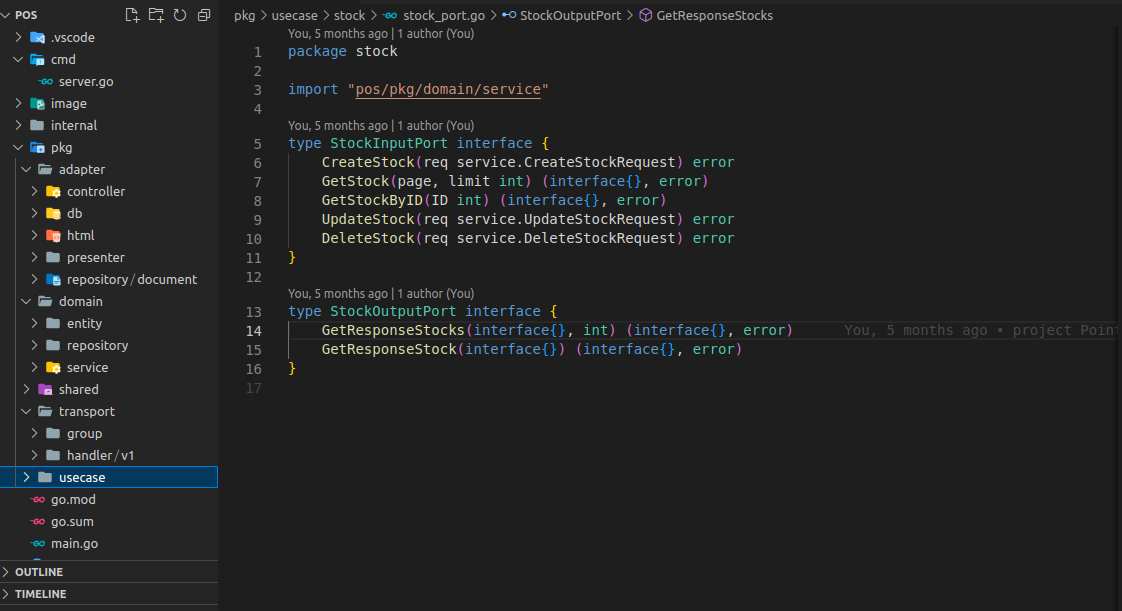
Gambar 15 menunjukkan hasil rancangan antarmuka pengguna untuk halaman *dashboard Point of sale* nya sendir. Pada rancangan antarmuka tersebut terdapat daftar produk-produk yang tersedia di toko tersebut. Lalu admin akan memilih setiap barang yang di hendak dibeli oleh pelanggan dan di bagian kanan terdapat daftar barang yang akan dibeli oleh pelanggan , disana pegawai juga bisa melakukan penambahan kuantitas ataupun pengurangan kuantitas barang. Terakhir aplikasi akan melakukan perhitungan total barang dan melakukan input data ke *database* lalu melakukan *print receipt*.

Gambar 16 menunjukkan hasil rancangan antarmuka pengguna untuk halaman transaksi. Pada rancangan antarmuka tersebut akan memuat informasi setiap transaksi yang dilakukan di halaman POS . Informasi yang di *record*  dalam tabel ini akan memberikan kita data mengenai product-product apa saja yang kerap dibeli oleh pelanggan dan informasi-informasi lainnya yang berkaitan tentang transaksi

Gambar 15 *Halaman Point Of Sale* Gambar 16 *Halaman Transaksi*

# Hasil/Pembahasan

Dalam perancangan aplikasi Point of sales penulis menggunakan bahasa pemograman Go versi 1.18 dengan framework Echo versi 4.9.0 dibagian *backend* dan menggunakan ReactJS versi 18.2.0 dibagian *frontend*. Pada saat pengembangan saya menemukan bahwa penggunaan *Clean Architecture* dalam pengembangan aplikasi memiliki beberapa keuntungan yang signifikan. Salah satunya adalah meningkatkan skalabilitas, testabilitas, maintainability, fleksibilitas, dan reusabilitas pada pengembangan aplikasi. Dengan pendekatan ini, aplikasi dapat dengan mudah di-maintain, di-Skala, dan diuji, serta dapat diubah dengan mudah dan aman tanpa memengaruhi bagian lain dari aplikasi. Selain itu, kode yang dihasilkan dengan menggunakan *Clean Architecture* lebih mudah untuk digunakan kembali dalam proyek lain.

