

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

Sebagai tahap awal dalam mengembangkan sistem pengelolaan persediaan gudang di Toko Bangunan Duta Sae, kajian terhadap penelitian-penelitian terdahulu yang relevan sangat penting untuk dilakukan. Penelitian sebelumnya memberikan gambaran mengenai berbagai pendekatan yang telah digunakan dalam pengelolaan persediaan barang berbasis web, sekaligus membantu dalam mengidentifikasi keunggulan dan kelemahan dari masing-masing metode. Beberapa penelitian yang relevan untuk kajian ini antara lain adalah penelitian oleh Maulana dkk. (2023), serta penelitian oleh Afrianto dkk., (2020). Penelitian oleh Maulana dkk., (2023) berhasil merancang sistem pengelolaan stok di PT. Alwahijab menggunakan diagram konteks, DFD, ERD, dan struktur *database*. Sistem ini telah diuji menggunakan metode *blackbox* dan terbukti berfungsi dengan baik, meskipun masih diperlukan pengembangan lebih lanjut dalam aspek keamanan dan kinerja. Afrianto dkk., (2020), di sisi lain, mengembangkan sistem inventory berbasis P.O.S yang mencakup perancangan dengan UML serta diagram konteks, juga diuji dengan metode *blackbox*. Kedua penelitian ini memberikan wawasan tentang pengelolaan stok barang berbasis web dalam konteks yang berbeda, yaitu pada industri busana dan ritel, serta menyoroti pentingnya sistem web dalam pemantauan stok *real-time* untuk meminimalkan kesalahan pencatatan barang.

Penelitian serupa juga dilakukan oleh Wahyudin dan Bela (2021) yang merancang sistem inventori berbasis web untuk Agen Kianda dengan fitur yang mencakup tiga hak akses pengguna (Admin, Pramuniaga, dan Kasir) serta didesain untuk meminimalkan kesalahan *input* stok. Penelitian ini memperlihatkan efektivitas penggunaan sistem berbasis web dalam mengelola stok barang secara akurat dan memudahkan pelaporan stok. Selanjutnya, penelitian oleh Hidayat dan Ginting (2019) menyoroti kemudahan monitoring stok ikan, pengelolaan

pelanggan, serta pencatatan transaksi penjualan dan pembelian. Penelitian ini menegaskan pentingnya aksesibilitas data stok secara efisien untuk berbagai jenis usaha, termasuk pada bisnis ikan hias.

Hasil dari penelitian-penelitian di atas menunjukkan bahwa sistem informasi berbasis web mampu meningkatkan efisiensi pengelolaan stok, mengurangi risiko kesalahan manual dalam pencatatan, dan memudahkan pelaporan. Penelitian ini memberikan referensi penting bagi pengembangan sistem pengelolaan stok berbasis web di Toko Bangunan Duta Sae, yang memiliki fokus khusus pada pengelolaan stok gudang tanpa menerapkan sistem pelanggan atau harga produk, seperti yang diterapkan pada studi sebelumnya.

2.2. Kerangka Pemikiran

Sebagai landasan dalam pengembangan sistem, kerangka pemikiran ini bertujuan untuk memberikan pemahaman yang jelas mengenai tahapan-tahapan penting yang harus dilalui. Setiap tahapan dalam kerangka ini saling terkait dan berkontribusi untuk mencapai hasil yang optimal dalam pengelolaan persediaan gudang. Dengan mengikuti langkah-langkah sistematis yang diuraikan, diharapkan sistem yang dikembangkan dapat memenuhi kebutuhan pengguna secara efektif. Selain itu, kerangka pemikiran ini berfungsi sebagai panduan untuk memastikan bahwa sistem pengelolaan persediaan yang dihasilkan mampu memenuhi berbagai aspek kebutuhan pengguna, termasuk kemudahan penggunaan, kecepatan akses, dan ketepatan data. Dengan demikian, kerangka ini tidak hanya mengarahkan proses pengembangan tetapi juga memastikan bahwa hasil akhir dapat diandalkan dalam mendukung operasional pengelolaan persediaan secara keseluruhan. Berikut merupakan gambaran kerangka pemikiran dalam pengembangan sistem yang akan dilakukan, yang dapat dilihat pada Gambar 2.1



Gambar 2. 1 Kerangka Pemikiran

Kerangka pemikiran ini menggambarkan langkah-langkah sistematis dalam mengembangkan dan menerapkan sistem informasi, khususnya untuk pengelolaan persediaan gudang. Berikut adalah penjelasan singkat setiap langkah:

1. Latar Belakang Masalah

Tahap awal penelitian, yaitu menggambarkan isu atau permasalahan utama yang menjadi fokus penelitian. Penjelasan ini mencakup alasan pentingnya

masalah tersebut untuk diteliti, relevansinya, serta urgensinya untuk diselesaikan.

