Rancang Bangun Aplikasi Point Of Sales Untuk Toko Baju Menggunakan Laravel

Rendy Kamaluddin[1], Hadi Muhammad Yusuf[2], Cahyo Prianto[3].

Prodi D4 Teknik Informatika Sekolah Vokasi, Universitas Logistik dan Bisnis Internasional[1]
email: rd24rk@gmail.com
Prodi D4 Teknik Informatika Sekolah Vokasi, Universitas Logistik dan Bisnis Internasional[2]
email: hadimuhammadyusuf019@gmail.com
Prodi D4 Teknik Informatika Sekolah Vokasi, Universitas Logistik dan Bisnis Internasional[3]
email: cahyo@ulbi.ac.id

Abstrak

Aplikasi *Point of Sales (POS)* berbasis web dikembangkan untuk meningkatkan efisiensi operasional dan pengelolaan transaksi pada toko baju. Sistem ini dibangun menggunakan *Laravel* dan *MySQL* serta mencakup manajemen pengguna, produk, transaksi, gudang, dan laporan. Aplikasi mendukung tiga peran utama: *admin* mengelola pengguna dan laporan, kasir memproses penjualan, dan *staf* gudang mengawasi pergerakan stok. Pengembangan menggunakan metodologi *Agile* untuk memastikan fleksibilitas melalui iterasi dan evaluasi berkelanjutan. Pengujian *unit* memastikan keandalan sistem dalam aspek fungsionalitas dan antarmuka pengguna. Fitur utama mencakup pelacakan stok *real-time*, pencatatan barang masuk dan keluar, serta notifikasi *stok* rendah untuk mencegah kekurangan inventaris. Aplikasi ini meningkatkan akurasi transaksi, efisiensi pengelolaan stok, dan mendukung pengambilan keputusan yang lebih cepat dan tepat.

Kata Kunci: *Point of Sales*, *Laravel Framework*, Manajemen Stok Barang, Website Toko Baju, Transaksi-*Realtime*.

Abstract

The web-based Point of Sales (POS) application was developed to enhance operational efficiency and transaction management in clothing stores. Built with Laravel and MySQL, the system integrates user, product, transaction, warehouse, and report management. It supports three main roles: admins manage users and reports, cashiers process sales, and warehouse staff oversee inventory movement. The Agile methodology ensures adaptability through iterative development and continuous evaluation. Unit testing validates system reliability in terms of functionality and user interface performance. Key features include real-time stock tracking, inventory logging, and low-stock notifications to prevent shortages. This application improves transaction accuracy, streamlines inventory management, and supports faster and more precise decision-making.

Keywords: *Point Of Sales, Laravel Framework, Inventory Management , Clothing Store Website, Real-Time Transactions.*

1. PENDAHULUAN

Di era digital saat ini, kemajuan teknologi telah membawa perubahan signifikan dalam berbagai bidang, termasuk toko baju. Toko baju, sebagai bagian dari sektor ini, harus beradaptasi dengan kemajuan teknologi untuk meningkatkan efektivitas operasional dan kinerja bisnis. Pengelolaan inventaris dan transaksi penjualan yang dilakukan secara manual sering kali menyebabkan berbagai masalah, seperti kesalahan pencatatan data, keterlambatan dalam pemrosesan transaksi, dan kesulitan dalam menghasilkan laporan penjualan yang akurat (Ahnaf & Puspitasari, 2023), (Subianto et al., 2023). Masalah-masalah ini dapat mengganggu operasional toko, mengurangi kepuasan pelanggan, dan bahkan dapat menyebabkan kerugian bagi perusahaan (Kucharska, 2024).

Penerapan sistem *Point of Sale (POS)* menjadi salah satu solusi efektif untuk mengatasi masalah ini. *POS* adalah alat yang digunakan untuk mencatat transaksi penjualan, memantau stok barang, dan

menghasilkan laporan penjualan secara otomatis (Budiarti & Kamila, 2024). Dengan menggunakan sistem ini, toko dapat beroperasi dengan lebih efisien, akurat, dan terorganisir . Dalam konteks persaingan toko baju yang semakin ketat, inovasi teknologi ini menjadi kebutuhan penting untuk mendukung pertumbuhan bisnis (Hasbi & Istabul, 2023).

Pemilihan struktur yang tepat dalam pengembangan sistem *POS* sangat penting untuk menciptakan aplikasi yang andal dan efektif. *Laravel*, sebagai *framework* berbasis *PHP*, menawarkan kemudahan penggunaan, fitur keamanan yang kuat, dan dukungan untuk arsitektur *Model-View-Controller (MVC)*, (Salim & Frederica, 2020). Arsitektur *MVC* memisahkan logika bisnis, tampilan antarmuka, dan pengelolaan data, sehingga memudahkan proses pengembangan dan pemeliharaan aplikasi yang lebih kompleks (Firsty Giovanni Naoki Panca Putra et al., 2023). *Laravel* juga menyediakan fitur penting seperti validasi input dan manajemen sesi, serta kemampuan untuk diintegrasikan dengan berbagai jenis basis data, seperti *MySQL*, yang sangat membantu dalam pembuatan sistem *POS.*

Studi ini berfokus pada pengembangan aplikasi *POS* berbasis *web* yang menggunakan *framework Laravel* untuk membantu toko pakaian dalam mengelola transaksi dan stok barang. Aplikasi ini dirancang untuk mengelola proses penjualan, memantau stok barang, dan menghasilkan laporan penjualan yang menyeluruh. Sistem ini juga mendukung akses berbagai peran, seperti Admin, Kasir, dan Gudang, yang masing-masing memiliki hak akses dan fungsi sesuai dengan tugas mereka. Salah satu fitur utama aplikasi ini adalah pencatatan transaksi penjualan secara *real-time*, pengelolaan stok barang otomatis, dan notifikasi untuk stok barang yang mulai menipis. Integrasi fitur-fitur ini bertujuan untuk mempermudah operasi toko, mengurangi kesalahan manual, dan meningkatkan efisiensi bisnis secara keseluruhan.

