



Implementasi Sistem Informasi Pengelolaan Surat Jalan dan Invoice Menggunakan Framework CodeIgniter

Beny Rahman Hakim

Teknik Informatika, Politeknik Hasnur, Indonesia

*Penulis Korespondensi: benyrah@gmail.com

Abstract. *The development of information technology has become a primary pillar in enhancing corporate operational efficiency, where rapid and accurate data integration significantly determines business competitiveness. Case study in PT Gawi Anjung Bersama (GAB) revealed that the company is facing operational constraints because the management of delivery orders and invoices still relies on conventional office applications such as Microsoft Excel and Word. This procedure causes hurdles in data sharing among staff and results in low information security due to the absence of integrated access control settings. This research aims to design and develop the Gawi Anjung Bersama Information System (SIGAB), a web-based system implementing the CodeIgniter framework. The system is developed using Waterfall model, encompassing stages from analysis, design, and implementation to testing. The research flow starts with problem identification, primary and secondary data collection, and functional and non-functional requirements analysis. The system is designed using Unified Modeling Language (UML) modeling, specifically Use Case and Class Diagrams, as well as Entity Relationship Diagrams (ERD) for the database. The system interface is designed as wireframes on Figma web. This system separates access rights into three main actors namely owner, administrator, and staff, to ensure data security. Blackbox testing results, conducted after the system was hosted, indicate that all functional features operate as intended. The implementation of SIGAB is expected to optimize the management of delivery orders and invoices at PT GAB.*

Keywords: CodeIgniter; Delivery Order; ERD; Invoice; SIGAB.

Abstrak. Perkembangan teknologi informasi telah menjadi pilar utama dalam meningkatkan efisiensi operasional perusahaan, dimana integrasi data yang cepat dan akurat sangat menentukan daya saing bisnis. Studi kasus pada PT Gawi Anjung Bersama (GAB) menunjukkan bahwa perusahaan menghadapi kendala operasional sebab pengelolaan surat jalan dan invoice (tagihan) masih mengandalkan aplikasi perkantoran konvensional seperti Microsoft Excel dan Word. Prosedur ini menyebabkan hambatan dalam berbagi data antar pegawai, serta rendahnya keamanan informasi akibat ketiadaan pengaturan hak akses yang terintegrasi. Penelitian ini bertujuan untuk melakukan rancang bangun Sistem Informasi Gawi Anjung Bersama (SIGAB) berbasis web menerapkan framework CodeIgniter. Metode penelitian yang digunakan adalah model Waterfall yang mencakup tahap analisis, desain, implementasi, hingga pengujian. Alur penelitian dimulai dari identifikasi masalah, pengumpulan data primer dan sekunder, hingga analisis kebutuhan fungsional dan non-fungsional. Sistem didesain menggunakan pemodelan Unified Modeling Language (UML), meliputi use case dan class diagram, serta Entity Relationship Diagram (ERD) untuk perancangan basis data. Antarmuka sistem didesain dalam bentuk wireframe dengan aplikasi Figma web. Sistem ini membagi hak akses ke dalam tiga aktor utama, yaitu owner (pemilik), administrator, dan staf untuk menjamin keamanan data. Hasil pengujian blackbox, setelah sistem di-hosting, menunjukkan bahwa seluruh fitur fungsional dapat berjalan sebagaimana mestinya. Implementasi SIGAB diharapkan mampu mengoptimalkan pengelolaan surat jalan dan invoice pada PT GAB.

Kata kunci: CodeIgniter; ERD; Faktur; SIGAB; Surat Perintah Pengiriman.

1. LATAR BELAKANG

Teknologi informasi kini tidak lagi sekadar menjadi pendukung, melainkan instrumen fundamental yang menentukan reliabilitas operasional perusahaan di berbagai sektor industri. Integrasi data yang presisi dan distribusi informasi yang cepat menjadi parameter krusial dalam menjaga stabilitas rantai pasok dan daya saing bisnis (Pratama & Hartono, 2023). Namun nyatanya, banyak entitas bisnis menengah yang masih terjebak dalam pola manajemen

administratif konvensional. PT GAB (Gawi Anjung Bersama) merupakan salah satu perusahaan yang bergerak di bidang distribusi produk berskala besar, yang telah membangun reputasi melalui kerja sama strategis dengan perusahaan ternama, serta menangani berbagai permintaan pasar. Sebagai distributor yang melayani klien dengan standar operasional tinggi, pengelolaan dokumen transaksi seperti surat jalan dan *invoice* (tagihan) menjadi jantung dari proses bisnis perusahaan.

Pada manajemen administrasi di PT Gawi Anjung Bersama ditemukan permasalahan yakni ketergantungan tinggi terhadap penggunaan perangkat lunak perkantoran umum seperti Microsoft Excel dan Word dalam mengelola data transaksi harian. Berdasarkan hasil observasi dan wawancara yang dilakukan penulis bersama owner, administrator, dan staf operasional, teridentifikasi bahwa prosedur kerja ini menimbulkan inefisiensi yang signifikan. Dokumen yang dibuat secara manual sering kali menyebabkan duplikasi data, risiko kesalahan pengetikan (*human error*), serta kesulitan dalam melakukan pencarian data historis. Selain itu, prosedur pengiriman file yang harus dilakukan secara manual antar staf memicu hambatan komunikasi dan memperlambat alur kerja (Mulyani, 2022).

