

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1. Tinjauan Pustaka

Tinjauan pustaka melibatkan penyelidikan kajian-kajian terdahulu atau yang telah dilakukan sebelumnya, bertujuan untuk membandingkannya dengan penelitian saat ini dan penelitian yang akan datang. Fungsi dari tinjauan literatur adalah untuk memberikan sumber informasi yang relevan sehingga peneliti dapat menghindari plagiarisme selama proses penelitian. Selain itu, tinjauan literatur diperlukan untuk mengidentifikasi perbedaan konten di dalam setiap karya ilmiah. Beberapa penelitian sebelumnya terkait dengan fokus penelitian ini, melibatkan elemen-elemen seperti judul, perbandingan (*comparing*), kontras (*contrasing*), kritik (*criticize*), sintesis (*synthesize*), dan ringkasan (*summary*).

Tabel 2. 1 Tabel Tinjauan Pustaka

No.	Judul	<i>Comparing</i>	<i>Contrasting</i>	<i>Critisize</i>	<i>Synthesize</i>	<i>Summarize</i>
1.	<i>Design and Build a Quality Assurance Document Archiving Application Using the Rapid Application Development</i> [7].	Pengembangan sistem berbasis <i>website</i> menggunakan <i>Framework</i> <i>Laravel</i> dan metode <i>Rapid Application Development</i> (RAD) dalam pengembanganny a.	Memiliki objek penelitian sistem pengarsipan dokumen <i>Quality Assurance</i> .	Saran yang diberikan bersifat umum dan bisa mendapat manfaat dari lebih banyak spesifik. Misalnya, menyebutkan fitur-fitur mana yang perlu pengembangan lebih lanjut atau bagaimana situs web dapat diintegrasikan dengan sistem informasi manajemen yang sudah ada.	Penelitian lanjutan dapat mengevaluasi efektivitas sebenarnya dari implementasi sistem ini dalam meningkatkan efisiensi pengarsipan dan proses <i>quality assurance</i> .	Penelitian ini menghasilkan Sistem Pengarsipan Dokumen <i>quality assurance</i> Web dengan R.A.D dan <i>Laravel</i> . Sistem efisien dalam pengelompokan dan pencarian dokumen, diuji tanpa kesalahan. Pengembangan mendatang fokus pada fungsionalitas, antarmuka pengguna, integrasi MIS, dan penelitian di institusi pendidikan lain.
2.	<i>Designing a web-based career system using the</i>	Pengembangan sistem berbasis <i>website</i> menggunakan	Menggunakan metode <i>waterfall</i> dan memiliki objek pencarian	Tidak adanya metode pengujian yang digunakan dalam menguji	Penelitian lanjutan dapat dilakukan untuk menganalisis	Penelitian ini membahas pembuatan aplikasi pencarian pekerjaan

No.	Judul	<i>Comparing</i>	<i>Contrasting</i>	<i>Critisize</i>	<i>Synthesize</i>	<i>Summarize</i>
	<i>laravel framework</i> [8]	<i>framework laravel.</i>	sistem pekerjaan dan rekrutmen karyawan untuk berbagai industri.	fungsionalitas sistem.	dampak nyata dari penggunaan aplikasi ini pada pemohon dan industri. Ini mencakup evaluasi efektivitas dalam mengisi posisi pekerjaan, respons waktu, dan peningkatan proses rekrutmen.	berbasis <i>website</i> dengan menggunakan kerangka kerja Laravel dan metode <i>waterfall</i> .
3.	<i>Development of a Web Based Inventory Information System</i> [2].	Pengembangan sistem berbasis <i>website</i> untuk manajemen stok.	Menggunakan metode penelitian <i>Waterfall</i>	Tidak adanya metode pengujian yang digunakan dalam menguji fungsionalitas sistem.	Penelitian lanjutan dapat dilakukan untuk mengevaluasi efisiensi sistem dengan pengujian skenario penggunaan dengan tujuan memastikan kinerja sistem optimal.	Penelitian ini membahas pengembangan aplikasi web berbasis Yii <i>Framework</i> untuk meningkatkan efisiensi manajemen stok, Dengan model <i>waterfall</i> dan fokus pada pencatatan, pemrosesan, dan pelaporan data inventaris.