Selain itu, sistem ini dimaksudkan untuk menghasilkan laporan penjualan yang akurat dan mudah dipahami, yang akan membantu pemilik toko membuat keputusan yang lebih baik tentang bisnis mereka. Studi ini dilakukan untuk mencari solusi untuk beberapa masalah utama yang dihadapi oleh pemilik toko pakaian, antara lain:

1. Pengelolaan inventaris yang masih dilakukan secara manual sering menyebabkan kesalahan pencatatan.
2. Kesulitan dalam mencatat transaksi penjualan secara cepat dan akurat.
3. Kurangnya alat untuk menghasilkan laporan penjualan yang informatif.

Sistem *POS* yang dikembangkan dalam penelitian ini bertujuan untuk menyederhanakan pengelolaan toko dan meningkatkan keakuratan data. Sistem ini mengintegrasikan pengelolaan transaksi penjualan, manajemen inventaris produk, dan pembuatan laporan keuangan dalam satu *platform* yang terintegrasi, sehingga dapat memberikan solusi yang menyeluruh bagi operasional toko. Diharapkan bahwa penelitian ini akan berkontribusi pada pengembangan solusi digital di industri ritel, khususnya yang berkaitan dengan manajemen toko pakaian. Dengan menerapkan teknologi yang tepat, pemilik toko dapat meningkatkan layanan pelanggan, meningkatkan efisiensi operasional, dan meningkatkan daya saing bisnis mereka, (Saputra & Fadlila, 2023). Sistem *POS* yang dikembangkan dalam penelitian ini dapat disesuaikan dan dikembangkan untuk berbagai jenis bisnis, sehingga memberikan manfaat yang lebih luas.

Sebagai kesimpulan, pengembangan aplikasi *POS* berbasis *Laravel* adalah langkah yang efektif untuk mengatasi berbagai masalah yang berkaitan dengan penjualan dan pengelolaan inventaris. Sistem ini tidak hanya mempermudah operasi harian, tetapi juga membantu bisnis berkembang dengan memberikan data yang akurat dan meningkatkan efisiensi kerja. Untuk memperluas fungsi, langkah strategis dapat mencakup pengembangan tambahan, seperti penggabungan metode pembayaran digital dan pengembangan aplikasi berbasis *mobile*.

2. METODE PENELITIAN

Metode penelitian yang sistematis dan terstruktur sangat penting dalam menyelesaikan permasalahan melalui pendekatan ilmiah. Dalam konteks ini, penelitian yang dilakukan harus mencakup langkah-langkah yang jelas, mulai dari identifikasi masalah hingga penarikan kesimpulan berdasarkan data yang dikumpulkan. Proses ini tidak hanya bertujuan untuk memberikan jawaban yang valid dan dapat dipercaya, tetapi juga untuk mendukung pengambilan keputusan yang tepat dalam konteks yang lebih luas. Penelitian yang terorganisir dengan baik memungkinkan analisis yang lebih akurat dan hasil yang lebih relevan, yang pada gilirannya dapat meningkatkan efektivitas pengambilan keputusan di berbagai bidang, termasuk dalam pengembangan sistem aplikasi (Pinton & Torres, 2020), (Rangavittal, 2022).

Dalam pengembangan sistem aplikasi, langkah-langkah penelitian harus dirancang untuk memastikan bahwa data yang dikumpulkan terorganisir dengan baik. Hal ini mencakup perumusan tujuan yang jelas, eksplorasi literatur terkait, implementasi sistem, pengujian, serta evaluasi hasil. Setiap tahapan ini berkontribusi pada pemahaman yang lebih baik mengenai masalah yang dihadapi dan solusi yang dapat diterapkan. Penelitian yang baik juga harus mempertimbangkan konteks organisasi dan budaya yang ada, karena faktor-faktor ini dapat mempengaruhi keberhasilan implementasi solusi yang diusulkan (Alaidaros et al., 2021), (Zakrzewska et al., 2022).

Pendekatan *Agile Development* dipilih dalam penelitian ini karena sifatnya yang iteratif dan kolaboratif. Metode ini memungkinkan pengembang untuk merespons perubahan kebutuhan pengguna dengan cepat dan efektif, yang sangat penting dalam lingkungan yang dinamis dan tidak pasti. *Agile Development* tidak hanya berfokus pada pengembangan perangkat lunak, tetapi juga mencakup aspek manajerial dan budaya organisasi yang diperlukan untuk mendukung transisi yang sukses ke dalam metodologi ini. Penelitian menunjukkan bahwa keberhasilan implementasi *Agile* sering kali bergantung pada kepemimpinan transformasional dan investasi dalam literasi *Agile* di seluruh organisasi (Dragičević & Bošnjak, 2019). Dengan pendekatan ini, proses penelitian dipecah menjadi beberapa siklus kerja yang memungkinkan adanya penyempurnaan terus-menerus di setiap tahap implementasi. Penjelasan detail mengenai metode *agile* sebagai berikut.

2.1 Metode Agile Development

Metode *Agile Development* merupakan pendekatan pengembangan perangkat lunak yang menekankan fleksibilitas, kolaborasi tim, dan pengembangan berkelanjutan dalam literasi pendek. Pendekatan ini mempermudah pengembang untuk merespons perubahan kebutuhan pengguna dengan cepat dan efektif.

Tahapan-tahapan dalam *Agile Development* meliputi(Schwaber & Sutherland, 2017):

1. *Requirements*
   Pada tahap ini, dilakukan identifikasi kebutuhan pengguna berdasarkan masukan dari pemilik toko pakaian, gudang, dan kasir. Proses ini melibatkan analisis kebutuhan yang akan menjadi dasar untuk menentukan fitur-fitur utama aplikasi, seperti manajemen stok, transaksi penjualan, dan pencetakan struk. Kebutuhan ini kemudian dirinci menjadi *backlog* untuk direncanakan dalam *sprint*.
2. *Design*
   Perancangan aplikasi dilakukan dengan membuat sketsa *user interface* menggunakan *tools Figma*. Selain itu, dibuat pula diagram *Unified Modeling Language (UML)* untuk memvisualisasikan alur data dan proses sistem, termasuk diagram *use case*, dan *class* diagram. Perancangan ini bertujuan memastikan aplikasi dirancang sesuai kebutuhan pengguna.
3. *Development*
   Proses *development* melibatkan pengkodean aplikasi menggunakan *framework Laravel.* Tahap ini dimulai dengan mengembangkan fitur utama, seperti autentikasi pengguna, pencatatan transaksi, dan manajemen stok barang. Penggunaan *Laravel* memungkinkan pengembang

memanfaatkan arsitektur *Model-View-Controller (MVC)* untuk memisahkan logika bisnis, data, dan tampilan.