2. Rumusan Masalah

Merangkum latar belakang menjadi suatu pernyataan masalah yang akan dijadikan dasar dan bahan utama dalam penelitian.

3. Metode *Prototyping*

Mengembangkan sistem berbasis web melalui tahapan analisis, desain, prototipe.

4. Pengujian Prototipe Awal

Sistem diuji oleh pemilik untuk memastikan kesesuaian; jika ada umpan balik, kembali ke tahap implementasi perencanaan untuk perbaikan.

5. Implementasi Sistem

Menerapkan sistem berbasis web yang telah disempurnakan dengan fitur pencatatan stok otomatis dan laporan.

6. Pengujian Sistem

Memastikan sistem sesuai kebutuhan melalui pengujian *User Acceptance Testing* (UAT).

7. Sistem Terimplementasi

Menghasilkan sistem siap pakai yang meningkatkan efisiensi dan akurasi pengelolaan stok.

2.3. Sistem Stok

2.3.1. Sistem Manajemen Stok

Menurut Maulana dkk. (2023), sistem manajemen stok adalah alat yang dirancang untuk mengatasi masalah pengelolaan stok secara manual dan meningkatkan akurasi data persediaan. Sistem ini memungkinkan perusahaan untuk memantau stok secara lebih efektif, membantu dalam pengambilan keputusan yang lebih baik terkait ketersediaan dan pengisian ulang barang.

Afrianto dkk. (2020) mengartikan sistem manajemen inventaris sebagai sebuah solusi untuk mengurangi kesalahan manual dalam pengelolaan stok dan meningkatkan efisiensi dalam pelaporan. Sistem ini memungkinkan pelacakan stok secara *real-time* dan menghasilkan laporan stok dengan lebih cepat dan akurat.

Menurut Wahyudin dan Bela (2021), manajemen stok adalah elemen krusial dalam operasional perusahaan yang berfungsi untuk mengelola persediaan barang di gudang secara efektif, terutama barang-barang yang akan dijual kepada konsumen. Dengan memanfaatkan sistem informasi inventori berbasis web, perusahaan dapat melakukan pencatatan dan pengecekan stok secara lebih efisien, mengurangi risiko penumpukan stok yang tidak terkontrol, serta meningkatkan efektivitas operasional.

Menurut Yulianto dkk., (2021) yang mengutip dari Rahmawati, (2017), Sistem inventaris adalah sekumpulan kebijakan dan pengendalian, yang memonitor tingkat inventaris, dan menentukan tingkat mana yang harus dijaga, bila stok harus diisi kembali dan berapa banyak yang harus dipesan.

Menurut Wijaya dkk., (2021) yang mengutip dari Rahmawati, (2017), Pengertian dari stok atau persediaan adalah suatu aktivitas yang meliputi barang-barang milik perusahaan dengan maksud untuk dijual dalam periode usaha yang normal.

Sedangkan menurut Hidayat dan Ginting (2019), sistem manajemen stok adalah alat berbasis web yang dirancang untuk memudahkan pemantauan stok dan pengelolaan inventaris, dengan tujuan mengurangi kesalahan manusia serta meningkatkan aksesibilitas data stok untuk pengambilan keputusan yang lebih efisien.

2.3.2. Metode Perhitungan Stok

Metode *perpetual inventory* adalah sistem pencatatan yang memungkinkan perusahaan untuk terus memantau persediaan barang secara langsung (Ramanda, 2023). Di Toko Bangunan Duta Sae, penerapan metode ini memfasilitasi pencatatan barang yang masuk dan keluar secara terus-menerus, sehingga perusahaan dapat mengawasi jumlah stok yang tersedia dengan akurat. Metode ini juga meminimalkan risiko kesalahan pencatatan yang sering muncul pada metode konvensional (Ramanda, 2023). Rumus perhitungan laporan total stok barang di Gudang Toko Bangunan Duta Sae disajikan pada Persamaan (2.1)

$$\begin{aligned} \text{Total Stok} = & (\text{Stok Awal} + \text{Barang Masuk} - \text{Barang Keluar} \\ & - \text{Barang Retur Pemasok}) + \text{Selisih Stok Opname} \end{aligned} \quad (2.1)$$