Masalah keamanan dan integritas data juga ditemukan, sebab file-file tersebut dapat diakses atau diubah oleh siapapun tanpa adanya pembatasan hak akses. Penelitian ini bertujuan untuk melakukan rancang dan bangun sistem informasi berbasis web sebagai solusi dalam mengatasi kendala tersebut. Sistem Informasi Gawi Anjung Bersama (SIGAB) dirancang untuk optimalisasi pengelolaan surat jalan dan *invoice*, sekaligus menyediakan manajemen hak akses terkendali guna menjamin akuntabilitas setiap pengguna.

SIGAB dikembangkan menggunakan *framework* (kerangka kerja) CodeIgniter versi 3 (CI3) sebagai komponen *backend*-nya. CI3 digunakan sebab karakteristiknya yang *small footprint* atau sangat ringan, sehingga tidak membebani sumber daya *server* perusahaan. Selain itu, CI3 menggunakan paradigma *Model-View-Controller* (MVC) yang efisien dalam memisahkan logika bisnis dengan tampilan antarmuka (Siddique & Ali, 2024). Hal ini bermanfaat agar memudahkan pemeliharaan (*maintenance*) kode dan pengembangan fitur di masa mendatang. Dukungan dokumentasi yang stabil dan keamanan bawaan terhadap serangan *Cross-Site Request Forgery* (CSRF) dan *SQL Injection* menjadikan CodeIgniter 3 sebagai pilihan yang rasional dan handal untuk kebutuhan operasional PT Gawi Anjung Bersama yang menuntut stabilitas tinggi.

Metode yang digunakan dalam penelitian ini adalah Waterfall, yaitu pendekatan sekuensial yang sistematis untuk memastikan setiap tahapan pengembangan sistem terdokumentasi dengan baik (Pressman & Maxim, 2020). Alur penelitian dimulai dari

identifikasi masalah melalui observasi langsung di lapangan. Selanjutnya, dilakukan pengumpulan data primer melalui wawancara bersama *owner* (pemilik usaha) hingga staf, serta data sekunder melalui studi pustaka. Tahap analisis mencakup pendefinisian kebutuhan fungsional dan kebutuhan non-fungsional. Pada desain sistem, digunakan pemodelan *Unified Modeling Language* (UML) yang mencakup *use case diagram* untuk mendefinisikan interaksi aktor dalam sistem, serta *class diagram* untuk struktur sistem (Fowler, 2021). Database sistem dirancang dengan *Entity Relationship Diagram* (ERD) agar relasi antar tabel data surat jalan dan *invoice* tetap konsisten. Adapun antarmuka sistem didesain menggunakan *wireframe* pada Figma web guna mempermudah pembuatan tampilan sistem dengan bahasa HTML, CSS dan Javascript. Tahap akhir adalah implementasi kode program dan pengujian menggunakan metode *blackbox testing* untuk memastikan setiap fitur sistem berjalan sesuai kebutuhan pengguna. Hasil dari penelitian ini diharapkan mampu meningkatkan kinerja operasional perusahaan agar SIGAB tidak hanya menjadi solusi teknis, tetapi juga motor penggerak bagi PT Gawi Anjung Bersama dalam meningkatkan profesionalisme pelayanan terhadap mitra-mitra strategisnya.

2. KAJIAN TEORITIS

Dalam memetakan kebaruan dan posisi riset ini, penulis merujuk pada beberapa studi terkait yang telah dilakukan sebelumnya. Pertama, penelitian oleh (Saputra, 2022) mengembangkan sistem administrasi gudang yang membuktikan bahwa digitalisasi dokumen dapat mereduksi waktu pencarian data hingga 70%. Kedua, studi yang dilakukan (Rahmawati, 2023) mengenai sistem penagihan berbasis web menunjukkan bahwa penerapan framework mempercepat proses rekonsiliasi keuangan perusahaan distribusi. Ketiga, penelitian oleh (Kurniawan, 2024) mengimplementasikan model Waterfall pada sistem manajemen surat jalan, namun fokusnya terbatas pada pelacakan logistik tanpa integrasi *invoice*. Keempat, penelitian (Arisandi, 2025) menyoroti pentingnya kontrol akses berbasis peran (*Role-Based Access Control*) pada sistem informasi perusahaan untuk mencegah kebocoran data. Kelima, penelitian (Hidayat, 2025) menggunakan PHP murni untuk sistem administrasi, namun menyarankan penggunaan *framework* untuk keamanan yang lebih baik.

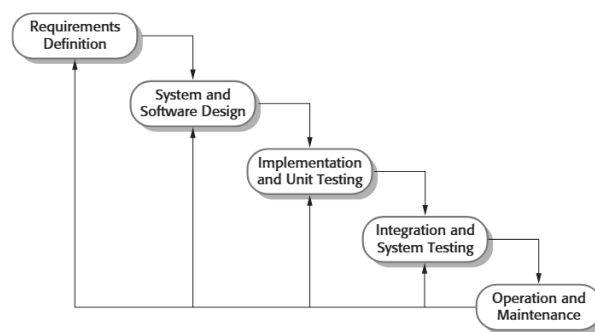
Berdasarkan tinjauan pustaka yang telah dilakukan, diketahui bahwa sebagian besar sistem yang ada hanya berfokus pada salah satu aspek (logistik saja atau keuangan saja) dan jarang yang mengintegrasikan alur spesifik antara surat jalan dengan penerbitan *invoice* secara otomatis dalam satu pintu akses sesuai kebutuhan distributor produk. SIGAB hadir untuk mengisi celah tersebut dengan mengintegrasikan kedua dokumen krusial tersebut dalam satu

ekosistem web yang memiliki pembagian peran spesifik untuk *owner*, administrator dan staf, sehingga aliran data menjadi *linear* juga tervalidasi.

