

PERANCANGAN SISTEM INVENTORY MOTOR BERBASIS WEB UNTUK MANAJEMEN STOK TERDESENTRALISASI DAN OTORISASI KEUANGAN (STUDI KASUS : PT. ARMADA TUNASJAYA)

Rangga Yuda Pratama, Aditya Galih Sulaksono

Sistem Informatika, Universitas Merdeka Malang

Jalan Terusan Dieng, Malang, Indonesia

ranggayuda2003@gmail.com

ABSTRAK

Perusahaan dengan banyak cabang membutuhkan sistem manajemen inventaris yang efisien untuk memastikan kelancaran operasional. PT. Armada TunasJaya menghadapi tantangan dalam mengelola stok secara real-time dan menerapkan otorisasi keuangan yang fleksibel. Penelitian ini bertujuan merancang sistem berbasis web untuk mendukung manajemen stok terdesentralisasi dan otoritas keuangan. Metode pengembangan menggunakan model waterfall. Hasil menunjukkan sistem ini mampu meningkatkan efisiensi pengelolaan stok dan memberikan laporan secara real-time kepada pusat tanpa menghambat fleksibilitas cabang. Penelitian ini memberikan kontribusi signifikan terhadap pengembangan teknologi informasi di bidang manajemen inventaris.

Kata kunci : *sistem inventaris, manajemen stok, otoritas keuangan, web-based system*

1. PENDAHULUAN

Inventory adalah persediaan barang atau bahan yang disimpan untuk tujuan tertentu, misalnya untuk memenuhi kebutuhan operasional organisasi, disebut inventaris. Setiap barang yang dijual setiap hari dicatat dalam manajemen inventory[1].

PT. Armada Tunasjaya didirikan pada tahun 1995 sebagai bagian dari strategi PT. Armada Group untuk diversifikasi industri otomotifnya. Untuk memenuhi permintaan pasar kendaraan bermotor yang sedang tumbuh pesat di Indonesia pada saat itu, perusahaan berkonsentrasi pada penjualan mobil roda dua dan roda empat. PT. Armada Tunas Jaya cepat memperoleh pangsa pasar yang besar di industri otomotif nasional.

PT. Armada Tunasjaya, yang berafiliasi dengan Ahass Honda Motor, berlokasi di Jalan Jenderal Sudirman 199, Jawa Tengah. Perusahaan ini menawarkan purna jual, suku cadang, dan penjualan motor. Selain itu, sebagai bagian dari operasinya di Magelang, Jawa Tengah, PT. Armada Tunas Jaya juga menyimpan data tentang pembelian dan penjualan motor.

PT. Armada Tunasjaya menghadapi sejumlah masalah selama operasinya, terutama terkait dengan manajemen stok motor. Dalam bisnis jual beli motor, manajemen stok yang efektif sangat penting untuk memastikan ketersediaan yang cukup dan mencegah kekurangan atau kelebihan stok. Setiap cabang harus dapat mengelola stok motor secara mandiri, sementara kantor pusat hanya dapat menerima laporan dari semua cabang tanpa perlu sinkronisasi data secara langsung.

Selain itu, prosedur otorisasi serta persetujuan keuangan perlu diimplementasikan menggunakan cara yang tidak menghambat fleksibilitas operasional cabang. Pengembangan software manajemen persediaan motor yang efektif akan membantu dalam mencapai tujuan tersebut dengan menyediakan fitur

yang mendukung akurasi, efisiensi, serta laporan yang memadai ke pusat.

2. TINJAUAN PUSTAKA

2.1. Sistem Informasi

Sistem informasi adalah perangkat lunak yang dirancang untuk mengelola dan mengolah data sehingga menghasilkan informasi yang bermanfaat. Misalnya, sistem informasi e-raport digunakan dalam pendidikan untuk mengelola data akademik seperti nilai siswa, data guru, dan kelas. Penggunaan sistem informasi ini bertujuan untuk meningkatkan efisiensi pengelolaan data dan mengoptimalkan cara siswa, guru, dan orang tua menerima informasi[2], [3].

Secara lebih luas, sistem Informasi dapat didefinisikan sebagai rangkaian terorganisir dari komponen-komponen yang saling terkait, yang bekerja bersama-sama untuk mengumpulkan, menyimpan, mengolah, dan memproses data menjadi informasi yang berguna untuk mendukung pengambilan keputusan, pengendalian, dan operasionalisasi dalam suatu organisasi[4].

2.2. Manajemen Inventory

Menurut Sadono Sukirno pada bukunya yang berjudul "Pengantar Bisnis", Manajemen inventory berarti mengelola atau mengatur barang-barang agar jumlahnya tidak berlebihan sehingga biaya dapat ditekan serendah mungkin, tetapi jumlahnya juga harus selalu cukup untuk memenuhi permintaan pelanggan[5].