Dengan rumus ini, stok akhir menggambarkan jumlah barang yang tersedia setelah mempertimbangkan semua transaksi dalam periode tertentu. Metode ini memberikan perusahaan akses ke informasi persediaan yang selalu mutakhir, sehingga keputusan terkait pengadaan dan manajemen barang dapat diambil dengan lebih tepat dan cepat, yang pada akhirnya meningkatkan efektivitas operasional secara keseluruhan, (Rianita, 2021).

2.4. Dasar Teori

2.4.1. *Unified Modeling Language (UML)*

Dalam tahap perencanaan pengembangan sistem informasi, pemilihan diagram dan notasi yang tepat sangat krusial untuk memastikan pemahaman yang mendalam dan jelas mengenai kebutuhan sistem. Salah satu alat yang efektif dalam proses ini adalah UML (*Unified Modeling Language*). UML memungkinkan pengembang dan pemangku kepentingan untuk memvisualisasikan struktur dan perilaku sistem, yang pada gilirannya meningkatkan komunikasi dan kolaborasi di antara tim pengembang. Dengan penerapan UML, pengembang dapat membuat representasi yang jelas dari sistem yang sedang dirancang, sehingga dapat lebih mudah mendeteksi masalah yang mungkin muncul sejak awal dan mempercepat proses pengembangan.

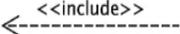
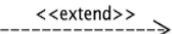
UML (*Unified Modeling Language*) merupakan bahasa pemodelan standar yang digunakan untuk memodelkan, menspesifikasikan, membangun, dan mendokumentasikan artefak dari sistem perangkat lunak. UML menawarkan notasi yang konsisten dan terstruktur, sehingga memungkinkan pengembang untuk merancang, menganalisis, dan mendokumentasikan sistem informasi dengan lebih efektif (Narulita dkk., 2024). Dalam penelitian ini, akan diterapkan tiga dari sembilan jenis diagram UML, yaitu *use case diagram*, *activity diagram*, dan *sequence diagram*.

2.4.2. *Use case Diagram*

Use case diagram merupakan representasi visual yang menggambarkan berbagai komponen, seperti aktor, *use case*, dan hubungan antar komponen tersebut (Hasanah dan Untari, 2020). Berbagai simbol atau notasi digunakan untuk menggambarkan fungsionalitas sistem dalam *use case diagram*. Dengan adanya *use*

case diagram, analisis dapat lebih mudah dalam menyusun kebutuhan (*requirement*) untuk pengembangan sistem. Diagram ini digunakan untuk menjelaskan desain sistem kepada pengguna serta merancang semua fitur yang ada dalam sistem yang akan dikembangkan (Hasanah dan Untari, 2020). Pada Tabel 2.1 terdapat penjelasan mendetail mengenai masing-masing simbol dalam *use case diagram*.

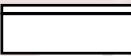
Tabel 2. 1 Simbol *Use case Diagram*

NO	Nama	Simbol	Deskripsi
1	Aktor		Entitas eksternal yang berinteraksi dengan sistem, seperti pengguna atau sistem lain.
2	<i>Use case</i>		Representasi fungsi atau layanan yang disediakan oleh sistem untuk aktor.
3	Asosiasi		Hubungan antara aktor dan <i>use case</i> yang menunjukkan interaksi.
4	Include		Relasi yang menunjukkan bahwa suatu <i>use case</i> mengandung <i>use case</i> lain sebagai bagian wajib.
5	Extend		Relasi yang menunjukkan bahwa <i>use case</i> dapat memperluas fungsi lain secara opsional.

2.4.3. *Activity Diagram*

Activity diagram menggambarkan aliran proses atau aktivitas dalam sistem yang akan dikembangkan, mulai dari tahap awal, keputusan-keputusan yang diambil di dalam sistem, hingga cara suatu proses berakhir (Hasanah dan Untari, 2020). Diagram ini juga memperlihatkan proses-proses paralel yang berlangsung saat sistem dieksekusi. Setiap langkah atau tahapan yang terjadi dalam sistem diilustrasikan dalam diagram ini. Setiap *use case* minimal memiliki satu *activity diagram* yang sesuai. *Activity diagram* dirancang berdasarkan satu atau beberapa *use case* yang terdapat pada *diagram use case*. Sementara itu, *activity diagram* merepresentasikan proses yang berlangsung dalam suatu sistem, *use case diagram* menggambarkan cara aktor berinteraksi dengan sistem untuk melaksanakan aktivitas (Hasanah dan Untari, 2020). Penjelasan terperinci mengenai setiap simbol dalam *activity diagram* dapat dilihat pada Tabel 2.2.

