

PERANCANGAN SISTEM & USER INTERFACE UNTUK APLIKASI KASIR DAN INVENTARIS PADA BENGKEL BARU MOTOR SPORT

Rafael Wun¹⁾, Tony²⁾

^{1, 2)} Program Studi Sistem Informasi Fakultas Teknologi Informasi Universitas Tarumanagara
Jl. Letjen S. Parman No. 1, Jakarta Barat 11440 Indonesia
email : rafael.825210075@stu.untar.ac.id¹⁾, tony@fti.untar.ac.id²⁾

ABSTRACT

An easy-to-use and efficient interface is essential for cashiering and inventory management systems, especially in motorcycle workshops that have unique requirements for tracking goods and processing sales transactions. This research focuses on designing a web-based application interface to streamline sales transactions and inventory management at the Baru Motor Sport workshop. The User Interface (UI) simplifies core processes, including recording incoming and outgoing goods, managing inventory, and processing sales transactions in real-time. Vue.js is used to create a responsive and intuitive front-end, supported by Tailwind CSS for a modern design. The design process followed the Scrum methodology, which emphasizes iterative feedback, flexibility, and continuous improvement to align the UI with user needs. The interface accommodates three main roles: Admin, Inventory Staff, and Cashier Staff. Each role's UI is customized to facilitate user-specific tasks, such as inventory tracking for Inventory Staff and transaction handling for Cashier Staff, while Admin oversees user management and views sales reports. This UI design aims to improve operational efficiency, enhance usability, and support accurate and efficient workflows in the workshop.

Key words

cashier, inventory, motorcycle, user interface, Design

1. Pendahuluan

Digitalisasi telah memicu perubahan yang signifikan dalam dunia bisnis, terutama dengan penerapan teknologi informasi untuk meningkatkan efisiensi dan akurasi operasional, serta mengurangi ketergantungan pada metode manual yang rentan terhadap berbagai kesalahan [1]. Bagi usaha kecil dan menengah (UKM), seperti bengkel motor, sistem digital menjadi kebutuhan penting untuk meningkatkan daya saing serta efisiensi operasional [2]. Hal ini dapat diterapkan pada bengkel Baru Motor Sport Tenggarong, yang menghadapi tantangan pengelolaan stok barang secara manual, yang sering kali menyebabkan ketidakakuratan data.

Sejak berdiri pada tahun 2014, masih mengandalkan pencatatan manual untuk pengelolaan transaksi penjualan dan stok barang, yang berisiko menyebabkan terjadinya

kekurangan atau penumpukan stok yang tidak diperlukan, yang mempengaruhi peningkatan biaya operasional. Dengan semakin tingginya permintaan layanan bengkel seiring bertambahnya jumlah kendaraan roda dua di Indonesia, BMS memerlukan solusi digital yang dapat mengoptimalkan pengelolaan stok dan transaksi serta memberikan respon yang lebih cepat terhadap kebutuhan pelanggan.

Salah satu solusi untuk mengatasi masalah ini adalah perancangan *user interface* (UI) untuk aplikasi kasir dan inventaris berbasis web. Perancangan ini menggunakan teknologi Figma untuk desain *wireframe* dan *mockup* aplikasi, yang akan dikembangkan dengan Vue.js dan Tailwind CSS untuk meningkatkan akurasi pencatatan, mempercepat proses transaksi, dan menghasilkan laporan yang lebih tepat. Dengan menggunakan metodologi Scrum dalam perancangan sistem dan UI aplikasi, sehingga dapat dilakukan secara iteratif, yang memungkinkan penyesuaian berkelanjutan berdasarkan umpan balik pengguna untuk menghasilkan antarmuka yang efektif, *user-friendly*, dan sesuai dengan kebutuhan operasional bengkel.

Pada perancangan sistem dan *user interface* untuk aplikasi kasir dan inventaris pada bengkel Baru Motor Sport, menggunakan beberapa referensi yang dapat dilihat sebagai berikut:

Maximillian *et al.* [3] menyoroti pengembangan aplikasi kasir berbasis desktop yang dirancang khusus untuk mendukung pencatatan transaksi di bengkel motor. Aplikasi ini dikembangkan menggunakan Python, PyQt5, dan MySQL sebagai basis datanya. Fitur utamanya mencakup pengelolaan data transaksi dan pembuatan laporan keuangan, yang secara signifikan membantu dalam meningkatkan efisiensi operasional bengkel. Implementasi aplikasi *desktop* ini memperlihatkan bahwa teknologi sederhana namun fungsional dapat memberikan dampak positif pada peningkatan produktivitas dan pengelolaan bisnis.

Abdurrahman [4] mengembangkan aplikasi kasir berbasis Android untuk mempermudah proses transaksi di toko ritel melalui digitalisasi pencatatan penjualan. Aplikasi ini dibangun menggunakan bahasa pemrograman Java dan SQLite sebagai basis datanya. Fitur-fitur utama yang ditawarkan mencakup pengelolaan produk, pembuatan laporan penjualan, dan kemampuan mencetak struk

dengan printer Bluetooth. Aplikasi ini bertujuan untuk memberikan kemudahan dalam manajemen produk dan pelaporan penjualan secara lebih efisien dan efektif, sehingga meningkatkan kelancaran operasional toko ritel. Low *et al.* [5] mengembangkan sebuah aplikasi inventori yang bertujuan mengoptimalkan manajemen stok dan pencatatan barang. Aplikasi ini dibangun menggunakan PHP, Bootstrap, dan MySQL, serta memiliki lima modul utama, yaitu *login*, inventori, kasir, pemasok, dan pembuatan laporan. Fitur-fitur ini memungkinkan pengguna untuk mengelola data produk, pemasok, dan melihat laporan inventaris secara rinci. Dengan modul yang lengkap, aplikasi ini memberikan kemudahan dalam mengelola stok dan meningkatkan efisiensi pencatatan inventaris pada bisnis pengguna.