Pada buku yang sama, Sadono Sukirno menyatakan bahwa manajemen inventory harus diterapkan mulai dari barang mentah, barang sedang pakai, dan barang sudah jadi. Mengingat bahwa menyimpan tiga jenis barang tersebut sama dengan investasi barang yang cukup besar, pengontrolan

inventory sangat penting untuk mencegah biaya investasi inventory menjadi terlalu besar[5].

2.3. Inventory

Inventory adalah persediaan barang atau bahan yang disimpan untuk tujuan tertentu, misalnya untuk memenuhi kebutuhan operasional organisasi, disebut inventaris. Setiap barang yang dijual setiap hari dicatat dalam manajemen inventory[1].

Persediaan ini termasuk barang mentah, bahan, atau barang jadi yang disimpan untuk digunakan di masa mendatang dan membantu bisnis menghadapi dan mengantisipasi perubahan harga di pasar. Setiap barang yang dijual setiap hari dicatat dalam inventory management untuk memastikan bahwa produk tersedia sesuai permintaan pelanggan[6].

2.4. Teknologi Web

Salah satu teknologi yang paling banyak digunakan untuk membangun sistem informasi berbasis web adalah website; mereka memungkinkan pengolahan data lebih cepat dan efisien, sehingga lebih efisien dan efektif. Selain itu, teknologi web memungkinkan kecerdasan buatan (AI) untuk memproses data lebih cepat dan tepat guna, yang meningkatkan kemampuan pengolahan dan penyampaian data secara otomatis dari satu sumber ke sumber lain[2].

Teknologi web yang terus berkembang menjadi pilihan utama dalam berbagai industri, termasuk manajemen inventori, karena memungkinkan pengembangan aplikasi yang dapat diakses secara global yang memungkinkan pengguna untuk memantau, mengelola, dan memproses data dari mana saja dan kapan saja. Dalam konteks manajemen inventori motor, teknologi web menawarkan banyak keuntungan, termasuk skalabilitas, integrasi data secara real-time, dan kemudahan akses.

2.5. PHP

PHP, juga dikenal sebagai Preprocessor Hypertext, adalah bahasa pemrograman server-side yang sangat disukai untuk membuat aplikasi web dinamis. PHP, yang dikenal karena kesederhanaannya, memungkinkan pengembang menangani permintaan klien dengan cepat dan dinamis. Selain itu, komunitas PHP yang besar menawarkan akses ke berbagai pustaka dan kerangka kerja, yang membuat proses pengembangan aplikasi web lebih mudah dan lebih cepat. PHP adalah pilihan utama untuk pengembangan aplikasi berbasis web karena ekosistemnya yang luas dan kemudahan penggunaan[7].

2.6. Framework Laravel

Laravel adalah framework PHP populer yang mendukung arsitektur MVC, memisahkan logika dan tampilan, pengelola basis data terintegrasi melalui Eloquent ORM, serta fitur keamanan bawaan seperti terhadap CSRF dan SQL injection. dengan sistem routing yang mudah dan autentikasi lengkap, Laravel

menjadi pilihan tepat untuk sistem inventory berbasis web aman dan terstruktur[8].

2.7. MySQL

MySQL adalah sistem manajemen basis data relasional yang sangat disukai dan sering digunakan bersama PHP dalam aplikasi berbasis internet. Sebagai sistem open-source, MySQL kompatibel dengan berbagai sistem operasi dan menawarkan keandalan dan efisiensi dalam mengelola basis data yang besar dan kompleks. MySQL juga membantu pengembang mengelola data dengan perintah dasar berbasis SQL (Bahasa Pertanyaan Struktural), yang membuatnya populer di kalangan pengembang teknologi[9].

MySQL dapat secara efektif menghubungkan perangkat lunak aplikasi ke database server dengan menggunakan bahasa SQL, yang memungkinkannya mendukung berbagai operasi database, seperti menyimpan, mengambil, dan memanipulasi data, dengan cepat dan aman. Karena kemampuan untuk diintegrasikan dengan aplikasi web, ini adalah pilihan utama untuk pengembangan sistem berbasis data di seluruh dunia[10].

2.8. Blackbox Testing

Blackbox testing merupakan metode pengujian perangkat lunak yang berfokus pada evaluasi spesifikasi fungsional sistem tanpa mempertimbangkan struktur atau implementasi kode, teknik ini sering digunakan untuk mendeteksi kesalahan dalam fungsionalitas perangkat lunak, seperti antarmuka pengguna, pengelolaan data, serta validasi dan pengolahan input, sehingga dapat memastikan bahwa perangkat lunak berjalan dengan kebutuhan pengguna dan spesifikasi yang telah ditemukan[11].