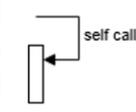
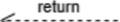
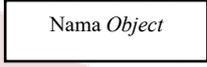
Tabel 2. 2 Simbol *Activity Diagram*

NO	Nama	Simbol	Deskripsi
1	Mulai/ <i>Initial Node</i>		Simbol awal dari alur aktivitas, biasanya ditandai dengan titik hitam kecil.
2	Aktivitas		Menunjukkan pekerjaan atau tindakan yang dilakukan di dalam sistem.
3	Akhir/ <i>Final Node</i>		Simbol akhir dari alur aktivitas, menandakan akhir proses.
4	Keputusan/ <i>Decision Node</i>		Digunakan untuk menunjukkan percabangan alur berdasarkan kondisi tertentu.
5	Penggabungan/ <i>Join Node</i>		Relasi yang menunjukkan bahwa <i>use case</i> dapat memperluas fungsi lain secara opsional.
6	Fork Node		Membagi satu alur menjadi beberapa alur paralel.
7	Lajur <i>(Swimlane)</i>		Membagi aktivitas berdasarkan peran atau departemen untuk mengidentifikasi tanggung jawab.

2.4.4. *Sequence Diagram*

Sequence diagram berfungsi untuk memodelkan interaksi dinamis antar objek dalam sistem, menggambarkan urutan pesan yang dikirim antar objek dan kolaborasi mereka dalam menyelesaikan suatu fungsi atau proses, sehingga membantu pengembang memahami aliran informasi dan kontrol di antara elemen-elemen sistem (Widyastuti dan Luis, 2022). Selain itu, pendapat lain menyatakan bahwa menyatakan bahwa *sequence diagram* merepresentasikan kolaborasi dinamis antar objek, memperlihatkan rangkaian pesan yang dikirim di antara objek dan interaksi yang terjadi dalam sistem yang dibangun (Sinambela dkk., 2024). Penjelasan mendetail mengenai setiap simbol dalam *sequence diagram* disajikan pada Tabel 2.3.

Tabel 2. 3 Simbol *Sequence Diagram*

NO	Nama	Simbol	Deskripsi
1	Garis Hidup (<i>Lifeline</i>)		Garis vertikal yang menunjukkan keberadaan objek selama proses berlangsung.
2	Aktor		Menandakan entitas eksternal yang berinteraksi dengan sistem.
3	<i>Message</i> (Pesan)		Menunjukkan komunikasi atau tindakan yang terjadi antar objek atau komponen, biasanya berupa interaksi dari satu simbol ke simbol lainnya.
4	<i>Self-Message</i>		Panah yang berawal dan berakhir pada objek yang sama, menggambarkan pemanggilan fungsi pada objek itu sendiri.
5	<i>Activation</i> (Garis Aktivasi)		Garis tebal vertikal pada <i>lifeline</i> yang menunjukkan durasi suatu objek aktif menjalankan proses tertentu.
6	<i>Fragment</i> (<i>alt</i> , <i>opt</i>)		Simbol yang membungkus beberapa pesan untuk menunjukkan kondisi khusus dalam interaksi.
7	<i>Return Message</i>		Garis panah putus-putus yang menunjukkan pesan balasan atau konfirmasi dari suatu proses.
8	<i>Object</i>		Objek adalah elemen atau entitas yang merepresentasikan komponen tertentu dalam sistem, biasanya ditempatkan di bagian atas <i>lifeline</i> .

2.4.5. *Hypertext Markup Language* (HTML)

Menurut Anamisa dan Mufarroha (2022), *Hypertext Markup Language* (HTML) adalah bahasa *markup* standar yang digunakan untuk membuat struktur halaman web. HTML menyediakan elemen-elemen dasar seperti teks, gambar, tautan, dan tabel, yang diatur dalam bentuk *tag*. Dengan menggunakan HTML, pengembang web dapat mendefinisikan konten dan *layout* suatu halaman, sehingga memungkinkan penampilan web untuk menampilkan informasi secara terorganisir dan interaktif.

