BAB II

KAJIAN LITERATUR

2.1 Konsep Sistem Informasi

Konsep sistem informasi menjadi landasan teoritis penting karena menjelaskan bagaimana organisasi membangun mekanisme terstruktur untuk mengelola data operasional agar dapat menghasilkan informasi yang berguna bagi pelaksanaan kegiatan dan pengambilan keputusan. Pada praktiknya, sistem informasi tidak hanya dipahami sebagai aplikasi, melainkan sebagai rangkaian proses yang mengoordinasikan aktivitas pengumpulan data, pengolahan, penyimpanan, dan penyajian informasi sehingga alur kerja organisasi lebih terkendali serta mudah dipantau.

Selain itu, pembahasan konsep sistem informasi diperlukan agar pengembangan sistem tidak berhenti pada implementasi teknis, tetapi juga memperhatikan keterkaitan proses bisnis, data yang dikelola, serta tujuan informasi yang ingin dihasilkan. Kebutuhan organisasi terhadap ketepatan, kecepatan, dan integrasi informasi dalam proses bisnis mendorong pemanfaatan sistem informasi sebagai sarana pengelolaan data yang lebih terstruktur dan terpusat sehingga memudahkan pemantauan aktivitas dan mengurangi ketidaksesuaian pencatatan [8]. Dengan dasar pengantar ini, subbab berikutnya menguraikan konsep sistem dan informasi sebagai elemen pembentuk utama sebelum membahas sistem informasi sebagai kesatuan.

2.1.1 Sistem

Sistem merupakan satu kesatuan yang tersusun atas elemen/komponen (subsistem) yang saling berhubungan dan bekerja sama untuk mencapai tujuan tertentu. Karena adanya keterkaitan antar komponen, perubahan pada satu bagian dapat memengaruhi bagian lain dan berdampak pada kinerja keseluruhan sistem, sehingga sistem perlu dipahami sebagai struktur yang terintegrasi, bukan kumpulan bagian yang berdiri sendiri [9], [10].

Dalam konteks organisasi, sistem umumnya dijelaskan melalui unsur seperti batas sistem (*system boundary*), lingkungan luar (*environment*), masukan (*input*), proses (*process*), dan keluaran (*output*). Batas sistem berfungsi mendefinisikan ruang lingkup sistem (membedakan apa yang termasuk sistem dan di luar sistem), sedangkan lingkungan luar mencakup faktor eksternal yang dapat memengaruhi sistem. Mekanisme *input–process–output* menggambarkan transformasi kerja sistem, yaitu *input* diolah melalui aturan/prosedur

tertentu sehingga menghasilkan *output* sesuai tujuan, dan keberadaan pengendalian diperlukan agar proses berjalan sesuai target serta mengurangi penyimpangan [9].

Secara teoritis, sistem dapat diklasifikasikan untuk memudahkan analisis, misalnya sistem fisik dan sistem abstrak, sistem deterministik dan sistem probabilistik, serta sistem terbuka dan sistem tertutup. Sistem deterministik memiliki perilaku yang relatif dapat diprediksi, sedangkan sistem probabilistik dipengaruhi ketidakpastian sehingga hasilnya tidak selalu dapat dipastikan. Sistem tertutup relatif minim interaksi dengan lingkungan, sedangkan sistem terbuka berinteraksi dengan lingkungan serta menerima masukan dan menghasilkan keluaran; klasifikasi ini membantu peneliti menentukan karakter sistem yang dikaji sehingga pendekatan perancangan dan pengendalian dapat disesuaikan dengan konteks organisasi [9], [10].

2.1.2 Informasi

Informasi merupakan data yang telah diolah sehingga menjadi lebih terstruktur, bermakna, dan berguna bagi pengguna. Data pada dasarnya masih berupa fakta mentah (misalnya angka stok, catatan transaksi, nama barang), sedangkan informasi muncul setelah data diproses melalui aktivitas seperti pengelompokan, perhitungan, penyaringan, dan penyajian agar dapat dipahami serta dimanfaatkan dalam konteks tertentu [11]. Dalam praktik sistem informasi, informasi diposisikan sebagai keluaran (*output*) yang memberikan nilai tambah karena membantu pengguna memahami kondisi dan menyusun tindakan berdasarkan data yang sudah "siap pakai".

Nilai guna informasi sangat ditentukan oleh konteks pemakaian dan kebutuhan keputusan. Artinya, data yang sama dapat menghasilkan informasi yang berbeda tergantung tujuan pengolahan, aturan bisnis, serta cara penyajiannya (misalnya ringkasan persediaan per periode, per lokasi gudang, atau per proyek). Karena itu, pengolahan data tidak sekadar mengubah bentuk, tetapi juga menambahkan struktur dan interpretasi agar informasi yang dihasilkan relevan untuk tugas pengguna, termasuk untuk memantau keadaan, membandingkan kondisi antar waktu, dan mengevaluasi pencapaian target [11].

Kualitas informasi menjadi syarat penting agar informasi benar-benar dapat mendukung pengambilan keputusan. Informasi yang berkualitas umumnya dicirikan oleh akurat (minim kesalahan), relevan (sesuai kebutuhan pengguna), tepat waktu (tersedia saat dibutuhkan), dan lengkap (mewakili aspek penting yang diperlukan) [12]. Jika karakteristik ini tidak terpenuhi, maka informasi berisiko menimbulkan salah interpretasi, keterlambatan

tindakan, serta keputusan yang tidak tepat, terutama pada proses operasional yang membutuhkan pembaruan data secara cepat dan konsisten [12].

2.1.3 Sistem Informasi

Sistem informasi merupakan suatu sistem terorganisasi yang digunakan untuk mendukung kebutuhan organisasi melalui pengelolaan data menjadi informasi, sehingga aktivitas operasional dan administrasi dapat berjalan lebih terarah serta informasi yang dibutuhkan pengguna dapat tersedia sesuai konteks pekerjaan [10]. Dengan demikian, sistem informasi tidak hanya dipahami sebagai aplikasi, tetapi sebagai mekanisme kerja yang memadukan unsur manusia, prosedur, dan sarana teknologi untuk menghasilkan informasi yang bernilai guna. Peran ini menjadikan sistem informasi penting dalam membantu organisasi menjaga keteraturan proses bisnis, karena aliran data dan informasi dapat diolah secara lebih terstandar dibandingkan pencatatan manual [10].

Secara fungsional, sistem informasi menjalankan alur pengelolaan data yang mencakup pengumpulan/pencatatan data (*input*), pemrosesan data (*processing*) seperti validasi, pengelompokan, perhitungan, dan pembaruan status, dilanjutkan penyimpanan terstruktur (*storage*) agar data dapat ditelusuri kembali, lalu penyajian keluaran (*output*) dalam bentuk laporan atau informasi yang dapat dimanfaatkan pengguna [10]. Alur ini diperlukan agar data tidak sekadar tersimpan, tetapi dapat diolah secara konsisten menjadi informasi yang mendukung pekerjaan dan keputusan. Selain itu, penyimpanan terstruktur memungkinkan data historis digunakan kembali untuk kebutuhan pelaporan berkala, penelusuran transaksi, serta evaluasi kondisi operasional berdasarkan periode tertentu [10].

