

BAB II

KAJIAN LITERATUR

2.1 Aplikasi

Aplikasi adalah perangkat lunak yang dirancang untuk menjalankan fungsi tertentu sesuai kebutuhan pengguna. Istilah aplikasi berasal dari kata *application* yang berarti penerapan atau penggunaan [7]. Dalam konteks teknologi informasi, aplikasi merupakan program siap pakai yang dibuat untuk mempermudah pekerjaan manusia, baik dalam bentuk *desktop*, *web*, maupun *mobile*. Aplikasi berperan sebagai antarmuka antara pengguna dan sistem, sehingga memungkinkan pengolahan data dan komunikasi [8].

Aplikasi juga dapat diartikan sebagai penerapan atau penggunaan suatu konsep, data, permasalahan, maupun pekerjaan ke dalam media atau sarana tertentu yang berfungsi untuk menghasilkan bentuk baru yang lebih terstruktur. Secara umum, aplikasi merupakan perangkat lunak yang dirancang dan dioperasikan secara khusus serta terpadu sesuai dengan fungsinya, sehingga dapat langsung digunakan oleh pengguna [9].

2.2 Website

Website atau situs web adalah sebuah aplikasi yang diakses melalui internet atau jaringan komputer dengan menggunakan protokol HTTP (*HyperText Transfer Protocol*) atau HTTPS (*HyperText Transfer Protocol Secure*) [10]. *Website* merupakan kumpulan halaman web yang saling terhubung dan dapat diakses melalui *web browser*. Setiap halaman web pada *website* terdiri dari dokumen-dokumen digital yang berisi teks, gambar, audio dan video [11].

Sistem berbasis web merupakan solusi modern untuk meningkatkan efisiensi dalam manajemen data dan penyebaran informasi. *Website* memiliki keunggulan dalam memberikan akses yang fleksibel kepada pengguna kapan saja dan dari mana saja, sehingga membuatnya menjadi pilihan utama dalam pengembangan sistem informasi di berbagai industri [12]. *Website* terdiri dari dua bagian utama, *frontend* yaitu (bagian antarmuka pengguna) dan *backend* (bagian *server* yang mengelola logika dan *database*). Sistem informasi berbasis web mengintegrasikan berbagai teknologi untuk menciptakan solusi digital yang mengelola data dan informasi secara efisien [13], [14].

Jenis-jenis *website* dapat diklasifikasikan berdasarkan beberapa aspek, yaitu sifat, tujuan penggunaan, serta bahasa pemrogramannya. Berdasarkan sifatnya, *website* dibagi

menjadi dua kategori, yaitu *website dinamis* dan *website statis*. *Website dinamis* memiliki konten yang diperbarui secara berkala, seperti pada *platform* toko *online* atau layanan *internet banking*. Sebaliknya, *website statis* jarang mengalami perubahan konten dan umumnya digunakan untuk halaman informasi sederhana seperti *landing page*. Contoh *website* berdasarkan tujuan penggunaan atau pembuatannya yaitu *personal website* yang digunakan untuk menampilkan informasi individu, *corporate website* yang dimiliki oleh perusahaan, *portal website* yang menyediakan beragam layanan seperti berita dan email, serta *website media sharing* yang memungkinkan pengguna saling berbagi konten seperti gambar atau video dan musik. Berdasarkan pemrogramannya, *website* dapat dibedakan menjadi dua jenis. Pertama, *website server-side*, yaitu *website* yang memerlukan pemrosesan di sisi *server* dan biasanya dikembangkan menggunakan bahasa pemrograman seperti PHP atau ASP. Kedua, *website client-side*, yaitu *website* yang dapat berjalan langsung melalui *browser* tanpa memerlukan *server* untuk proses eksekusinya [15].

Penggunaan *website* dalam pengembangan aplikasi manajemen gudang memberikan beberapa keuntungan:

- a. Aksesibilitas tinggi, dapat diakses dari berbagai perangkat tanpa instalasi.
- b. Kolaborasi, memungkinkan banyak pengguna atau divisi berbeda bekerja secara bersamaan.
- c. Skalabilitas, mudah dikembangkan sesuai kebutuhan bisnis.