2.4.6. Cascading Style Sheets (CSS)

Menurut Anamisa dan Mufarroha (2022), *Cascading Style Sheets (CSS)* adalah bahasa yang digunakan untuk mendesain dan mengatur tampilan halaman web. CSS memungkinkan pengembang web untuk memisahkan konten dari presentasi, sehingga memudahkan pengelolaan dan pemeliharaan tampilan situs. Dengan CSS, pengembang dapat mengatur berbagai elemen, seperti warna, *font*, *layout*, dan *spacing*, memberikan kebebasan untuk menciptakan desain yang menarik dan responsif. Selain itu, CSS mendukung konsep *cascading*, di mana aturan-aturan yang lebih spesifik dapat menimpa aturan yang lebih umum, memungkinkan kontrol yang lebih besar terhadap gaya visual halaman web.

2.4.7. Hypertext Preprocessor (PHP)

Menurut Khanna (2020), *Hypertext Preprocessor (PHP)* adalah bahasa pemrograman *server-side* yang dirancang untuk pengembangan web. PHP awalnya diciptakan untuk membuat halaman *web* dinamis dan dapat di-*embed* ke dalam HTML, sehingga memudahkan pengembang untuk membangun aplikasi web dengan cepat. PHP menjadi populer karena kemampuannya dalam memproses data dari sisi server, menangani formulir, dan berinteraksi dengan berbagai jenis basis data. Selain itu, PHP bersifat *open-source*, yang menjadikannya pilihan yang ekonomis dan fleksibel untuk pengembangan aplikasi web yang dapat dijalankan pada berbagai *platform* dan sistem operasi.

2.4.8. Javascript

Menurut Anamisa dan Mufarroha (2022), *JavaScript* adalah bahasa pemrograman yang digunakan untuk membuat halaman web menjadi interaktif dan dinamis. Sebagai salah satu teknologi inti dalam pengembangan web, JavaScript memungkinkan pengembang untuk menambahkan elemen interaktif seperti animasi, pengolahan data *input* pengguna, dan pembaruan konten secara *real-time* tanpa perlu memuat ulang halaman. Dengan dukungan luas dari berbagai *browser*, JavaScript dapat digunakan di sisi klien (*client-side*) untuk meningkatkan pengalaman pengguna dan juga dapat diterapkan di sisi server (*server-side*).

2.4.9. Bootstrap

Menurut Anamisa dan Mufarroha (2022), *Bootstrap* adalah *framework* CSS yang dirancang untuk mempermudah pengembangan antarmuka pengguna *User Interface (UI)* yang *responsive* dan *mobile-first*. Dengan menyediakan kumpulan komponen dan kelas CSS yang siap pakai, *Bootstrap* memungkinkan pengembang untuk dengan cepat membangun halaman web yang memiliki tampilan yang konsisten di berbagai perangkat. *Framework* ini juga dilengkapi dengan *JavaScript* plugins yang dapat menambahkan interaktivitas, serta mendukung kustomisasi untuk menyesuaikan tema dan gaya sesuai kebutuhan proyek.

2.4.10. AdminLTE

Menurut Nata dkk., (2023), *AdminLTE* adalah salah satu *template* administrasi berbasis HTML yang dirancang menggunakan *framework* CSS *Bootstrap*. *Template* ini menyediakan berbagai komponen antarmuka pengguna (UI) yang memudahkan pengembang dalam membangun halaman administrasi aplikasi web. *AdminLTE* menawarkan tampilan yang responsif, sehingga dapat diakses dengan baik di berbagai perangkat, baik desktop maupun *mobile*.

2.4.11. Server-side Scripting

Menurut Nauli dkk., (2024), *server-side scripting* adalah teknik pemrograman yang dijalankan di sisi *server* untuk menghasilkan konten dinamis dan berinteraksi dengan *database* sebelum mengirimkan hasilnya ke klien. Dalam proses ini, skrip yang ditulis dalam bahasa pemrograman seperti PHP, Python, atau Ruby dieksekusi di *server*, memungkinkan pengembang untuk menangani logika bisnis, mengelola data pengguna, dan memproses permintaan dari klien. Hasil eksekusi skrip tersebut dikirim ke *browser* pengguna dalam bentuk HTML, sehingga pengguna dapat melihat konten yang relevan dan interaktif sesuai dengan permintaan mereka.