3. METODE PENELITIAN

Metode Waterfall

Pengembangan sistem informasi SIGAB dilakukan melalui alur kerja *Software Development Life Cycle* (SDLC), yaitu proses standar yang digunakan oleh organisasi untuk membangun, memelihara, dan mengganti suatu sistem informasi guna meningkatkan kualitas perangkat lunak (Pressman & Maxim, 2020). Salah satu metode klasik yang sangat populer dalam ekosistem SDLC adalah model Waterfall. Metode ini merupakan pendekatan sekuensial linier dimana setiap fase harus diselesaikan sepenuhnya sebelum melangkah ke fase berikutnya, sehingga memastikan dokumentasi dan struktur proyek yang sangat disiplin (Dhika et al., 2022). Pada penelitian ini, tahapan Waterfall dibatasi mulai dari tahap analisis kebutuhan (*requirements*) hingga tahap pengujian sistem (*system testing*) saja, tanpa mencakup tahap pemeliharaan (*maintenance*). Pembatasan ini dilakukan karena fokus utama riset adalah pada validasi rancang bangun dan fungsionalitas sistem informasi di lingkungan operasional PT Gawi Anjung Bersama. Gambar 1 mengilustrasikan pemodelan Waterfall (Sommerville, 2021).



Gambar 1. Tahapan metode Waterfall menurut Sommerville.

Tahapan metode Waterfall dapat bervariasi, tergantung sumber rujukan yang digunakan. Setiap versinya memiliki kekurangan dan kelebihan. Penelitian ini memilih versi Sommerville karena tahapannya yang mudah diikuti. Adapun uraian tentang Waterfall menurut Soumerville adalah sebagai berikut:

Requirements Definition (Penentuan Kebutuhan)

Sering pula disebut *requirements analysis and definition* (definisi/penentuan dan analisis kebutuhan) atau tahap analisis. Pada fase ini, seluruh layanan, batasan, dan tujuan sistem ditetapkan melalui konsultasi dengan pengguna sistem, yang kemudian didefinisikan secara

rinci untuk berfungsi sebagai spesifikasi sistem (Sirajuddin & Souwakil, 2022). Proses ini merupakan langkah krusial untuk memastikan bahwa pengembang memahami sepenuhnya permasalahan bisnis yang dihadapi oleh PT GAB sebelum solusi teknis dirancang (Fachrizal & Ismail, 2023).

System and Software Design (Desain Sistem dan Perangkat Lunak)

Biasa dikenal dengan sebutan tahap desain. Proses ini melibatkan identifikasi dan deskripsi abstraksi sistem mendasar, seperti pemodelan UML (*Unified Modelling Language*), relasi database, antarmuka sistem, serta hubungannya satu sama lain (Rosa & Shalahuddin, 2022).

Implementation and Unit Testing (Implementasi dan Pengujian Unit)

Lebih populer dengan istilah proses implementasi. Pada tahap ini, desain perangkat lunak direalisasikan sebagai satu set program atau unit program melalui proses pengkodean (Sirajuddin & Souwakil, 2022).

Integration and System Testing (Integrasi dan Pengujian Sistem)

Unit program atau program individu diintegrasikan dan diuji sebagai sistem yang lengkap untuk memastikan bahwa persyaratan perangkat lunak telah dipenuhi. Setelah pengujian sistem selesai dan dinyatakan valid melalui metode *blackbox*, perangkat lunak siap di-hosting dan digunakan oleh pihak PT GAB (Rosa & Shalahuddin, 2022).

Operation and Maintenance (Operasi dan Pengelolaan)

Tahap ini merupakan fase terlama dalam siklus hidup di mana sistem dipasang dan digunakan secara praktis oleh pengguna. Pemeliharaan melibatkan koreksi kesalahan yang tidak ditemukan pada tahap awal, meningkatkan implementasi unit sistem, dan memperluas layanan sistem seiring munculnya persyaratan baru (Sirajuddin & Souwakil, 2022).

Pengumpulan Data

Tahapan pengumpulan data dalam penelitian ini dilakukan secara sistematis untuk memperoleh informasi yang akurat dan relevan dengan kebutuhan sistem pada PT Gawi Anjung Bersama. Peneliti mengategorikan sumber data ke dalam dua jenis utama, yaitu data primer yang diperoleh langsung dari lapangan dan data sekunder yang berasal dari referensi pendukung yang telah ada sebelumnya.

Data primer dikumpulkan melalui teknik observasi dan wawancara mendalam untuk memahami proses bisnis pengelolaan surat jalan dan *invoice* saat ini. Observasi dilakukan dengan mengamati secara langsung bagaimana staf administrasi mengelola dokumen menggunakan perangkat lunak konvensional, sementara wawancara dilakukan dengan pihak

owner dan administrator untuk mengidentifikasi kendala operasional serta kebutuhan fungsional sistem SIGAB yang akan dikembangkan.