4. *Testing*
   Pengujian dilakukan secara iteratif untuk memastikan setiap fitur berfungsi sesuai spesifikasi. Metode unit testing dengan *PHPUnit* digunakan untuk menguji komponen aplikasi secara terisolasi. *Code coverage* digunakan untuk mengukur cakupan eksekusi kode. Selain itu, pengujian langsung oleh pengguna (admin, kasir dan gudang) untuk memperoleh masukan dan memastikan kemudahan penggunaan aplikasi.
5. *Deployment*
   Aplikasi yang telah selesai diuji dan disempurnakan di *deploy* untuk digunakan oleh pengguna. Sistem ini dirancang untuk mendukung operasional toko pakaian, termasuk transaksi penjualan, pencatatan barang masuk, dan notifikasi stok rendah.
6. *Review*
   Setelah setiap *sprint*, dilakukan evaluasi terhadap hasil pengembangan. Masukan dari pengguna dan pemilik toko dianalisis untuk perbaikan di *sprint* berikutnya. Tahap ini penting untuk memastikan aplikasi berkembang sesuai kebutuhan pengguna dan bisnis.

3. HASIL DAN PEMBAHASAN

Pada bagian hasil pembahasan ini, akan diuraikan mengenai "Aplikasi *Point of Sales* Toko Baju Menggunakan *Laravel*", sebuah *platform* yang dirancang untuk mengelola inventaris, mencatat transaksi penjualan, dan menyediakan laporan penjualan secara otomatis bagi toko pakaian.

3.1 Analisis Kebutuhan

Penelitian ini menggunakan komunikasi dengan cara *brainstorming* yang melibatkan tim penulis dan pengembang yang akan merancang sistem ini untuk mengumpulkan data tentang sistem dan fungsi yang perlu diimplementasikan. Pengumpulan data dilakukan dengan merujuk pada jurnal. Fungsi kebutuhan menentukan tindakan yang harus dilakukan sistem untuk mendukung operasi dan interaksi pengguna. "Aplikasi *Point Of SaIes* Untuk Toko Baju Menggunakan *Laravel*" memiliki kebutuhan fungsional berikut.

Tabel 1. Kebutuhan Fungsionalitas

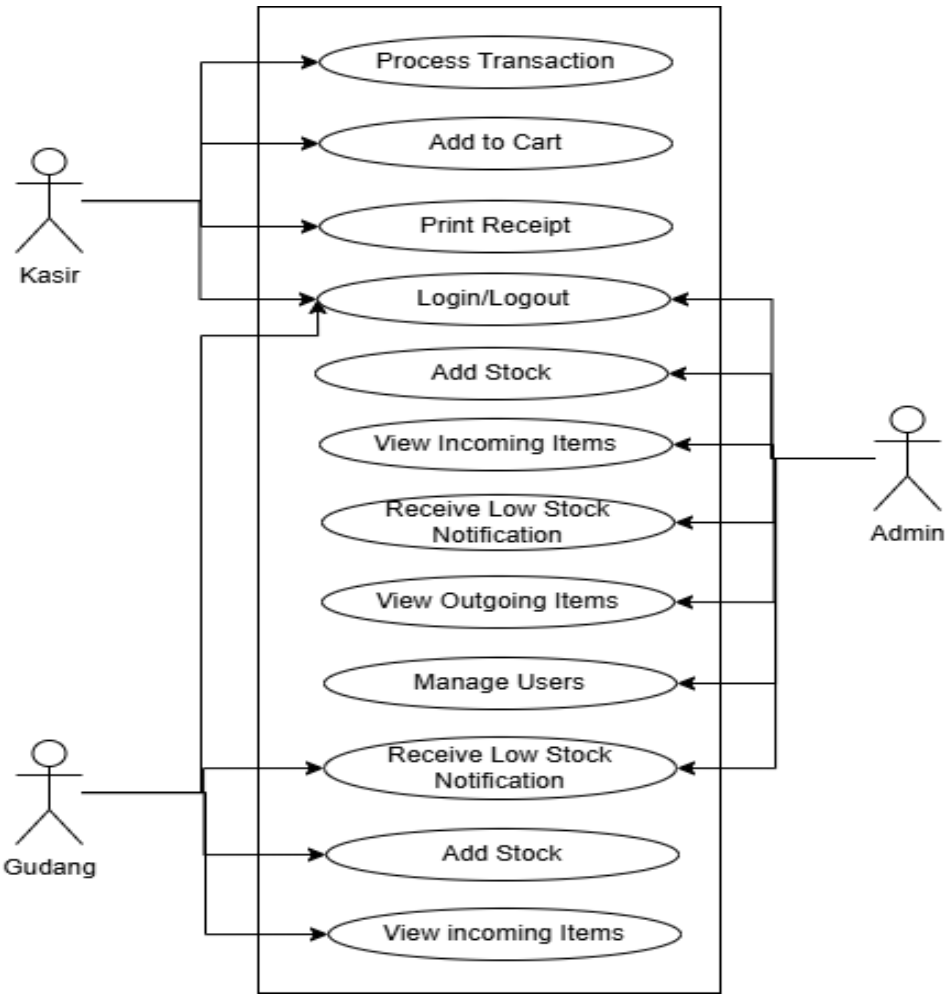
Kebutuhan Fungsional	Penjelasan
Pengelolaan Data Pengguna	Admin dapat menambah, mengedit, menghapus, dan melihat data pengguna sesuai dengan hak aksesnya.
Pengelolaan Transaksi Penjualan	Admin dapat memantau dan mengelola seluruh transaksi penjualan yang dicatat oleh kasir.
Pembuatan dan Cetak Laporan Penjualan	Admin dapat membuat dan mencetak laporan penjualan untuk memantau performa toko secara berkala.
Proses Transaksi Penjualan oleh Kasir	Kasir dapat melakukan transaksi penjualan, menghitung total harga, dan mencetak struk pembelian.
Melihat Riwayat Transaksi Barang Masuk	Gudang dapat mengakses riwayat transaksi barang masuk untuk memantau pergerakan stok.
Pencatatan Barang Masuk dan Pembaruan Stok	Gudang mencatat barang yang masuk dan secara otomatis memperbarui stok di sistem.
Notifikasi Stok Rendah	Sistem akan memberikan notifikasi kepada gudang jika stok barang mendekati batas minimum.