No.	Judul	<i>Comparing</i>	<i>Contrasting</i>	<i>Critisize</i>	<i>Synthesize</i>	<i>Summarize</i>
4.	Rancang Bangun Sistem Manajemen Transaksi Dan Stok Barang Toko Kutus-Kutus Bajra[9]	Pengembangan sistem manajemen stok barang berbasis <i>website</i> menggunakan <i>framework</i> laravel.	Memiliki objek manajemen transaksi dan stok barang tanpa metode pengembangan khusus.	Penelitian kurang mendalami pertanyaan UMUX yang mendapat penilaian rendah tanpa memberikan analisis atau saran konkrit untuk meningkatkan pengalaman pengguna pada aspek yang dinilai kurang baik.	Penelitian selanjutnya dapat memfokuskan perbaikan pada aspek-aspek yang dinilai rendah. Pengoptimalan antarmuka pengguna perlu diperhatikan untuk mengurangi frustrasi dan meningkatkan efisiensi waktu.	Penelitian ini membahas pembangunan aplikasi web manajemen transaksi dan stok di Toko Kutus-Kutus Bajra menggunakan Laravel, dengan penekanan pada pengoptimalan antarmuka pengguna dan pilihan metode pengujian.
5.	Desain Aplikasi Tracer Study Berbasis Web menggunakan Laravel <i>Framework</i> [10]	Mengembangkan aplikasi berbasis <i>website</i> menggunakan <i>framework</i> laravel.	Memiliki objek penelitian sistem Tracer study yang mengukur dan mencari kinerja lulusan Universitas Raharja.	Pada penelitian ini tidak menggunakan metode pengembangan secara khusus.	Penelitian lanjutan yang dapat dilakukan adalah mengembangkan sistem dengan <i>framework</i> yang berbeda seperti menggunakan codeigniter atau yang lainnya, untuk dapat membandingkan	Penelitian ini membahas pengembangan aplikasi tracer study menggunakan <i>Framework</i> Laravel untuk membantu universitas melacak data alumni dan mengukur kinerja lulusan. Desain sistem mencakup fitur seperti

No.	Judul	<i>Comparing</i>	<i>Contrasting</i>	<i>Critisize</i>	<i>Synthesize</i>	<i>Summarize</i>
					hasil yang didapatkan.	manajemen akun pengguna, data alumni, kuesioner, dan pembuatan laporan.
6.	Perancangan Sistem Informasi <i>Inventory</i> Penjualan <i>Sparepart</i> Berbasis Web Menggunakan Metode Rad[11]	Perancangan <i>website Inventory</i> menggunakan <i>framework</i> Laravel dan metode RAD.	Pada Penelitian tidak menggunakan pengujian <i>usability</i> .	Tidak adanya pengujian <i>usability</i> untuk menguji penggunaan sistem yang telah dirancang.	Penelitian selanjutnya dapat dilakukan pengujian pada sistem sehingga dapat mengetahui <i>user experience</i> dari pengguna sistem.	Penelitian ini Membahas tentang perancangan sistem informasi <i>inventory</i> penjualan <i>sparepart</i> berbasis <i>website</i> menggunakan metode rad di Perusahaan CV. Supra Jaya Motor untuk membantu pengelolaan data menjadi lebih cepat, akurat, dan efisien.
7.	Rancang Bangun Sistem Informasi <i>Inventory</i> Menggunakan Metode <i>Rapid Application Development</i>	Perancangan dan pembangunan <i>website Inventory</i> menggunakan <i>framework</i> Laravel dan metode RAD.	Memiliki subjek penelitian pada toko <i>furniture</i> dan tidak menggunakan pengujian <i>usability</i> .	Penelitian tidak melakukan pengujian <i>usability</i> pada sistem yang telah dibangun.	Penelitian lanjutan dapat dilakukan untuk mengevaluasi efisiensi sistem dengan pengujian skenario penggunaan	Penelitian ini membahas tentang rancang bangun sistem informasi menggunakan <i>framework</i> Laravel dan metode RAD, untuk meningkatkan

No.	Judul	<i>Comparing</i>	<i>Contrasting</i>	<i>Critisize</i>	<i>Synthesize</i>	<i>Summarize</i>
	(RAD) Pada Toko Abadi Jaya[12]				dengan tujuan memastikan <i>user experience</i> yang optimal.	kecepatan dan efisiensi proses bisnis yang ada.
8.	Pembangunan <i>Website</i> Manajemen Inventori Bahan Baku Makanan Pada Rumah Makan Alam Mutiara Menggunakan Metode RAD (<i>Rapid Application Development</i>)[13]	Pembangunan <i>website Inventory</i> menggunakan metode RAD.	Memiliki objek penelitian berupa persediaan bahan baku makanan dan subjek penelitian pada restoran dan tidak menggunakan pengujian <i>usability</i> .	Penelitian tidak melakukan pengujian <i>usability</i> pada sistem yang telah dibangun sehingga tidak adanya pengujian pada peningkatan dari proses bisnis yang ada.	Penelitian lanjutan dapat dilakukan untuk mengevaluasi efisiensi sistem dengan pengujian skenario penggunaan dengan tujuan memastikan bahwa sistem telah meningkatkan efisiensi proses bisnis yang ada.	Penelitian ini merancang sistem informasi manajemen persediaan bahan baku berbasis <i>website</i> menggunakan metode RAD di rumah makan Alam Mutiara. Sistem ini meningkatkan efisiensi dan efektivitas pengelolaan bahan baku, memudahkan pencatatan dan pelaporan, serta memastikan ketersediaan bahan baku untuk meningkatkan kepuasan pelanggan.