Anisah [6] mengembangkan aplikasi pengelolaan inventaris menggunakan metode *Rapid Application Development* (RAD). Aplikasi ini dibangun dengan PHP dan MySQL, serta dirancang untuk memudahkan pengguna dalam mencatat barang masuk dan keluar, serta melihat laporan stok secara cepat melalui antarmuka yang sederhana. Metode RAD mendukung proses pengembangan yang cepat dan responsif terhadap kebutuhan pengguna, menjadikan aplikasi ini efisien dalam membantu pengelolaan inventaris.

Kinselton [7] mengembangkan aplikasi *web* untuk meningkatkan efisiensi pencatatan penjualan barang pada *My Sport* Indonesia. Aplikasi ini memungkinkan pengguna untuk mengelola data produk, transaksi penjualan, serta melihat laporan penjualan secara otomatis. Dikembangkan menggunakan HTML, CSS, PHP, dan MySQL, aplikasi ini mengadopsi metode SDLC *waterfall* dalam proses pengembangannya, yang memastikan pengembangan yang terstruktur dan sistematis untuk mencapai hasil yang optimal dalam manajemen penjualan.

2. Tinjauan Pustaka

2.1. Bengkel Motor

Berdasarkan Kamus Besar Bahasa Indonesia (KBBI), bengkel adalah tempat yang digunakan untuk memperbaiki kendaraan, seperti mobil, sepeda motor, dan lainnya. Dengan meningkatnya jumlah kendaraan bermotor, bengkel menjadi sektor yang sangat vital bagi masyarakat Indonesia, khususnya bengkel yang mengkhususkan diri pada perawatan sepeda motor. Sepeda motor menjadi jenis kendaraan yang paling banyak digunakan di Indonesia, terbukti dari data Badan Pusat Statistik (BPS) yang mencatat lebih dari 120 juta unit sepeda motor pada tahun 2021 hingga 2022. Jumlah ini jauh lebih banyak dibandingkan dengan mobil yang hanya sekitar 15 juta unit [8]. Keberadaan bengkel motor sangat penting untuk menjaga kondisi sepeda motor agar tetap optimal. Perawatan rutin yang dilakukan di bengkel dapat membantu pengendara memastikan kendaraannya selalu dalam kondisi prima, sehingga dapat mengurangi risiko kerusakan saat berkendara di jalan [9].

2.2. User Interface (UI)

Desain UI merupakan elemen-elemen visual yang membentuk tampilan dari sebuah produk digital, yang bertujuan untuk menciptakan pengalaman pengguna yang efisien dan menyenangkan. Desain UI meliputi pengaturan tata letak, penggunaan warna, tipografi, ikonografi, dan elemen-elemen grafis lainnya yang membantu pengguna berinteraksi dengan sistem atau aplikasi. Tujuan utama dari desain antarmuka pengguna adalah untuk menyederhanakan interaksi antara pengguna dan perangkat lunak, membuatnya intuitif, dan meningkatkan kenyamanan pengguna saat mengakses fitur-fitur yang tersedia [10].

2.3. Website

Website adalah sekumpulan halaman web yang saling terhubung dan dapat diakses melalui *World Wide Web*, yang merupakan jaringan informasi global yang memungkinkan pengguna untuk memperoleh berbagai informasi dari internet. Halaman *web* atau *web pages* merupakan komponen dasar dari sebuah website, yang dibangun menggunakan HTML. Setiap halaman dalam website memiliki identifikasi unik yang dapat diakses melalui URL, yang terdiri dari beberapa bagian penting, yaitu protokol, nama *domain*, dan *subdomain*. Nama *domain* berfungsi sebagai alamat yang memudahkan browser untuk menemukan dan menampilkan file yang diinginkan. Sedangkan *subdomain* digunakan untuk mengatur dan membagi konten dalam website menjadi bagian-bagian yang lebih terstruktur [11][12].

2.4. Vue JS

Vue JS adalah sebuah *framework* JavaScript yang dirancang khusus untuk membangun antarmuka pengguna (UI) dengan pendekatan berbasis komponen. Setiap elemen dalam UI dapat dipisahkan menjadi komponen yang dapat digunakan kembali (*reusable*), yang mempermudah pengembangan dengan memberikan struktur yang jelas dan memudahkan pemeliharaan. Dengan sistem reaktif yang dimilikinya, Vue.js dapat mengelola perubahan data dan secara otomatis memperbarui tampilan sesuai dengan perubahan tersebut. Vue.js juga dilengkapi dengan berbagai fitur tambahan, seperti Vue *Router* untuk manajemen *routing* dan Vuex untuk manajemen *state* pada aplikasi yang lebih kompleks. Fitur komponen yang dapat digunakan kembali ini juga mempermudah dalam mendeteksi dan memperbaiki kesalahan yang muncul selama proses pengembangan, karena setiap komponen dikembangkan secara terpisah [13].

2.5. Tailwind CSS

Tailwind CSS adalah sebuah *framework* yang menawarkan fleksibilitas tinggi dalam pembuatan desain UI, dengan pendekatan *utility-first*. Pendekatan ini memungkinkan pengembang untuk langsung menerapkan gaya pada elemen atau *tag* HTML menggunakan atribut *class*. Tailwind mendukung desain responsif dengan

memungkinkan penyesuaian gaya untuk berbagai ukuran layar melalui penambahan *prefix*. Selain itu, Tailwind juga mendukung mode gelap, yang memungkinkan transisi antara mode terang dan gelap sesuai dengan preferensi pengguna [14].