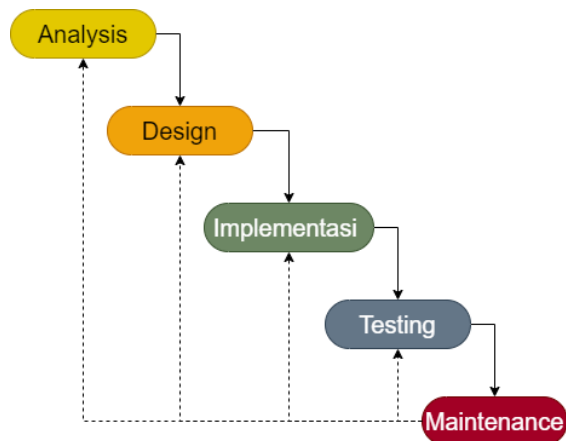
2.9. System Usability Scale (SUS)

System Usability Scale (SUS) merupakan metode evaluasi praktis yang digunakan untuk mengukur tingkat kegunaan sebuah informasi berdasarkan perspektif pengguna. SUS menggunakan kuesioner dengan 10 pertanyaan yang dirancang untuk mengevaluasi efektivitas, efisiensi, dan kepuasan pengguna terhadap sistem. Dengan skala Likert dari "Sangat Tidak Setuju" sampai "Sangat Setuju", skor SUS dihitung pada rentang 0 – 100 untuk menentukan tingkat penerimaan sistem melalui kategori seperti acceptable, marginal, atau not acceptable. Selain itu, SUS memberikan penilaian tambahan berupa grade scale dan adjective rating, yang memberikan wawasan lebih mendalam tentang persepsi pengguna terhadap sistem[12].

3. METODE PENELITIAN

Penelitian ini menggunakan metode pengembangan Waterfall. Model SDLC waterfall, yang dikenal sebagai model sekuensial linier atau siklus hidup klasik yang menawarkan metode pendekatan siklus hidup perangkat lunak yang berurutan. Proses ini dimulai dengan analisis, desain,

implementasi & Coding, integrasi & testing, dan berakhir pada tahap maintenance[13].



Gambar 1. Metode Waterfall

Teknik pengumpulan data dikumpulkan melalui berbagai teknik, termasuk wawancara dan observasi. Untuk melakukan observasi ini, kami melihat secara langsung proses manajemen inventori PT. Armada Tunasjaya, dengan penekanan khusus pada manajemen stok motor di berbagai cabang.

Tujuan observasi ini adalah untuk mendapatkan pemahaman yang lebih baik tentang bagaimana proses pencatatan barang masuk dan keluar dilakukan, serta untuk menemukan masalah dengan sistem manual yang sedang digunakan.

Wawancara dilakukan dengan staf terkait, terutama General Affair yang bertanggung jawab atas pengelolaan gudang dan aset perusahaan. Wawancara ini bertujuan untuk mendapatkan informasi lebih mendalam mengenai:

Tabel 1. Kebutuhan Sistem

No	Pertanyaan
1	Apa saja fitur yang diperlukan untuk membantu pengelolaan stok motor dan spare parts di gudang setiap cabang?
2	Apakah diperlukan akses sistem yang berbeda untuk staf gudang (operasional) dan manager atau kepala cabang?
3	Seberapa penting bagi anda untuk memiliki fitur laporan grafik atau statistika tentang stok barang?
4	Bagaimana sistem sebaiknya menangani stock motor yang berbeda jenis, warna, atau spesifikasi?

Tabel 2. Permasalahan yang dihadapi

No	Pertanyaan
1	Apa kendala utama yang sering anda temui saat mencatat barang masuk atau keluar?
2	Apakah ada kendala dalam memantau stok motor dan spare parts yang tersebar di berbagai cabang? Jika ya, bagaimana biasanya hal ini diatasi?
3	Seberapa sering terjadi keterlambatan dalam pembaharuan data stok motor dan spare parts saat ada barang masuk dan keluar? Apa penyebab utamanya?
4	Bagaimana cara anda menangani kesalahan pencatatan stok saat ini? Apakah prosesnya sulit?

Tabel 3. Harapan terhadap sistem

No	Pertanyaan
1	Apakah sistem harus dapat menghasilkan laporan stok secara real-time? Jika ya, dalam format apa yang paling nyaman bagi anda?
2	Apakah ada sistem lain yang pernah anda gunakan, dan apa fitur terbaik dari sistem tersebut?
3	Sejauh mana anda berharap sistem dapat mempercepat proses pencatatan motor dan spare parts masuk dan keluar tanpa mengorbankan akurasi data?
4	Apa harapan anda terkait tingkat keamanan data dalam sistem, seperti pengaturan hak akses untuk mencegah kesalahan atau penyalahgunaan data stok?

Metode System Usability Scale (SUS) Metode ini digunakan untuk mengevaluasi kegunaan aplikasi Inventory Motor. Responden menilai 10 pertanyaan menggunakan skala Likert 1-5, mencakup aspek kemudahan, konsistensi, dan hambatan penggunaan. Skor dihitung dengan memisahkan pernyataan positif dan negatif, menghasilkan nilai akhir yang mengukur tingkat penerimaan dan kegunaan aplikasi.