Perbedaan yang terdapat dalam penelitian ini dengan 8 penelitian dalam tinjauan pustaka terdapat pada subjek penelitiannya, yaitu pada perusahaan distribusi ikan. Selain itu *framework* yang digunakan dalam penelitian juga memiliki versi yang berbeda. Pada penelitian sebelumnya masih menggunakan *framework* laravel dibawah versi 10 yang terbaru, sedangkan pada penelitian ini menggunakan *framework laravel* versi 10. Pada penelitian sebelumnya juga masih banyak yang belum melakukan pengujian usability pada hasil sistem yang telah dibangun, sedangkan dalam penelitian ini dilakukan pengujian *usability* terutama dalam hal kemudahan penggunaan sistem menggunakan metode *Single Ease Question* (SEQ).

2.2. Landasan Teori

2.2.1. Manajemen Stok

Manajemen stok atau *inventory* memiliki peran penting dalam mengelola operasional suatu organisasi atau perusahaan. Stok bahan baku melibatkan persediaan material mentah yang diperlukan untuk memulai proses produksi, sedangkan stok barang dalam proses produksi mencakup item yang sedang melalui tahap pengerjaan. Stok *maintenance/repair/operating supplier* (MROs) terdiri dari item yang mendukung kelancaran operasional, seperti suku cadang atau peralatan pemeliharaan. Terakhir, stok barang jadi mencakup produk yang telah selesai diproses dan siap untuk didistribusikan atau dijual kepada pelanggan[14][15]. Pentingnya sistem stok tidak hanya terletak pada penyediaan barang yang memadai, tetapi juga pada efisiensi pengelolaan persediaan[14].

2.2.2. Website

Website adalah kumpulan halaman atau dokumen digital yang dapat diakses melalui internet[15]. Setiap halaman dalam *website* dapat berisi berbagai jenis konten seperti teks, gambar, video, dan elemen interaktif lainnya. *Websites* biasanya saling terhubung melalui tautan atau *hyperlink*, membentuk struktur informasi yang terorganisir. *Website* memiliki berbagai tujuan, termasuk menyajikan informasi, menyediakan produk atau layanan, memfasilitasi

komunikasi, dan memberikan pengalaman *online*. Pengembangan dan pengelolaan *website* melibatkan penggunaan teknologi seperti HTML, CSS, dan *JavaScript*, yang membantu membentuk tata letak, gaya, dan interaktivitas halaman. *Website* dapat bersifat statis, dengan konten tetap, atau dinamis, yang memungkinkan konten diperbarui secara berkala. *Website* tidak hanya menjadi wadah digital untuk berbagai jenis konten dan interaksi *online*, tetapi juga mencerminkan perpaduan teknologi dan desain yang diperlukan untuk memberikan pengalaman pengguna yang efektif dan memenuhi berbagai tujuan, baik itu penyampaian informasi, komunikasi, maupun pelayanan produk dan layanan secara dinamis[16].

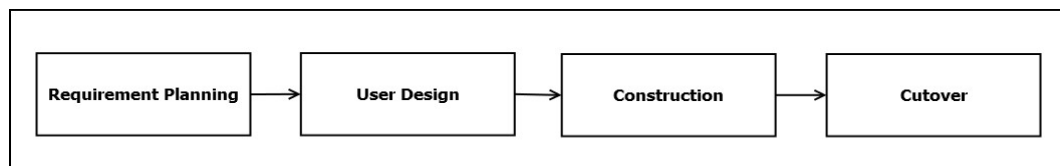
2.2.3. *System Development Life Cycle (SDLC)*

Software Development Life Cycle (SDLC) adalah suatu kerangka kerja yang digunakan dalam dunia pengembangan perangkat lunak untuk memandu proses dari konsepsi ide hingga implementasi dan pemeliharaan aplikasi. SDLC menggambarkan serangkaian tahapan yang dilalui oleh tim pengembang untuk mencapai tujuan akhirnya. Tahap-tahap tersebut melibatkan identifikasi kebutuhan, perencanaan, analisis, desain, implementasi, pengujian, dan pemeliharaan. Setiap tahap memiliki peranannya masing-masing dalam memastikan bahwa perangkat lunak yang dihasilkan memenuhi standar kualitas, keamanan, dan kebutuhan pengguna. Konsep SDLC menjadi landasan yang memungkinkan pengembangan perangkat lunak dilakukan secara terstruktur dan efisien[17].