3. Metode Penelitian

Metode penelitian yang digunakan dalam pengembangan aplikasi kasir dan inventaris berbasis *web* pada bengkel Baru Motor Sport (BMS) adalah metode *Scrum*. *Scrum* adalah *framework* yang digunakan untuk mengelola pekerjaan pada suatu proyek, dengan kemampuan adaptasi terhadap masalah yang kompleks [15]. Metode ini dipilih karena sifatnya yang adaptif terhadap perubahan kebutuhan pengguna dan kemampuannya dalam membagi pekerjaan menjadi *sprint* pendek yang terukur. Proses pengembangan aplikasi dimulai dengan tahap *planning*, dengan mengumpulkan kebutuhan sistem melalui wawancara dengan pemilik bengkel, informasi yang dikumpulkan kemudian disusun pada *product backlog*, yang berisi daftar dan fitur yang akan diimplementasikan dalam aplikasi.

Berikut dilakukan penjadwalan *sprint/sprint planning*, dengan memilih fitur-fitur yang akan dikerjakan dalam *sprint*. Setiap *sprint* memiliki durasi 3 hingga 4 minggu, yang dilaksanakan secara iteratif/berulang hingga seluruh fitur pada *backlog* selesai. Pada akhir setiap *sprint*, dilakukan sesi *review* dengan pemilik bengkel untuk meninjau hasil pengembangan dan mengumpulkan *feedbacks*, yang digunakan untuk memperbarui *backlog* dan perencanaan *sprint* berikut. Proses umum metode *Scrum* dapat dilihat pada **Gambar 1**.



Gambar 1 Gambaran umum metode *Scrum*

Selain itu, dalam penelitian ini akan menggunakan *Unified Modeling Language (UML)*. *UML* adalah bahasa pemodelan standar yang digunakan dalam perancangan sistem perangkat lunak dengan memvisualisasikan struktur dan perilaku sistem pada diagram. *UML* menyediakan berbagai notasi diagramatik yang dapat digunakan untuk menentukan spesifikasi sistem yang dikembangkan kepada para stakeholders [16]. Pada penelitian ini, akan menggunakan *Use Case Diagram*, *Activity Diagram*, *Class Diagram*, dan *Sequence Diagram*.

Use Case Diagram digunakan untuk menggambarkan interaksi antara aktor/pengguna dengan sistem, yang membantu dalam memahami fungsionalitas utama yang

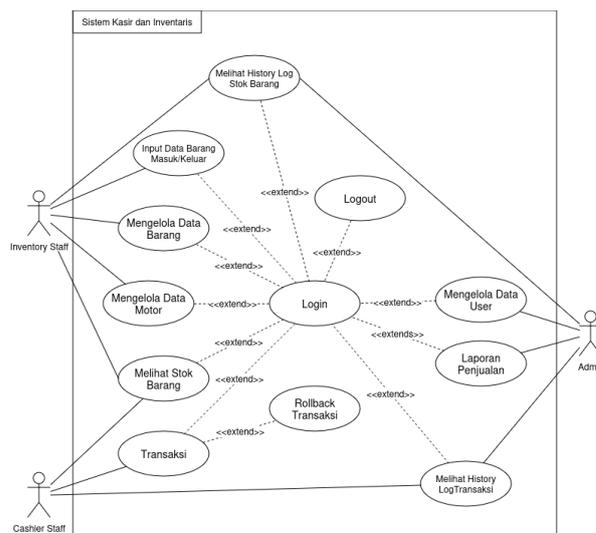
diharapkan dari aplikasi. *Activity Diagram* digunakan untuk mengidentifikasi aktivitas-aktivitas dalam setiap kebutuhan fungsional sistem secara berurutan dan hasil yang diharapkan dari setiap tindakan. *Class Diagram* digunakan untuk menggambarkan struktur internal aplikasi, termasuk entitas/kelas beserta atribut dan hubungan antar entitas. Sementara itu *Sequence Diagram*, digunakan mengidentifikasi setiap objek yang terlibat dengan suatu interaksi dalam suatu skenario/*use case* secara berurutan [16].

4. Hasil Perancangan

Hasil perancangan aplikasi kasir dan inventaris berbasis *web* pada bengkel Baru Motor Sport mencakup beberapa diagram *UML*. Diagram-diagram ini digunakan untuk memodelkan fitur dan interaksi utama dalam aplikasi, serta mendokumentasikan arsitektur sistem secara menyeluruh. Adapun diagram yang disertakan dalam hasil perancangan meliputi *use case diagram*, *activity diagram*, *class diagram*, dan *sequence diagram*.

4.1. Use Case Diagram

Gambar 2 menunjukkan tiga aktor yang terlibat dalam aplikasi meliputi, *Admin*, *Inventory Staff*, dan *Cashier Staff*. *Admin*, peran dengan otoritas tertinggi mampu mengakses keseluruhan fitur, namun pada diagram digambarkan beberapa fitur unik yang hanya dapat diakses oleh *Admin*. Sementara itu, *Inventory Staff* yang bertanggung jawab dalam pengelolaan stok dan data barang meliputi, pengelolaan data motor, data barang, melihat stok barang, dan *history log* barang, serta *input* stok barang yang masuk dan/atau keluar. Sedangkan *Cashier Staff*, yang bertugas dalam melakukan transaksi dapat mengakses beberapa fitur mencakup transaksi, *rollback* transaksi, melihat stok barang, dan *history log* transaksi.