Tabel 4. Pertanyaan kuesioner

No	Pertanyaan
Q1	Saya merasa aplikasi Inventory Motor akan sering saya gunakan
Q2	Saya merasa aplikasi Inventory Motor sulit digunakan
Q3	Saya merasa aplikasi Inventory Motor mudah digunakan
Q4	Saya membutuhkan bantuan orang lain atau teknisi untuk memahami cara menggunakan aplikasi ini
Q5	Saya merasa fitur-fitur pada aplikasi Inventory Motor berjalan sesuai dengan harapan
Q6	Saya merasa terdapat banyak hal yang tidak konsisten pada aplikasi Inventory Motor
Q7	Saya merasa orang lain akan mudah memahami cara menggunakan aplikasi Inventory Motor
Q8	Saya merasa aplikasi Inventory Motor membingungkan untuk digunakan
Q9	Saya merasa tidak ada hambatan dalam menggunakan aplikasi Inventory Motor
Q10	Saya perlu membiasakan diri terlebih dahulu sebelum dapat menggunakan aplikasi Inventory Motor dengan baik

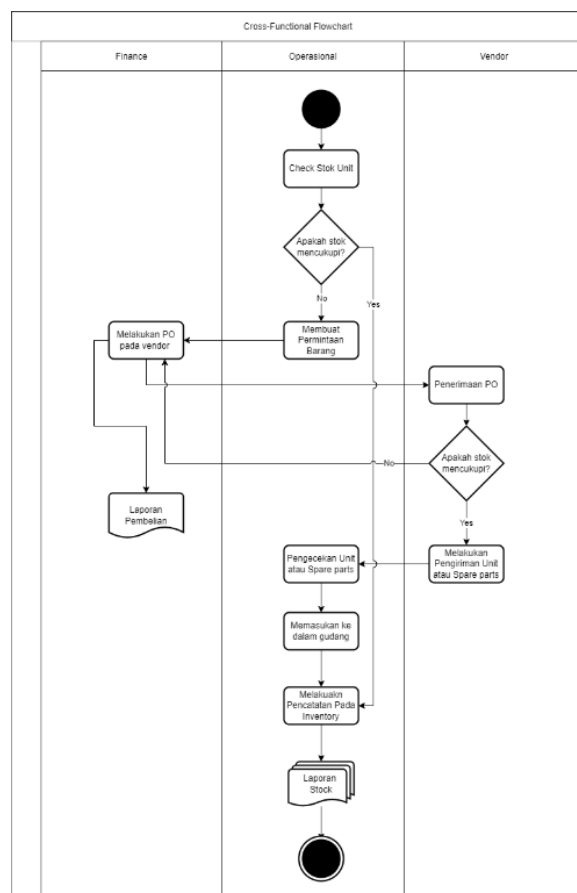
3.1. Use Case Diagram

Admin dapat mengelola pengguna, memperbarui data stock, dan melihat laporan masuk/keluar. Operasional dapat mengelola dan mencatat barang masuk/keluar serta mengakses informasi stok. Finance dapat membuat invoice, memeriksa tagihan/faktur, mengakses laporan outstanding PO, dan mengirim data ke pusat, sales dapat menerima barang, melihat stok dan mengakses riwayat permintaan barang



Gambar 2. Use Case Diagram

3.2. Activity Diagram Inventory



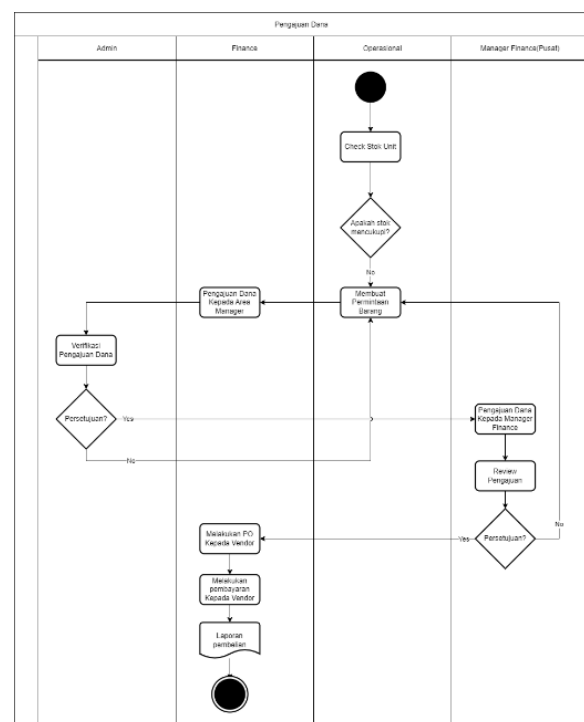
Gambar 3. Activity Diagram Inventory

Berikut adalah penjelasan secara umum mengenai Flowchart Inventory yang ditampilkan:

- Finance melakukan pemesanan (PO) ke vendor.

- Vendor menerima PO dan mengecek ketersediaan stok unit atau spare part.
- Jika stok mencukupi, vendor melakukan pengiriman unit atau spare part ke operational.
- Jika stok tidak mencukupi, vendor akan membuat permintaan barang ke operational.
- Operational menerima permintaan barang dan melakukan pengecekan unit atau spare parts.
- Operational memasukkan unit atau spare part ke gudang.
- Operational melakukan pencatatan pada inventory.
- Operational membuat laporan stock.