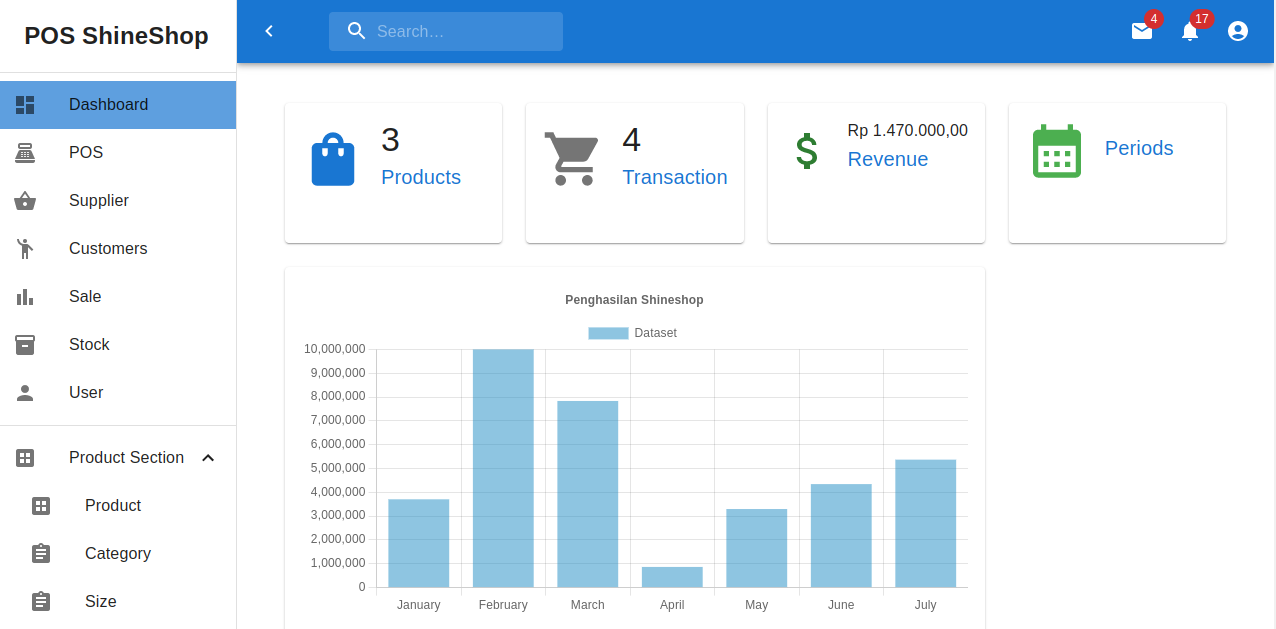
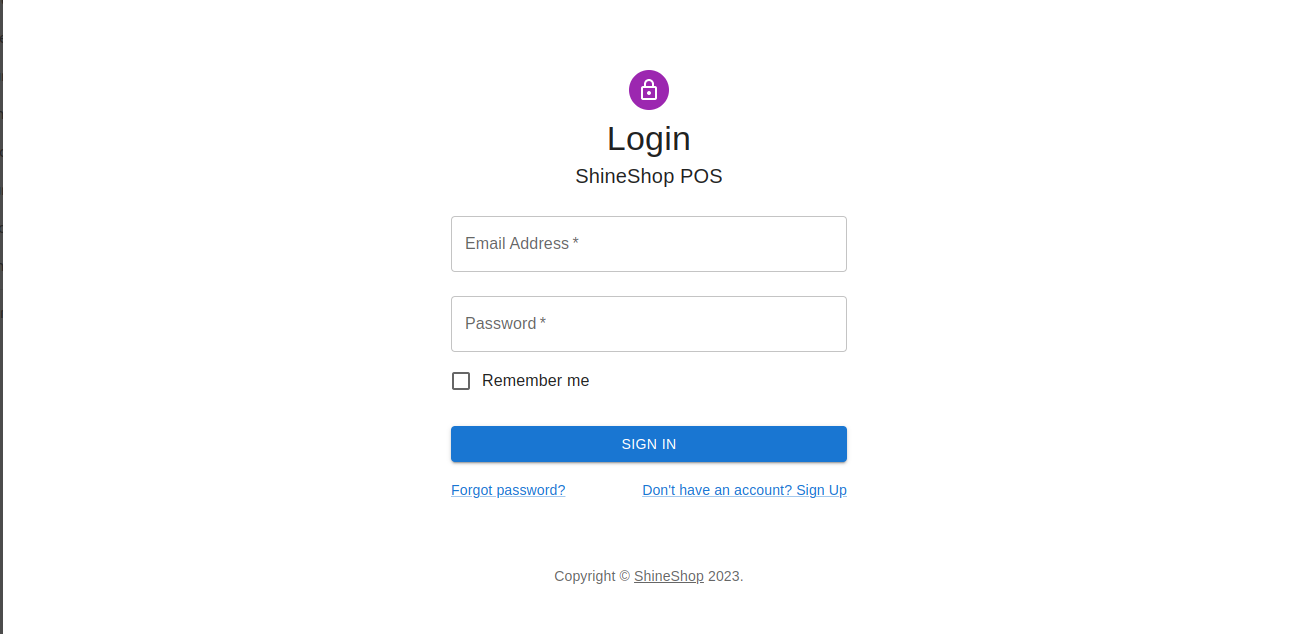


Gambar 17 Implementasi Clean Achitecture

Gambar 17 menunjukkan contoh pengimplementasian *Clean Architecture* dalam kode aplikasi. Pada gambar tersebut, terlihat bahwa kode aplikasi dibagi menjadi beberapa lapisan yang masing-masing memiliki tanggung jawabnya sendiri, yaitu lapisan presentasi, lapisan domain, dan lapisan infrastruktur.

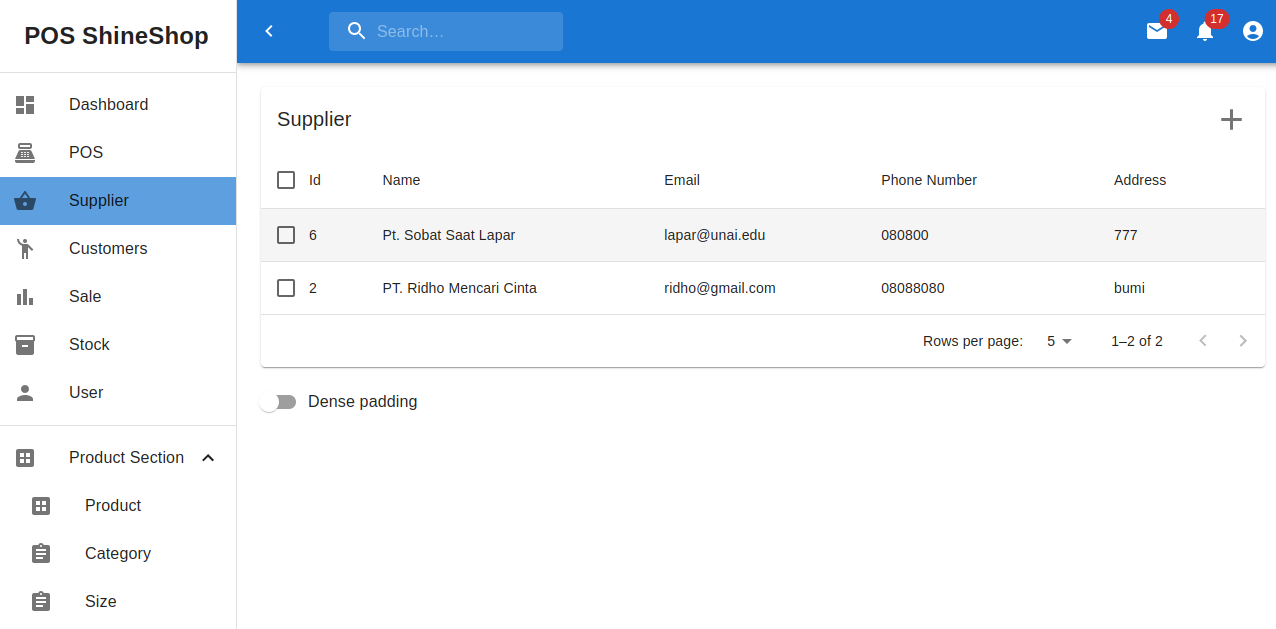
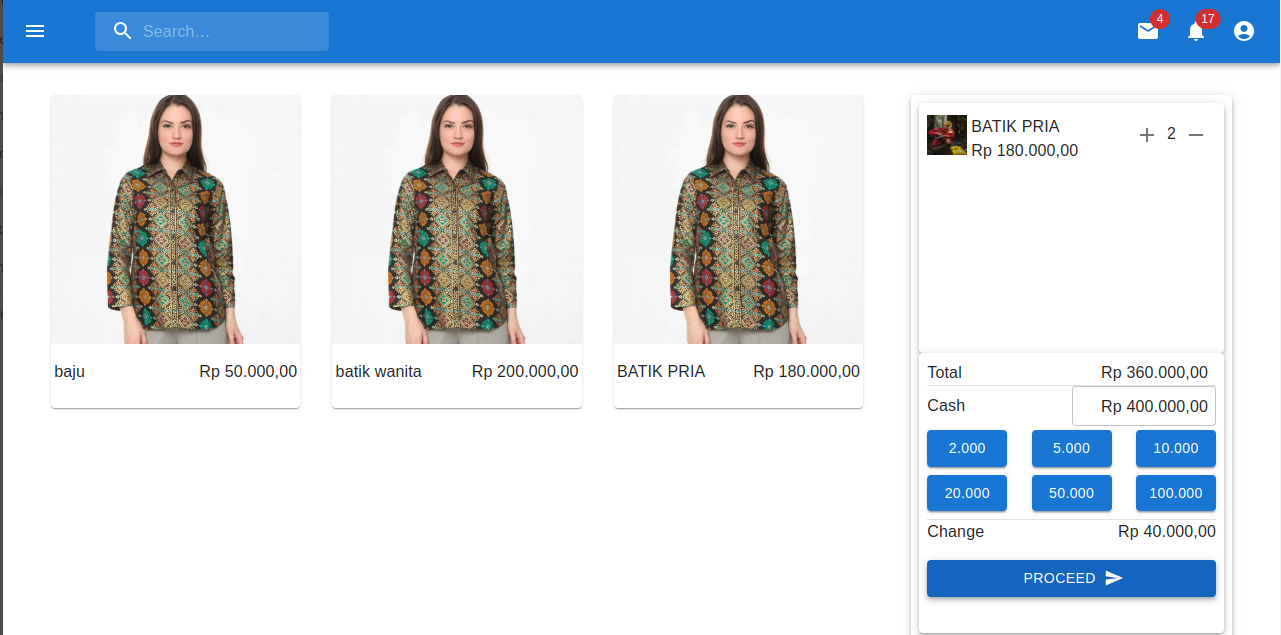
Gambar 18 menunjukkan hasil implementasi dari halaman *login* pada aplikasi *point-of-sale* yang meminta pengguna untuk memasukkan informasi *login*, seperti nama pengguna dan kata sandi. Halaman ini sangat penting untuk menjaga keamanan aplikasi dan informasi yang disimpan pada aplikasi, serta memungkinkan pengembang untuk melacak aktivitas pengguna pada aplikasi dan meningkatkan kinerja aplikasi.