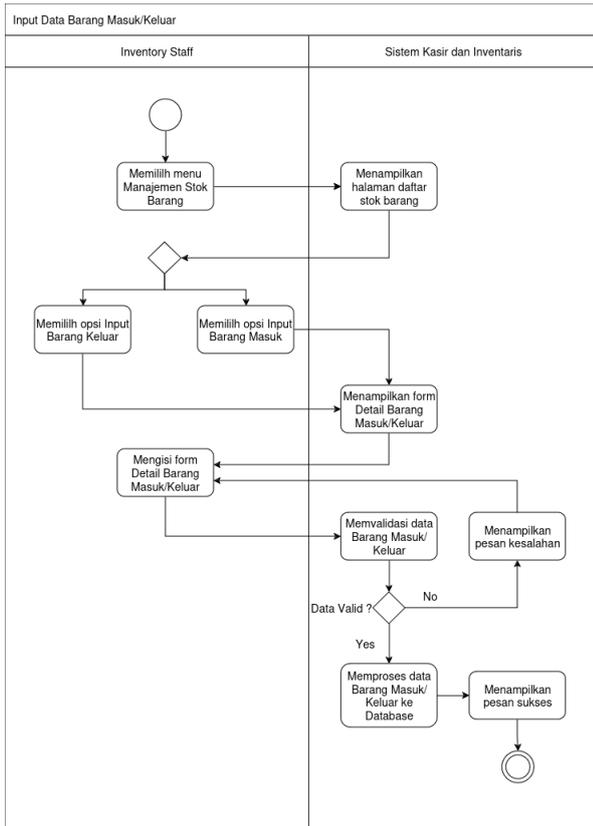


Gambar 2 *Use Case Diagram*

4.2. Activity Diagram

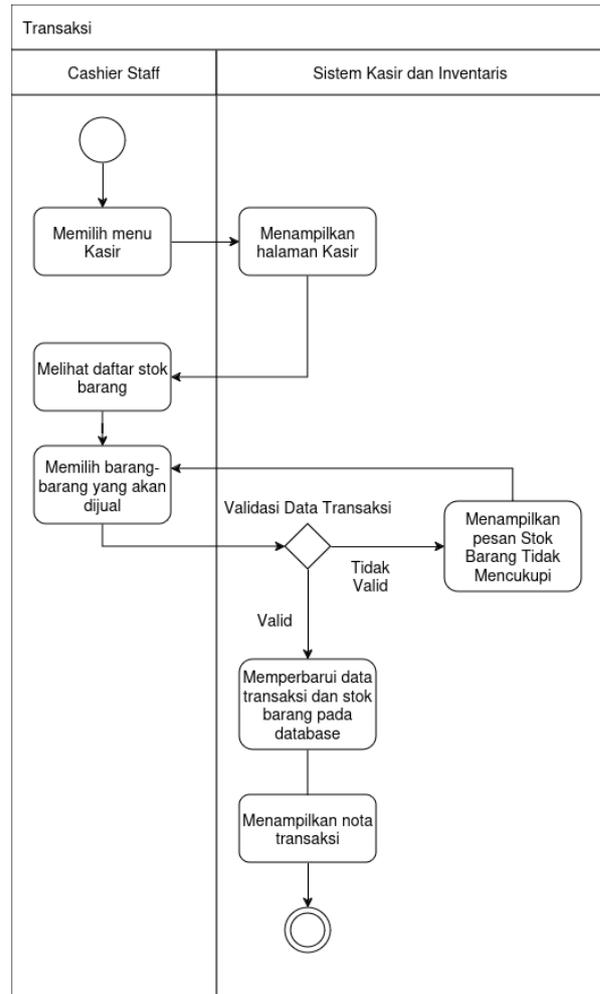
Gambar 3 menunjukkan alur aktivitas yang dilakukan oleh *inventory staff* saat mencatat barang yang masuk ke

dalam stok ataupun keluar dari stok untuk keperluan servis. Proses dimulai dengan mengakses halaman pengelolaan barang, lalu, sistem menampilkan daftar barang dari *database*. Kemudian, aktor dapat memilih untuk memasukkan data barang masuk atau barang keluar dengan menekan tombol yang disediakan. Dan aktor akan melakukan *input* data barang meliputi, nama barang, dan jumlah penambahan/pengurangan stok. Setelah proses input selesai, sistem akan memvalidasi dan menyimpan data tersebut dalam *database* jika valid, serta menginformasikan pembaruan stok dengan mencatat pembaruan pada *history log* barang.



Gambar 3 Activity Diagram Input Data Barang

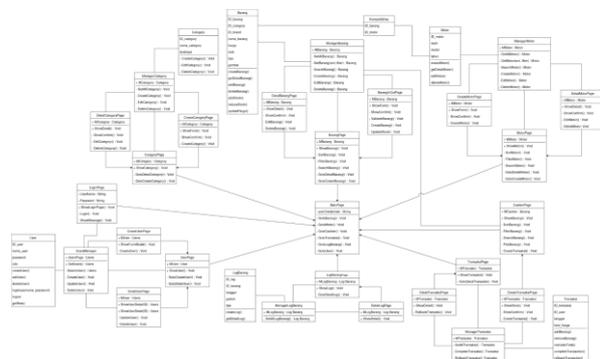
Gambar 4 menunjukkan alur proses yang terjadi saat *cashier staff* melakukan transaksi penjualan, yang dimulai dari mengakses halaman kasir, kemudian aktor akan memilih barang, dan memasukkan jumlah barang yang akan dijual. Setelah pemilihan barang, sistem secara otomatis memeriksa kembali ketersediaan stok dan menghitung total harga transaksi tersebut. Kemudian, sistem akan mengurangi stok barang yang terjual, mencatat *log* transaksi dan menghasilkan struk transaksi sebagai bukti.