3.3. Activity Diagram Pengajuan Dana

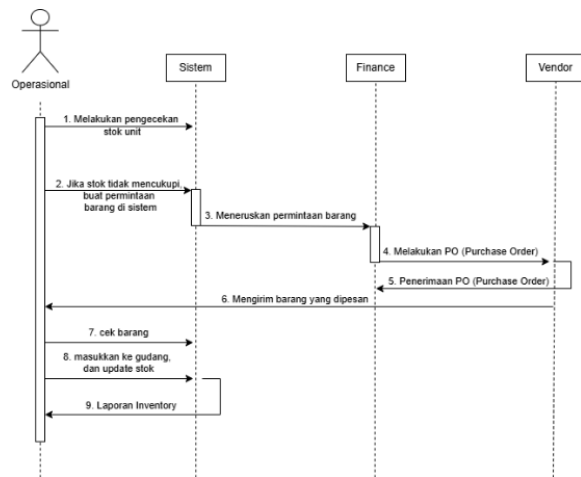


Gambar 4. Activity Diagram Pengajuan Dana

Berikut adalah penjelasan secara umum mengenai Flowchart Pengajuan Dana yang ditampilkan:

- Finance melakukan pemesanan (PO) ke vendor.
- Vendor menerima PO dan mengecek ketersediaan stok unit atau spare part.
- Jika stok mencukupi, vendor melakukan pengiriman unit atau spare part ke operational.
- Jika stok tidak mencukupi, vendor akan membuat permintaan barang ke operational.
- Operational menerima permintaan barang dan melakukan pengecekan unit atau spare parts.
- Operational memasukkan unit atau spare part ke gudang.
- Operational melakukan pencatatan pada inventory.
- Operational membuat laporan stock.

3.4. Sequence Diagram Inventory



Gambar 5. Sequence Diagram Inventory

Diagram sequence di atas menunjukkan alur proses pengelolaan stok barang, mulai dari pemeriksaan stok hingga penerimaan barang dan pembaruan data stok. Berikut adalah penjelasan umum tentang bagaimana proses ini terjadi:

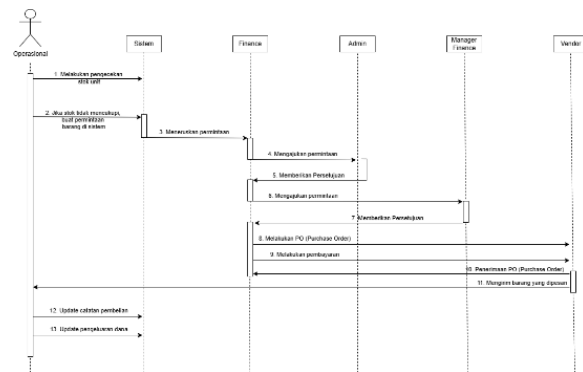
- a. Pengecekan Stok (Operasional) Bagian operasional memantau stok barang di gudang.
- b. Pengajuan Permintaan Barang (Operasional) Bagian operasional memasukkan permintaan barang ke sistem jika stok kurang.
- c. Sistem Mengirim Permintaan Barang ke Bagian Keuangan Sistem mengirimkan permintaan barang ke bagian keuangan.
- d. Bagian Keuangan Membuat Permintaan Pembelian (PO) untuk memesan barang dari vendor.
- e. Vendor Menerima PO Vendor menerima PO dan memprosesnya.
- f. Vendor Mengirim Barang Vendor mengirimkan barang yang dipesan ke bagian operasional.
- g. Pemeriksaan Barang (Operasional) Operasional memeriksa barang yang diterima untuk memastikan sesuai pesanan.
- h. Pembaruan Stok Barang dimasukkan ke gudang dan di update di sistem setelah pemeriksaan.
- i. Laporan Inventory: Bagian operasional melaporkan stok terbaru.

3.5. Sequence Diagram Pengajuan Dana

Berikut adalah penjelasan umum mengenai proses yang digambarkan dalam sequence diagram pengajuan dana:

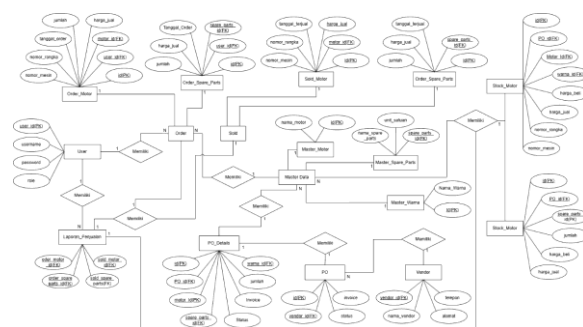
- a. Melakukan pengecekan stok unit: Bagian operasional memeriksa stok barang di gudang.
- b. Membuat permintaan barang: Operasional dapat mengajukan permintaan barang melalui sistem jika stok kurang.
- c. Meneruskan permintaan: Bagian keuangan menerima permintaan melalui sistem.
- d. Mengajukan Permintaan: Keuangan mengajukan permintaan persetujuan ke bagian administrasi.