2.2.4. *Rapid Application Development (RAD)*

James Martin memperkenalkan *Rapid Application Development (RAD)*, suatu pendekatan pengembangan sistem yang didesain untuk memberikan hasil yang berkualitas tinggi dengan waktu pengembangan yang lebih singkat dibandingkan dengan siklus hidup tradisional[18]. Pendekatan ini mencakup metode pengembangan perangkat lunak untuk menyediakan proses pengembangan yang lebih cepat. Dalam penerapan RAD, pengembangan aplikasi dapat dilakukan

dengan cepat karena melibatkan partisipasi aktif para pemakai yang berperan sebagai pengambil keputusan pada setiap tahapan pengembangan. Hal ini memungkinkan sistem yang dikembangkan dapat memenuhi keinginan para pemakai, mengurangi kebutuhan untuk pengembangan ulang setelah tahap implementasi. Pendekatan RAD, dengan demikian, memberikan kemampuan untuk menghasilkan sistem dengan efisiensi waktu yang tinggi dan responsif terhadap kebutuhan pengguna[3]. Siklus hidup RAD memiliki empat fase, untuk membantu memastikan bahwa para pengembang membangun hal yang benar-benar dibutuhkan oleh pengguna yang dapat dilihat pada Gambar 2.1[18].



Gambar 2. 1 Siklus Hidup RAD[18]

1) *Requirement Planning*

Fase *Requirement Planning* melibatkan pengguna akhir berpengetahuan tinggi dalam menentukan fungsi-fungsi sistem. Dalam fase ini, diskusi terstruktur mengenai masalah bisnis dilakukan dengan bimbingan profesional Sistem Informasi. Keterlibatan pengguna dan eksekutif yang tepat penting untuk menjalankan perencanaan persyaratan dengan cepat dan komprehensif. Fase ini bertujuan membentuk pemahaman yang jelas tentang fungsi yang diinginkan dari sistem, menjadi dasar langkah-langkah selanjutnya dalam pengembangan perangkat lunak[18].

2) *User Design*

Fase *User Design Phase* adalah langkah penting dalam pengembangan perangkat lunak di mana pengguna akhir turut serta secara aktif dalam merancang bagian non-teknis dari sistem. Fase Desain Pengguna bertujuan memastikan desain yang disepakati pengguna dapat langsung diwujudkan ke tahap pembuatan sistem (Fase *Construction*)[18].

3) *Construction*

Fase *Construction* dalam siklus pengembangan perangkat lunak adalah tahap di mana kode program dibuat. Fase *Construction* memungkinkan konstruksi sistem dengan cepat, dengan pengujian yang dilakukan sepanjang proses untuk memastikan produk akhir memenuhi kebutuhan pengguna saat diimplementasikan[18].

4) *Cutover*

Fase *cutover* adalah tahap dalam siklus pengembangan perangkat lunak yang melibatkan implementasi dan migrasi dari sistem lama ke sistem baru. Pada fase ini, dilakukan serangkaian tindakan termasuk pengujian, pelatihan pengguna, dan operasi bersamaan dengan sistem lama hingga sistem baru berjalan dengan efektif. Tujuannya adalah memastikan transisi yang lancar dan sukses dengan dampak negatif yang minimal. Persiapan untuk fase *cutover* dimulai pada tahap awal pengembangan untuk meminimalkan risiko dan memastikan kelancaran implementasi[18].

2.2.5. *Database*

Database adalah suatu kumpulan data yang saling terhubung secara logis dan deskriptif, dirancang untuk memenuhi kebutuhan informasi dalam suatu organisasi. *Database* memiliki sifat penyimpanan data dengan format yang independen dan fleksibel. Keuntungan utama dari penggunaan *database* adalah bahwa data didefinisikan secara terpisah dari program aplikasi yang mengaksesnya. Artinya, *database* dapat diakses dan digunakan oleh berbagai program aplikasi tanpa mempengaruhi struktur atau format data yang terdapat di dalamnya. Selain itu, *database* juga memungkinkan perluasan (lingkup) tanpa memberikan dampak negatif pada program-program yang menggunakan data dari *database* tersebut. *Database* memberikan kemudahan dalam manajemen dan akses data secara efisien untuk mendukung kebutuhan informasi organisasi[19].

Database memiliki dua komponen penting, yaitu data dan informasi, dengan tujuan utama mengelola data sehingga dapat menjadi informasi yang diinginkan dan memfasilitasi proses pengambilan, penghapusan, dan pengeditan data secara

efektif, efisien, dan akurat. *Database Management System (DBMS)* atau Manajemen Basis Data merupakan perangkat lunak yang dirancang untuk mengelola dan mengeksekusi kueri (query) basis data. DBMS mencakup semua aspek dari pembuatan *database* hingga operasi-operasi di dalamnya, seperti pengelolaan entri data, pengeditan, penghapusan, pembuatan laporan, dan sebagainya, dengan tujuan menjalankan tugas-tugas tersebut secara efektif dan efisien. Contoh perangkat lunak DBMS meliputi MySQL, Oracle, Sybase, dBase, MS. SQL, dan PostgreSQL, dan MongoDB. Keberadaan *database* dan DBMS menjadi krusial dalam konteks manajemen informasi, memungkinkan organisasi untuk mengoptimalkan pengolahan data, meminimalkan kesalahan, dan memaksimalkan nilai informasi yang dihasilkan, sehingga mendukung kelancaran berbagai operasi dan pengambilan keputusan yang strategis[20].