Gambar 2.1 Sistem Informasi *[13]*

Komponen sistem informasi mencakup perangkat keras (*hardware*), perangkat lunak (*software*), pengguna/pengelola (*brainwar*e), basis data (*database*), prosedur (*procedure*), serta jaringan/komunikasi (*network*) [14]. *Hardware* menyediakan sarana fisik pemrosesan, *software* menjalankan logika aplikasi, *brainware* mengoperasikan dan mengendalikan penggunaan sistem, *database* menjadi pusat penyimpanan data terstruktur, prosedur mengatur alur kerja agar penggunaan sistem konsisten, dan *network* memungkinkan pertukaran data terutama pada lingkungan *multiuser* [14]. Keterpaduan antar komponen menjadi kunci karena ketidaksesuaian pada satu komponen dapat memengaruhi kinerja komponen lainnya; misalnya prosedur yang tidak baku dapat menyebabkan *input* tidak seragam, sementara pengelolaan *database* yang kurang baik dapat memicu ketidakkonsistenan data, yang pada akhirnya menurunkan kualitas informasi serta menghambat efektivitas proses kerja dan pengambilan keputusan. Dalam sistem yang digunakan banyak pengguna, pengaturan prosedur dan jaringan juga membantu menjaga sinkronisasi data, sehingga informasi yang ditampilkan tetap sesuai dengan kondisi yang tercatat di sistem [14].

2.2 Konsep Inventaris

Inventaris (persediaan) merupakan aset atau barang yang disimpan organisasi guna memenuhi kebutuhan operasional dan permintaan, serta mencegah gangguan layanan akibat kekurangan barang. Inventaris perlu dikelola secara efektif karena pergerakan stok memengaruhi kontinuitas operasi seperti produksi dan distribusi. Sebagian literatur menjelaskan bahwa manajemen persediaan harus menjamin ketersediaan stok serta mengatur mekanisme seperti *safety stock* dan *reorder point* untuk menjaga keseimbangan antara tingkat pelayanan dan biaya persediaan [15].

Pengelolaan inventaris secara manual kerap menghadapi tantangan seperti pencatatan yang tidak akurat, risiko data salah/silang, serta lambatnya informasi stok untuk pengambilan keputusan. Oleh karena itu, penelitian pada sistem informasi persediaan barang menunjukkan bahwa penerapan sistem informasi yang terstruktur dapat mendukung pencatatan barang masuk/keluar, pengawasan stok, dan penyusunan laporan yang *real-time* sehingga membantu efisiensi operasional perusahaan dagang.

2.2.1 Barang Masuk

Barang masuk (pembelian) merupakan proses penambahan persediaan barang ke dalam gudang yang berasal dari aktivitas pengadaan dari pemasok, yang mencakup tahapan

pemesanan, penerimaan, pemeriksaan kesesuaian jumlah dan kualitas, serta pencatatan ke dalam sistem inventaris. Proses ini memiliki peran penting dalam menjaga akurasi data persediaan karena setiap transaksi yang dicatat akan memperbarui jumlah stok sekaligus mencatat harga perolehan barang sebagai dasar dalam perhitungan harga pokok barang (HPP). Ketelitian dalam pencatatan sangat diperlukan untuk mencegah terjadinya selisih antara stok fisik dan data sistem, serta mendukung pengendalian internal dan pengambilan keputusan manajerial yang berkaitan dengan persediaan, terutama dalam sistem informasi persediaan yang terkomputerisasi [16].

2.2.2 Barang Keluar

Barang keluar merupakan proses pengurangan persediaan barang dari gudang yang terjadi akibat aktivitas distribusi atau penjualan, yang secara langsung memengaruhi jumlah stok yang tersedia dalam sistem inventaris. Dalam praktiknya, setiap transaksi barang keluar harus dicatat secara sistematis untuk menjaga kesesuaian antara data sistem dan kondisi fisik di lapangan. Pencatatan ini tidak hanya berfungsi untuk memantau arus keluar barang, tetapi juga menjadi dasar dalam penyusunan laporan persediaan serta mendukung proses pengambilan keputusan manajerial. Sistem informasi persediaan yang terkomputerisasi mampu meningkatkan akurasi pencatatan barang keluar, mempercepat penyajian laporan, serta meminimalkan kesalahan yang sering terjadi pada sistem manual [17].

2.2.3 Metode Penilaian Persediaan

Dalam pengelolaan persediaan, perusahaan juga perlu menetapkan metode pengeluaran/penilaian persediaan agar pencatatan biaya barang keluar dan nilai persediaan akhir dilakukan secara konsisten, terutama ketika harga pembelian berubah antar waktu. Metode yang umum digunakan antara lain FIFO (*First In First Out*), LIFO (*Last In First Out*), dan AVG (*Weighted Average*/rata-rata tertimbang). FIFO mengasumsikan barang yang pertama masuk akan dikeluarkan terlebih dahulu sehingga sesuai untuk barang yang sensitif terhadap lama penyimpanan. LIFO mengasumsikan barang yang terakhir masuk dikeluarkan terlebih dahulu. Sementara itu, AVG menentukan biaya unit berdasarkan rata-rata tertimbang dari persediaan yang tersedia sehingga menghasilkan nilai biaya yang lebih stabil ketika terjadi fluktuasi harga pembelian [18].

Metode penilaian persediaan yang digunakan dalam panglong milik CV. Bintang Berjaya Sejahtera adalah metode *Average* (rata-rata tertimbang). Metode ini dipilih karena pergerakan stok barang yang terjadi bersifat tercampur, di mana setiap barang masuk pada

waktu yang berbeda akan digabungkan dengan stok yang sudah ada tanpa dibedakan berdasarkan waktu perolehannya. Pada metode ini, harga pokok barang dihitung berdasarkan rata-rata nilai persediaan yang tersedia, yaitu dengan membagi total nilai persediaan dengan jumlah unit barang, sehingga setiap unit barang yang keluar dihitung dengan harga pokok yang sama. Dengan pendekatan tersebut, sistem informasi inventaris dapat menghasilkan perhitungan harga pokok barang yang lebih akurat dan konsisten, serta mendukung penyajian informasi stok yang sesuai dengan kondisi aktual untuk keperluan pengendalian persediaan dan pengambilan keputusan.

2.3 Sistem Informasi Inventaris Barang

Sistem informasi inventaris barang adalah sistem yang dirancang untuk mengelola persediaan barang secara terintegrasi, mulai dari pendataan barang, pencatatan transaksi barang masuk dan barang keluar, pengendalian stok, hingga penyusunan laporan inventaris. Sistem ini bertujuan memastikan ketersediaan barang sesuai kebutuhan operasional, mencegah kekurangan (*stockout*) maupun kelebihan stok (*overstock*), serta meminimalkan risiko kehilangan dan ketidaksesuaian data.