ISSN: 0216-2539 (Print)
E-ISSN: 2656-4157 (Online)

3.2 Perancangan Sistem

Aplikasi ini dirancang menggunakan berbagai metode *Class Diagram*, *Usecase Diagram* dan *Flowchart*, yang semuanya merupakan bagian dari Bahasa Pemodelan Terpadu *Unifield Modelling Language*. Pendekatan ini dipilih untuk memastikan bahwa perancangan sistem terstruktur dengan baik.

3.2.1 *Use Case Diagram*

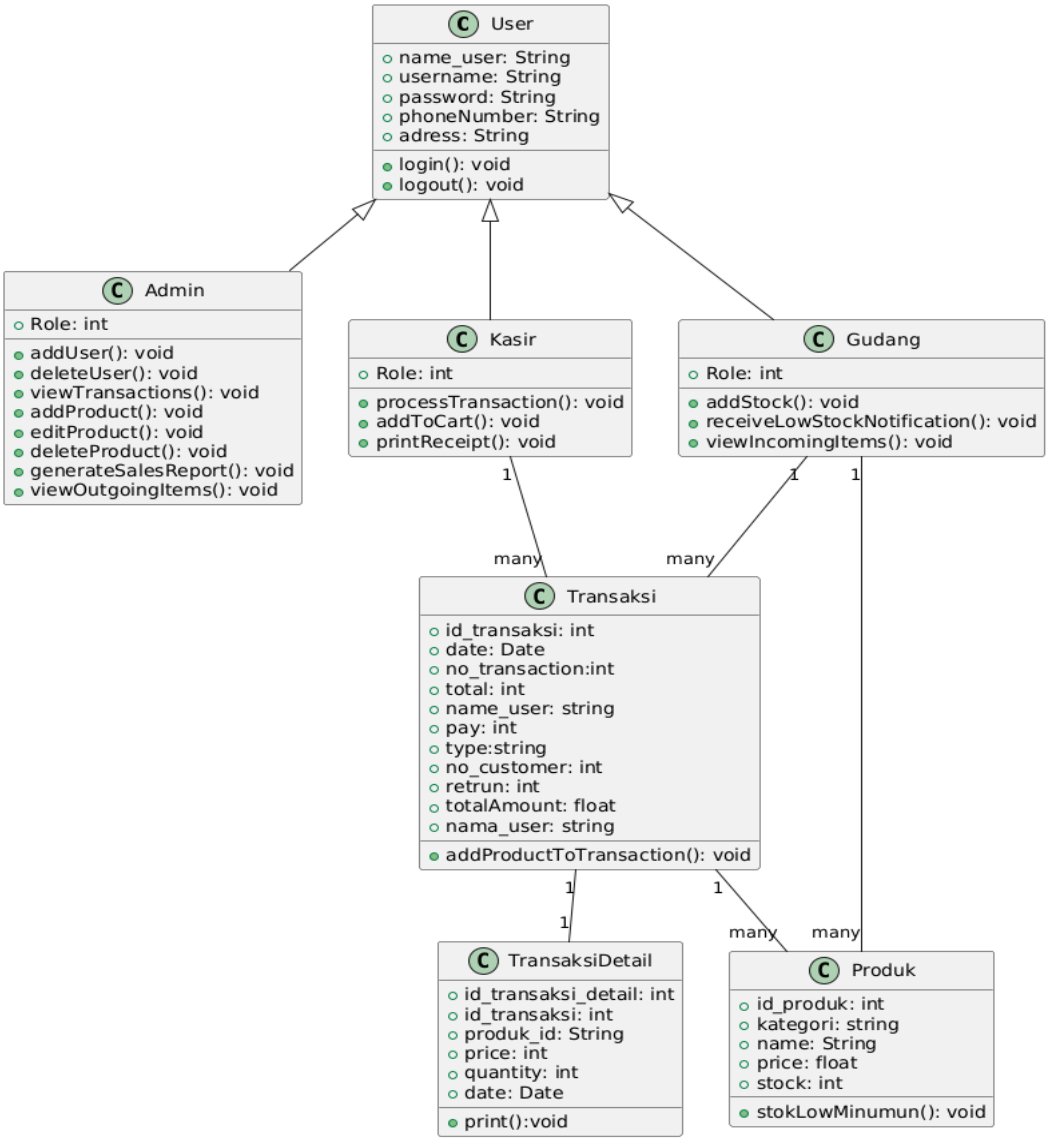


Gambar 1. Use Case Diagram

Aplikasi Point of Sales (POS) ini memiliki struktur menu pada gambar 1 yang terdiri dari tiga peran utama: Admin, Kasir, dan Gudang. Masing-masing peran melakukan tugas dan tugas yang berbeda. Untuk mengelola data pengguna dan produk melalui fitur *CRUD*, admin harus memantau riwayat transaksi, melihat barang yang keluar, mencetak laporan penjualan, dan melakukan proses login dan logout. Kasir bertanggung jawab untuk menambah produk ke dalam keranjang belanja, memproses transaksi penjualan, mencetak struk pembelian, dan mengelola sesi *login* dan *logout*. Sementara itu, Gudang bertanggung jawab untuk menambah stok barang, memantau stok barang yang keluar, menerima, dan mengubah stok barang. Pembagian tugas ini dibuat untuk menjamin bahwa proses operasional toko berjalan dengan baik dan terorganisir.