Gambar 4 Activity Diagram Transaksi

4.3. Class Diagram

Gambar 5, menggambarkan entitas/objek utama dan hubungan antar objek melalui halaman yang ada pada aplikasi, meliputi *User*, *Category*, *Barang*, *Motor*, *Transaksi* dan *Log Barang*.

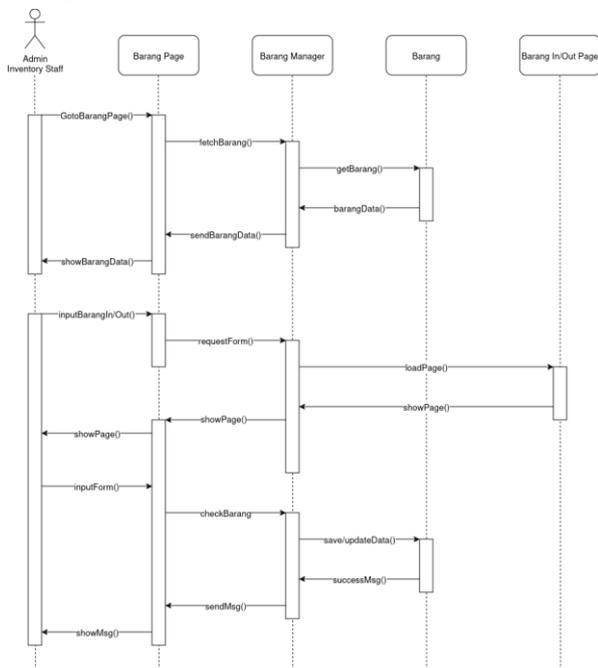


Gambar 5 Class Diagram

4.4. Sequence Diagram

Gambar 6, digambarkan alur proses pencatatan barang masuk dan keluar yang melibatkan interaksi antara beberapa objek, yaitu *Inventory Staff*, *Halaman Barang*, *Barang Manager*, *Barang*, dan *Halaman BarangInOut*. Proses dimulai ketika *Inventory Staff* mengakses Halaman

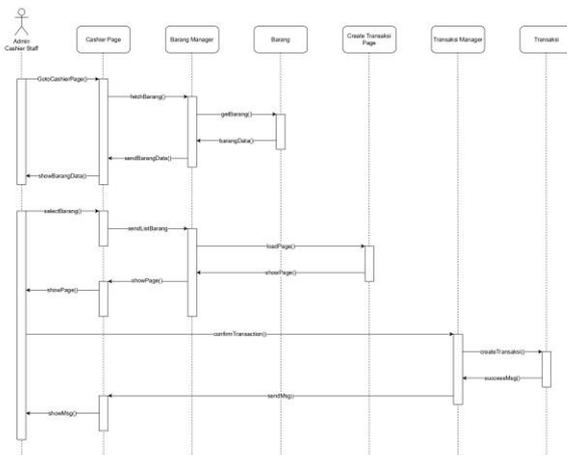
Barang untuk melihat daftar barang yang ada. Sistem kemudian akan melakukan permintaan **fetchBarang()** ke Barang Manager, yang berfungsi untuk mengambil data barang dari sistem. Barang Manager akan mengirimkan permintaan **getBarang()** yang mengembalikan data seluruh barang yang tersedia pada Halaman Barang, sehingga data tersebut bisa ditampilkan kepada *Inventory Staff*. Setelah barang-barang ditampilkan, *Inventory Staff* memiliki opsi untuk melakukan pencatatan barang masuk atau keluar. *Inventory Staff* dapat menggunakan fitur input data barang yang ada di Halaman *BarangIn/Out*. Proses ini dimulai dengan memilih barang yang akan diubah stoknya (baik untuk barang masuk maupun barang keluar), kemudian memasukkan jumlah barang yang relevan. Setelah input data barang selesai, sistem akan memproses dan memperbarui stok barang melalui *Barang Manager*. Perubahan ini disimpan dalam basis data dan akan tercatat dalam *log* stok barang, memastikan bahwa setiap perubahan tercatat dengan akurat.



Gambar 6 Sequence Diagram BarangIn/Out

Gambar 7, digambarkan alur proses transaksi penjualan pada sistem kasir. Proses ini dimulai oleh *cashier* mengakses halaman *Cashier* untuk melihat daftar barang yang tersedia di sistem. Untuk mendapatkan data barang, *Cashier Page* akan mengirimkan permintaan **fetchBarang()** ke *Barang Manager*, yang kemudian melanjutkan permintaan tersebut ke objek *Barang* melalui fungsi **getBarang()**. *Barang* akan mengembalikan data semua barang yang tersedia kepada *Barang Manager*, yang kemudian menampilkannya kembali pada halaman *cashier*. Dengan informasi tersebut, *Cashier* dapat melihat dan memilih barang-barang yang akan ditransaksikan. Setelah memilih barang yang ingin dijual, *Cashier* kemudian mengkonfirmasi transaksi melalui *Create Transaksi Page*. *Create Transaksi Page* akan mengirimkan permintaan **createTransaksi()** ke

Transaksi Manager, yang bertanggung jawab untuk memproses transaksi tersebut. *Transaksi Manager* kemudian menyimpan transaksi baru di objek *Transaksi* dan juga memperbarui pada stok barang melalui interaksi dengan *Barang Manager*. Setelah transaksi berhasil diproses, *Cashier Page* akan menampilkan struk transaksi yang mencantumkan detail barang yang dibeli, jumlah, harga, dan total yang harus dibayar. Proses ini mengakhiri keseluruhan transaksi.