2.2.6. *Hypertext Preprocessor (PHP)*

Hypertext Preprocessor (PHP) adalah sebuah bahasa pemrograman berbentuk skrip yang secara khusus dirancang untuk membangun aplikasi *website* dinamis. Karakteristik utama PHP adalah penempatannya di sisi *server*, di mana kode PHP dieksekusi secara langsung oleh *server website*. Proses pemrosesan terjadi saat pengguna mengakses halaman *website* yang mengandung kode PHP. *Server* memproses kode tersebut, menjalankan logika bisnis, berinteraksi dengan basis data, dan menghasilkan *output* dinamis. Hasil pemrosesan ini kemudian dikirimkan kepada *website browser* klien, yang menampilkan halaman web yang dibangun sesuai dengan logika yang diimplementasikan dalam kode PHP. PHP memungkinkan pembentukan tampilan yang responsif dan berubah-ubah berdasarkan permintaan terkini. PHP yang fokus pada sisi *server*, mampu memberikan pengalaman *website* dinamis yang memadai, memisahkan logika aplikasi dari tampilan untuk memudahkan pengembangan dan pemeliharaan kode[21].

2.2.7. *Hyper Text Markup Language (HTML)*

Hypertext Markup Language (HTML) adalah bahasa markup yang digunakan untuk membuat dan merancang struktur dasar sebuah halaman *website*. HTML memberikan cara untuk menyusun dan menyematkan berbagai elemen, seperti teks, gambar, tautan, formulir, dan elemen multimedia lainnya, dalam sebuah dokumen *website*[16]. Bahasa ini memungkinkan pengembang *website* untuk mendefinisikan tentang konten suatu halaman *website* akan ditampilkan dan diatur oleh *browser*. HTML bekerja dengan menggunakan serangkaian *tag* atau tanda yang ditempatkan di dalam dokumen, dan setiap *tag* memiliki fungsi khusus yang menentukan peran dan penampilan dari elemen yang diapitnya. HTML tidak hanya menjadi fondasi teknis untuk merancang halaman *website*, tetapi juga menjadi kunci kreativitas pengembang dalam menyajikan informasi secara visual dan interaktif melalui struktur yang terorganisir dan intuitif[22].

2.2.8. Laravel framework

Laravel adalah sebuah kerangka kerja pengembangan aplikasi *website* yang sangat diminati dan kuat dalam ekosistem PHP. Didesain untuk memfasilitasi proses pengembangan aplikasi, Laravel menawarkan sejumlah fitur dan alat yang memungkinkan para pengembang untuk menciptakan aplikasi *website* dengan efisiensi, keamanan, dan keterkelolaan yang optimal. Keunggulan Laravel mencakup sistem *routing* yang fleksibel, integrasi manajemen otentikasi, dan Kemampuan untuk bermigrasi basis data dapat menyederhanakan pengelolaan skema basis data dan mempermudah proses pemindahan data. Salah satu fitur utama yang membedakan Laravel adalah *Eloquent*, sebuah *Object-Relational Mapping (ORM)* yang terintegrasi, memungkinkan pengembang berkomunikasi dengan basis data melalui sintaks PHP yang bersih dan mudah dipahami. Laravel juga menggunakan *Blade* sebagai *template engine* memberikan kemudahan dalam pengembangan tampilan dengan sintaks yang mudah dipahami. Kelebihan Laravel juga terlihat dari sintaksis yang ekspresif dan dokumentasi yang kaya, membuatnya menjadi pilihan utama bagi banyak pengembang yang ingin mempercepat pengembangan aplikasi *website*[4].

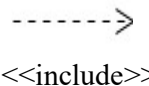
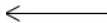
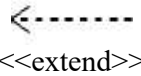
2.2.9. Unified Modeling Language (UML)

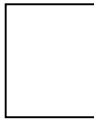
Unified Modeling Language (UML) adalah bahasa pemodelan visual yang secara luas digunakan dalam pengembangan perangkat lunak untuk merancang, memodelkan, dan mendokumentasikan sistem[23]. UML menyediakan notasi grafis standar yang memungkinkan para pengembang secara jelas dan efektif menggambarkan struktur, perilaku, dan interaksi di antara komponen-komponen suatu sistem perangkat lunak[24]. Berbagai jenis diagram UML, seperti *class diagram*, *use case diagram*, dan *Activity diagram*, memberikan cara yang terstandarisasi untuk menyajikan aspek-aspek kritis dari perangkat lunak. Penggunaan UML membantu tim pengembangan dalam memahami kompleksitas sistem, meningkatkan komunikasi internal, dan menyediakan dokumen visual yang diperlukan untuk perencanaan, implementasi, dan pemeliharaan perangkat lunak. UML sebagai alat pemodelan yang umum digunakan, membantu menciptakan desain perangkat lunak yang konsisten, serta memfasilitasi kerjasama yang efektif di antara anggota tim pengembangan[25].