Pengelolaan inventaris merupakan serangkaian aktivitas untuk menganalisis dan memantau persediaan secara berkala guna meminimalkan kesalahan pencatatan, ketidaksesuaian stok, maupun kehilangan barang pada perusahaan [19]. Pada sistem terkomputerisasi, pengelolaan inventaris umumnya mencakup beberapa fungsi utama:

1. pencatatan *master data* (barang, kategori, satuan, lokasi, pemasok, pelanggan),
2. pencatatan transaksi (penerimaan, pengeluaran, retur, penyesuaian stok),
3. pengendalian stok (minimum stok, peringatan stok, mutasi barang),
4. pelaporan (kartu stok, rekap transaksi, laporan persediaan periode tertentu),
5. kontrol akses pengguna sesuai peran (*role-based access control*).

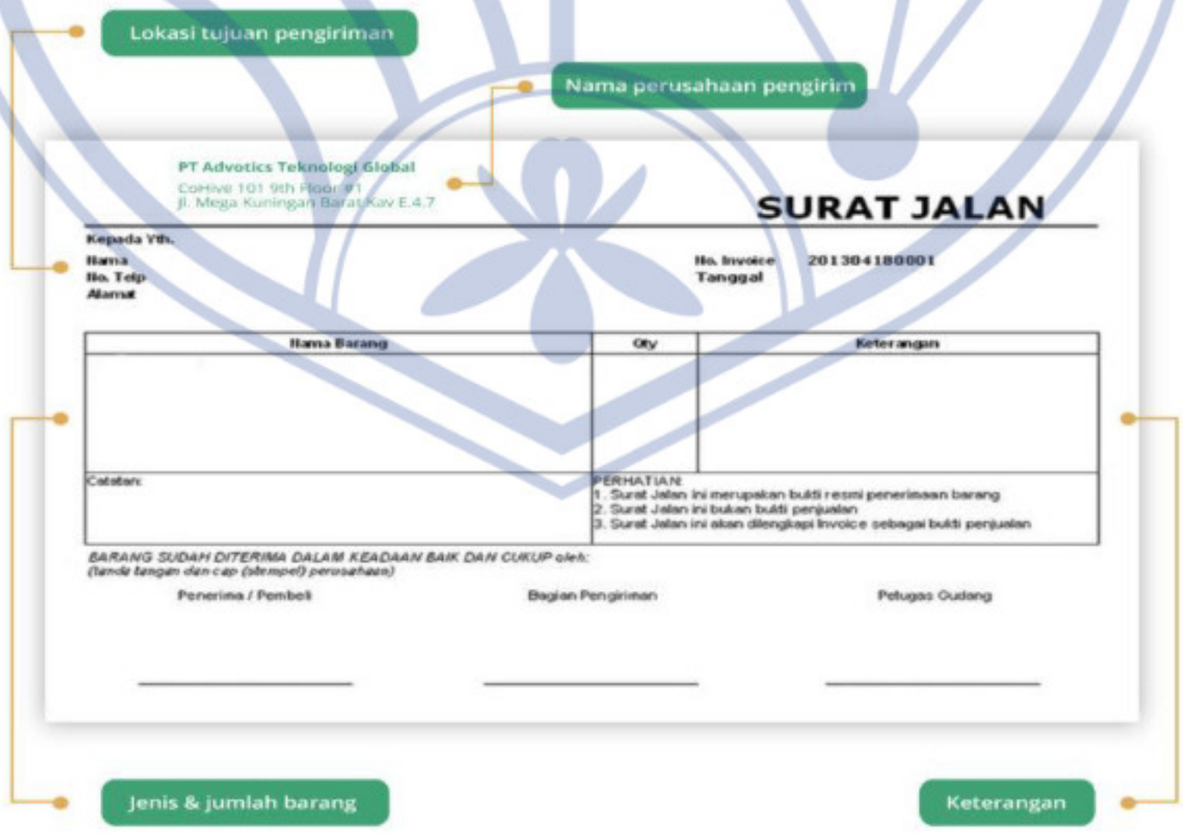
Sistem informasi inventaris yang baik juga memerlukan mekanisme validasi data (misalnya validasi jumlah, satuan, dan lokasi), konsistensi penomoran dokumen, serta pencatatan waktu dan pengguna yang melakukan transaksi. Dengan demikian, data inventaris dapat dipertanggungjawabkan, mudah ditelusuri, dan mendukung pengambilan keputusan seperti perencanaan pengadaan, evaluasi penggunaan barang, dan analisis perputaran stok.

2.4 Surat Jalan dan *Invoice*

Dalam konteks sistem informasi inventaris dan distribusi, dokumen transaksi berperan sebagai dokumen pengendalian yang menjembatani aktivitas operasional (pergerakan fisik barang) dan aktivitas administratif/akuntansi (pencatatan transaksi). Dua dokumen yang umum digunakan dalam proses pengiriman dan penagihan adalah surat jalan dan *invoice*. Keberadaan keduanya membantu organisasi menjaga keterlacakan (*traceability*) aliran barang dan konsistensi data transaksi, karena setiap pengeluaran barang idealnya memiliki bukti pengiriman serta dasar penagihan yang terdokumentasi. Implementasi sistem informasi akuntansi pada proses jasa pengiriman juga menunjukkan pentingnya pencatatan dan pengelolaan dokumen transaksi berbasis sistem agar administrasi lebih tertata dan mudah ditelusuri kembali [20].

Seiring digitalisasi proses bisnis, pengelolaan surat jalan dan *invoice* dalam sistem informasi tidak hanya bertujuan menyimpan dokumen, tetapi juga memastikan validitas data, standarisasi penomoran, pengendalian status proses (misalnya *draft*–disetujui–dikirim–diterima–selesai), serta rekonsiliasi antar dokumen untuk mencegah ketidaksesuaian antara stok, pengiriman, dan tagihan. Dalam konteks sistem informasi akuntansi, pengendalian proses penagihan (*invoice/billing*) juga ditekankan untuk meningkatkan ketepatan dan efisiensi pembayaran melalui prosedur yang lebih terstruktur dan terdokumentasi [21].

2.4.1 Surat Jalan



Gambar 2.2 Contoh Surat Jalan *[22]*

Surat jalan merupakan dokumen yang digunakan sebagai bukti pengiriman barang dari satu pihak ke pihak lain. Dokumen ini berfungsi sebagai alat kontrol distribusi, bukti

operasional bahwa barang telah dikirim, serta dasar verifikasi penerimaan di pihak tujuan. Informasi yang umumnya termuat pada surat jalan meliputi nomor dokumen, tanggal, identitas pengirim dan penerima, alamat tujuan, rincian barang (nama, jumlah, satuan), keterangan pengiriman, serta kolom otorisasi (misalnya tanda tangan petugas/kurir dan penerima). Secara operasional, surat jalan membantu memastikan bahwa kuantitas dan jenis barang yang dikirim sesuai dengan dokumen, sehingga mendukung ketertelusuran (*traceability*) pergerakan barang keluar dari gudang hingga diterima pihak tujuan.