2.4.12. JavaScript Object Notation (JSON)

Menurut Nurbarkah (2021), *JavaScript Object Notation (JSON)* adalah format pertukaran data yang ringan dan mudah dibaca oleh manusia serta mudah diparse oleh mesin. *JSON* digunakan untuk menyimpan dan mentransfer data dalam

struktur objek yang terdiri dari pasangan kunci-nilai. Format ini sangat populer dalam pengembangan aplikasi web, terutama dalam komunikasi antara *server* dan klien, karena kemampuannya untuk menangani data kompleks dengan cara yang sederhana dan efisien. Dengan dukungan luas dari berbagai bahasa pemrograman, JSON menjadi pilihan utama untuk pertukaran data dalam aplikasi berbasis web.

2.4.13. Laravel

Menurut Khanna (2020), Laravel merupakan salah satu *framework* PHP yang menawarkan berbagai alat dan struktur yang mempermudah pengembangan aplikasi web. Laravel dikenal sebagai *framework* yang fleksibel dan efisien, khususnya dalam konsep routing yang diperkenalkannya, serta berbagai fitur lainnya seperti *Eloquent ORM* untuk manajemen basis data dan *Blade* untuk *templating*. Kemudahan ini menjadikan Laravel banyak digunakan oleh pengembang untuk membangun aplikasi web yang kompleks dalam waktu yang lebih singkat.

2.4.14. SQLite

Menurut Ardian dan Setyawati (2024), SQLite adalah sistem manajemen basis data yang bersifat ringan dan berukuran kecil, yang dirancang untuk digunakan dalam aplikasi *embedded* atau aplikasi yang membutuhkan *database* yang sederhana. SQLite tidak memerlukan server terpisah, sehingga dapat langsung diintegrasikan ke dalam aplikasi, seperti aplikasi *mobile* atau *desktop*. Dengan menggunakan SQLite, pengembang dapat dengan mudah menyimpan, mengelola, dan mengambil data tanpa memerlukan konfigurasi yang rumit, menjadikannya pilihan populer untuk usaha mikro kecil menengah (UMKM) yang membutuhkan solusi basis data yang efisien dan mudah digunakan.

2.4.15. Simple Random Sampling.

Menurut Rahmani (2022), menggunakan metode *simple random sampling* untuk memastikan setiap individu dalam populasi memiliki peluang yang sama untuk dipilih sebagai responden. Populasi penelitian mencakup karyawan yang menggunakan SIA (Sistem Informasi Akuntansi), dengan sampel diambil secara acak untuk mengurangi bias dan meningkatkan representativitas. Metode ini dipilih karena sederhana, efektif, dan memungkinkan generalisasi hasil ke seluruh

populasi. Dengan pendekatan ini, penelitian bertujuan untuk mengidentifikasi faktor-faktor yang memengaruhi kualitas SIA secara objektif dan memberikan rekomendasi yang relevan untuk perbaikan sistem di organisasi.

Menurut Kusuma (2024), teori *random sampling* dijelaskan sebagai teknik pengambilan sampel di mana setiap individu dalam populasi memiliki kesempatan yang sama untuk dipilih sebagai bagian dari sampel. Metode ini bertujuan untuk memastikan bahwa sampel yang dihasilkan representatif terhadap populasi pengguna aplikasi SIMDA di Dinas Pendidikan. Penelitian ini memanfaatkan *random sampling* untuk mendapatkan data yang tidak bias dan mewakili berbagai pengguna aplikasi, sehingga evaluasi pengalaman pengguna aplikasi dapat dilakukan secara objektif. Teknik ini didasarkan pada prinsip bahwa pengambilan sampel secara acak mampu mengurangi kemungkinan bias seleksi dan memberikan hasil yang lebih generalisabel terhadap populasi. Dengan demikian, *random sampling* menjadi metode yang tepat untuk memastikan keberagaman responden dan meningkatkan validitas hasil penelitian yang menggunakan metode UEQ.

2.4.16. Pengujian *User Acceptance Testing* (UAT).

Menurut Parameswari dkk., (2023), dalam pengembangan aplikasi monitoring stok barang, pengujian *User Acceptance Testing* (UAT) penting dilakukan untuk memastikan bahwa aplikasi tersebut sesuai dengan kebutuhan pengguna. Proses ini melibatkan pengguna langsung dalam berinteraksi dengan aplikasi untuk mengevaluasi fungsi dan kemudahan penggunaannya.