Gambar 19 menunjukkan halaman dashboard pada aplikasi point of sale yang menampilkan grafik penjualan total produk pakaian. Grafik tersebut menunjukkan jumlah produk yang terjual dalam rentang waktu tertentu. Grafik tersebut juga dilengkapi dengan informasi terkait produk yang terjual, seperti kategori produk, harga, dan jumlah produk yang terjual.



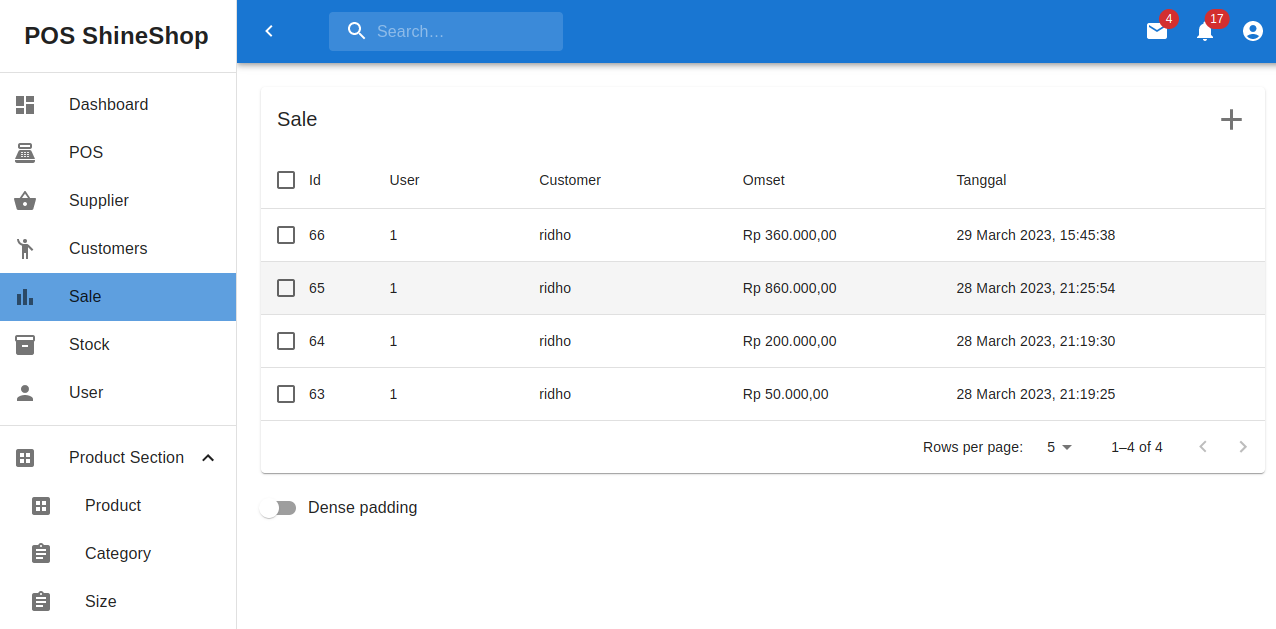
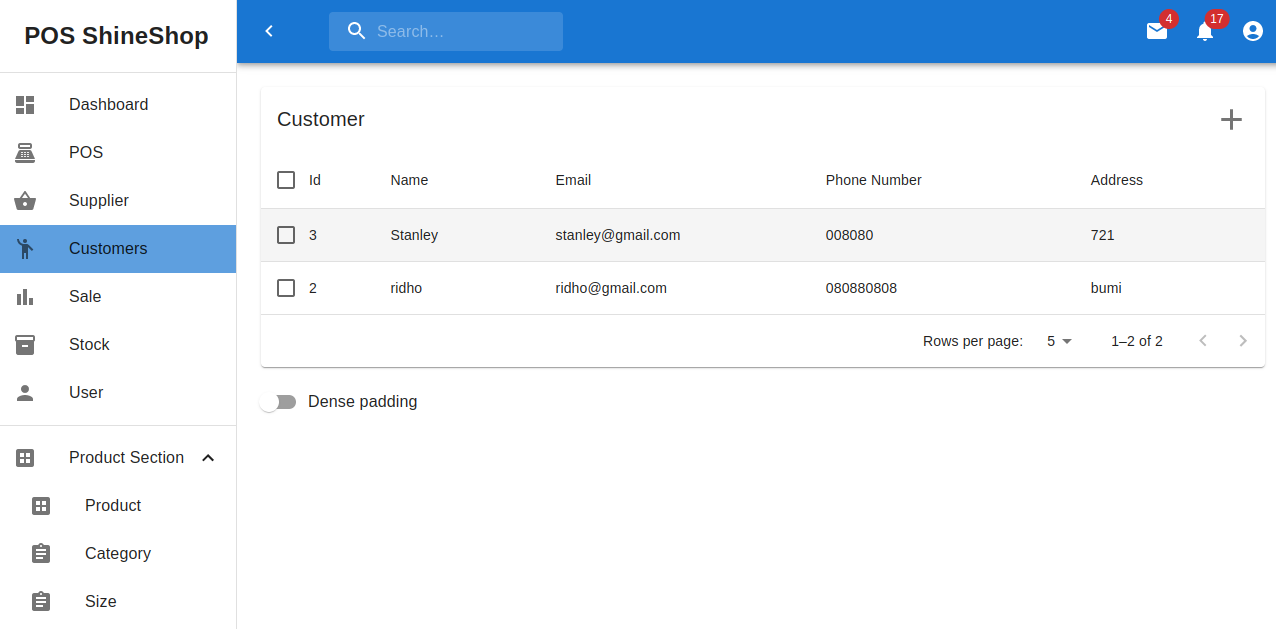
Gambar 18 Halaman Login Gambar 19 Halaman Dashboard