- e. Persetujuan manajer: Jika permintaan sesuai, manajer memeriksanya dan memberikan persetujuan.
- f. Pengajuan ke manajer keuangan: Setelah admin menyetujuinya, permintaan dikirim ke manajer keuangan.
- g. Persetujuan dari manajer keuangan: Manajer keuangan memberikan persetujuan akhir.
- h. Membuat Permintaan Pembelian (PO): Keuangan membuat permintaan pembelian dan mengirimkannya ke vendor.
- i. Melakukan pembayaran: Keuangan membayar vendor. Penerimaan Pesanan: Vendor menerima pesanan dan memprosesnya.
- j. Mengirim barang: Barang dikirim ke operasional oleh vendor.
- k. Update stok: Setelah barang diterima, operasional memperbarui stok di sistem.
- l. Update pengeluaran: Untuk mencatat pembayaran, sistem mengubah catatan pengeluaran dana.



Gambar 6. Sequence Diagram Pengajuan Dana

3.6. Perancangan Entity Relationship Diagram



Gambar 7. Entity Relationship Diagram

Diagram ERD (Entity Relationship Diagram) ini menggambarkan model data yang digunakan dalam sistem penjualan motor. Terdapat beberapa entitas utama yang saling terhubung, yaitu:

- a. User: Entitas ini mewakili pengguna sistem, yang memiliki atribut seperti id, username, password, dan role.
- b. Order_Motor: Entitas ini mewakili pesanan motor yang dilakukan oleh pengguna, yang

- memiliki atribut seperti id, tanggal_order, dan nomor_rangka.
- Sold_Motor:** Entitas ini mewakili motor yang terjual, yang terhubung dengan entitas Order_Motor dan memiliki atribut seperti id, nama_motor, dan unit_satuan.
 - Master_Motor:** Entitas ini mewakili data induk motor, yang memiliki atribut seperti id, nama_motor, dan harga_jual.
 - Master_Warna:** Entitas ini mewakili data warna motor, yang terhubung dengan entitas Master_Motor.
 - Order_Spare_Parts:** Entitas ini mewakili pesanan suku cadang yang dilakukan oleh pengguna, yang terhubung dengan entitas Master_Motor.
 - Stock_Motor:** Entitas ini mewakili stok motor yang tersedia, yang terhubung dengan entitas Master_Motor.
 - Vendor:** Entitas ini mewakili vendor yang memasok suku cadang, yang memiliki atribut seperti id, nama_vendor, dan alamat.
 - Invoice:** Entitas ini mewakili faktur penjualan motor, yang terhubung dengan entitas Sold_Motor dan Vendor.

Hubungan antar entitas dalam ERD ini menunjukkan relasi yang kompleks dalam sistem penjualan motor, seperti relasi one-to-many antara Order_Motor dan Sold_Motor, serta relasi many-to-many antara Order_Spare_Parts dan Master_Motor.

3.7. Class Diagram

Class Diagram ini menggambarkan struktur dan hubungan antara kelas-kelas yang ada dalam sistem penjualan motor. Berikut penjelasan mengenai masing-masing kelas:

- Sold_Spare_Parts dan Order_Spare_Parts:** Kelas-kelas ini berkaitan dengan transaksi penjualan suku cadang. Sold_Spare_Parts mewakili suku cadang yang sudah terjual, sementara Order_Spare_Parts mewakili pesanan suku cadang yang dilakukan oleh pengguna.
- Sold_Motor dan Order_Motor:** Kelas-kelas ini berkaitan dengan transaksi penjualan motor. Sold_Motor mewakili motor yang sudah terjual, sementara Order_Motor mewakili pesanan motor yang dilakukan oleh pengguna.
- User:** Kelas ini mewakili pengguna sistem, yang dapat melakukan berbagai operasi seperti pemesanan motor dan suku cadang.
- Laporan_Penjualan:** Kelas ini mewakili laporan penjualan, yang dapat dibuat berdasarkan data-data dari kelas Sold_Motor dan Sold_Spare_Parts.
- Master_Spare_Parts, Master_Motor, dan Master_Warna:** Kelas-kelas ini mewakili data induk suku cadang, motor, dan warna motor yang tersedia dalam sistem.

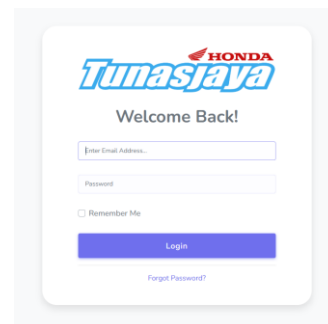
- Stock_Motor:** Kelas ini mewakili stok motor yang tersedia, yang terhubung dengan kelas Master_Motor dan PO_Details.
- Vendor:** Kelas ini mewakili vendor yang memasok suku cadang, yang terhubung dengan kelas PO.
- PO_Details dan PO:** Kelas-kelas ini berkaitan dengan proses pembelian suku cadang dan motor dari vendor.