Gambar 7 Sequence Diagram Transaksi Penjualan

4.5. Database Design

Selain melakukan pemodelan sistem, perancangan aplikasi kasir dan inventaris juga menggunakan *conceptual* dan *logical database design*. *Conceptual database design* bertujuan untuk mengidentifikasi *entities*, *relationships* serta *attributes* [17]. Proses diawali dengan mengidentifikasi *entities* yang menggambarkan setiap objek dalam sistem, hasil identifikasi dapat dilihat pada **Tabel 1**. Diikuti dengan identifikasi hubungan/*relationship* (R) antar *entity* (E), serta *multiplicity* (M) yang dapat dilihat pada **Tabel 2**.

Tabel 2 Identifikasi Entity

Entity	Deskripsi	Kemunculan
Motor	Informasi tentang model motor yang tersedia untuk stok aksesoris di bengkel BMS.	Data motor dapat ditambahkan dan dimodifikasi oleh <i>admin</i> , dan <i>inventory staff</i>
Kompatibilitas	Informasi tentang kompatibilitas antara barang dengan model motor	Data kompatibilitas yang muncul pada saat pengisian data barang dengan memilih tipe barang jika tidak

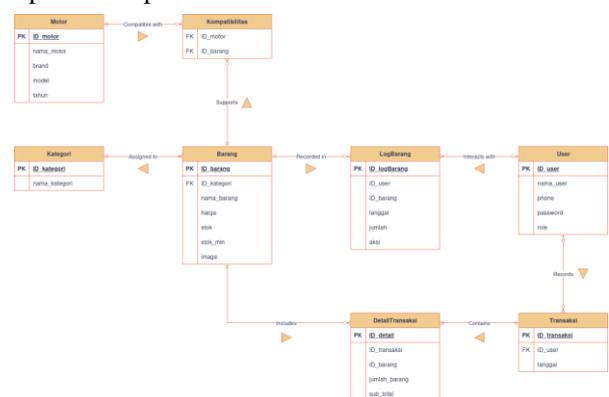
		universal maka admin dan/atau inventory staff akan mengisi model motor yang kompatibel dengan barang yang dimasukkan
Barang	Informasi tentang barang yang disediakan bengkel BMS	Data barang dapat ditambahkan dan dimodifikasi oleh <i>admin</i> , dan <i>inventory staff</i>
Kategori	Informasi tentang kategori barang, seperti tipe-tipe aksesoris, <i>sparepart</i> , dan jenis barang lainnya yang tersedia di bengkel.	Data kategori barang dikelola oleh admin dan inventory staff.
LogBarang	Informasi tentang perubahan stok barang yang terjadi, termasuk penambahan barang masuk, barang keluar dan penjualan barang	Data <i>stocks history log</i> dapat diakses (<i>view</i>) oleh <i>admin</i> dan <i>inventory staff</i>
Transaksi	Informasi tentang transaksi penjualan produk pada bengkel BMS	Data transaksi dapat ditambahkan oleh <i>cashier staff</i> dan diperiksa oleh <i>admin</i>
DetailTransaksi	Rincian lebih lanjut dari setiap transaksi yang mencakup item barang yang dibeli, harga, dan jumlah unit yang terlibat	Data detail transaksi dapat ditambahkan oleh <i>cashier staff</i> , dan diperiksa oleh admin.

	dalam transaksi tersebut.	
<i>User</i>	Informasi tentang pengguna aplikasi kasir dan inventaris pada bengkel BMS	Data pengguna dapat ditambahkan dan dimodifikasi oleh <i>admin</i>

Tabel 2 Identifikasi *Relationship* antar *Entity*

<i>E</i>	<i>M</i>	R	<i>M</i>	<i>E</i>
Motor	1..1	<i>Compatible with</i>	0..*	Kompatibilitas
Barang	1..1	<i>Supports</i>	0..*	Kompatibilitas
Barang	1..*	Assigned to	1..1	Kategori
Barang	1..1	Recorded in	0..*	LogBarang
User	0..1	Interacts with	0..*	LogBarang
User	0..1	Records	0..*	Transaksi
Transaksi	1..1	Contains	1..*	Detail Transaksi
Barang	1..1	Includes	0..*	Detail Transaksi

Logical Database Design bertujuan untuk merancang skema relasi antar tabel yang telah diidentifikasi pada tahap *Conceptual Database Design* dalam bentuk *entity relationship diagram* [17]. Hasil perancangan ER diagram dapat dilihat pada **Gambar 8**.

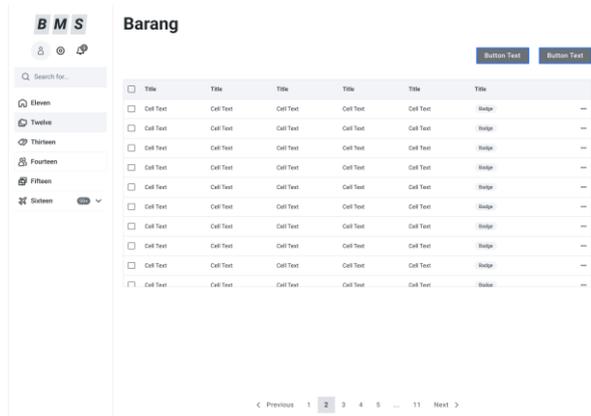


Gambar 8 ER Diagram

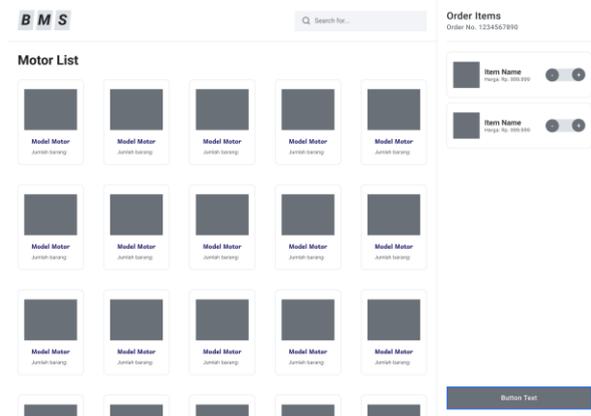
4.6. Wireframe

Wireframe merupakan sketsa awal dari *layout* desain UI yang berfungsi sebagai panduan dasar dalam