3.2.2 *Class Diagram*

UML Class Diagram pada gambar 2 dirancang dalam pengembangan website ini terdiri dari beberapa kelas utama, yaitu *User*, Admin, Kasir, Gudang, Transaksi, Transaksi detail, dan Produk**.** Setiap kelas memiliki atribut dan metode yang mendukung fungsionalitas sistem. *User* berfungsi sebagai parent *class* bagi Admin, Kasir, dan Gudang, yang memiliki peran dalam pengelolaan pengguna, transaksi, dan stok. Kelas Transaksi memiliki relasi *one-to-many* dengan TransaksiDetail dan *many-to-many* dengan Produk untuk memastikan pencatatan yang akurat. *UML Class Diagram* ini memastikan struktur sistem yang terorganisir dan efisien, sehingga mendukung pengembangan yang optimal sesuai kebutuhan fungsionalitasnya.

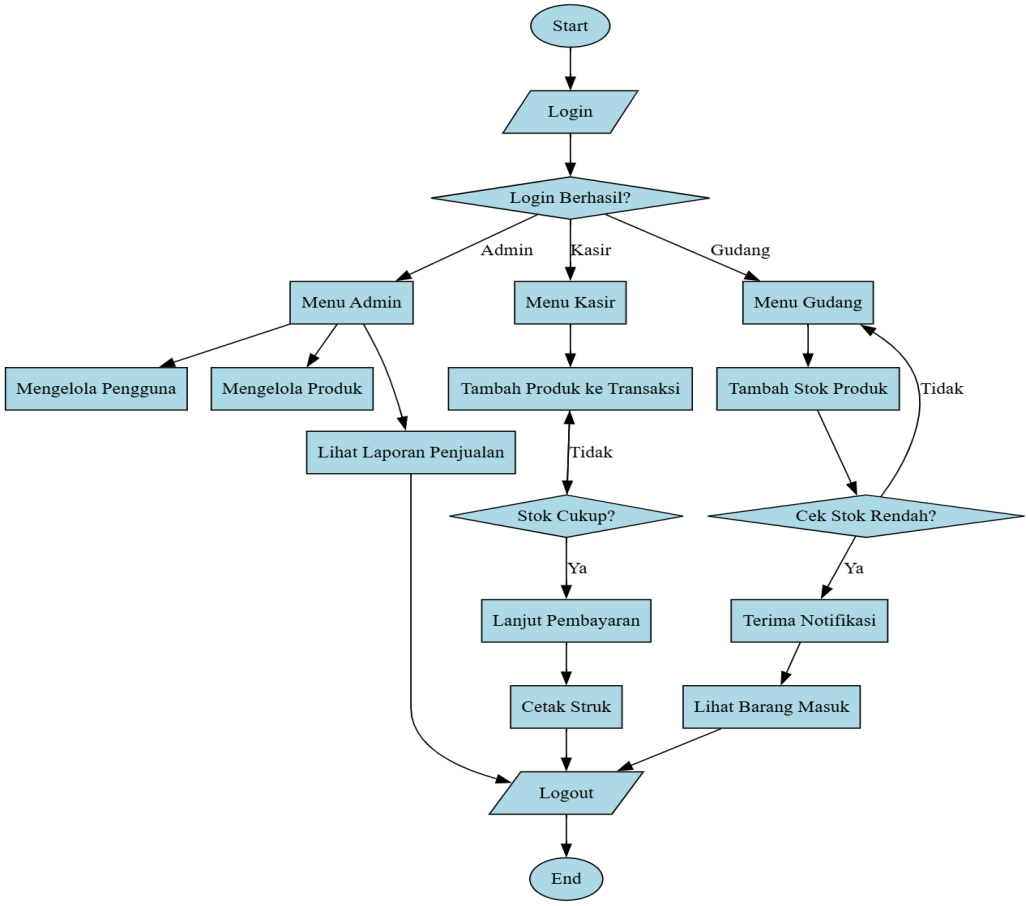


Gambar 2. Class Diagram

3.2.3 *FlowChart*

Pada perancangan pada gambar 3 *Flowchart* sistem ini, pengguna memulai proses dengan melakukan *login*, di mana sistem akan memverifikasi keabsahan *username* dan *password* yang dimasukkan. Jika *login* tidak valid, pengguna diminta untuk mengulang proses *login*. Setelah berhasil *login*, pengguna dapat memilih peran yang sesuai dengan hak aksesnya, yaitu Admin, Kasir, atau Gudang. Admin memiliki akses untuk mengelola pengguna dan produk, Kasir bertanggung jawab untuk memproses

transaksi, dan Gudang dapat mengelola stok serta menerima notifikasi terkait stok produk. Dalam proses transaksi, Kasir menambahkan produk yang dipilih, dan sistem akan memverifikasi ketersediaan stok produk. Jika stok mencukupi, transaksi dapat dilanjutkan hingga pembayaran selesai dan struk dicetak. Selain itu, Admin dapat menambah, mengedit, atau menghapus produk, sementara Gudang dapat menambah stok serta menerima notifikasi jika stok produk rendah. Setelah selesai, pengguna dapat *logout* dari sistem.



Gambar 3. *FlowChart*

3.3 *Development*

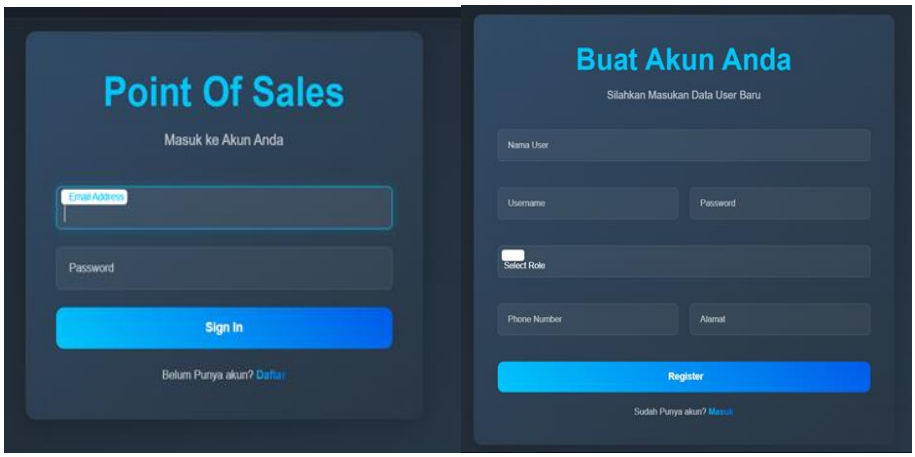
User Interface adalah komponen sistem yang memungkinkan pengguna berinteraksi langsung dengan sistem. Tampilan visual produk *User Interface* berfungsi sebagai penghubung antara sistem dan pengguna. Ini dapat berupa situs *web*, aplikasi, atau jenis perangkat lunak lainnya. *User Interface* terdiri dari semua komponen interaktif dan visual, seperti tombol, ikon, menu, dan tampilan layar yang digunakan pengguna untuk berinteraksi dengan aplikasi atau perangkat. Dengan membuat interaksi menjadi lebih mudah, efisien, dan intuitif, desain antarmuka pengguna yang baik sangat penting. Dalam perangkat lunak, *User Interface* mencakup elemen seperti layout, warna, tipografi, dan kontrol,
Seperti tombol dan kolom input.