2.2.10. Use Case Diagram

Use Case Diagram adalah suatu jenis diagram dalam *Unified Modeling Language* (UML) yang bertujuan menggambarkan interaksi antara aktor (seperti manusia atau entitas lain) dengan sistem. Pada diagram ini, setiap fungsionalitas sistem direpresentasikan oleh elips dengan nama *use case* di dalamnya, sedangkan aktor direpresentasikan oleh ikon manusia atau entitas lain di luar elips *use case*. Garis panah menghubungkan aktor dengan *use case*, menunjukkan bagaimana aktor berinteraksi dengan sistem melalui berbagai fungsionalitas atau tindakan yang diwakili oleh *use case*. Adapun keterangan lengkap mengenai *node* yang digunakan dalam *use case diagram* terdapat pada table 2.2 dibawah. *Use Case Diagram* memberikan pandangan tingkat tinggi tentang skenario fungsional dan proses bisnis dalam sistem, termasuk interaksi antara aktor dan fungsionalitas apa saja yang dapat dilakukan oleh masing-masing akto. Diagram ini sangat berguna dalam menggambarkan bagaimana sistem berperilaku dari perspektif pengguna atau aktor yang berinteraksi dengan sistem tersebut, berikut adalah *node-node* yang digunakan dalam *use case diagram*[26]:

Tabel 2. 2 Node Use Case Diagram[26]

No	Simbol	Keterangan
1	<i>Use Case</i> 	Simbol yang menggambarkan interaksi-interaksi antar aktor (pengguna) dan tujuannya untuk menjelaskan fungsionalitas dan perilaku sistem yang akan dikembangkan.
2	<i>Actor</i> 	Menggambarkan peran khusus yang dimainkan oleh pengguna saat terlibat dalam interaksi dengan <i>use case</i> .
3	<i>Dependency</i> 	Hubungan ini mengacu pada suatu perubahan dimana suatu unsur yang mandiri akan berpengaruh terhadap unsur yang tidak memiliki otonomi.
4	<i>Include</i> 	Simbol ini mewakili hubungan di mana <i>use case</i> tambahan terkandung di dalam <i>use case</i> lain yang mana <i>use case</i> tersebut ditambahkan dengan <i>use case</i> utama yang digunakan untuk melakukan beberapa fungsionalitas.
5	<i>Generalization</i> 	Menggambarkan objek anak mengadopsi perilaku dan struktur data yang diwarisi dari objek induk yang berada di atasnya.
6	<i>Extend</i> 	Menggambarkan <i>use case</i> target melibatkan perluasan fungsionalitas dari <i>use case</i> sumber pada titik yang telah ditentukan sebelumnya.
7	<i>Collaboration</i> 	Menggambarkan interaksi antara aturan-aturan dan elemen-elemen lain dalam sebuah <i>use case</i> yang saling bekerja bersama untuk menciptakan tindakan yang lebih besar daripada jumlah dan kontribusi individu yang disebut sinergi
8	<i>Note</i>	Mengacu entitas fisik yang ada saat aplikasi sedang berjalan, penggunaan tersebut mewakili sumber daya komputasi.

No	Simbol	Keterangan
		
9	<i>System</i> 	Menggambarkan sebuah paket yang memberikan gambaran sistem dengan cakupan yang terbatas.
10	<i>Association</i> 	Menggambarkan koneksi antara satu objek dengan objek lainnya.

2.2.11. Activity Diagram

Activity diagram adalah suatu jenis diagram dalam *Unified Modeling Language* (UML) yang berfungsi untuk menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis. Diagram ini digunakan untuk memvisualisasikan serangkaian aktivitas atau tindakan yang dilakukan oleh sistem atau proses, menunjukkan bagaimana aktivitas tersebut saling berinteraksi dan diurutkan. *Activity Diagram* memberikan gambaran visual tentang proses yang terjadi di tingkat tinggi, dengan fokus pada aktivitas-aktivitas dan transisi antaraktivitas dalam sistem atau proses tersebut. Mirip dengan *flowchart*, diagram ini membantu pemodelan dan pemahaman terhadap jalannya suatu proses atau alur kerja. *Activity Diagram* sangat berguna dalam merancang, memodelkan, dan memahami proses bisnis atau operasional, serta memberikan pemahaman yang lebih baik tentang bagaimana sistem atau proses bekerja. Adapun keterangan lengkap mengenai *node* yang digunakan dalam *Activity diagram* terdapat pada table 2.3 dibawah ini[27]:

Tabel 2. 3 Node *Activity Diagram*[27]

No	Simbol	Keterangan
1	<i>Initial Node</i> 	Bagaimana objek suatu aktivitas pada sistem diawali.