2.4.2 *Invoice*

Gambar 2.3 Contoh *Invoice [23]*

Invoice merupakan dokumen penagihan yang digunakan pada transaksi penjualan atau pembelian barang. *Invoice* memuat rincian transaksi seperti nomor *invoice*, tanggal, identitas pihak terkait, rincian barang/jasa, harga satuan, kuantitas, total nilai, pajak (jika berlaku), serta ketentuan pembayaran. Dalam administrasi perusahaan, *invoice* berperan penting untuk pencatatan akuntansi, kontrol piutang/hutang, dan rekonsiliasi transaksi. Keberadaan *invoice* juga mendukung transparansi transaksi karena seluruh komponen biaya dan kewajiban pembayaran terdokumentasi secara jelas.

2.4.3 Integrasi Surat Jalan dan *Invoice*

Integrasi surat jalan dan *invoice* pada sistem informasi inventaris memberikan manfaat pada konsistensi data transaksi. Surat jalan merepresentasikan pergerakan fisik barang (barang keluar), sedangkan *invoice* merepresentasikan konsekuensi administratif/keuangan dari transaksi tersebut. Digitalisasi dokumen bertujuan mempercepat proses, meningkatkan akurasi, serta memudahkan penelusuran status transaksi. Selain itu, sistem dapat menerapkan pengendalian seperti penomoran otomatis, status dokumen (*draft*,

disetujui, dikirim, diterima, selesai), serta penyimpanan arsip dokumen dalam bentuk digital untuk mendukung audit dan pelaporan. Integrasi ini juga memungkinkan validasi silang, misalnya memastikan *invoice* hanya dapat diterbitkan untuk pengiriman yang sudah tercatat pada surat jalan, sehingga mengurangi risiko inkonsistensi antara data stok dan data transaksi keuangan.

2.5 *Website*

Website merupakan kumpulan halaman digital yang berisi informasi berupa teks, animasi, gambar, suara, video, atau gabungan dari semuanya yang terkoneksi oleh internet [24]. Pengembangan sistem berbasis *website* pada umumnya dibangun dengan teknologi standar web seperti HTML sebagai struktur dokumen, CSS sebagai pengaturan tampilan (*presentation layer*), serta JavaScript sebagai bahasa utama pada sisi klien untuk mengendalikan interaksi pengguna dan dinamika antarmuka. Pada sistem informasi modern, pengembangan *website* tidak hanya berupa halaman statis, tetapi lebih sering berbentuk aplikasi web yang memiliki logika bisnis, manajemen *state*, dan komunikasi data dengan server melalui protokol HTTP/HTTPS menggunakan API (misalnya REST).

Dalam penelitian ini, pengembangan sistem informasi inventaris berbasis *website* menggunakan Angular sebagai *framework* pada sisi klien (*front-end*). Angular merupakan *framework* yang berbasis TypeScript, yaitu *superset* dari JavaScript yang menambahkan fitur seperti *static typing*, *class*, *interface*, dan dukungan paradigma pemrograman berorientasi objek, sehingga meningkatkan keterbacaan kode, konsistensi struktur program, dan memudahkan pemeliharaan aplikasi skala menengah hingga besar [25]. Penggunaan Angular juga mendorong arsitektur aplikasi yang modular melalui konsep *component-based architecture*, di mana antarmuka pengguna dibangun dari komponen-komponen kecil yang dapat digunakan ulang (*reusable*), sehingga pengembangan fitur seperti manajemen barang, transaksi masuk/keluar, serta laporan inventaris menjadi lebih terstruktur.

Selain bahasa dan *framework*, Angular menyediakan mekanisme penting yang mendukung pengembangan sistem informasi, seperti *data binding* (sinkronisasi data antara tampilan dan logika), *dependency injection* (pengelolaan dependensi agar kode lebih teruji dan rapi), *routing* (pengaturan navigasi halaman tanpa memuat ulang seluruh halaman seperti pada *Single Page Application*/SPA), serta integrasi komunikasi data ke server melalui modul seperti *HttpClient* [26]. Untuk pengolahan data dan interaksi pengguna, Angular juga dapat dikombinasikan dengan *library* pendukung (misalnya RxJS) sehingga proses seperti

pencarian cepat, *filtering*, validasi *input*, dan pembaruan tampilan dapat dilakukan secara reaktif dan efisien.

Sistem informasi berbasis *website* memiliki keunggulan utama berupa aksesibilitas dan fleksibilitas, karena pengguna dapat mengakses sistem melalui peramban (*browser*) tanpa instalasi khusus, selama terhubung ke jaringan. Pada implementasi *multiuser*, sistem *website* mendukung penggunaan secara bersamaan, pengaturan hak akses berdasarkan peran (*role-based access control*), serta pembaruan sistem yang terpusat pada server sehingga lebih mudah dipelihara. Dalam konteks sistem inventaris, platform *website* sesuai karena kebutuhan akses data inventaris bersifat dinamis dan berkelanjutan, mencakup pencatatan transaksi secara *real-time*, pencarian data cepat, dan penyajian laporan yang dapat diunduh sesuai kebutuhan operasional.

Agar sistem berbasis Angular berjalan optimal dalam konteks sistem informasi inventaris, beberapa aspek teknis perlu diperhatikan. Pertama, keamanan melalui autentikasi dan otorisasi (misalnya *token-based authentication*), pembatasan akses berdasarkan peran, serta perlindungan dari serangan umum web seperti XSS dengan sanitasi *input* pada sisi tampilan. Kedua, konsistensi dan integritas data melalui validasi *input* di sisi klien dan tetap diperkuat di sisi server, sehingga mengurangi data ganda atau tidak valid. Ketiga, kinerja melalui optimasi pemanggilan API, *pagination* pada tabel data, serta strategi *caching* bila diperlukan untuk mengurangi beban server dan mempercepat respon aplikasi. Keempat, desain antarmuka responsif, sehingga sistem dapat diakses dari berbagai perangkat (*desktop* maupun *mobile*) tanpa mengurangi keterbacaan dan kemudahan penggunaan.

2.6 *Unified Modeling Language* (UML)

Unified Modeling Language (UML) merupakan bahasa pemodelan visual berstandar internasional yang digunakan dalam analisis dan perancangan sistem perangkat lunak secara terstruktur. UML menyediakan himpunan notasi diagram yang formal untuk memvisualisasikan berbagai aspek sistem, termasuk struktur statis dan perilaku dinamisnya, sehingga memfasilitasi komunikasi yang efektif antara analis, perancang, dan pemangku kepentingan lainnya dalam siklus hidup pengembangan perangkat lunak. Studi empiris pada pengembangan sistem menunjukkan bahwa penggunaan UML dapat meningkatkan kejelasan dalam pemodelan kebutuhan pengguna, mendukung dokumentasi desain sistem yang komprehensif, serta memperkuat keterpaduan antara spesifikasi konseptual dan perancangan teknis implementasi sistem [27].