User Experience adalah seluruh pengalaman pengguna yang subjektif saat berinteraksi dengan produk. Ini mencakup bagaimana orang melihat dan bertindak terhadap sistem serta pengalaman mereka saat menggunakannya. Dalam tulisan ini, perancangan atau pembangunan aplikasi dilakukan secara utuh dengan mempertimbangkan komponen *User Interface* dan *User Experience* yang disebutkan.

ISSN: 0216-2539 (Print)
E-ISSN: 2656-4157 (Online)

Tujuan dari setiap langkah dalam proses perancangan adalah untuk membuat antarmuka yang tidak hanya praktis, tetapi juga menarik dan mudah digunakan. Berikut adalah beberapa foto dari hasil implementasi atau tampilannya, yang menunjukkan bagaimana desain *User Interface* dan *User Experience* diintegrasikan secara keseluruhan dalam aplikasi.

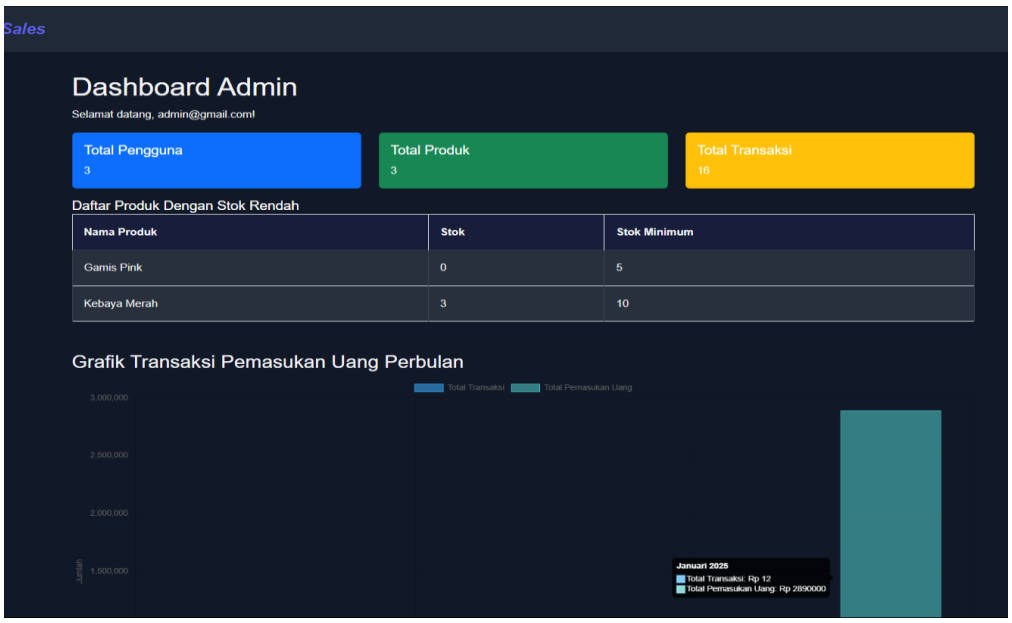
3.3.1 Tampilan *Login* dan *register*



Gambar 4. Tampilan *Login* dan *Register* Website

Halaman login merupakan pintu masuk utama ke dalam aplikasi *Point of Sales*. Halaman login ini menjadi komponen penting dalam menjaga keamanan dan memastikan hanya pengguna yang memiliki hak akses yang dapat masuk ke dalam sistem. Jika belum memiliki akun pegawai kasir diarahkan ke halaman *register* pada gambar 4 dimana mereka harus mengisi data diri untuk membuat akun baru.

3.3.2 Tampilan Admin



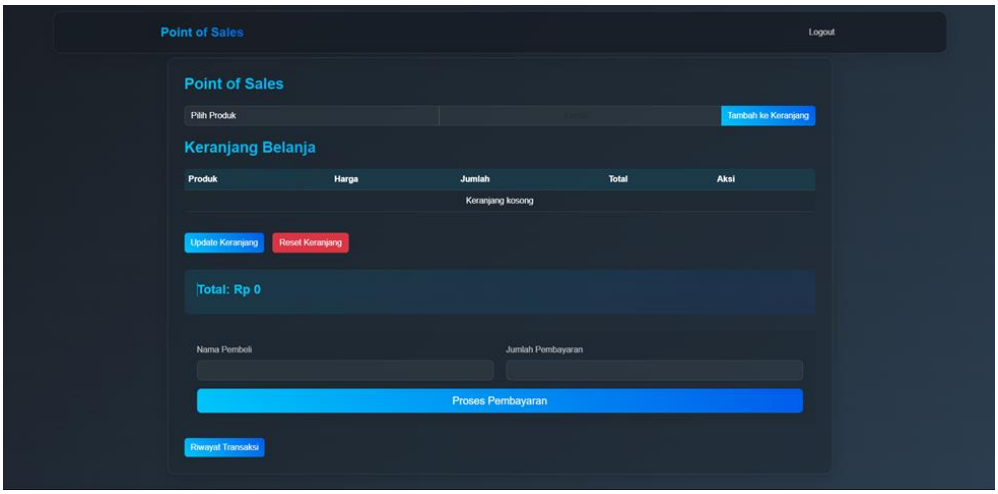
ISSN: 0216-2539 (Print)
E-ISSN: 2656-4157 (Online)

Gambar 5. Tampilan Admin

Gambar 5 merupakan halaman dashboard sebagai admin yang menampilkan data seperti, total pengguna, total produk, total transaksi, daftar produk stok rendah, grafik bulan penjualan. Yang terdapat juga *side bar* yang akan diarahkan ke halaman mengelola user, produk, melihat transaksi maupun cetak berdasarkan hari dan bulan.