Selain data lapangan, peneliti juga menghimpun data sekunder melalui studi pustaka untuk memperkuat landasan teoretis penelitian. Proses ini melibatkan penelaahan berbagai literatur, seperti buku teks mengenai rekayasa perangkat lunak, jurnal ilmiah terkait *framework* CodeIgniter, serta dokumen internal perusahaan yang berkaitan dengan format standar surat jalan dan *invoice*.

4. HASIL DAN PEMBAHASAN

Bagian ini membahas tentang penerapan metode Waterfall versi Sommerville dalam proses rancang bangun web SIGAB berbasis *framework* CodeIgniter. Tahapan dimulai dari analisis (*requirements definition*), desain (*system and software design*), implementasi (*implementation and unit testing*), dan pengujian (*integration and system testing*) sistem.

Analisis

Analisis dalam penelitian ini menggunakan pendekatan kebutuhan fungsional (*functional requirements*) dan non-fungsional (*non-functional requirements*) guna mendefinisikan apa yang harus dilakukan sistem beserta batasannya.

Kebutuhan Fungsional Sistem

Kebutuhan fungsional merupakan pernyataan layanan yang harus disediakan oleh sistem, mencakup bagaimana sistem harus bereaksi terhadap input tertentu serta bagaimana sistem berperilaku pada situasi yang spesifik (Sirajuddin & Souwakil, 2022). Fungsi utama dari analisis ini adalah untuk mendefinisikan secara rinci seluruh prosedur kerja, fitur, dan kapabilitas teknis yang wajib dimiliki oleh perangkat lunak guna memenuhi ekspektasi pengguna (Pressman & Maxim, 2020). Dalam penelitian ini, analisis kebutuhan fungsional menjadi sangat krusial untuk memastikan bahwa transisi dari pengelolaan manual berbasis Excel ke sistem informasi dapat mengakomodasi seluruh alur kerja surat jalan dan invoice secara akurat. Penentuan kebutuhan ini didasarkan pada hasil studi pustaka, observasi dan wawancara bersama *owner* serta staf PT GAB. Tabel 1 berikut menunjukkan hasil analisis kebutuhan fungsional SIGAB.

Tabel 1. Analisis Kebutuhan Fungsional Sistem.

Aktor	Sistem
Owner	<ul style="list-style-type: none"> a. Mampu memproses login sebagai owner. b. Mampu mengubah profil dan katasandi. c. Mampu mengelola data administrator dan staf, seperti tambah, edit, hapus beserta mengatur hak aksesnya. d. Mampu menampilkan halaman beranda yang memuat data jumlah RfQ (<i>Request for Quotation</i>), PO (<i>Purchase Order</i>), <i>invoice</i>, surat jalan dan pengguna, serta pendapatan tahun berjalan. e. Mampu mengelola data RfQ, seperti tambah data secara manual, edit, hapus, impor (tambah) data via file Excel, cetak dan ekspor data ke format Excel. f. Mampu mengubah, ekspor dan cetak data PO. g. Mampu menampilkan, menerbitkan, edit, cetak dan unduh (sesuai format perusahaan), serta hapus data surat jalan. h. Mampu menampilkan, menerbitkan, edit, cetak dan unduh (sesuai format perusahaan), serta hapus data <i>invoice</i>. i. Mampu mengubah profil usaha. j. Mampu memantau aktivitas seluruh pegawai (administrator dan staf) yang melakukan tambah, edit, dan hapus data (RfQ, PO, surat jalan, serta <i>invoice</i>) melalui halaman <i>Activity Log</i>. k. Mampu menghapus seluruh data <i>Activity Log</i>.
Administrator	<ul style="list-style-type: none"> a. Mampu memproses login sebagai administrator. b. Mampu mengubah profil dan katasandi. c. Mampu mengelola data staf (kecuali <i>owner</i>), seperti tambah, edit, hapus dan mengatur hak akses pengguna. d. Mampu menampilkan halaman beranda yang memuat data jumlah RfQ (<i>Request for Quotation</i>), PO (<i>Purchase Order</i>), <i>invoice</i>, surat jalan dan pengguna, serta pendapatan tahun berjalan. e. Mampu mengelola data RfQ, seperti tambah data secara manual, edit, hapus, impor (tambah) data via file Excel, cetak dan ekspor data ke format Excel. f. Mampu mengubah, ekspor dan cetak data PO. g. Mampu menampilkan, menerbitkan, edit, cetak dan unduh (sesuai format perusahaan), serta hapus data surat jalan. h. Mampu menampilkan, menerbitkan, edit, cetak dan unduh (sesuai format perusahaan), serta hapus data <i>invoice</i>. i. Mampu mengubah profil usaha. j. Mampu memantau aktivitas administrator dan staf (kecuali <i>owner</i>) yang melakukan tambah, edit, dan hapus data (RfQ, PO, surat jalan, serta <i>invoice</i>) melalui halaman <i>Activity Log</i>. k. Mampu menghapus seluruh data <i>Activity Log</i>.
Staf	<ul style="list-style-type: none"> a. Mampu memproses login sebagai staf. b. Mampu mengubah profil dan katasandi. c. Mampu menampilkan halaman beranda yang memuat data jumlah RfQ (<i>Request for Quotation</i>), PO (<i>Purchase Order</i>), <i>invoice</i>, surat jalan dan pengguna, serta pendapatan tahun berjalan. d. Mampu mengelola data RfQ sesuai hak akses, seperti tambah data secara manual, edit, hapus, impor (tambah) data via file Excel, cetak dan ekspor data ke format Excel. e. Mampu mengubah, ekspor dan cetak data PO sesuai hak akses. f. Mampu menampilkan, menerbitkan, edit, cetak dan unduh (sesuai format perusahaan), serta hapus data surat jalan sesuai hak akses. g. Mampu menampilkan, menerbitkan, edit, cetak dan unduh (sesuai format perusahaan), serta hapus data <i>invoice</i> sesuai hak akses. h. Mampu memantau aktivitasnya saat melakukan tambah, edit, dan hapus data (RfQ, PO, surat jalan, serta <i>invoice</i>) melalui halaman <i>Activity Log</i>.