Dalam konteks penelitian dan pengembangan sistem informasi, pemanfaatan UML terbukti memberikan kontribusi signifikan dalam memperjelas hubungan fungsional, alur proses, serta interaksi antar komponen sistem sebelum fase implementasi kode dilakukan. Dengan menerapkan diagram-diagram UML seperti *Use Case Diagram*, *Activity Diagram*, dan diagram lain yang relevan, perancang sistem dapat menurunkan risiko miskomunikasi, meminimalkan kesalahan desain, serta mempermudah proses validasi kebutuhan fungsional dengan pihak pengguna atau klien. Oleh karena itu, UML tidak hanya berfungsi sebagai alat bantu teknis, tetapi juga sebagai instrumen ilmiah yang memperkuat metodologi rekayasa perangkat lunak secara keseluruhan [27].

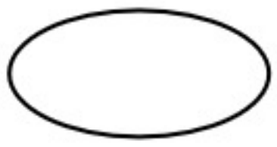
2.6.1 *Use Case Diagram*

Use Case Diagram merupakan salah satu diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk memodelkan kebutuhan fungsional sistem berdasarkan interaksi antara aktor dan sistem yang dikembangkan. Diagram ini merepresentasikan peran aktor sebagai entitas eksternal yang berinteraksi dengan sistem serta fungsi atau layanan yang harus disediakan untuk memenuhi kebutuhan tersebut. Dalam tahap analisis sistem, *Use Case Diagram* berfungsi untuk mendefinisikan batas sistem (*system boundary*), mengidentifikasi skenario penggunaan, serta memetakan hubungan antar fungsi melalui relasi seperti *include* dan *extend*. Diagram ini bersifat konseptual dan tidak menjelaskan mekanisme internal maupun urutan proses secara teknis, melainkan menekankan pada abstraksi kebutuhan pengguna terhadap sistem. Oleh karena itu, *Use Case Diagram* menjadi dasar dalam penyusunan spesifikasi kebutuhan perangkat lunak dan menjadi acuan pada tahap perancangan sistem berikutnya [27].

Tabel 2.1 Simbol *Use Case Diagram*

No	Simbol	Nama	Keterangan
1		*Actor*	Merepresentasikan entitas eksternal yang berinteraksi dengan sistem. *Actor* dapat berupa pengguna (*user*), sistem lain, atau perangkat eksternal yang berhubungan langsung dengan sistem.
2		*Generalization*	Mendefinisikan relasi pewarisan (*inheritance*) antara dua actor atau dua *use case*, di mana salah satu elemen merupakan bentuk spesialisasi dari elemen lainnya.

3	**<< include >>** **- - - - - - - - ->**	*Include*	Menunjukkan bahwa suatu *use case* secara wajib menyertakan *use case* lain sebagai bagian dari alur prosesnya. Artinya, setiap kali *use case* utama dijalankan, maka *use case* yang di-*include* akan selalu ikut dieksekusi tanpa pengecualian. Relasi ini digunakan untuk merepresentasikan fungsi umum yang sering digunakan oleh beberapa *use case*, sehingga mendukung prinsip modularitas serta menghindari pengulangan spesifikasi kebutuhan yang sama dalam sistem.
4	**<< extend >>** **- - - - - - - - ->**	*Extend*	Menunjukkan bahwa suatu *use case* dapat memperluas atau menambahkan perilaku terhadap *use case* lain dalam kondisi tertentu. *Use case* utama tetap dapat dijalankan secara mandiri tanpa harus menjalankan *use case* ekstensi, karena relasi ini bersifat opsional dan bergantung pada terpenuhinya kondisi tertentu. Dengan demikian, <<*extend*>> digunakan untuk memodelkan fitur tambahan atau variasi perilaku yang tidak selalu terjadi dalam setiap eksekusi proses utama.
5	———	*Association*	Menunjukkan hubungan interaksi antara actor dengan *use case* tertentu. Garis ini mengindikasikan keterlibatan *actor* dalam menjalankan fungsi sistem tersebut.
6		*System*	Menyatakan batasan sistem dalam relasinya dengan aktor-aktor yang berada di luar sistem serta fitur-fitur yang disediakan di dalam sistem. Notasi ini menunjukkan ruang lingkup sistem yang dimodelkan.

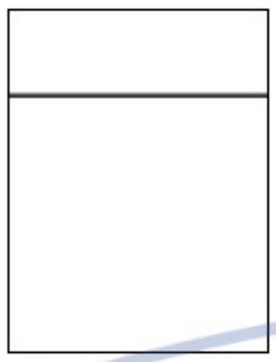
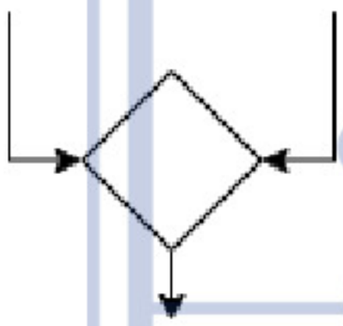
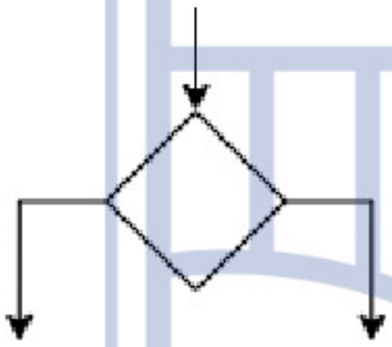
7		*Use Case*	Menggambarkan fungsi atau layanan yang disediakan oleh sistem. *Use case* merepresentasikan kebutuhan fungsional yang diharapkan pengguna terhadap sistem yang dibangun.

2.6.2 *Activity Diagram*

Activity Diagram merupakan salah satu diagram perilaku dalam *Unified Modeling Language* (UML) yang digunakan untuk memodelkan alur aktivitas atau proses kerja sistem secara berurutan dan logis. Diagram ini menggambarkan urutan tindakan (*activity*), percabangan keputusan (*decision*), proses paralel (*fork* dan *join*), serta kondisi awal dan akhir suatu proses dalam sistem. Berbeda dengan *Use Case Diagram* yang berfokus pada kebutuhan fungsional dari perspektif aktor, *Activity Diagram* menekankan pada representasi alur kontrol (*control flow*) yang menunjukkan bagaimana suatu proses dijalankan dari awal hingga selesai. Dalam pengembangan sistem, *Activity Diagram* digunakan untuk memvisualisasikan *workflow* atau proses bisnis sehingga mempermudah analisis terhadap logika proses, identifikasi kemungkinan kondisi alternatif, serta penyusunan rancangan sistem yang lebih sistematis sebelum tahap implementasi dilakukan [27].