Gambar 20 menunjukkan halaman point-of-sale yang merupakan halaman utama aplikasi point-of-sale. Halaman ini memungkinkan pengguna untuk melakukan berbagai tindakan terkait dengan melihat produk dan melakukan pencatatan transaksi setelahnya melakukan print struk. Gambar 21 menunjukkan halaman supplier dengan fitur CRUD untuk mengelola supplier .



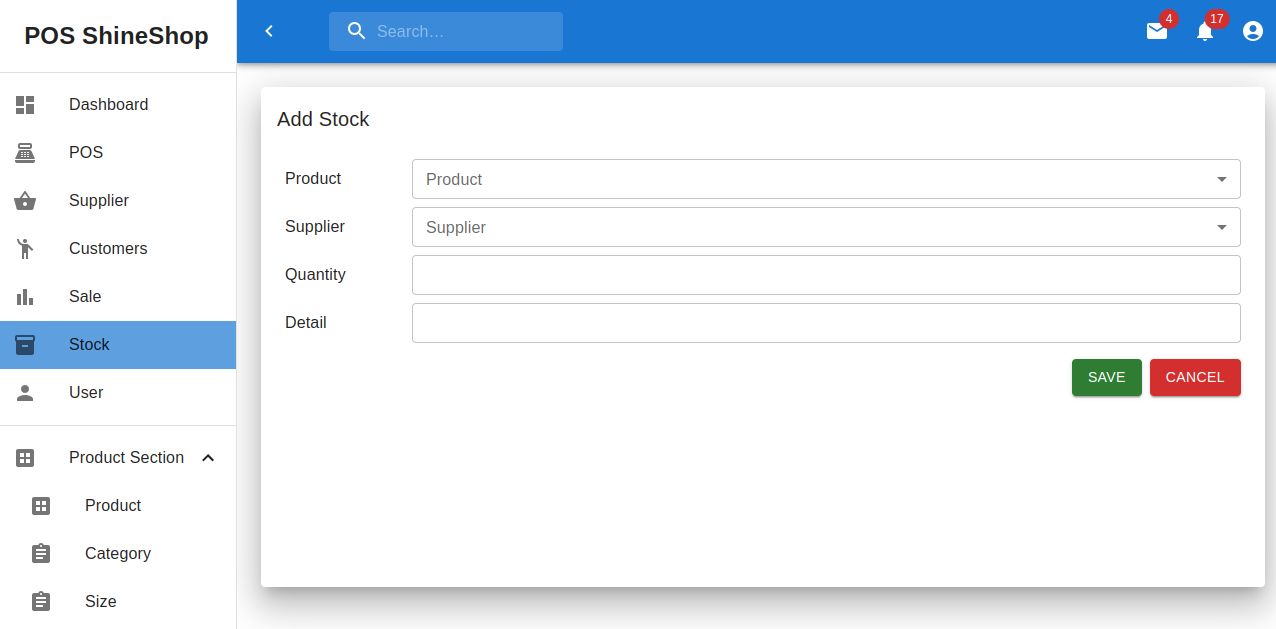
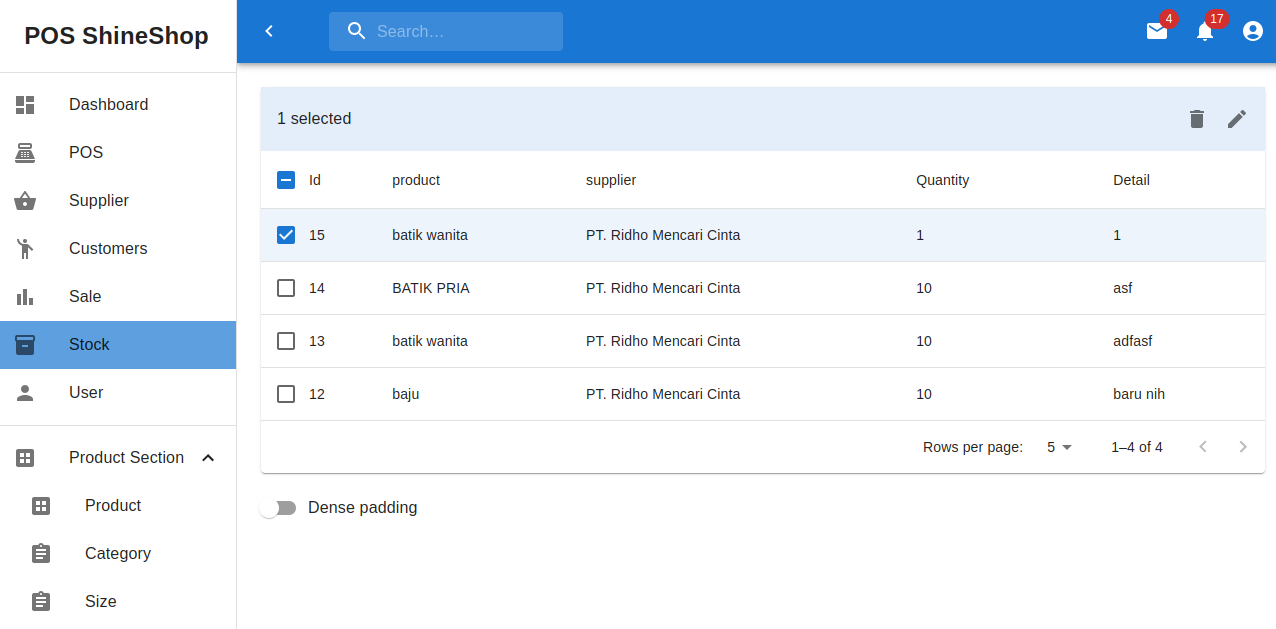
Gambar 20 Halaman *Point-of-Sales*  Gambar 21 Halaman Supplier

Gambar 22 menjelaskan halaman customer dengan fitur CRUD yang memungkinkan pengguna untuk membuat, membaca, memperbarui, dan menghapus data pelanggan pada aplikasi point-of-sale. Gambar 23 menunjukan pengimplementasian halam *sale* dimana pengguna dapat melihat transaksi berdasarkan *date-range* yang dipilih ,secara *default* aplikasi akan menunjukan transaksi bulanan dan pengguna dapat melakukan *export* menjadi file *excel* sesuai dengan periode transaksi yang dipilih.