No	Simbol	Keterangan
2	<i>Activity Final</i>  <i>Node</i>	Menggambarkan suatu aktivitas pada sistem dibentuk atau diakhiri.
3	<i>Activity</i> 	Menggambarkan interaksi antarmuka kelas yang saling terhubung.
4	<i>Action</i> 	Menunjukkan kondisi sistem yang mencerminkan implementasi suatu tindakan.
5	<i>Fork Node</i> 	Perubahan suatu aliran menjadi beberapa aliran pada tahapan tertentu.
6	<i>Decision/Merge</i> 	Merupakan suatu percabangan asosiasi, di mana terdapat penggabungan lebih dari satu aktivitas menjadi satu kesatu.

2.2.12. Sequence Diagram

Sequence Diagram adalah jenis diagram dalam *Unified Modeling Language* (UML) yang dirancang untuk menggambarkan proses yang dilakukan oleh pengguna atau objek dalam sistem informasi. Diagram ini fokus pada penggambaran interaksi antar objek berdasarkan urutan waktu, memberikan representasi visual tentang tahapan-tahapan proses yang dilakukan. *Sequence Diagram* menjelaskan bagaimana objek-objek berinteraksi satu sama lain dengan mengirimkan pesan-pesan, dan susunan diagram ini dibuat berdasarkan urutan waktu, memudahkan pemahaman mengenai urutan langkah-langkah atau tahapan proses. Diagram ini tidak hanya menjelaskan interaksi objek, tetapi juga dapat

digunakan sebagai alat untuk mendukung atau melengkapi *Use Case Diagram*. *Sequence Diagram* memungkinkan para pengembang perangkat lunak untuk memahami bagaimana fungsionalitas sistem atau *use case* diimplementasikan dalam tahapan-tahapan yang berurutan, sehingga membantu dalam merancang sistem yang sesuai dengan kebutuhan, Adapun keterangan lengkap mengenai *node* yang digunakan dalam *sequence diagram* terdapat pada table 2.4 dibawah ini[27]:

Tabel 2. 4 Node Sequence Diagram[27]

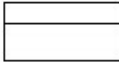
No	Simbol	Keterangan
1	<i>LifeLine</i> 	Menggambarkan suatu entitas objek saling berhubungan satu sama lain.
2	<i>Message</i> 	Menjelaskan interaksi antar objek meliputi informasi mengenai kegiatan yang sedang terjadi.
3	<i>Message</i> 	Menjelaskan rincian mengenai komunikasi antar objek-objek, termasuk informasi terkait kegiatan yang terjadi.

2.2.13. Class Diagram

Class Diagram merupakan jenis diagram dalam *Unified Modeling Language* (UML) yang berfungsi sebagai pemodelan struktur basis data dan kelas objek dalam suatu sistem. Diagram ini digunakan untuk mendefinisikan kelas-kelas yang membentuk basis data, menyajikan informasi mengenai atribut, metode, serta hubungan antar kelas. Setiap kelas direpresentasikan oleh suatu persegi panjang yang mencakup detail karakteristik kelas tersebut. Object *class* digunakan untuk merepresentasikan entitas atau konsep dalam sistem, dan setiap *class* memiliki

atribut sebagai property yang menggambarkan ciri-ciri kelas tersebut. *Class Diagram* membantu dalam menggambarkan alur data dan interaksi antar objek-objek melalui hubungan antar kelas seperti asosiasi, komposisi, atau generalisasi. Diagram ini penting untuk merencanakan, merancang, dan memahami struktur keseluruhan sistem dari segi definisi kelas-kelas yang dibuat untuk membangun sistem perangkat lunak, Berikut adalah Tabel 2.5 yang berisi keterangan dari *node* yang digunakan dalam *class diagram*[27]:

Tabel 2. 5 Node Class Diagram[27]

No	Simbol	Keterangan
1	<i>Class</i> 	Mengacu pada sekumpulan objek dengan karakteristik dan fungsi yang sama.
2	<i>Collaboration</i> 	Menggambarkan rangkaian tindakan yang dilaksanakan oleh sistem sehingga hasil dapat diukur oleh aktor yang berkaitan.
3	<i>Realization</i> 	Menggambarkan tindakan atau operasi yang dilakukan oleh objek.
4	<i>Nary Association</i> 	Merujuk pada usaha untuk menghindari hubungan asosiasi yang melibatkan lebih dari dua objek.
5	<i>Association</i> 	engilustrasikan keterkaitan antara satu objek dengan objek-objek yang lainnya.
6	<i>Dependency</i> 	Hubungan ini mengacu pada suatu perubahan dimana suatu unsur yang mandiri akan berpengaruh terhadap unsur yang tidak memiliki otonomi.
7	<i>Generalization</i> 	Menggambarkan objek anak mengadopsi perilaku dan struktur data yang diwarisi dari objek induk yang berada di atasnya.