Tabel 2.2 Simbol *Activity Diagram*

No	Gambar	Nama	Keterangan
1		*Activity*	Merepresentasikan proses atau rangkaian kegiatan utama dalam sistem dan digambarkan dengan persegi panjang bersudut membulat.
2		*Action*	*Action* adalah langkah kerja spesifik yang bersifat *atomic* dan menunjukkan eksekusi suatu tugas dalam alur aktivitas.
3		*Initial Node*	*Initial Node* merupakan titik awal proses yang digambarkan dengan lingkaran hitam penuh dan menandai dimulainya *workflow.*
4		*Activity Final Node*	*Activity Final Node* adalah simbol akhir proses yang digambarkan dengan lingkaran hitam berlapis dan menunjukkan bahwa seluruh aktivitas telah selesai.

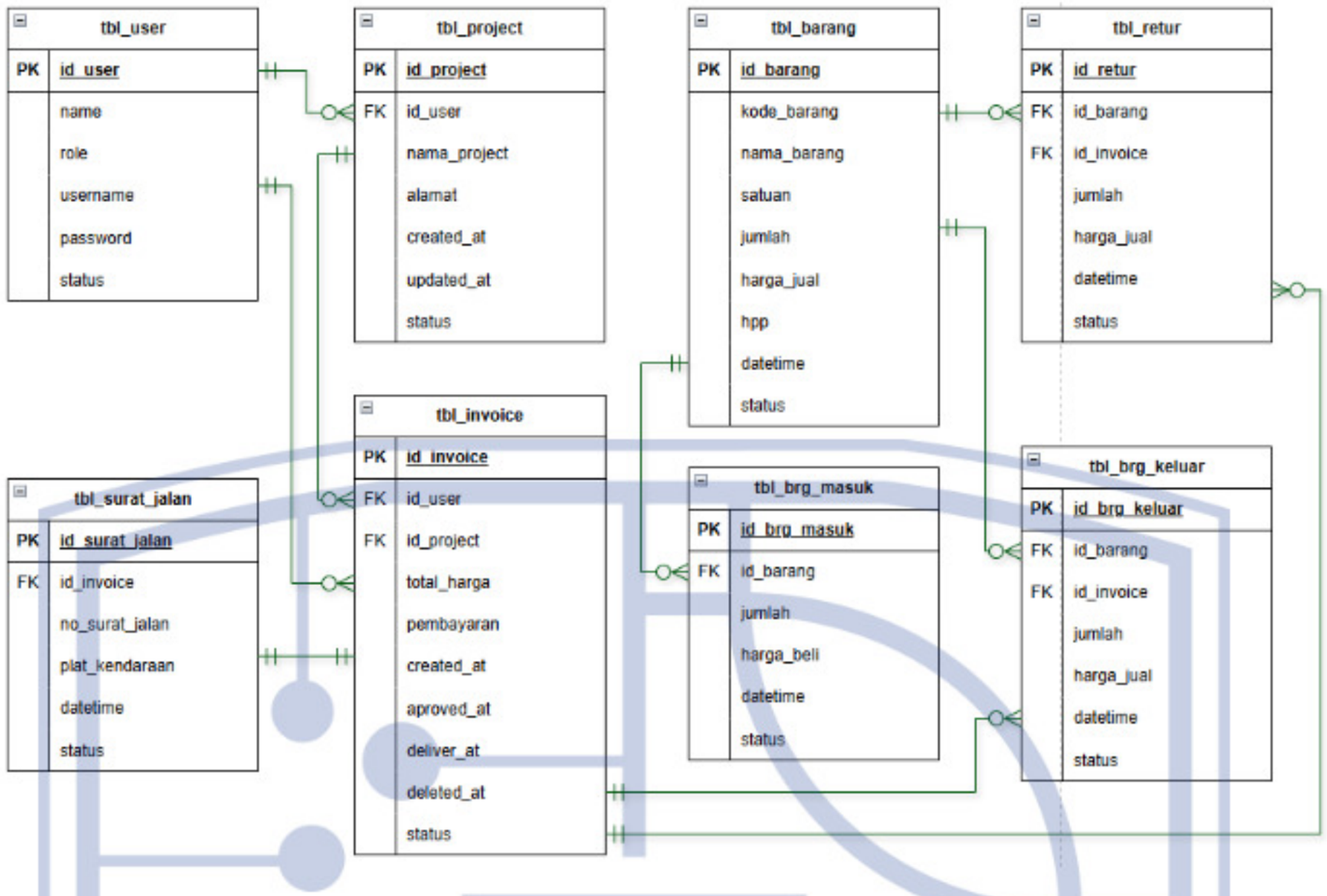
5		*Fork Node*	*Fork Node* berfungsi membagi satu aliran menjadi beberapa aliran paralel yang berjalan secara bersamaan, biasanya digambarkan dengan garis tebal horizontal atau vertikal.
6		*Swimlanes*	*Swimlanes* digunakan untuk memisahkan tanggung jawab berdasarkan aktor atau unit organisasi, sehingga memperjelas siapa yang menjalankan setiap aktivitas dalam diagram.
7		*Decision Node*	*Decision Node* adalah simbol berbentuk belah ketupat yang digunakan untuk membuat percabangan berdasarkan kondisi tertentu (seperti *if–else*). *Node* ini memiliki satu aliran masuk dan beberapa aliran keluar, di mana hanya satu jalur yang dipilih sesuai dengan kondisi yang terpenuhi.
8		*Merge Node*	*Merge Node* adalah simbol berbentuk belah ketupat yang berfungsi menggabungkan kembali beberapa alur percabangan menjadi satu alur. *Node* ini memiliki beberapa aliran masuk dan satu aliran keluar, dan digunakan setelah proses percabangan selesai agar alur dapat berlanjut.

2.7 Basis Data dan PostgreSQL

Basis data (*database*) adalah kumpulan data yang saling berhubungan dan disimpan secara terorganisir pada media penyimpanan tertentu sehingga dapat diakses, dikelola, dan diperbarui secara efisien untuk mendukung kebutuhan informasi [28]. Dalam sistem informasi, basis data berperan sebagai pusat penyimpanan (*central repository*) yang menjaga konsistensi, integritas, dan ketersediaan data, terutama pada sistem yang melibatkan banyak pengguna dan transaksi yang berkelanjutan seperti sistem informasi inventaris.

Penggunaan basis data memungkinkan penyimpanan data secara terpusat sehingga dapat mengurangi redundansi, menjaga konsistensi antar entitas data, serta meningkatkan integritas dan keamanan informasi. Pada sistem inventaris barang, basis data umumnya menyimpan data *master* (misalnya barang, kategori, satuan, lokasi, pengguna, dan hak akses) serta data transaksi (misalnya barang masuk, barang keluar, mutasi, surat jalan, dan *invoice*). Pada pendekatan basis data relasional, struktur data direpresentasikan dalam bentuk tabel

yang saling terhubung melalui relasi menggunakan kunci utama (*primary key*) dan kunci tamu (*foreign key*), sehingga keterkaitan antar data dapat dijaga secara sistematis [29].