2.3 Manajemen Gudang

Manajemen gudang merupakan bidang yang berfokus pada pengelolaan aktivitas penyimpanan serta distribusi barang di dalam gudang. Proses ini mencakup pengaturan dan pengawasan terhadap arus masuk dan keluarnya barang, sehingga ketersediaan dan penataan stok dapat terjaga dengan baik [16]. Perusahaan yang menerapkan sistem manajemen gudang yang baik akan lebih efisien dalam pengelolaan persediaan serta mampu menunjang kelancaran proses distribusi dalam lingkungan bisnis yang kompetitif [17]. Manajemen gudang yang baik menuntut akurasi pencatatan barang masuk dan keluar serta pemeliharaan stok yang tepat. Ketidaksesuaian data antara sistem dan kondisi fisik di gudang berpotensi menimbulkan masalah seperti keterlambatan pengiriman dan kehilangan barang yang dapat merugikan perusahaan [18].

Penerapan sistem informasi yang terstruktur dan sistematis sangat diperlukan untuk memantau arus keluar masuk barang serta kapasitas produksi. Informasi yang dihasilkan dari sistem tersebut kemudian dapat dimanfaatkan dengan mudah oleh seluruh pihak yang

membutuhkan data terkait manajemen pergudangan. Selama ini, perusahaan masih mengelola pencatatan dan pengolahan data menggunakan cara yang sederhana, yaitu dengan aplikasi seperti Microsoft Excel dan Microsoft Word dalam kegiatan operasional sehari-hari. Pengelolaan data produksi yang mencakup puluhan jenis barang dengan kapasitas produksi hingga ratusan ribu unit menimbulkan beban kerja yang tinggi. Kebutuhan akan laporan dan penyajian data yang akurat serta *real-time* sangat penting, namun sulit dipenuhi dengan sistem yang digunakan saat ini [19]. Manajemen gudang yang efektif merupakan salah satu aset utama perusahaan. Gudang berfungsi sebagai divisi kunci yang mendukung perencanaan pengendalian persediaan barang secara optimal. Peran ini sangat penting untuk menjamin kelancaran operasional [20].

Secara umum, manajemen pergudangan terdiri dari serangkaian aktivitas yang saling terintegrasi untuk mendukung kelancaran arus barang. Aktivitas manajemen gudang terdiri dari [21]:

a. Kegiatan Administrasi

Kegiatan ini terdiri dari pengeluaran dan pemasukan dana yang ada dalam perusahaan yang diatur dalam aktivitas administrasi. Semua kegiatan pendanaan diupayakan tercatat seluruhnya untuk memudahkan pelaksanaan pengembangan produksi perusahaan kaitannya dengan manajemen gudang.

b. Penerimaan barang

Aktivitas penerimaan barang diatur sepenuhnya untuk bisa memetakan besaran dana dan produk yang didapatkan oleh perusahaan. Kegiatan ini terdiri dari penurunan barang dari kendaraan pengiriman, pemeriksaan material sesuai dengan daftar pengiriman (*packing list*), serta pemeriksaan kualitas untuk memutuskan apakah barang tersebut dapat diterima atau disimpan di dalam gudang.

c. Penyimpanan barang

Kegiatan ini bertujuan menjamin keamanan seluruh aset dari berbagai risiko di lingkungan gedung, sekaligus berfungsi sebagai fasilitas strategis dalam sistem logistik perusahaan.

d. Pengemasan barang

Pengemasan barang ini dilakukan untuk memastikan kondisi barang yang dikirimkan ke konsumen dalam kondisi yang baik.

e. Pengeluaran barang

Pengeluaran barang dilakukan melalui pencatatan yang sistematis untuk menjamin alur produksi yang tertib dan efisien. Melalui pengendalian operasional, biaya, dan personalia,

aktivitas ini mendukung sistem *supply chain management* dalam mengoptimalkan distribusi, sehingga perusahaan memperoleh kepastian data yang akurat pada setiap arus barang yang keluar dari gudang.