2.2.14. *Blackbox Testing*

Black Box Testing merupakan teknik pengujian perangkat lunak yang berfokus pada spesifikasi fungsional tanpa memperhatikan struktur internal program. *Black Box Testing* dilakukan dengan penguji mendefinisikan kondisi masukan dan menguji spesifikasi fungsional program. Tujuan utama dari *Black Box Testing* adalah menunjukkan bagaimana fungsi perangkat lunak beroperasi, termasuk pengujian *input* dan *output* data, serta menjaga keakuratan informasi yang disimpan dan yang berasal dari sumber eksternal. *Black Box Testing* mengidentifikasi jenis kesalahan, seperti fungsi yang tidak benar atau tidak ada, kesalahan antarmuka, kesalahan pada struktur data dan akses basis data, kesalahan performansi, dan kesalahan inisialisasi maupun terminasi[5]. Penerapan *Black Box Testing* tidak hanya terfokus pada spesifikasi fungsional perangkat lunak, tetapi juga memberikan pemahaman yang mendalam tentang fungsi-fungsi tersebut berinteraksi, menjaga keakuratan informasi, dan mengidentifikasi berbagai jenis kesalahan potensial. Melalui pendekatan ini, penguji dapat mengoptimalkan kualitas perangkat lunak dengan memastikan bahwa fungsi-fungsi yang diuji beroperasi sesuai dengan harapan dan memenuhi standar keandalan yang diinginkan[5].

2.2.15. *Single Ease Question*

Single Ease Question (SEQ) adalah metode kuesioner yang terlibat dalam pengujian usability setelah pengguna menyelesaikan suatu tugas atau berinteraksi dengan produk atau sistem tertentu. Metode ini memungkinkan para fasilitator atau peneliti untuk dengan cepat memperoleh tanggapan dari responden mengenai sejauh mana responden menghadapi kesulitan atau menemui kemudahan dalam menyelesaikan tugas yang diberikan. SEQ digunakan karena memberikan

kemudahan dan efisiensi dalam mendapatkan umpan balik, memungkinkan evaluasi yang langsung dan kontekstual terhadap pengalaman pengguna pasca-interaksi. Pendekatan ini membantu mengidentifikasi secara lebih tepat masalah atau kesulitan spesifik yang mungkin timbul selama proses interaksi[6]. Berikut pada Gambar 2.2 adalah contoh kuisisioner dari metode *Single Ease Question*[28]:

Overall, how difficult or easy did you find this task?

Very Difficult	2	3	4	5	6	Very Easy
1						7
☐	☐	☐	☐	☐	☐	☐

Gambar 2. 2 Contoh kuisisioner *Single Ease Question*[28]

Skala penilaian pada SEQ memiliki rentang dari 1 hingga 7 poin, di mana nilai 1 menandakan bahwa tugas dianggap sangat sulit, sementara nilai 7 menunjukkan bahwa tugas dianggap sangat mudah. Fleksibilitas skala ini memungkinkan variasi dalam penilaian, memungkinkan responden memberikan nuansa yang lebih kaya terkait tingkat kesulitan atau kemudahan yang dialami responden. Setelah mendapatkan penilaian dari responden, interpretasi hasil SEQ dilakukan dengan mempertimbangkan nilai rata-rata. Jika nilai rata-rata SEQ ≤ 5 , hal ini menandakan bahwa tugas tersebut dianggap memiliki tingkat kemudahan yang rendah menurut persepsi responden. Sebaliknya, nilai rata-rata yang lebih tinggi menunjukkan bahwa tugas dianggap lebih mudah oleh responden. Pendekatan ini memberikan pemahaman mendalam tentang pengalaman pengguna dan dapat menjadi panduan berharga dalam mengembangkan produk atau sistem yang lebih sesuai dengan kebutuhan dan preferensi pengguna. Penggunaan *Single Ease Question* dalam pengujian usability memungkinkan pengumpulan tanggapan yang cepat dan langsung dari responden setelah menyelesaikan tugas dapat memberikan wawasan yang mendalam tentang tingkat kemudahan atau kesulitan dalam interaksi pengguna dengan suatu produk atau sistem. *Single Ease Question* tidak hanya mempercepat proses evaluasi usability, tetapi juga memungkinkan identifikasi masalah secara spesifik, mendukung pengembangan produk yang lebih

sesuai dengan kebutuhan dan preferensi pengguna[6]. Berikut adalah rumus rata rata yang digunakan untuk menghitung nilai *single ease question*[29]:

$$\tilde{x} = \frac{1}{n} \sum_{i=1}^n a_i \quad (2.1)$$

$\tilde{x}$: Mean

n : banyak data

a_i : data ke i

i : data ke 1 sampai data ke n