**Gambar 22** Halaman *Customer* **Gambar 23** Halaman *Sale*

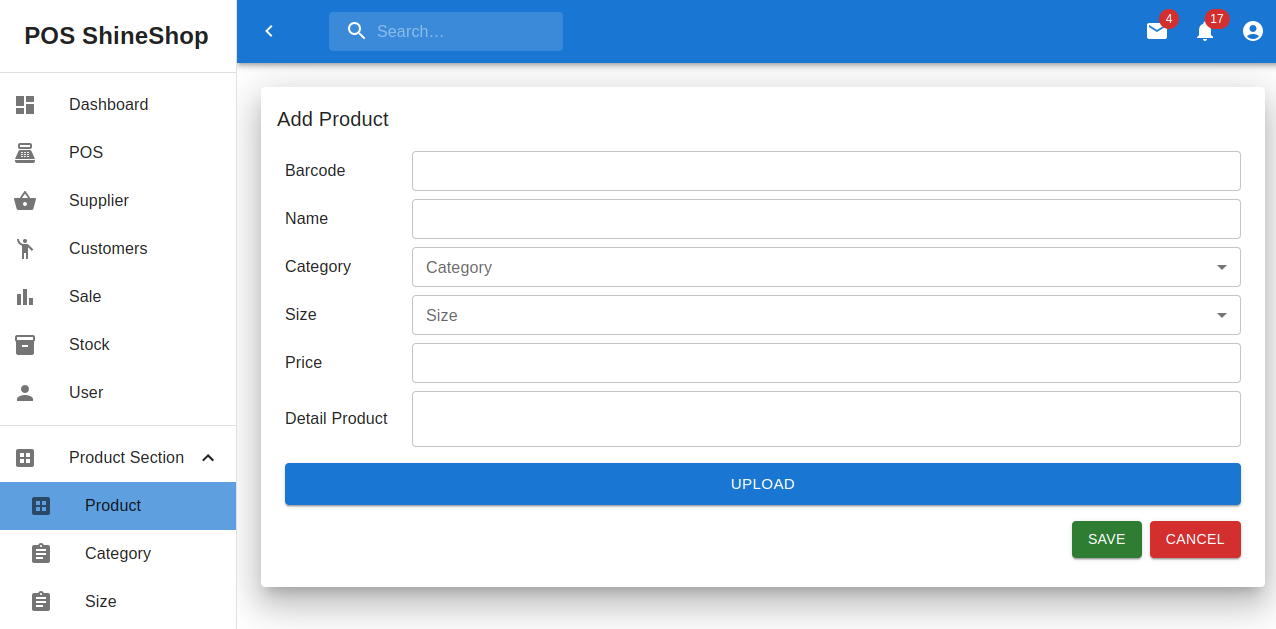
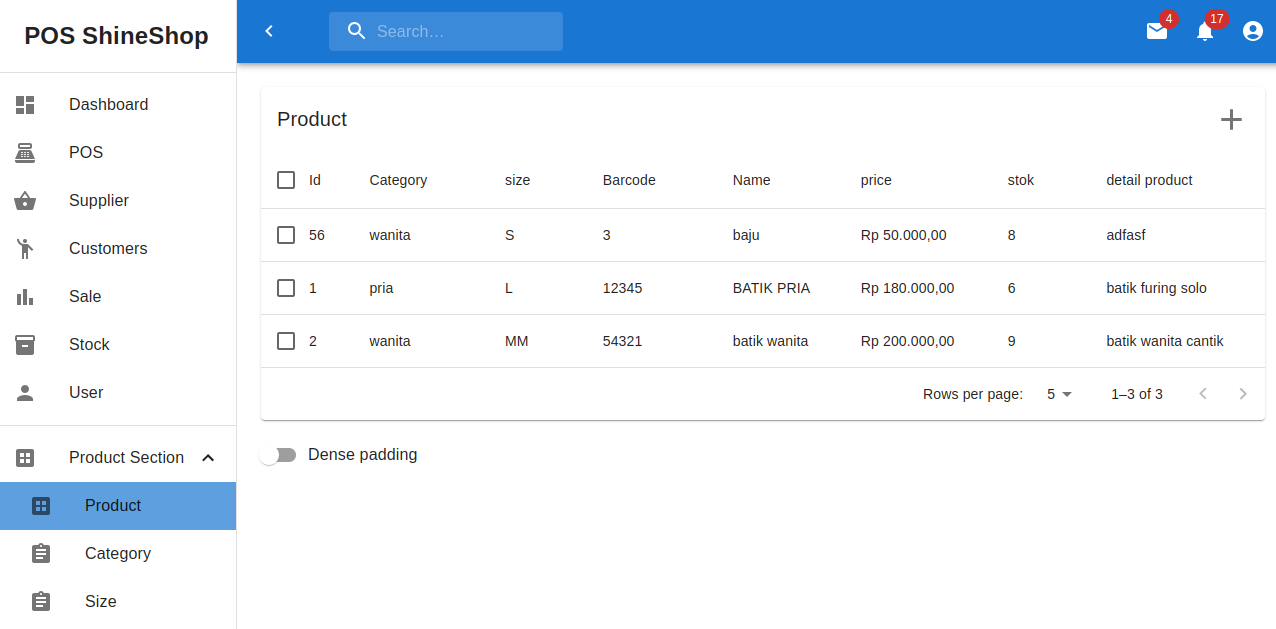
Gambar 24 menunjukkan halaman barang masuk pada aplikasi point-of-sale yang memungkinkan pengguna untuk melihat barang masuk yang telah diterima oleh toko. Pada halaman ini, pengguna dapat melihat informasi tentang barang masuk, seperti nama barang, jumlah barang, dan tanggal masuk. Gambar 25 merupakan halaman menambahkan stok dengan memilih product apa yang masuk dan ukuran dan kuantitas dari barang.