Adapun dalam manajemen pergudangan, terdapat prosedur penanganan barang yang tidak sesuai melalui mekanisme retur sebagai berikut:

1. Retur pembelian

Retur pembelian adalah proses pengembalian produk yang telah dibeli kepada pihak pemasok akibat kerusakan atau ketidaksesuaian spesifikasi produk dengan pesanan awal [22].

2. Retur penjualan

Retur penjualan adalah proses penerimaan kembali produk dari pihak pelanggan yang umumnya terjadi akibat kecacatan fisik selama masa pengiriman atau ketidaksesuaian spesifikasi produk dengan pesanan awal [23].

2.4 Artificial Intelligence

Artificial Intelligence (AI) merupakan bidang ilmu komputer yang mempelajari bagaimana cara komputer memiliki kemampuan kognitif yang terkait dengan kecerdasan manusia seperti kemampuan mengenali pola, kemampuan belajar, dan kemampuan dalam mengambil keputusan [24]. Alih-alih hanya mengandalkan algoritma berbasis aturan yang kaku, perkembangan AI terkini lebih menekankan pada sistem adaptif yang mampu belajar dan berkembang seiring dengan bertambahnya informasi.

Dalam manajemen bisnis, AI berfungsi sebagai lapisan inteligensi yang memungkinkan sistem beroperasi secara otonom untuk mencapai efisiensi dan produktivitas yang optimal. Melalui kemampuan tersebut, implementasi AI tidak hanya meningkatkan kualitas layanan dan produk, tetapi juga mampu memberikan keunggulan strategis dalam memprediksi tren serta peluang pasar [25]. Pergeseran paradigma dari sistem berbasis aturan (*rule-based*) menuju sistem adaptif ini mendasari perkembangan AI modern yang berorientasi pada data (*data-driven*). Pada pendekatan ini, fokus utama pengembangan adalah menciptakan sistem yang mampu mempelajari representasi data secara mandiri untuk menghasilkan keputusan atau prediksi yang akurat dalam berbagai konteks yang dinamis.

2.5 Machine Learning

Machine Learning adalah bidang penelitian ilmiah yang mempelajari algoritma dan model statistik yang digunakan oleh sistem komputer untuk menyelesaikan tugas-tugas

tertentu dengan memanfaatkan pola dan inferensi [26]. *Machine learning* secara umum dibagi menjadi 4, yaitu [27]:

- a. *Supervised Learning*. Menggunakan *dataset* yang telah memiliki label, di mana label tersebut berfungsi sebagai tag pengenalan bagi setiap data. Contoh implementasi dari metode ini adalah klasifikasi email *spam*.
- b. *Unsupervised Learning*: Menggunakan *dataset* yang tidak memiliki label. Dalam model ini, sistem belajar secara mandiri untuk mengidentifikasi pola atau struktur tertentu guna mengelompokkan data tersebut.
- c. *Semi-Supervised Learning*: Merupakan gabungan antara metode *supervised* dan *unsupervised learning*. Pada model ini, *dataset* yang digunakan untuk pelatihan terdiri dari sebagian data yang memiliki label dan sebagian lainnya tidak.
- d. *Reinforcement Learning*: Model pembelajaran yang menggunakan sistem imbalan dan sanksi (*reward and penalties*). Melalui mekanisme ini, model belajar untuk mengambil tindakan yang dapat memaksimalkan perolehan *reward* serta menghindari *penalties*.

2.6 Deep Learning

Deep learning adalah cabang dari *machine learning* yang menggunakan jaringan saraf tiruan dengan banyak lapisan neuron untuk mengekstraksi fitur yang kompleks dan mewakili data input yang kompleks [28]. Perbedaan utama dari *machine learning* dan *deep learning*, yaitu pada *machine learning*, *feature extraction* perlu dirancang secara manual, yang mana memakan banyak sekali waktu dan tenaga. Sedangkan *deep learning* akan secara otomatis mengekstrak *features* untuk klasifikasi. Hal ini menjadikan *deep learning* sebagai fondasi utama dalam pengembangan sistem cerdas yaitu *Large Language Models* (LLM) yang menjadi otak bagi *Agentic AI*. Secara umum, arsitektur dasar *deep learning* terdiri dari tiga bagian utama [29]:

1. *Input Layer*: Lapisan yang menerima data mentah (misalnya piksel gambar atau numerik).
2. *Hidden Layers*: Serangkaian lapisan di mana proses komputasi matematis terjadi untuk mengenali pola-pola abstrak melalui mekanisme bobot dan bias.
3. *Output Layer*: Lapisan terakhir yang menghasilkan prediksi, klasifikasi, atau keputusan berdasarkan pemrosesan di lapisan sebelumnya.