Gambar 8. Class Diagram

4. HASIL DAN PEMBAHASAN

4.1. Halaman Login



Gambar 11. Halaman Login

Menampilkan UI Halaman Login. User yang memiliki role superadmin, admin, finance, sales, dan operasional. Disini user harus memasukan email dan password sebagai autentikasi untuk dapat mengakses website dan fitur-fitur sesuai dengan role masing-masing.

4.2. Dashboard





Gambar 12. Dashboard

Dashboard yang dirancang untuk menampilkan keseluruhan data dalam bentuk grafik dari Stock, Orders, dan Sold. Dimana informasi pada Dashboard ini akan menyesuaikan secara otomatis dengan data tahun berjalan (real-time) sehingga memudahkan pemantauan kinerja berdasarkan tahun yang relevan.

4.3. Pengujian Blackbox

Untuk memastikan bahwa sistem yang telah dirancang dapat berjalan dengan baik, penulis meminta seorang staff untuk melakukan uji coba terhadap sistem. Uji coba ini bertujuan untuk mengevaluasi fungsionalitas dan kinerja sistem secara menyeluruh, serta memastikan bahwa semua fitur berfungsi sesuai dengan yang diharapkan. Hasil dari uji coba ini akan menjadi acuan untuk perbaikan atau penyempurnaan sistem jika ditemukan kekurangan atau bug. Dengan demikian, diharapkan sistem yang dikembangkan memberikan manfaat optimal bagi operasional perusahaan.

Tabel 5. Hasil Pengujian Blackbox

No	Fitur Yang Diuji	Hasil Pengujian
1	Halaman Login	Berhasil
2	Dashboard	Berhasil
3	User Management	Berhasil
4	Vendor	Berhasil
5	Purchase Order	Berhasil
6	Purchase Order Details	Berhasil
7	Stocks	Berhasil
8	Sold Itemd	Berhasil
9	All Stocks Data	Berhasil
10	Order Motor	Berhasil
11	Order Spare Parts	Berhasil

No	Fitur Yang Diuji	Hasil Pengujian
12	Sales Report	Berhasil
13	Master Motor	Berhasil
14	Master Warna	Berhasil
15	Master Spare Parts	Berhasil

4.4. System Usability Scale (SUS)

Penilaian dalam System Usability Scale (SUS) dilakukan menggunakan skala Likert dengan rentang 1 hingga 5. Berikut adalah penjelasan skala digunakan:

Tabel 6. Skala Likert

No.	Skala Likert
1	Sangat Tidak Setuju (STS): 1
2	Tidak Setuju (TS): 2
3	Netral (N): 3
4	Setuju (ST): 4
5	Sangat Setuju (SS): 5

Skala ini digunakan untuk menggali persepsi responden mengenai kemudahan, konsisten, dan kegunaan aplikasi. Hasilnya akan digunakan untuk menghitung skor SUS yang merefleksikan tingkat penerimaan pengguna terhadap aplikasi.

Hasil skor SUS selanjutnya diinterpretasikan berdasarkan rentang penerimaan berikut:

- 0 – 60.9 = Not Acceptable
Sistem tidak dapat diterima karena terdapat banyak kekurangan dalam kegunaan dan fungsionalitas.
- 61 – 80.9 = Marginal
Sistem berada pada tingkat yang dapat diterima atau cukup, tetapi memerlukan perbaikan untuk meningkatkan pengalaman pengguna.
- 81 – 100 = Acceptable
Sistem diterima dengan baik dan memiliki tingkat kegunaan yang memadai untuk memenuhi kebutuhan pengguna.

Rentang ini memberikan gambaran tentang seberapa baik sistem diterima dan menjadi acuan untuk menentukan fokus perbaikan demi meningkatkan pengalaman pengguna.

Tabel 6. Hasil System Usability Scale

No. Res	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Jumlah	Total Skor
1	3	4	4	4	4	4	4	5	4	5	30	75.0
2	4	5	4	4	4	4	4	5	3	4	31	77.5
3	4	4	4	5	5	5	4	4	4	5	32	80.0
4	2	3	4	4	4	4	4	4	4	3	26	65.0
5	5	4	5	5	5	5	5	5	5	5	37	92.5
6	4	4	4	4	4	4	4	4	3	4	34	85.0
7	5	4	4	4	5	5	5	4	4	5	35	87.5
8	3	4	4	4	4	4	3	4	3	3	30	75.0
Total											255	637.5
Rata-rata											31.9	79.7

Hasil perhitungan System Usability Scale dilakukan dengan rumus berikut:

$X = \text{nilai pernyataan positif} - 1$

$Y = 5 - \text{nilai pernyataan negatif}$

$Z = \text{jumlah pernyataan}$

$n = \text{jumlah responden (Res)}$

$\text{total skor} = \text{jumlah} * 2.5$

$$\text{skor rata-rata} = \frac{\sum \text{Total Skor}}{n} = \frac{637.5}{8} = 79.7$$

Berdasarkan rumus diatas, diperoleh skor 79,7. Skor ini menunjukkan bahwa aplikasi Inventory Motor berada dalam kategori Marginal menurut System Usability Scale (SUS), yang berarti aplikasi cukup diterima tetapi masih ada ruang untuk perbaikan guna meningkatkan pengalaman pengguna[12].