Gambar 24 Halaman Stock Gambar 25 Halaman *Stock Add*

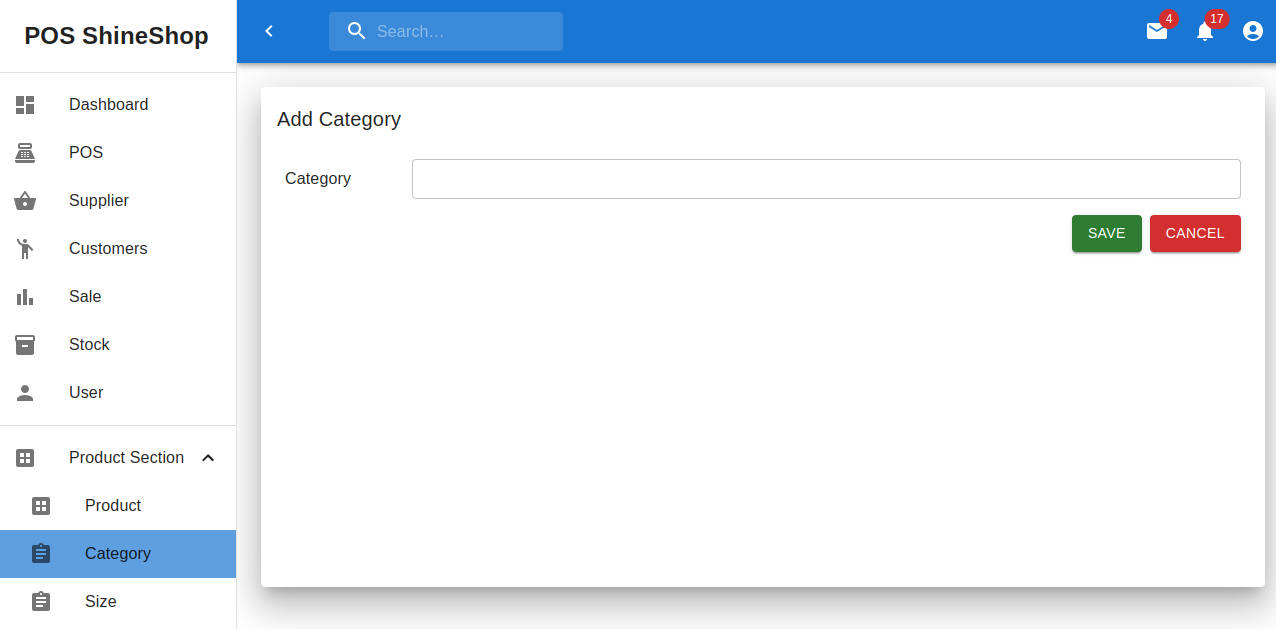
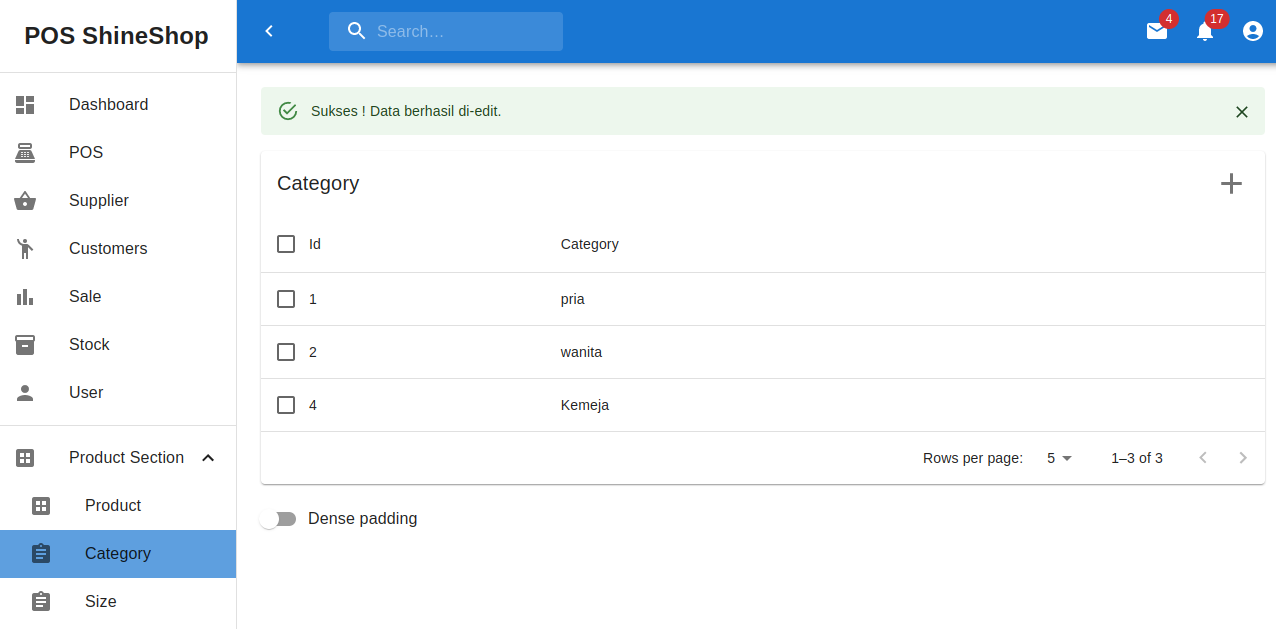
Gambar 26 menunjukkan halaman informasi barang pada aplikasi point-of-sale yang memungkinkan pengguna untuk melihat informasi lengkap tentang produk yang tersedia, seperti nama barang, deskripsi, harga, dan jumlah stok.

Gambar 27 menunjukkan halaman menambahkan produk pada aplikasi point-of-sale yang memungkinkan pengguna untuk menambahkan produk baru ke dalam aplikasi. Pada halaman ini, pengguna dapat memasukkan informasi tentang produk, seperti nama barang, deskripsi, harga, dan jumlah stok. Setelah pengguna memasukkan informasi produk dengan benar, aplikasi akan menambahkan produk tersebut ke dalam daftar produk yang tersedia.



Gambar 26 Halaman *Product* Gambar 27 Halaman *Product Add*

Gambar 28 menunjukkan halaman kategori pada aplikasi point-of-sale yang memungkinkan pengguna untuk mengelompokkan produk berdasarkan kategori tertentu. Gambar 29 menujukkan implementasi dalam menambahkan product baru.



Gambar 28 Halaman Category Gambar 29 Halaman *Category Add*

untuk pengujian terhadap aplikasi, penulis menggunakan metode Pengujian *Blackbox Testing* [21]⁠. Pengujian ini dimaksudkan untuk menguji fungsionalitas dari aplikasi. Berikut ini hasil dari pengujian dijelaskan didalam tabel.