Hak akses dalam SIGAB terbagi menjadi 4, yaitu lihat halaman, tambah, edit dan hapus data. Aturan tersebut berlaku untuk halaman-halaman tertentu yakni halaman Penawaran

(RfQ), Purchase Order (PO), Surat Jalan, dan Invoice. Pengaturan ini penting agar data yang diolah pada SIGAB dapat dipertanggungjawabkan di lingkup perusahaan.

Kebutuhan Non-Fungsional Sistem

Kebutuhan non-fungsional adalah batasan pada layanan atau fungsi yang ditawarkan oleh sistem, yang mencakup aspek-aspek seperti kendala waktu, standar operasional, hingga persyaratan teknis terkait perangkat keras dan perangkat lunak (Sirajuddin & Souwakil, 2022). Tabel 2 memperlihatkan hasil analisis kebutuhan non-fungsional sistem.

Tabel 2. Analisis Kebutuhan Non-fungsional Sistem.

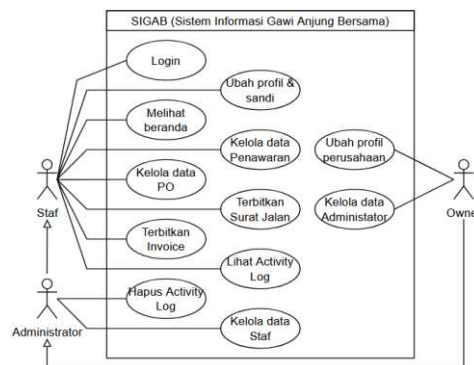
Aktor	Sistem
Keamanan (<i>Security</i>)	Mampu menerima atau menolak akses pengguna, serta menampilkan fitur-fitur sistem informasi sesuai hak akses.
Ketersediaan (<i>Availability</i>)	Mampu diakses 24/7 jam melalui jaringan internet.
Akuntabilitas (<i>Accountability</i>)	Mampu mencatat log aktivitas setiap pengguna yang melakukan penambahan, perubahan dan penghapusan data pada halaman-halaman tertentu.

Desain

Pada tahap ini, hasil analisis kebutuhan fungsional dan non-fungsional diterjemahkan menjadi desain sistem. Proses ini bertujuan agar meminimalisir kesalahan dan mempermudah tahap implementasi sistem. Desain yang digunakan dalam penelitian ini mencakup pemodelan UML (*use case* dan *class diagram*), ERD dan wireframe dari sistem yang akan dibangun.

Use Case Diagram

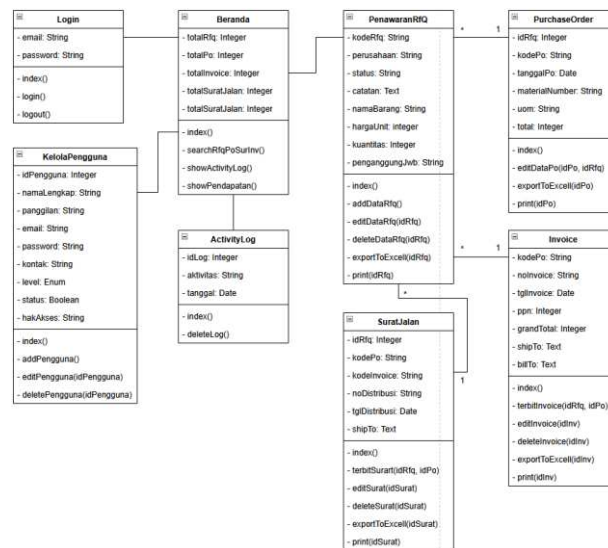
Diagram kasus penggunaan (*use case diagram*) adalah model grafis yang berfungsi untuk menggambarkan interaksi antara aktor dengan serangkaian kegiatan utama yang disediakan oleh sistem (Sirajuddin & Souwakil, 2022). Diagram ini menggunakan beberapa simbol standar seperti *actor* (aktor) untuk mewakili entitas luar, *use case* (kasus penggunaan) untuk mewakili fungsi sistem, serta *association* (asosiasi) untuk menunjukkan hubungan di antara keduanya (Rosa & Shalahuddin, 2022). Gambar 2 berikut memperlihatkan *use case diagram* dari SIGAB.



Gambar 2. Use case diagram sistem.