5. KESIMPULAN DAN SARAN

Sistem inventori motor berbasis web untuk PT. Armada Tunasjaya mendukung pengelolaan inventori cabang secara terdesentralisasi dengan fitur real-time, laporan otomatisasi, dan pengelolaan stok sesuai dengan spesifikasi. dengan skor 79,7, sistem ini memenuhi kebutuhan perusahaan namun dapat ditingkatkan dengan integrasi sistem keuangan atau ERP, fitur mobile-friendly, serta keamanan data yang lebih canggih. evaluasi berkala, pemeliharaan rutin dan pelatihan staff penting untuk memastikan sistem terus mendukung operasional secara efisien.

DAFTAR PUSTAKA

- [1] E. Affandi and T. Syahputra, "Pemodelan Uml Manajemen Sistem Inventory," *J-SISKO TECH*, vol. 1, no. 2, pp. 14–25, 2018.
- [2] D. Effendi and A. Sulaksono, "Implementasi Framework Laravel pada Perancangan Sistem Informasi E-Raport Berbasis Web SD Negeri 1 Kartoharjo," *JITET*, vol. 1, no. 11, pp. 815–822, Nov. 2021, doi: 10.17977/um068v1i112021p815-822.
- [3] V. P. Ramadhan, A. G. Sulaksono, and M. Andarwati, "Desain Sistem Jasa Titip Berbasis E-Commerce Untuk Memudahkan Para Penyedia Jasa Titip Kue," *Seminar Nasional Sistem Informasi*, 2018.
- [4] Ario Febri Putra Raharjo and Dio Restu Putra, "Sistem Informasi Profil Perusahaan Berbasis Web Pada Toko Dua Arah Dengan Metode Extreme Programming," 2023-07-26, vol. Vol.1 No.4, p. 392, Jul. 2023.
- [5] D. Widiyanto, "Perancangan Sistem Informasi Manajemen Inventori Berbasis Web (studi Kasus: Smk Ypt Purworejo)," *Jurnal Ekonomi dan Teknik Informatika*, vol. 10, no. 1, 2022.
- [6] A. Heryanto, H. Fuad, and D. Dananggi, "Rancang Bangun Sistem Informasi Inventory Barang Berbasis Web Studi Kasus di PT. Infinetworks Global Jakarta," *JURNAL SISFOTEK GLOBAL*, vol. 4, no. 2, 2014.
- [7] F. Marisa and T. G. Yuarita, "Perancangan Aplikasi Point of Sales (pos) Berbasis Web Menggunakan Metode Siklus Hidup Pengembangan Sistem," *JTMI*, vol. 3, no. 2, Sep. 2017, doi: 10.26905/jtmi.v3i2.1514.
- [8] Z. Subecz, "Web-development with Laravel framework," *Gradus*, vol. 8, no. 1, pp. 211–218, 2021, doi: 10.47833/2021.1.CSC.006.
- [9] E. Mulyani, M. S. Bunga, A. L. Gozali, A. Nurhanudin, and S. F. Udin, "Penerapan Business Logic Component dalam aplikasi penjualan dan edukasi produk hidroponik berbasis mobile," *J. of Information System and Application Development*, vol. 1, no. 2, pp. 89–100, Sep. 2023, doi: 10.26905/jisad.v1i2.11065.
- [10] J. Surya and F. Huda Aminuddin, *Pemrograman MYSQL Database With Streamlit Python*. PT. Sonpedia Publishing Indonesia, 2024.
- [11] M. S. Mustaqbal, R. F. Firdaus, and H. Rahmadi, "Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis (studi Kasus: Aplikasi Prediksi Kelulusan Snmptn)," *Jurnal Ilmiah Teknologi Informasi Terapan*, vol. 1, no. 3, 2015.
- [12] R. Ulfa, "Mengukur kepuasan pengguna sistem informasi bimbingan konseling (e-bk) menggunakan system usability scale (sus) di smk negeri 1 banda aceh," *Universitas Islam Negeri Ar-Raniry*, Mar. 2021.
- [13] I. G. E. Waisnawa, G. N. M. Nata, and K. H. S. Dewi, "Sistem Informasi Inventory Barang Pada Bali Jacosi berbasis Website Menggunakan Framework Laravel," *Spinter: Prosiding Seminar Hasil Penelitian Informatika Dan Komputer | Institut Teknologi Dan Bisnis Stikom Bali*, vol. 1, no. 2, 2024.