2.7 Large Language Model

Large Language Model (LLM) adalah sebuah model kecerdasan buatan berbasis *deep learning* yang memiliki kemampuan untuk memahami bahasa alami dan mengenali

pola dari data dengan jumlah yang besar [29]. Berbeda dengan media pembelajaran konvensional, LLM memiliki kemampuan untuk memahami bahasa dan menghasilkan teks yang relevan secara semantik [30]. Keunggulan performa LLM modern secara teknis didorong oleh penggunaan arsitektur *transformer* yang mengimplementasikan mekanisme *self-attention* [31]. Mekanisme ini memungkinkan model untuk melakukan pembobotan terhadap signifikansi setiap elemen dalam urutan data secara simultan, sehingga mampu menangkap dependensi kontekstual jarak yang sering kali luput pada model bahasa tradisional. Dengan jumlah parameter yang mencapai skala miliaran, LLM menunjukkan fenomena kemampuan emergensi, di mana model dapat melakukan penalaran kompleks dan dekomposisi instruksi secara logis.

Kemampuan penalaran kompleks ini dapat diperluas secara sistematis menggunakan metode *Chain of Thought*. Pendekatan ini mendekomposisi proses penalaran model ke dalam langkah-langkah modular, sehingga setiap tahapan keputusan logis dapat diikuti dan dievaluasi alurnya secara runtut [32]. Namun, pada sistem operasional yang menuntut waktu respons cepat, tahapan penalaran eksplisit ini dapat dinonaktifkan. Selain memproses penalaran secara internal, LLM juga dapat berinteraksi langsung dengan sistem eksternal menggunakan mekanisme *Function Calling*. Melalui mekanisme ini, LLM dapat menerjemahkan input bahasa alami pengguna menjadi representasi JSON terstruktur yang menentukan nama fungsi serta parameter spesifik untuk dieksekusi oleh layanan backend [33].

2.8 *Agentic AI*

Agentic AI merupakan paradigma baru dalam kecerdasan buatan yang menekankan pada *agency*, yakni kapasitas untuk melaksanakan tugas secara mandiri dengan pengawasan manusia yang minimal dan berorientasi tujuan dalam lingkungan yang dinamis. Perkembangan paradigma ini difasilitasi oleh kemajuan pesat dalam *machine learning*, *reinforcement learning*, dan *large language models* (LLMs) yang memungkinkan sistem untuk mengintegrasikan fungsi persepsi, perencanaan strategis, dan eksekusi tindakan dalam satu arsitektur yang terintegrasi [34].

Beberapa pilar teknologi yang membentuk *Agentic AI* meliputi:

1. *Reinforcement Learning* (RL), memungkinkan agen belajar melalui interaksi dan umpan balik.
2. *Multi Agent System* (MAS), mendukung koordinasi dan negosiasi antar agen.

3. *Symbolic Reasoning* dan *Automated Planning*, agen menggunakan representasi terstruktur dan metode berbasis logika untuk merencanakan urutan tindakan, memungkinkan pemikiran strategis jangka panjang.
4. *Cognitive Architectures*, model seperti SOAR, ACT-R, dan LIDA menirukan aspek-aspek kognisi manusia seperti memori, persepsi, dan pembelajaran yang membuat agen AI memiliki penalaran dan beradaptasi secara alami.
5. *Natural Language Processing* (NLP), memberdayakan agen untuk memproses bahasa manusia, memahami instruksi, dan melakukan interaksi dalam bentuk percakapan.
6. *Computer Vision* dan *Perception*, kemampuan pemrosesan visual untuk menafsirkan, memahami adegan, dan merespon isyarat lingkungan.
7. *Meta-Learning*, memberdayakan sistem *Agentic AI* untuk beradaptasi dengan cepat terhadap tugas-tugas baru [34].