pengembangan aplikasi. Wireframe digunakan untuk memastikan bahwa struktur dan navigasi aplikasi dapat diakses dan dipahami dengan mudah oleh user, tanpa terpengaruh elemen visual seperti warna, tipografi, atau detail desain lainnya. Wireframe digambarkan secara sederhana tanpa style yang detail, sehingga pengembang dapat lebih fokus pada organisasi konten, alur navigasi, dan konsistensi tata letak, sebelum melanjutkan ke tahap desain visual yang lebih mendetail [10].



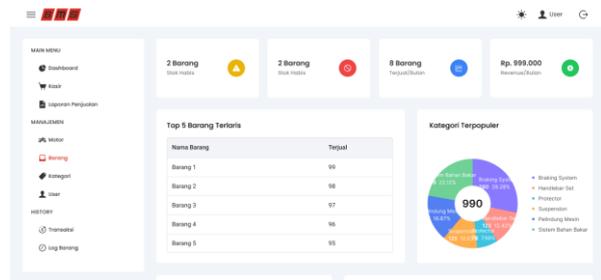
Gambar 9 Wireframe Table



Gambar 10 Wireframe Cashier

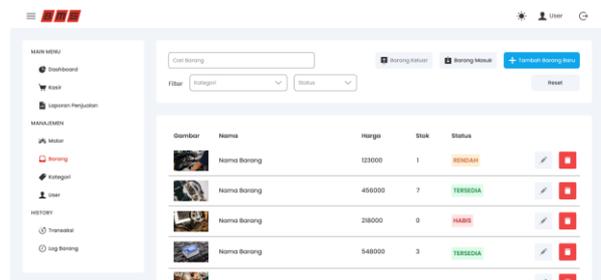
4.7. Mockup

Pada Gambar 11 ditunjukkan desain *mockup* halaman *dashboard* yang merupakan halaman utama setelah *Admin* berhasil login, yang berfungsi sebagai pusat ringkasan data, yang menampilkan informasi penting dalam bentuk visualisasi yang mudah dibaca seperti 5 barang yang terlaris, kategori terpopuler, jumlah stok barang yang rendah dan habis, jumlah barang terjual dan total pendapatan kotor. Dengan desain sederhana dan ringkas, *Admin* dapat dengan cepat memahami kondisi terkini inventaris dan penjualan tanpa harus masuk ke halaman lain. Pemilihan warna dan ikon dapat memperjelas visualisasi data untuk memudahkan pembacaan informasi yang padat.



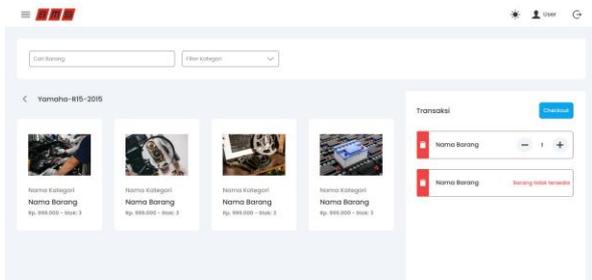
Gambar 11 Desain Mockup Dashboard

Pada Gambar 12 ditunjukkan desain *mockup* halaman barang, yang menampilkan daftar barang yang tersedia di bengkel, termasuk informasi penting seperti nama barang, harga, stok, status stok dan gambar. Desain *mockup* ini dirancang agar pengguna, terutama *inventory staff*, dapat dengan mudah mencari, menambah, dan memperbarui data barang. Halaman ini terdiri dari tabel daftar barang yang dilengkapi kolom untuk atribut barang. Di atas bagian komponen tabel terdapat fitur *search* dan *filter* yang memudahkan proses pencarian barang secara spesifik. Selain itu terdapat tombol “Tambah Barang Baru” yang dapat digunakan untuk mendaftarkan barang baru pada gudang, dan terdapat tombol “Barang Masuk” dan “Barang Keluar” yang digunakan untuk memperbarui stok barang yang ada pada gudang. Komponen tabel dan *filter* memungkinkan *Inventory Staff* untuk menavigasi dan memperbarui data barang dengan cepat. Dengan filter kategori dan *status*, staf dapat lebih mudah mengelola dan memantau barang sesuai dengan kebutuhan.



Gambar 12 Desain Mockup Halaman Barang

Berikutnya terdapat desain *mockup* menu kasir untuk staf kasir melakukan proses transaksi. Pada Gambar 13 dapat dilihat terdapat komponen *card* yang menampilkan gambar barang beserta harga dan stok untuk mempermudah kasir mengecek stok terlebih dahulu sebelum ditransaksikan, lalu pada sisi kanan terdapat panel *summary* yang menampilkan barang yang telah diseleksi dan *button* yang dapat mengubah kuantitas barang yang akan ditransaksikan. Halaman ini dirancang untuk meningkatkan kecepatan dan efisiensi proses pemilihan barang, dan *inputting* kuantitas barang. Desain ini berfokus pada kemudahan penggunaan, memungkinkan kasir untuk menyelesaikan transaksi dengan cepat. Elemen pencarian dan penambahan langsung membantu meminimalisir waktu pencarian barang di tengah antrian.