Class Diagram

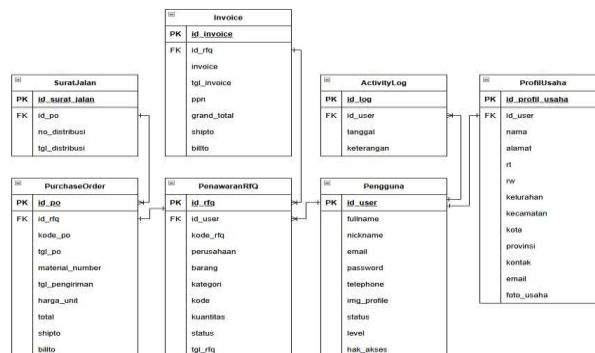
Disebut juga diagram kelas, merupakan model statis yang berfungsi untuk menggambarkan struktur sistem dengan merinci kelas-kelas yang ada, atribut, metode (*method/function*), serta hubungan antar objek dalam perangkat lunak (Booch et al., 2021). Diagram ini menggunakan simbol-simbol standar seperti kotak kelas yang terbagi menjadi tiga bagian (nama, atribut, dan method) serta garis association atau *multiplicity* (multiplisitas) untuk menunjukkan relasi antar kelas (Rosa & Shalahuddin, 2022). Gambar 3 berikut menunjukkan class diagram dari SIGAB.



Gambar 3. Class diagram sistem.

Entity Relationship Diagram (ERD)

Entity Relationship Diagram (diagram hubungan entitas) adalah model konseptual yang berfungsi untuk menggambarkan struktur logis serta hubungan antar data dalam sebuah basis data relasional (Elmasri & Navathe, 2021). Dengan adanya ERD, developer dapat memastikan bahwa integritas data tetap terjaga dan struktur tabel yang dibangun telah memenuhi kaidah normalisasi guna menghindari redundansi data (Raditya Putra, 2022). ERD untuk SIGAB dapat dilihat pada Gambar 4 berikut.



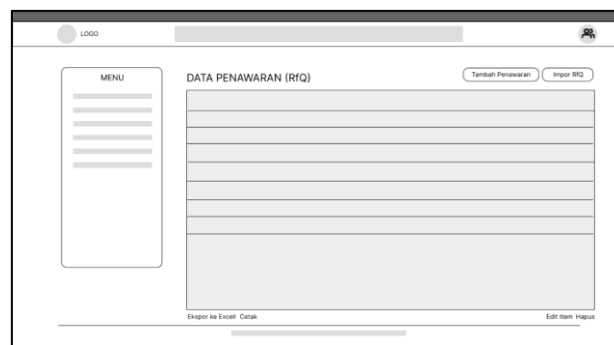
Gambar 4. Desain ERD sistem.

Wireframe

Wireframe (kerangka dasar) merupakan skema visual dua dimensi yang berfungsi untuk merepresentasikan struktur halaman, tata letak konten, serta fitur antarmuka (interface) sistem sebelum implementasi visual dilakukan (Garrett, 2021). Dalam penelitian ini, *wireframe* digunakan untuk memetakan alur interaksi pengguna pada sistem SIGAB secara presisi guna memastikan setiap fitur, seperti manajemen surat jalan dan *invoice*, memiliki tata letak yang rapih dan mudah dioperasikan oleh pengguna (Lazuardi & Hasibuan, 2023). Terdapat banyak antarmuka dari sistem yang dibangun, namun dalam penelitian ini dibatasi pada desain-desain *wireframe* pokok saja, seperti surat jalan, *invoice*, dan seterusnya.

Desain Halaman Penawaran (RfQ)

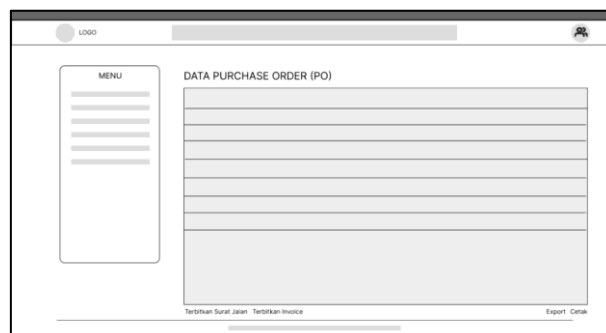
Halaman RfQ berguna untuk menambahkan data barang yang akan disuplai oleh perusahaan. Data dari halaman ini nantinya menjadi induk untuk halaman-halaman lain dalam pengelolaan surat jalan dan *invoice*. Desain halaman RfQ dapat dilihat pada Gambar 5 berikut.



Gambar 5. Desain halaman Penawaran (RfQ).

Desain Purchase Order (PO)

Barang-barang suplai yang diinput melalui halaman RfQ harus memperoleh kode PO, yang menandakan pesanan sudah difiksasi oleh pelanggan. Data lain yang juga ditambahkan pengguna SIGAB pada halaman ini yakni status suplai dan tanggal PO. Gambar 6 berikut mengilustrasikan desain halaman PO.



Gambar 6. Desain halaman Purchase Order (PO).

Desain Surat Jalan

Setelah mendapat kode PO, barang-barang kemudian didistribusikan ke pembeli melalui prosedur internal PT GAB. Prosedur tersebut memerlukan surat jalan agar dapat dilaksanakan. Halaman ini berguna untuk menerbitkan surat yang dimaksud. Adapun desain halaman Surat Jalan dapat dilihat pada Gambar 7 berikut.