Selain itu, halaman dashboard ini memiliki sidebar yang berfungsi sebagai navigasi yang memungkinkan admin mengakses berbagai halaman pengelolaan sistem. Halaman-halaman ini termasuk halaman di mana administrator dapat mengelola data pengguna, menambahkan atau memperbarui informasi produk, melihat daftar transaksi yang telah terjadi, dan mencetak laporan transaksi berdasarkan rentang waktu tertentu, seperti hari atau bulan.

3.3.3 Tampilan Admin



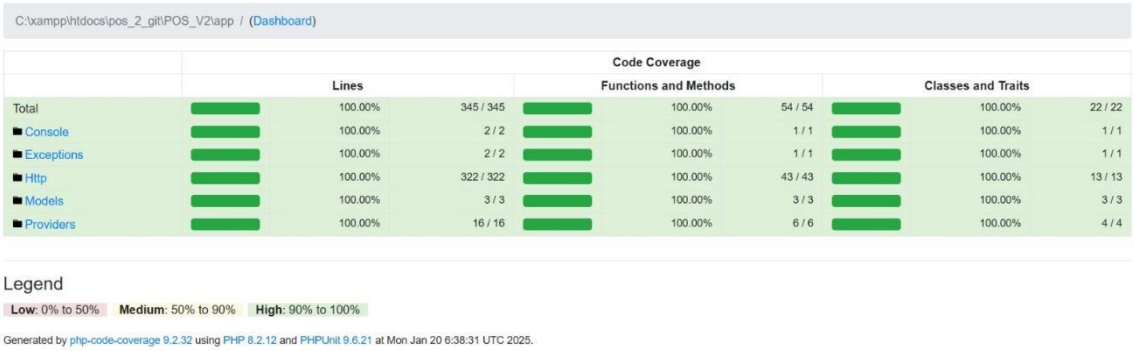
Gambar 6. Tampilan Kasir Website

Pada gambar 6 merupakan tampilan halaman dashboard kasir yang terdapat fitur tambah produk kedalam keranjang lalu bisa mengedit dan menghapus produk yang sudah ditambahkan ke dalam keranjang yang berfungsi untuk menginput barang yang akan dibeli oleh pembeli yang dilakukan oleh kasir.

3.4 Pengujian

Aplikasi *Point of Sales* berbasis *Laravel* diuji menggunakan teknik *unit testing* dengan *PHPUnit* dan dianalisis menggunakan *code coverage*. Pengujian ini berfokus pada evaluasi fungsi dan model aplikasi dengan mengkaji kode program. Tujuannya adalah memverifikasi bahwa aplikasi beroperasi sesuai dengan persyaratan yang ditetapkan, dengan mempertimbangkan berbagai skenario masukan dan hasil yang diharapkan. Metode ini efektif dalam mengidentifikasi berbagai kesalahan, seperti fungsi yang tidak berjalan, bug pada *User Interface*, serta masalah struktur data. Dengan *code coverage*, kita dapat memastikan bahwa pengujian mencakup sebagian besar kode, sehingga aplikasi dapat mengolah input secara akurat dan menghasilkan output yang tepat. Hasil pengujian ditampilkan pada gambar 7.

ISSN: 0216-2539 (Print)
E-ISSN: 2656-4157 (Online)



Gambar 7. Hasil Pengujian *Code Coverage*

Hasil kode *code coverage* pada gambar 7 menunjukkan bahwa kode aplikasi secara keseluruhan telah diuji dengan cakupan sebesar seratus persen pada semua aspek. Ini termasuk baris kode (345/345), fungsi dan metode (54/54), dan kelas dan sifat (22/22). Setiap direktori utama, seperti *Http*, *Models*, *Console*, *Exceptions*, dan *Providers*, memiliki cakupan yang sempurna. Hal ini menunjukkan bahwa pengujian yang dilakukan menggunakan *PHPUnit* dan *php-code-coverage* menunjukkan bahwa aplikasi memiliki kualitas dan stabilitas yang sangat baik. Warna hijau dalam laporan menegaskan pengujian yang menyeluruh (90%-100%), mencerminkan kualitas tinggi dalam pengujian otomatis dan keandalan kode aplikasi.

code coverage Controllers pada aplikasi ditunjukkan pada gambar 7, Setiap *controller* memiliki cakupan pengujian sepenuhnya untuk baris kode, fungsi, dan kelas. Dalam hasil pengujian tidak menemukan baris atau fungsi yang terlewatkan. Beberapa *controller* utama yang diuji, seperti *AdminDashboardController.php*, *GudangDashboardController.php*, *KasirDashboardController.php*, *ProductController.php*, dan *UserController.php*, memiliki cakupan yang sempurna. Menurut laporan ini, semua fitur yang dikelola oleh pengendali telah diuji secara menyeluruh untuk memastikan keandalan dan stabilitas di bagian pengelolaan logika aplikasi.

Direktori *Middleware* menunjukkan bahwa beberapa file, seperti *AdminCheck.php*, *Authenticate.php*, *GudangCheck.php*, dan *KasirCheck.php*, memiliki cakupan pengujian sebesar 100% untuk baris kode, fungsi, dan kelas. Namun, sejumlah file *middleware* lainnya tidak teruji seperti *EncryptCookies.php*, *PreventRequestsDuringMaintenance.php*, dan *VerifyCsrfToken.php*. Hal ini menunjukkan bahwa hanya *middleware* yang terkait langsung dengan otorisasi atau validasi peran yang telah diuji, sementara *middleware* lainnya belum mendapatkan perhatian dalam pengujian. Pada model seperti fungsi/metode, dan kelas/traits. File yang diuji, yaitu *Kategori.php*, *Produk.php*, *Transaksi.php*, dan *TransaksiDetail.php*, telah tercover sepenuhnya.