Gambar 2.4 *Database*

Agar kualitas data tetap terpelihara, perancangan basis data perlu memperhatikan normalisasi (misalnya hingga bentuk normal ketiga/3NF) untuk meminimalkan duplikasi dan anomali data, serta penerapan aturan integritas (*constraints*) seperti NOT NULL, UNIQUE, dan FOREIGN KEY untuk mencegah inkonsistensi akibat *input* yang tidak valid. Selain itu, mekanisme *backup* dan *recovery* penting disiapkan untuk menjamin ketersediaan (*availability*) dan keberlanjutan operasional ketika terjadi kegagalan sistem atau kehilangan data. Dari sisi keamanan, pengaturan hak akses pengguna pada level aplikasi maupun *database* diperlukan agar kerahasiaan data tetap terjaga sesuai peran dan kebutuhan sistem.

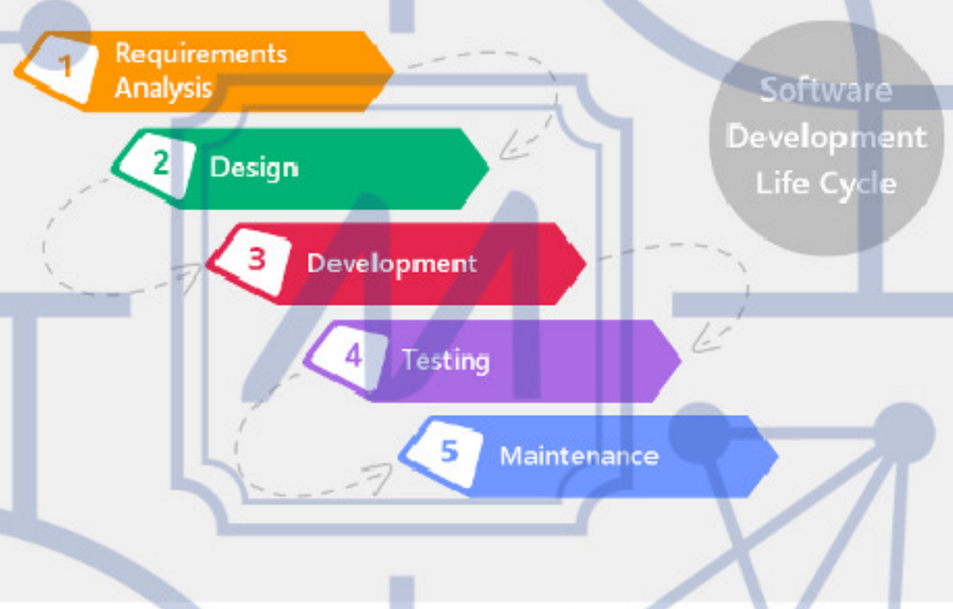
Dalam penelitian ini, pengelolaan basis data dilakukan menggunakan PostgreSQL, yaitu sistem manajemen basis data relasional (*Relational Database Management System*/RDBMS) yang bersifat *open source* dan banyak digunakan dalam pengembangan sistem informasi berbasis web. PostgreSQL menggunakan *Structured Query Language* (SQL) untuk operasi pengelolaan data seperti INSERT, UPDATE, DELETE, dan SELECT. PostgreSQL mendukung penggunaan *multiuser* serta mampu menangani data dalam jumlah besar sehingga sesuai untuk kebutuhan sistem informasi yang memerlukan penyimpanan terstruktur dan akses data yang konsisten [30]. Pada implementasi sistem inventaris, PostgreSQL dapat mengelola data *master* dan transaksi secara konsisten melalui dukungan *indexing* untuk mempercepat pencarian, constraints untuk menjaga integritas relasi antar *table*, serta fitur *transaction control* yang mendukung konsistensi data selama proses pengolahan berlangsung. Dengan perancangan skema yang baik dan optimasi *query* yang

tepat, PostgreSQL dapat memenuhi kebutuhan ketelitian, kecepatan akses, serta ketersediaan data dalam penyajian laporan inventaris.

2.8 Metode *Waterfall*

Metode *Waterfall* merupakan metode pengembangan sistem yang dilakukan secara bertahap dan berurutan. Setiap tahap dalam metode ini harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahap berikutnya. Perkembangan model ini berkembang secara sistematis dari satu tahap ke tahap lain dalam mode seperti air terjun [31].

Model ini menekankan dokumentasi pada setiap tahapan sehingga proses pengembangan lebih terstruktur, mudah dikendalikan, dan hasilnya dapat dievaluasi berdasarkan keluaran (*deliverable*) tiap tahap.



Gambar 2.5 Metode *Waterfall [32]*

Tahapan *Waterfall* pada pengembangan sistem umumnya meliputi:

1. Analisis kebutuhan sistem: mengidentifikasi kebutuhan fungsional dan nonfungsional, ruang lingkup, aktor/pengguna, serta spesifikasi proses bisnis. Keluaran tahap ini berupa dokumen kebutuhan (*requirement specification*).
2. Perancangan sistem: menyusun desain arsitektur, perancangan basis data (ERD, relasi tabel), perancangan antarmuka, serta rancangan alur proses (misalnya diagram UML).
3. Implementasi sistem: menerjemahkan rancangan ke dalam kode program, membangun modul-modul sistem, serta mengintegrasikan komponen aplikasi dengan *database*.
4. Pengujian sistem: melakukan pengujian untuk memastikan sistem berjalan sesuai kebutuhan, mencakup pengujian fungsional (misalnya *black-box*), validasi *input*, serta pengujian integrasi antar modul.

5. Pemeliharaan sistem: perbaikan *bug*, penyesuaian minor, dan peningkatan sistem setelah diterapkan agar sistem tetap andal dan relevan dengan kebutuhan operasional.

Metode *Waterfall* sesuai digunakan ketika kebutuhan sistem relatif jelas, ruang lingkup terdefinisi, serta perubahan kebutuhan tidak sering terjadi. Dalam penelitian ini, *Waterfall* dipilih untuk memastikan proses pengembangan sistem informasi inventaris berjalan sistematis dan terdokumentasi dengan baik, sehingga memudahkan penelusuran proses, evaluasi hasil, dan penyusunan laporan penelitian secara terstruktur.

2.9 *Black Box Testing*

Black box testing merupakan metode pengujian perangkat lunak yang dilakukan dengan menguji fungsi sistem berdasarkan masukan (*input*) dan keluaran (*output*) tanpa melihat struktur kode program di dalamnya. Metode ini digunakan untuk mengetahui apakah sistem telah berjalan sesuai dengan kebutuhan pengguna dan spesifikasi yang telah dirancang. Dalam *black box testing*, penguji hanya berfokus pada fungsi aplikasi, sehingga kesalahan yang dapat ditemukan meliputi kesalahan fungsi, kesalahan antarmuka, kesalahan akses data, serta ketidaksesuaian proses *input* dan *output*. Pengujian ini banyak digunakan pada sistem informasi karena lebih mudah diterapkan dan mampu memastikan bahwa setiap fitur pada aplikasi dapat berjalan sesuai dengan kebutuhan pengguna [33].