Dalam implementasinya, sistem *Agentic AI* tidak hanya bersifat reaktif, tetapi mampu bertindak otonom, merencanakan tindakan, berkolaborasi dengan pengguna, serta menyesuaikan perilakunya berdasarkan konteks dan tujuan tertentu [35]. Kapasitas ini memungkinkan sistem untuk mengambil keputusan operasional secara mandiri dan tetap adaptif meskipun menghadapi dinamika pada lingkungan kerja [36]. Lebih lanjut, kemampuan agen untuk beroperasi secara mandiri tanpa campur tangan manusia memberikan keuntungan signifikan dalam memproses data berskala besar secara *real-time*, sehingga memungkinkan sistem untuk beralih dari sekadar alat pemantau pasif menjadi entitas yang bertindak proaktif [37].

Untuk merealisasikan tingkat otonomi operasional tersebut, sistem *Agentic AI* sangat bergantung pada integrasi mekanisme *Function Calling* dan instruksi arsitektural yang ditanamkan melalui *system prompt*. *Function Calling* merupakan kapabilitas fundamental yang memungkinkan *Large Language Model* (LLM) untuk berinteraksi dengan lingkungan perangkat lunak eksternal dengan cara menerjemahkan bahasa alami menjadi format pemanggilan terstruktur yang dapat dieksekusi oleh aplikasi, sehingga agen dapat secara aktif menjalankan perintah spesifik atau menarik data dinamis di luar batasan data latih statistiknya [38]. Di sisi lain, agar agen dapat mengorkestrasi eksekusi tersebut secara akurat, *system prompt* berperan sebagai fondasi kognitif utama yang membatasi dan mengarahkan perilaku LLM. *System prompt* harus dikonstruksi secara komprehensif dengan memuat definisi tugas, deskripsi kemampuan alat (*tools*), skema parameter, serta elemen *chain-of-thought* yang secara presisi memandu logika agen dalam menganalisis

argumen dan memilih fungsi yang tepat, sehingga keandalan sistem dalam meminimalkan halusinasi fungsi di lingkungan industri dapat terjamin [39].

2.9 Metode Scrum

Scrum merupakan salah satu metode dalam pendekatan *Agile Software Development* yang menekankan pada pengembangan iteratif dan kolaboratif. Scrum memiliki sifat *iterative* dan *incremental*, yang memungkinkan pengembang untuk terus meningkatkan produk sesuai kebutuhan pasar dan pengguna. Metode ini bukan sekadar prosedur kaku, melainkan *framework* fleksibel yang dapat diadaptasi pada berbagai jenis proyek perangkat lunak [40].

Dalam penerapannya, Scrum terdiri atas beberapa peran penting, yaitu *Product Owner*, *Scrum Master*, dan *Development Team*. *Product Owner* bertanggung jawab terhadap prioritas dan visi produk serta keberhasilan rancangan produk, *Scrum Master* berperan sebagai fasilitator yang memastikan proses berjalan lancar, dan *Development Team* bertugas menyelesaikan pekerjaan teknis sesuai *backlog* yang telah disepakati [41].

Pada setiap tahapan dalam alur proses metode Scrum terdapat aktivitas kerja yang berada dalam suatu pola proses yang disebut *sprint* [42]. Setiap alur proses tersebut memiliki kegiatan sebagai berikut:

1. *Product Backlog*

Tahap awal melibatkan pengumpulan semua persyaratan dan fitur yang diinginkan pemangku kepentingan serta pengguna, yang disusun dalam *product backlog*.

2. *Sprint Planning*

Tim memilih item dari *product backlog* yang akan dikerjakan pada *sprint* berikutnya. Pemilihan berfokus pada fitur-fitur yang penting dan butuh segera disiapkan.

3. *Sprint Backlog*

Sprint backlog disusun dengan memecah *sprint* berdasarkan kebutuhan yang berasal dari *product backlog*, sehingga setiap *sprint* memiliki tujuan pengembangan fitur yang telah dirumuskan secara jelas.