Gambar 13 Desain *Mockup* Halaman Kasir

5. Kesimpulan & Saran

5.1 Kesimpulan

Aplikasi kasir dan inventaris yang dirancang untuk bengkel Baru Motor Sport memiliki potensi besar dalam membantu operasional yang lebih efisien dan terstruktur. Desain sistem yang mengedepankan kemudahan penggunaan akan mempermudah staf bengkel dalam mengelola stok barang dan melakukan transaksi penjualan. Antarmuka yang intuitif diharapkan dapat meningkatkan produktivitas serta mengurangi risiko kesalahan yang sering terjadi pada pencatatan manual, seperti ketidakakuratan stok dan keterlambatan dalam memproses penjualan.

Dengan adanya fitur-fitur yang difokuskan pada kebutuhan utama bengkel, seperti pengelolaan inventaris dan transaksi yang cepat, aplikasi ini dapat memberikan dampak langsung pada efisiensi harian. Pembagian akses sesuai peran pengguna (*Admin*, *Inventory Staff*, dan *Cashier Staff*) akan memastikan bahwa setiap pengguna dapat menjalankan tugasnya dengan lebih lancar dan terstruktur. Hal ini juga berpotensi untuk meningkatkan akurasi dan kecepatan dalam melayani pelanggan, yang pada akhirnya dapat berkontribusi pada kepuasan pelanggan dan daya saing bengkel.

5.2. Saran

Terdapat beberapa saran untuk pengembangan lebih lanjut yaitu, **Survei Berkala untuk Penyesuaian Fitur**, Mengadakan survei berkala terkait kepuasan pengguna dan kebutuhan fitur tambahan dapat menjadi cara yang efektif untuk memahami pengalaman mereka. Hasil survei dapat digunakan untuk menyesuaikan fitur aplikasi berdasarkan perubahan kebutuhan pengguna atau operasional bengkel dari waktu ke waktu. Dan **Pengembangan Fitur Backup dan Recovery Data**, Untuk menjaga keamanan data jangka panjang, pastikan aplikasi memiliki fitur backup dan *recovery* otomatis. Dengan fitur ini, data transaksi dan inventaris tetap terlindungi meskipun terjadi masalah teknis atau kehilangan data akibat perangkat.

REFERENSI

[1] E. SEUN, B. Gegeleso, T. Falana, A. ADERONKE, and B. Olabode, "Impact of information systems on operational efficiency: A comprehensive analysis," *Indian Journal of Computer Science and Engineering*, vol. 14, pp. 661–673, 08 2023.

[2] I. Ammar and D. Satria, "Digitalisasi sektor umkm dan dampaknya terhadap pertumbuhan ekonomi di indonesia," *Journal of Development Economic and Social Studies*, vol. 02, no. 4, pp. 716–726, 2023. [Online]. Available: <http://dx.doi.org/10.21776/jdess.2023.02.4.01>

[3] L. Maximillian, I. Sanjaya, and E. Alexander, "Perancangan aplikasi desktop kasir pada bengkel motor di surabaya," *E-Link: Jurnal Teknik Elektro dan Informatika*, vol. 19, no. 1, pp. 110–115, 2024.

[4] U. T. Abdurrahman et al., "Perancangan aplikasi kasir point of sales berbasis android menggunakan metode rapid application development untuk usaha retail," *Infotech: Jurnal Informatika & Teknologi*, vol. 1, no. 2, pp. 67–77, 2020.

[5] H. X. Low and M. M. Rejab, "A development of inventory management system for sin guan leong grocery store," *Applied Information Technology And Computer Science*, vol. 3, no. 2, pp.835–853, 2022.

[6] S. Anisah, "Implementasi metode rapid application development pada pengembangan aplikasi inventory barang," *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, vol. 7, no. 1, pp. 57–65, 2022.

[7] F. Kinselton and Tony, "Perancangan aplikasi pencatatan penjualan berbasis web pada my sport indonesia," *Jurnal Ilmu Komputer dan Sistem Informasi*, vol. 1, no. 1, pp. 1–8, 2023.

[8] Badan Pusat Statistik, "Perkembangan jumlah kendaraan bermotor menurut jenis (unit), 2021-2022," <https://www.bps.go.id/id/statistics-table/2/NTcjMg==/perkembangan-jumlah-kendaraan-bermotor-menurut-jenis--unit-.html>, 2024, diakses: 9 Agustus 2024.

[9] S. BARORI, "Perancangan sistem pencarian bengkel kendaraan bermotor di wilayah bandar lampung berbasis aplikasi android," *Jurnal Teknologi Pintar*, vol. 2, no. 7, 2022.

[10] F. Staiano, *Designing and Prototyping Interfaces with Figma: Elevate your design craft with UX/UI principles and create interactive prototypes*. Packt Publishing, 2023. [Online]. Available: <https://books.google.co.id/books?id=7zvrEAAAQBAJ>

[11] K. Wood, *Confident Web Design: Master the Fundamentals of Website Creation and Supercharge Your Career*. Kogan Page Publishers, 2018.

[12] M. I. Saad, *Otodidak Web Programming: Membuat Website Edutainment*. Elex Media Komputindo, 2020.

[13] Vue.js (n.d.) *Vue.js - the progressive javascript framework*. Accessed: 2024-08-31. [Online]. Available: <https://vuejs.org>

[14] Tailwind CSS (n.d.) *Tailwind css - rapidly build modern websites without ever leaving your html*. Accessed: 2024-08-31. [Online]. Available: <https://tailwindcss.com/>

[15] J. Sutherland and J. O. Coplien, *A Scrum book: The spirit of the game*. Pragmatic Bookshelf, 2019.

[16] S. Sundaramoorthy, *UML diagramming: a case study approach*. Auerbach Publications, 2022.

[17] T. Connolly, *Database Systems: A Practical Approach to Design*. Pearson Education Limited, 2015.