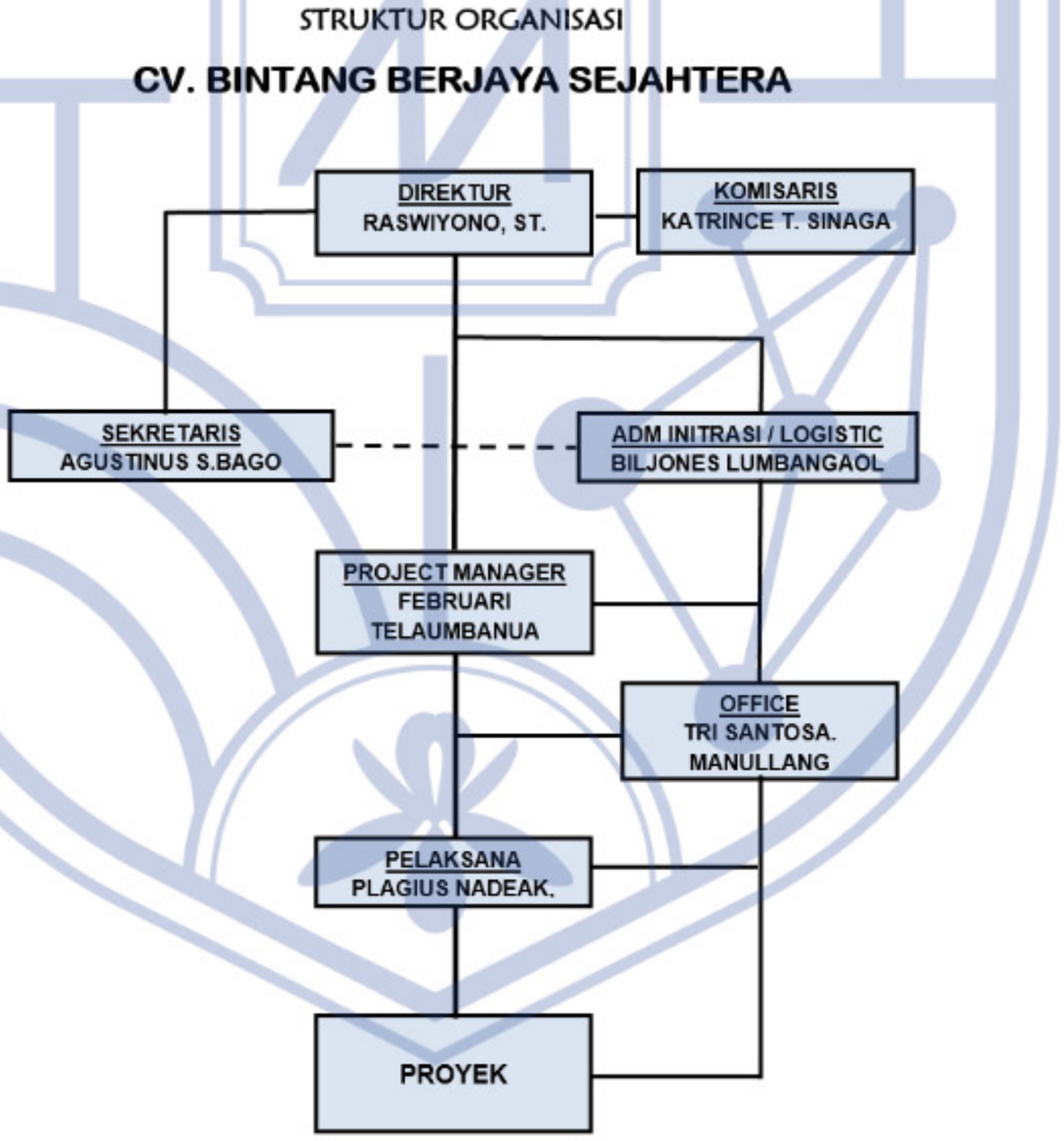
Secara umum, tahapan dalam *black box testing* dimulai dengan menentukan fungsi atau modul sistem yang akan diuji, kemudian menyusun skenario pengujian berdasarkan kebutuhan sistem. Setelah itu dilakukan proses pengujian dengan memberikan data masukan tertentu pada sistem untuk melihat apakah keluaran yang dihasilkan telah sesuai dengan hasil yang diharapkan. Tahap terakhir adalah melakukan evaluasi hasil pengujian untuk mengetahui apakah fungsi sistem dinyatakan valid atau masih terdapat kesalahan yang perlu diperbaiki. Dengan tahapan tersebut, *black box testing* dapat membantu pengembang dalam memastikan kualitas dan keandalan perangkat lunak sebelum sistem digunakan oleh pengguna [33] .

2.10 CV. Bintang Berjaya Sejahtera

CV. Bintang Berjaya Sejahtera (CV BBS) merupakan perusahaan yang bergerak pada bidang jasa konstruksi dan tercatat beroperasi di Kota Medan, Indonesia. Berdasarkan data profil perusahaan pada Indokontraktor, CV BBS terdaftar sebagai badan usaha kontraktor dengan cakupan layanan pada sektor konstruksi bangunan (konstruksi gedung hunian). Dalam pelaksanaan usahanya, perusahaan ini menjalankan aktivitas yang berkaitan

dengan perencanaan dan pelaksanaan pekerjaan konstruksi, yang pada umumnya mencakup pengelolaan kebutuhan proyek, koordinasi sumber daya, serta pemenuhan material dan layanan sesuai kebutuhan pekerjaan di lapangan.

Secara lokasi operasional, alamat kantor CV BBS berada di Jl. Matahari Raya No. 84A, Helvetia, Medan yang berfungsi sebagai pusat administrasi dan koordinasi. Berdasarkan pemetaan titik lokasi (pin) pada Google Maps, kantor tersebut berada pada koordinat geografis (3.61019, 98.63800). Selain itu, perusahaan memiliki gudang yang juga berfungsi sebagai panglong (depo/toko material bangunan) yang beralamat di Jl. Ghermenia Mesjid, Dusun IX Pasar V. Berdasarkan pemetaan titik lokasi (pin) pada Google Maps, panglong tersebut berada pada koordinat geografis (3.62330, 98.61111). Dalam konteks ini, "gudang/panglong" dimaknai sebagai lokasi penyimpanan sekaligus penyediaan material bangunan, sehingga aktivitas di tempat tersebut tidak hanya mencakup penyimpanan persediaan, tetapi juga penyiapan dan penyaluran material untuk kebutuhan proyek maupun permintaan lainnya.



Gambar 2.6 Struktur Organisasi

Berdasarkan uraian umum mengenai CV. Bintang Berjaya Sejahtera tersebut, untuk memperjelas pembagian peran, jalur koordinasi, serta tanggung jawab pada setiap bagian dalam menjalankan aktivitas operasional perusahaan, maka pada bagian berikut akan disajikan diagram struktur organisasi CV. Bintang Berjaya Sejahtera sebagai gambaran hubungan kerja dan alur komando di lingkungan perusahaan.