Gambar 7. Desain halaman Surat Jalan.

Desain Invoice

Selain surat jalan, pihak perusahaan juga akan menerbitkan *invoice* (surat tagihan) saat distribusi dilakukan. Halaman ini berguna untuk membuat surat tersebut. Gambar 8 berikut menampilkan desain dari halaman *invoice*.



Gambar 8. Desain halaman *Invoice*.

Implementasi

Pada tahap ini, hasil analisis dan desain sistem direalisasikan menjadi sebuah web berbasis framework CodeIgniter. Adapun yang diimplementasikan yakni database dan halaman-halaman dari web SIGAB.

Implementasi Database

Tahap ini merupakan realisasi dari rancangan ERD ke dalam sistem manajemen basis data MySQL. Di tahap ini, seluruh entitas, atribut, dan relasi yang telah direncanakan dikonversi ke dalam bentuk tabel-tabel fisik untuk menyimpan data operasional perusahaan secara terstruktur. Gambar 9 berikut memperlihatkan hasil implementasi database SIGAB.

Table	Action	Rows	Type	Collation	Size	Overhead
tb_corousel	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	36 K18	-
tb_invoice	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	36 K18	-
tb_loj	Browse Structure Search Insert Empty Drop	128	InnoDB	latin1_swedish_ci	36 K18	-
tb_po	Browse Structure Search Insert Empty Drop	18	InnoDB	utf8mb4_general_ci	36 K18	-
tb_profil_usaha	Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	36 K18	-
tb_rekapan_office	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	36 K18	-
tb_rfq	Browse Structure Search Insert Empty Drop	18	InnoDB	utf8mb4_general_ci	36 K18	-
tb_surat_jalan	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	36 K18	-
tb_user	Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	36 K18	-
9 tables	Sum		152 InnoDB	latin1_swedish_ci	344 K18	0.6

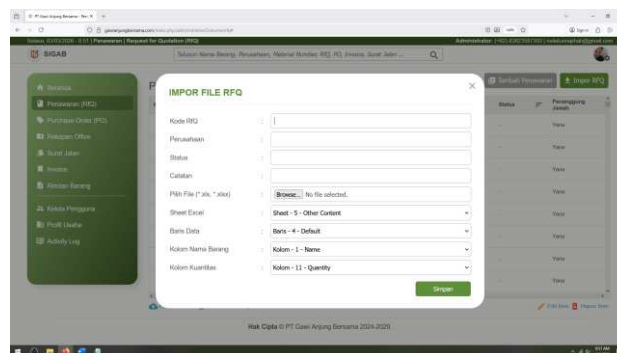
Gambar 9. Hasil implementasi database SIGAB.

Implementasi Sistem

Setelah database sistem dibangun, maka implementasi sistem dapat dilaksanakan sebagai pusat pengelolaan data SIGAB agar surat jalan, *invoice* juga data-data lainnya bisa diproses. Adapun halaman-halaman yang dibangun pada tahap ini diuraikan sebagai berikut.

a. Halaman Penawaran (RfQ)

Pada halaman ini, pengguna dapat menambahkan barang yang akan disuplai yakni dengan memasukkan kode RfQ, perusahaan pembeli, nama barang, kuantitas, status dan catatan. Input barang manual ini dapat dilakukan satu atau lebih produk. Selain itu, pengguna dapat pula menambahkan barang dengan cara impor data melalui file Excel sehingga lebih praktis untuk jumlah data yang besar. Tampilan halaman ini dapat dilihat pada Gambar 10 berikut.



Gambar 10. Implementasi halaman Penawaran (RfQ).

b. Halaman Purchase Order (PO)

Hanya barang yang sudah memiliki kode PO yang dapat diterbitkan surat jalan dan *invoice*-nya. Halaman ini berguna untuk menambahkan kode tersebut beserta kelengkapan data lainnya. Halaman ini dapat diamati pada Gambar 11 berikut.

Gambar 11. Implementasi halaman Purchase Order (PO).

c. Halaman Surat Jalan

Saat penerbitan surat jalan, pengguna perlu memasukkan data-data berupa kode PO, kode *invoice*, nama barang, nomor dan tanggal distribusi, serta alamat kirim. Gambar 12 berikut menampilkan halaman tersebut.

Gambar 12. Implementasi halaman Surat Jalan.

d. Halaman Invoice

Pada halaman *invoice*, pengguna menerbitkan surat ini dengan menambahkan data-data berupa kode PO, nama barang, tanggal, Pajak Pertambahan Nilai (PPn), *grand total*, alamat kirim dan alamat tagihan. Halaman ini dapat dilihat pada Gambar 13 berikut.

Gambar 13. Implementasi halaman Invoice.