Menurut Supriatna dkk., (2019) dalam tesisnya, *User Acceptance Testing* (UAT) adalah tahap pengujian akhir dalam pengembangan perangkat lunak yang dilakukan oleh pengguna akhir untuk memverifikasi apakah sistem telah memenuhi kebutuhan dan harapan mereka. UAT memastikan bahwa aplikasi atau sistem yang dikembangkan berfungsi sesuai spesifikasi yang disepakati dan siap digunakan dalam lingkungan nyata.

Menurut Yakub dkk., (2024), UAT merupakan metode pengujian yang fokus pada validasi bahwa sistem sesuai dengan kebutuhan pengguna. Pengujian ini membantu mengidentifikasi ketidaksesuaian dalam fungsionalitas bisnis yang

mungkin tidak terdeteksi oleh pengujian unit atau sistem, yang lebih berfokus pada komponen individual dan integrasi sistem secara teknis.

UAT bertujuan untuk mengukur sejauh mana sistem yang dikembangkan memenuhi persyaratan pengguna secara menyeluruh dan mengonfirmasi bahwa sistem siap digunakan dalam lingkungan nyata tanpa kendala teknis yang berarti.

Dalam hal ini, menurut Yakub dkk., (2024), pengujian *User Acceptance Testing* (UAT) diterapkan untuk mengukur lima komponen utama dari sistem, yaitu:

- a. *Learnability*: Tingkat kemudahan pengguna dalam menggunakan sistem untuk menyelesaikan tugasnya yang pertama kali menggunakan sistem.
- b. *Efficiency*: Kemampuan sistem untuk mendukung pengguna dalam menyelesaikan tugasnya dalam waktu yang relatif cepat waktu.
- c. *Memorability*: Tingkat kenyamanan pengguna dalam menggunakan sistem dengan baik setelah beberapa saat tidak menggunakannya.
- d. *Errors*: Kemungkinan error atau kesalahan pada sistem yang sedang digunakan oleh pengguna dan dapat pulih kembali dari kesalahan.
- e. *Satisfaction*: Tingkat kepuasan pengguna dalam menggunakan sistem, baik tampilan sistem maupun dalam penggunaannya.

Pertanyaan dalam kuesioner UAT terdapat pada Tabel 2.5, untuk rentang nilai kriteria interval berada di Tabel 2.4, dan berikut rumus dalam menghitung pembobotan UAT menurut (Yakub dkk., 2024) disajikan pada Persamaan (2.2) dan (2.3)

$$Q_n = \sum_{i=1}^5 F(i) \times scale(i) \quad (2.2)$$

$$P = \frac{Total Q_n}{N} / 5 \times 100\% \quad (2.3)$$

Di mana:

Q_n = Pertanyaan (1,2,3....n)

n = 1,2,3,.....10

F = Frekuensi Jawaban

$Scale$ = Skala Likert

P = Persentase

N =Total Responden

Tabel 2. 4 Presentase Nilai Interval (*Yakub dkk., 2024*)

Skor	Keterangan
0% – 19,99%	Sangat Tidak Baik
20% – 39,99%	Kurang Baik
40% – 59,99%	Netral
60% – 79,99%	Baik
80% – 100,00%	Sangat Baik

Tabel 2. 5 Pertanyaan Kuisoner Pengujian UAT

No	Pertanyaan
Komponen <i>Learnability</i>	
1	Apakah sistem mudah dipahami?
2	Apakah fitur-fitur pada sistem mudah digunakan?
Komponen <i>Efficiency</i>	
3	Apakah sistem mudah digunakan untuk pencatatan secara online?
4	Apakah menurut anda sistem mudah di terapkan pada instansi tersebut?
Komponen <i>Memorability</i>	
5	Apakah sistem dapat melakukan proses pencatatan stok dengan baik?
6	Apakah tata letak menu dan navigasi sistem membantu Anda mengingat lokasi fitur tertentu?
Komponen <i>Errors</i>	
7	Apakah kesalahan yang terjadi pada sistem teratasi dengan baik?
8	Apakah fitur pada sistem berfungsi dengan baik?
Komponen <i>Satisfaction</i>	
9	Apakah sistem memiliki tampilan yang menarik?
10	Apakah pengalaman menggunakan sistem ini baik?