4. KESIMPULAN DAN SARAN

Dari penelitian yang dilakukan mengenai pengembangan aplikasi *Point of Sales* untuk toko baju menggunakan *Laravel*, dapat disimpulkan bahwa aplikasi ini berhasil mengatasi berbagai masalah operasional yang dihadapi oleh toko pakaian, seperti pengelolaan inventaris dan pencatatan transaksi secara manual yang sering menyebabkan kesalahan. Sistem *POS* yang dikembangkan mampu meningkatkan efisiensi operasional, akurasi transaksi, serta mempermudah pengelolaan stok barang dengan fitur-fitur seperti pencatatan barang masuk dan keluar, notifikasi stok rendah, serta laporan penjualan yang akurat dan mudah dipahami. Pendekatan metodologi *Agile* yang diterapkan selama pengembangan aplikasi memungkinkan iterasi dan perbaikan secara berkelanjutan untuk menyesuaikan kebutuhan pengguna. Pengujian unit dan keterlibatan pengguna dalam setiap tahap pengembangan juga memastikan bahwa aplikasi ini dapat digunakan dengan baik oleh berbagai peran (Admin, Kasir, Gudang) sesuai dengan hak akses dan tanggung jawab masing-masing. Dengan demikian, aplikasi *POS*

berbasis *Laravel* ini dapat diandalkan untuk meningkatkan kinerja dan efisiensi operasional toko pakaian.

Saran yang dapat diajukan untuk pengembangan aplikasi *POS* ini adalah pengembangan aplikasi mobile dan integrasi pembayaran digital. Dengan adanya aplikasi *POS* versi *mobile*, kasir dan staf toko dapat lebih mudah melakukan transaksi dan mengelola stok secara fleksibel tanpa terikat pada perangkat *desktop*. Selain itu, integrasi pembayaran digital seperti *e-wallet*, kartu kredit, atau metode pembayaran online lainnya akan mempermudah pelanggan dalam melakukan pembayaran, meningkatkan kenyamanan transaksi, serta mempercepat proses pembayaran. Kedua saran ini diharapkan dapat meningkatkan efisiensi dan pengalaman pengguna dalam menjalankan operasional toko pakaian.

5. REFERENSI

Alaidaros, H., Omar, M., & Romli, R. (2021). A Review on the Methods of Evaluating the New Approaches Proposed in the Agile Context. *International Journal of Computer and Information Technology(2279-0764)*, *10*(1), 5–10. https://doi.org/10.24203/ijcit.v10i1.54

Ahnaf, F. A., & Puspitasari, E. D. (2023). Descriptive Analysis of Inventory of Equipment and Materials for Biology Laboratory Activities in High Schools throughout Banjarnegara Regency. Symposium on Biology Education (Symbion), 3, 285. https://doi.org/10.26555/symbion.11745

Kucharska, B. (2024). Consumers of different generations towards innovative technologies in customer service in retail. *Ekonomia*, *29*(3), 19–33. https://doi.org/10.19195/2658-1310.29.3.2

Budiarti, R. P. N., & Kamila, F. I. R. (2024). Transformative Implementation of Android-Based Point of Sale System at Shafira's Buffet Stall. *E3S Web of Conferences*, *482*. https://doi.org/10.1051/e3sconf/202448202002

Dragičević, Z., & Bošnjak, S. (2019). Agile architecture in the digital era: Trends and practices. *Strategic Management*, *24*, 12–33. https://doi.org/10.5937/StraMan1902011D

Firsty Giovanni Naoki Panca Putra, Jaswadi, & Ludfi Djajanto. (2023). Design and build a Point of Sales (POS) application model using the agile method for super mama frozen food. *World Journal of Advanced Engineering Technology and Sciences*, *9*(2), 142–156. https://doi.org/10.30574/wjaets.2023.9.2.0212

Hasbi, H. M., & Istabul, M. R. (2023). Design of a Website-Based Point-of-Sale Information System at "Vapeboss" Vape Shop Using the Waterfall Method. *Journal of Social Research*, *2*(9), 3146–3153. https://doi.org/10.55324/josr.v2i9.1366

Pinton, M., & Torres, A. S. (2020). Human aspects of agile transition in traditional organizations. *Journal of Technology Management and Innovation*, *15*(3), 62–73. https://doi.org/10.4067/s0718-27242020000300062

Rangavittal, P. B. (2022). Agile Software Development in Large Organizations: Challenges and Strategies for Success. *Journal of Engineering and Applied Sciences Technology*, *4*(4), 1–7. https://doi.org/10.47363/jeast/2022(4)232

Salim, S., & Frederica, D. (2020). *How is the Impact of Non-Cash Payment System on Sales of Micro, Small and Medium Enterprise?* *151*(Icmae), 188–191. https://doi.org/10.2991/aebmr.k.200915.044

Saputra, M., & Fadlila, R. M. (2023). An Effective Open ERP System for Automation in Financial Reporting for SMEs based on Service Oriented Architecture. *International Journal on Informatics Visualization*, *7*(3–2), 2007–2015. https://doi.org/10.30630/joiv.7.3-2.2367

Schwaber, K., & Sutherland, J. (2017). The Scrum Guide: The Definitive The Rules of the Game. *Scrum.Org and ScrumInc*, *November*, 19. http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-US.pdf

Subianto, T. A. D. P., Wiratama, J., & Halim, F. A. (2023). The Development of web-based Cashier and Inventory Information Systems using Prototyping Model on Micro, Small, and Medium Enterprise (MSMEs) in Indonesia. *JOINS (Journal of Information System)*, *8*(1), 80–89. https://doi.org/10.33633/joins.v8i1.7983

Wulandari, R., Maesaroh, S., & Erwansyah, E. (2021). Point Of Sales (POS) Application On Clients of

PT. Esensi Solusi Buana. *CCIT Journal*, *14*(1), 48–56. https://doi.org/10.33050/ccit.v14i1.1092

Zakrzewska, M., Jarosz, S., Piwowar-Sulej, K., & Sołtysik, M. (2022). Enterprise agility – its meaning, managerial expectations and barriers to implementation – a survey of three countries. *Journal of Organizational Change Management*, *35*(3), 488–510. https://doi.org/10.1108/JOCM-02-2021-0061