8	Kelola Invoice	Menerbitkan, edit, hapus, cetak, dan unduh dokumen <i>invoice</i> .	Dokumen <i>invoice</i> berhasil dibuat secara otomatis dan siap ditagihkan.	Valid
9	Profil Usaha	Mengubah informasi identitas perusahaan (<i>owner/admin</i>).	Informasi perusahaan pada format surat dan sistem berhasil diperbarui.	Valid
10	Activity Log	Memantau riwayat tambah, edit, dan hapus data yang dilakukan pegawai.	Sistem mencatat setiap aksi pengguna di tabel <i>log</i> .	Valid

5. KESIMPULAN DAN SARAN

Berdasarkan hasil dan pembahasan penelitian, dapat disimpulkan bahwa SIGAB berhasil menjawab permasalahan pengelolaan surat jalan dan invoice yang ada di PT Gawi Anjung Bersama. Perpindahan dari aplikasi konvensional berbasis Microsoft Excel dan Word menjadi sistem informasi berbasis CodeIgniter terbukti mampu menyelesaikan kendala duplikasi data, risiko human error, serta hambatan komunikasi antar staf khususnya dalam berbagi data surat jalan dan invoice. Penggunaan arsitektur Model-View-Controller (MVC) menghasilkan sistem yang ringan, stabil, dan mudah dikelola. Dari sisi keamanan, penerapan hak akses yang terbagi menjadi aktor owner, administrator, dan staf telah berhasil menjamin integritas data sesuai dengan tanggung jawab masing-masing pengguna. Hasil pengujian blackbox testing menunjukkan bahwa seluruh fitur fungsional sistem dapat berjalan sesuai kebutuhan. Meskipun demikian, penelitian ini masih memiliki keterbatasan yakni belum adanya fitur pemantauan posisi armada secara real-time (waktu nyata), juga integrasi sistem pembayaran elektronik. Oleh karena itu, penelitian selanjutnya disarankan untuk mengembangkan modul pelacakan logistik berbasis Global Positioning System (GPS) serta mengintegrasikan payment gateway (gerbang pembayaran) untuk mempercepat rekonsiliasi pembayaran. Di sisi lain, pengembangan aplikasi ke arah platform mobile berbasis android maupun iOS juga direkomendasikan agar aksesibilitas data bagi pengemudi dan manajemen dapat dilakukan dengan lebih fleksibel di luar kantor.

DAFTAR REFERENSI

- Arisandi, B. (2025). Keamanan data berbasis role access pada web perusahaan. *Jurnal Sains Dan Informatika*, 12(1), 88-96.
- Booch, G., Rumbaugh, J., & Jacobson, I. (2021). *The Unified Modeling Language user guide* (2nd ed.). Addison-Wesley.
- Dhika, H., Dessy, N., & Putra, M. G. L. (2022). Penerapan metode waterfall pada sistem informasi penjualan berbasis web. *Jurnal Teknologi Dan Sistem Informasi*, 3(1), 1-8. <https://doi.org/10.54259/satesi.v1i1.4>
- Elmasri, R., & Navathe, S. B. (2021). *Fundamentals of database systems* (7th ed.). Pearson.

- Fachrizal, M. R., & Ismail, S. J. I. (2023). Implementasi model waterfall pada rancang bangun sistem informasi administrasi. *Jurnal Informatika Dan Teknologi*, 6(1), 22-30.
- Fowler, M. (2021). *UML distilled: A brief guide to the standard object modeling language* (3rd ed.). Addison-Wesley.
- Garrett, J. J. (2021). *The elements of user experience: User-centered design for the web and beyond* (2nd ed.). New Riders.
- Hidayat, F. (2025). Komparasi framework MVC dalam pengembangan sistem administrasi. *Jurnal Tekno Komputer*, 10(2), 33-41.
- Kurniawan, M. (2024). Implementasi metode waterfall dalam pengembangan sistem surat jalan. *Jurnal Pengembangan Teknologi Informasi*, 8(3), 55-64.
- Lazuardi, A. J., & Hasibuan, M. S. (2023). Penerapan user centered design dalam perancangan antarmuka sistem informasi. *Jurnal Informatika Dan Teknologi*, 6(2), 45-53.
- Mulyani, S. (2022). *Analisis dan perancangan sistem informasi manajemen perusahaan* (3rd ed.). Abdi Sistematika.
- Pratama, & Hartono, R. (2023). Transformasi digital dan peran sistem informasi dalam skalabilitas bisnis modern. *Jurnal Sistem Informasi Dan Teknologi*, 9(2), 45-52. <https://doi.org/10.54259/satesi.v2i1.784>
- Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill Education.
- Raditya Putra, I. G. L. A. (2022). Perancangan basis data relasional pada sistem informasi administrasi. *Jurnal Informatika Dan Rekayasa Perangkat Lunak*, 3(2), 112-120.
- Rahmawati, L. (2023). Optimasi sistem penagihan menggunakan framework PHP pada sektor industri. *Jurnal Informatika Ekonomi*, 11(4), 101-110.
- Rosa, A. S., & Shalahuddin, M. (2022). *Rekayasa perangkat lunak terstruktur dan berorientasi objek*. Informatika.
- Saputra, D. (2022). Digitalisasi administrasi gudang pada perusahaan distribusi logistik. *Jurnal Riset Teknologi Informasi*, 7(1), 12-20.
- Siddique, M. S., & Ali, B. A. (2024). Analisis performa framework PHP dalam pengembangan aplikasi web skala menengah. *Jurnal Sistem Komputer Dan Informatika*, 5(1), 12-20.
- Sirajuddin, H. K., & Souwakil, A. S. (2022). Analisis dan perancangan sistem informasi pengolahan data penjualan berbasis web. *Jurnal Ilmiah ILKOMINFO: Ilmu Komputer & Informatika*, 5(1), 34-42.
- Sommerville, I. (2021). *Software engineering* (10th ed.). Pearson Education.