Tabel 1 Hasil Pengujian Aplikasi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Pengujian** | **Skenario Pengujian** | **Luaran yang diharapkan** | **Hasil Pengujian** |
| 1 | *Login* | Memasukan *email password* | Ketika autentikasi berhasil maka masuk kehalaman *dashboard* | [✓] Valid  [ ] Invalid |
| 2 | *Logout* | Menekan tombol *logout* | Kembali kehalaman *login* | [✓] Valid  [ ] Invalid |
| 3 | *Dasboard* | Admin Melihat kilasan transaksi | Halaman *dashboard* menampilkan kilasan product dan transaksi dalam kurun waktu tahunan | [✓] Valid  [ ] Invalid |
| 4 | *Size* | Masuk ke laman *size* | Aplikasi menampilkan seluruh *size* | [✓] Valid  [ ] Invalid |
| Menambahkan *size* baru | Aplikasi menyimpan *size* baru di *database* | [✓] Valid  [ ] Invalid |
| *Edit size* baru | Aplikasi Menyimpan perubahan *size* pada *database* | [✓] Valid  [ ] Invalid |
| *delete size* | Aplikasi menghapus *size* yang di pilih dari *database* | [✓] Valid  [ ] Invalid |
| 5 | *Category* | Masuk ke laman *category* | Aplikasi menampilkan seluruh *category* | [✓] Valid  [ ] Invalid |
| Menambahkan *category* baru | Aplikasi menyimpan *category* baru di *database* | [✓] Valid  [ ] Invalid |
| Edit category baru | Aplikasi menyimpan perubahan *category* pada *database* | [✓] Valid  [ ] Invalid |
| *Delete category* | Aplikasi menghapus *category* yang di pilih dari *database* | [✓] Valid  [ ] Invalid |
| 6 | *Product* | Masuk ke laman *product* | Aplikasi menampilkan seluruh *product* | [✓] Valid  [ ] Invalid |
| Menambahkan *product* baru | Aplikasi menyimpan *product* baru di *database* | [✓] Valid  [ ] Invalid |
| *Edit product* Baru | Aplikasi menyimpan perubahan *product* pada *database* | [✓] Valid  [ ] Invalid |
| *Delete product* | Aplikasi menghapus *product*  yang di pilih dari *database* | [✓] Valid  [ ] Invalid |
| 6 | *Stock* | Masuk ke laman stok | Aplikasi menampilkan seluruh stok | [✓] Valid  [ ] Invalid |
| Menambahkan stok baru | Aplikasi menyimpan stok baru dan menambahkan stok di *product* | [✓] Valid  [ ] Invalid |
| *Edit* stok Baru | Aplikasi menyimpan perubahan stok pada *database* | [✓] Valid  [ ] Invalid |
| *Delete* *stock* | Aplikasi menghapus stok yang di pilih dari *database* | [✓] Valid  [ ] Invalid |
| Memberikan  *Alert stock* | Aplikasi akan menampilkan pesan alert jika suatu barang sudah kurang dari 5 pcs | [✓] Valid  [ ] Invalid |
| 7 | *Customer* | Masuk ke laman *customer* | Aplikasi menampilkan seluruh *customer* | [✓] Valid  [ ] Invalid |
| Menambahkan customer baru | Aplikasi menyimpan *customer* baru di *database* | [✓] Valid  [ ] Invalid |
| *Edit customer* baru | Aplikasi Menyimpan perubahan *customer* pada *database* | [✓] Valid  [ ] Invalid |
| *Delete customer* | Aplikasi menghapus *customer* yang di pilih dari *database* | [✓] Valid  [ ] Invalid |
| 8 | *Supplier* | Masuk ke laman *supplier* | Aplikasi menampilkan seluruh s*upplier* | [✓] Valid  [ ] Invalid |
| Menambahkan *supplier* baru | Aplikasi menyimpan *supplier* baru di *database* | [✓] Valid  [ ] Invalid |
| Edit supplier baru | Aplikasi menyimpan perubahan *supplier* pada *database* | [✓] Valid  [ ] Invalid |
| *Delete supplier* | Aplikasi menghapus *customer* yang di pilih dari *database* | [✓] Valid  [ ] Invalid |
| 9 | Pos | Masuk ke laman pos | Aplikasi menampilkan seluruh *product* beserta gambar-gambarnya | [✓] Valid  [ ] Invalid |
| Menambahkan *product* ke keranjang | Aplikasi menyimpan transaksi baru dan mengurangi stok sesuai dengan *product* | [✓] Valid  [ ] Invalid |
| *print* detail transaksi | Aplikasi melakukan *print* transaksi yang dipilih | [✓] Valid  [ ] Invalid |
| 10 | *User* | Menambahkan *user* baru | Aplikasi menyimpan *user* baru ke *database* | [✓] Valid  [ ] Invalid |

# Kesimpulan/Saran

# Berdasarkan hasil penelitian dan pembahasan perancangan aplikasi Point of Sale berbasis web menggunakan Golang dan Reactjs. Maka penulis menyimpulkan bahwa Aplikasi berjalan dengan baik dan diharapkan dapat membantu dalam pencatatan dan monitoring stok product. Pemilik bisnis dapat dengan mudah mengawasi transaksi dan mengawasi produk yang memberikan kontribusi terhadap omset bisnis. Aplikasi ini diharapkan meningkatkan *user experience* pelanggan. Aplikasi ini juga dapat dengan mudah digunakan oleh pengguna dan terakhir dengan hadirnya aplikasi ini diharapkan dapat meningkatkan efektifitas dan produktifitas bisnis.

# Beberapa saran pengembangan yang dapat dilakukan yaitu Mengembangkan tampilan agar sistem terlihat lebih *responsive* jika digunakan pada perangkat *mobile* dan mengembangkan aplikasi *mobile* juga. Menambahkan fitur promosi yang akan mengirimkan pesan WhatApp otomatis ke *customer* jika ada *product* keluaran terbaru. Menambahkan fitur *scan barcode*. Mengintegrasi dengan *payment gateway provider*

# Daftar Pustaka

[1] P. Sosial and E. Susanti, “PELATIHAN DIGITAL MARKETING DALAM UPAYA PENGEMBANGAN USAHA BERBASIS TEKNOLOGI PADA UMKM DI DESA SAYANG KECAMATAN JATINANGOR,” *Sawala J. Pengabdi. Masy. Pembang. Sos. Desa dan Masy.*, vol. 1, no. 2, pp. 36–50, Jul. 2020, doi: 10.24198/SAWALA.V1I2.26588.

[2] D. Hendarsyah, S. Tinggi, I. Ekonomi, and S. Bengkalis, “E-Commerce Di Era Industri 4.0 Dan Society 5.0,” *IQTISHADUNA J. Ilm. Ekon. Kita*, vol. 8, no. 2, pp. 171–184, Dec. 2019, doi: 10.46367/IQTISHADUNA.V8I2.170.