4. *Sprint*

Sprint adalah periode waktu tetap di mana tim menyelesaikan item-item terpilih. Selama *sprint*, tim melakukan *daily standup* untuk berbagi informasi dan mengatasi hambatan yang muncul.

5. *Sprint Review*

Setelah *sprint* selesai, tim mengadakan *sprint review* untuk mendemonstrasikan hasil pekerjaan kepada pemangku kepentingan dan mengumpulkan umpan balik yang akan digunakan untuk perencanaan *sprint* berikutnya.

6. *Sprint Retrospective*

Tim melakukan *sprint retrospective* untuk melakukan evaluasi proses kerja, mengidentifikasi aspek yang berjalan baik dan yang perlu ditingkatkan, serta merencanakan perbaikan untuk *sprint* mendatang [43].

2.10 *Black Box Testing*

Black Box Testing adalah metode pengujian perangkat lunak yang dilakukan tanpa memeriksa bagian dalam atau struktur kode sistem. Pengujian ini hanya difokuskan pada *input* dan *output* yang dihasilkan dengan tujuan memastikan bahwa sistem beroperasi sesuai dengan spesifikasi fungsional yang telah ditetapkan [44]. Penggunaan metode *black box testing* memiliki keunggulan yaitu penguji tidak dituntut untuk memahami bahasa pemrograman tertentu yang digunakan dalam pengembangan aplikasi. Proses pengujian dilakukan dengan meninjau sistem dari sudut pandang pengguna akhir, sehingga tim pengembang dan penguji dapat berkolaborasi untuk memastikan aplikasi berfungsi dengan baik dan telah memenuhi kebutuhan pengguna [45].

Dalam pengujiannya, *black box testing* memiliki beberapa teknik yang dapat diterapkan, dan salah satu yang paling umum digunakan adalah teknik *Equivalence Partitioning*. Teknik ini merupakan pendekatan pengujian dengan cara memberikan data masukan pada setiap form yang terdapat dalam suatu sistem atau perangkat lunak. *Equivalence partitioning* dilakukan dengan membagi data masukan ke dalam dua kelompok, yaitu *input* yang *valid* dan *input* yang tidak *valid*, sehingga dapat dipastikan bahwa *output* yang dihasilkan telah sesuai dengan skenario atau kasus uji yang dirancang. Penerapan teknik ini membantu mengurangi kemungkinan terjadinya cacat pada perangkat lunak [46].

Untuk memastikan pengujian berjalan secara sistematis, pelaksanaan *black box testing* harus mengikuti tahapan-tahapan yang terstruktur. Tahapan pengujian tersebut terdiri dari [47]:

a. Analisis kebutuhan sistem

Mengidentifikasi seluruh kebutuhan fungsional perangkat lunak berdasarkan spesifikasi modul operasional dan batasan peran dari masing-masing pengguna maupun agen sistem.

b. Penentuan fungsi yang diuji

Menetapkan batasan fitur dan antarmuka yang akan dievaluasi, seperti modul manajemen data utama, pencatatan transaksi masuk dan keluar, serta fungsionalitas eksekusi oleh sistem.

c. Penyusunan data uji (*test data*)

Merancang parameter pengujian dengan menentukan klasifikasi data masukan yang valid maupun tidak valid pada setiap *form* aplikasi, misalnya pengujian menggunakan duplikasi ID pendataan, format jumlah yang tidak sesuai, atau parameter eksekusi yang tidak valid.

d. Pelaksanaan pengujian (*execution*)

Menjalankan aplikasi dan memberikan data masukan sesuai dengan skenario uji yang telah dirancang, kemudian mencatat setiap hasil keluaran yang diberikan oleh sistem.

e. Perbandingan hasil aktual dan hasil yang diharapkan

Mengevaluasi kesesuaian antara respons aktual dari sistem dengan spesifikasi hasil yang diharapkan guna memvalidasi keberhasilan suatu fungsi.

f. Evaluasi hasil pengujian

Menganalisis hasil yang tidak sesuai untuk menemukan potensi kesalahan pada validasi masukan atau logika antarmuka, sehingga dapat dijadikan dasar pelaporan untuk perbaikan perangkat lunak.