[3] S. Fransisca, N. Putri, and M. Kom, “Pemanfaatan Teknologi RFID untuk Pengelolaan Inventaris Sekolah Dengan Metode (R&D),” *J. Mhs. Apl. Teknol. Komput. dan Inf.*, vol. 1, no. 1, pp. 72–75, Jul. 2019, Accessed: Apr. 03, 2023. [Online]. Available: https://ejournal.pelitaindonesia.ac.id/JMApTeKsi/index.php/JOM/article/view/395

[4] A. P. Y. K. Wardani and N. A. S. Darmawan, “Peran Financial Technology pada UMKM: Peningkatan Literasi Keuangan Berbasis Payment Gateway,” *J. Ilm. Akunt. dan Humanika*, vol. 10, no. 2, pp. 170–175, Aug. 2020, doi: 10.23887/JIAH.V10I2.25947.

[5] F. Marisa and T. G. Yuarita, “PERANCANGAN APLIKASI POINT OF SALES (POS) BERBASIS WEB MENGGUNAKAN METODE SIKLUS HIDUP PENGEMBANGAN SISTEM,” *J. Teknol. dan Manaj. Inform.*, vol. 3, no. 2, Sep. 2017, doi: 10.26905/JTMI.V3I2.1514.

[6] M. D. Rahmatya, “SISTEM INFORMASI PENJUALAN PADA KANTIN X,” *J. Manaj. Inform.*, vol. 7, no. 2, Oct. 2017, doi: 10.34010/JAMIKA.V7I2.531.

[7] D. H. Simbolon, and J. I. Sihotang, “Pengembangan Sistem Informasi POS ( Point of Sales ) Berbasis Android Pada Toko Pokdakan Mitra Bersama Menggunakan *Framework* Flutter”., Fakultas Teknologi Informasi ., Universitas Advent Indonesia , 2022/

[8] G. Pamungkas and H. Yuliansyah, “RANCANG BANGUN APLIKASI ANDROID POS (POINT OF SALE) KAFE UNTUK KASIR PORTABLE DAN BLUETOOTH PRINTER,” *JST (Jurnal Sains dan Teknol.*, vol. 6, no. 1, p. 200, Mar. 2017, doi: 10.23887/JSTUNDIKSHA.V6I1.8828.

[9] Y. Christian and K. Kelvin, “Perancangan Dan Penerapan Sistem Pos (Point Of Sale) Berbasis Web Pada Warung Zikry,” *ConCEPt - Conf. Community Engagem. Proj.*, vol. 1, no. 1, pp. 61–66, Apr. 2021, Accessed: Nov. 22, 2022. [Online]. Available: https://journal.uib.ac.id/index.php/concept/article/view/4610

[10] A. A. A. Donovan and B. W. Kernighan, “The Go programming language”.

[11] Katherine Cox-Buday et al., Concurrency in Go (Italic), O'Reilly, 2017, pp. 3-11.

[12] Facebook Inc., "React - A JavaScript library for building user interfaces," Facebook Inc., Menlo Park, CA, USA, 2021. Accessed on: May 4, 2023. [Online]. Available: https://reactjs.org/

[13] Djatmiko, D., & Nugroho, A. Y. (2019). React JS Untuk Pemula. Yogyakarta: Gava Media. Halaman 78-80.

[14] Robert C. Martin, *Clean Code: A Handbook of Agile Software Craftsmanship20092Robert C. Martin. Clean Code: A Handbook of Agile Software Craftsmanship . Prentice‐Hall, 2008. £27.99, ISBN: 9‐780‐13235‐088‐4*, vol. 38, no. 6. 2009.

[15] K. S. Haryana, “PENERAPAN AGILE DEVELOPMENT METHODS DENGAN FRAMEWORK SCRUM PADA PERANCANGAN PERANGKAT LUNAK KEHADIRAN RAPAT UMUM BERBASIS QR-CODE,” *J. Comput. Bisnis*, vol. 13, no. 2, pp. 70–79, Dec. 2019, doi: 10.55281/JCB.V13I2.202.

[16] A. C. Hutauruk and A. F. Pakpahan, “Perancangan Sistem Informasi Organisasi Kemahasiswaan Berbasis Web pada Universitas Advent Indonesia Menggunakan Metode Agile Development (Studi Kasus: Universitas Advent Indonesia),” *CogITo Smart J.*, vol. 7, no. 2, pp. 315–328, Dec. 2021, doi: 10.31154/COGITO.V7I2.328.315-328.

[17] “What Is Agile Scrum Methodology? - businessnewsdaily.com.” https://www.businessnewsdaily.com/4987-what-is-agile-scrum-methodology.html (accessed Dec. 27, 2022).

[18] S. Ashmore and K. Runyan, *Introduction to Agile Methods*, 1st ed. New Jersey: Pearson Education, Inc, 2015.

[19] M. Muslihudin and Oktafiano, *Analisis dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML*, 1st ed. Yogyakarta: CV ANDI OFFSET, 2016.

[20] R. Artikel, J. Panjaitan, A. Fernando Pakpahan, J. Kolonel Masturi No, K. Bandung Barat, and J. Barat, “Perancangan Sistem E-Reporting Menggunakan ReactJS dan Firebase,” *J. Tek. Inform. dan Sist. Inf.*, vol. 7, no. 1, pp. 2443–2229, Apr. 2021, doi: 10.28932/JUTISI.V7I1.3098.

[21] W. N. Cholifah, Y. Yulianingsih, and S. M. Sagita, “Pengujian Black Box Testing pada Aplikasi Action & Strategy Berbasis Android dengan Teknologi Phonegap,” *STRING (Satuan Tulisan Ris. dan Inov. Teknol.*, vol. 3, no. 2, pp. 206–210, Dec. 2018, doi: 10.30998/STRING.V3I2.3048.

**DAFTAR RIWAYAT HIDUP**

**I. BIODATA PENULIS**

Nama : Daniel Reinhard Capah

Tempat/Tanggal Lahir : Porsea, 28 Februari 2001

Agama : Kristen Advent

Alamat : Desa Parbuluan VI, Kec: Parbuluan, Kab: Dairi, Sumatera Utara

Nomor HP : 0853-6866-8964

Email : [danielcapah777@gmail.com](mailto:danielcapah777@gmail.com)

**II**. **SERTIFIKAT**

Information Technology Specialist Certification (ITS) Databases

**III. PENDIDIKAN**

2020 - 2023 : Universitas Advent Indonesia

2019 – 2020 : Perguruan Tinggi Advent Surya Nusantara Pematang Siantar

2016 – 2019 : SMK Swasta Advent Air bersih Medan

2013 – 2016 : SMP Swasta Advent Lumban Gambiri

2007 – 2013 : SD Swasta Advent Simodong

**IV. PENGALAMAN ORGANISASI**

Januari 2023 – April 2023 , Itverse, Cybersecurity (Anggota)

**V. PENGALAMAN BEKERJA**

Januari 2022 – Juli 2022, UPT Komputer Universitas Advent Indonesia (Divisi Jaringan)

Agustus 2022 – Desember 2022, PT. Generasi Anak Muda Berkarya (Backend Developer)