

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Penelitian mengenai situs *website* yang menawarkan sistem pemesanan tiket wisata sudah banyak dilakukan. Dalam penelitian ini, penulis memanfaatkan beberapa referensi dari penelitian sebelumnya sebagai panduan dan rujukan. Berikut penelitian terdahulu yang menurut penulis terdapat relasi dengan penelitian yang akan dilakukan.

Berdasarkan hasil penelitian yang dilakukan oleh Dasril Aldo, Widya Lelisa Army, dkk pada tahun 2022. Menerangkan bahwa memiliki permasalahan bagi wisatawan yakni minimnya promosi dan informasi pariwisata Kota Batam. Peneliti kemudian mengembangkan sistem informasi terpadu pariwisata yang dilengkapi dengan fitur estimasi biaya, daftar destinasi di setiap daerah tujuan wisata, menu kalender acara, serta halaman utama *website* yang menyajikan deskripsi tentang Kota Batam dan galeri gambar wisata. Sistem ini dirancang berbasis *website* menggunakan *framework CodeIgniter*, *PHP* sebagai bahasa pemrogramannya, dan *Mysql* untuk penyimpanan data serta *xampp* sebagai *server*. Dengan demikian, sistem ini diharapkan menjadi media promosi utama pariwisata kota batam, agar dapat bersaing dengan wisata di kota-kota lain[8].

Pada penelitian yang dilakukan oleh Beatriks Diana Teang, NM Faizah dan Widyat Nurcahyo pada tahun 2023. Teridentifikasi masalah bahwa belum ada informasi berbasis web tentang objek wisata, kuliner, dan penginapan di Kabupaten Nagekeo. Menanggapi masalah ini, peneliti mengembangkan sistem informasi yang memudahkan wisatawan dalam memperoleh informasi tentang objek wisata, kuliner, dan penginapan di sekitar lokasi wisata di Kabupaten Nagekeo. Aplikasi ini dikembangkan menggunakan metode *Rapid Application Development (RAD)* dan pemodelan menggunakan *Data Flow Diagram (DFD)*[9].

Pada penelitian yang dilakukan oleh Nur Fuad Azizi, Indra Lukmana Sadri, dan Aristyo Hadikusuma pada tahun 2023. Menerangkan bahwa permasalahan dalam penelitian ini meliputi kebutuhan akan solusi kurangnya panduan dan sulit

dalam menggambarkan desain interior sesuai keinginan masyarakat, yang ingin diatasi melalui pengembangan *website* Housset dengan fitur seperti furnitur *online*, template ruangan, dan konsultasi bersama ahli desainer interior. Selain itu memperkuat keberadaan sebuah merk dengan menampilkan halaman *front-end* yang baik untuk mendapatkan kepercayaan pengguna. Untuk mengevaluasi pengujian aplikasi *website*, kuesioner dibagikan kepada lima orang yang dipilih. Hasil skor SUS sebesar 83,5 dengan indeks "Sangat Baik". Dalam mempertimbangkan kriteria tersebut, dapat ditarik kesimpulan bahwa *website* pada aplikasi ini memenuhi persyaratan[5].

Pada penelitian yang dilakukan oleh Oky Septian pada tahun 2019, dijelaskan bahwa penggunaan internet sebagai sarana untuk memperoleh informasi semakin meningkat. Internet juga dapat digunakan sebagai media promosi dan publikasi informasi. Berdasarkan kemudahan dan manfaat yang ditawarkan oleh internet, dikembangkan sistem *E-ticket* untuk memudahkan pengunjung museum dalam memesan tiket. Selain itu, sistem ini juga memudahkan pengelola dalam membuat laporan penjualan dan laporan pengunjung melalui *website*[4].

Pada penelitian yang dilakukan oleh Agus Salim, Rafly Yusrizal Amrie pada tahun 2021. Diidentifikasi bahwa sering kali objek wisata kecil yang berpotensi tidak tercakup dalam kriteria pengenalan dan penampilan objek wisata serta elemen penunjang lainnya. Selain itu, pelanggan sering kesulitan menyampaikan kebutuhan secara rinci tanpa melihat gambar yang jelas. Menanggapi masalah ini, peneliti bertujuan untuk membuat situs pariwisata yang memfasilitasi semua aspek penunjang untuk kegiatan pariwisata dengan tampilan dan pengalaman baru saat mempersiapkan akomodasi wisata. Dalam pengembangannya, peneliti menggunakan model *prototipe*, yang menjabarkan kebutuhan pelanggan secara rinci. Dengan adanya *website* ini, diharapkan dapat menjadi tolak ukur bagi pelanggan dalam melihat pilihan yang disajikan, sehingga menarik minat pelanggan dengan menyajikan informasi yang akurat[10].

Tabel 2. 1 Penelitian Sebelumnya

| No | Profil Artikel | Permasalahan | Metode | Hasil |
|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Dasril Aldo, Widya Lelisa Army, dkk. Judul artikel "Pengembangan Sistem Informasi Terpadu Industri Pariwisata Kota Batam Menerapkan <i>User Centered Design</i> Berbasis <i>Website</i> "[8]. Tahun 2022. | Minimnya promosi dan informasi pariwisata Kota Batam menjadi kendala bagi wisatawan untuk mendapatkan informasi tersebut. | <i>User Centered Design</i> (UCD). | Penelitian dengan menggunakan pengembangan <i>Framework CodeIgnite</i> dari aplikasi ini dan mampu mengimplementasikan pengembangan sistem informasi pariwisata Kota Batam berbasis <i>website</i> . |
| 2 | Beatriks Diana Teang, NM Faizah dan Widyat Nurcahyo. Judul artikel "Perancangan Aplikasi Sistem Informasi Pariwisata berbasis <i>Web</i> Di Kabupaten Nagekeo Provinsi Nusa Tenggara Timur Dengan Metode <i>Location Based Service</i> (LBS)"[9]. Tahun 2023. | Belum adanya <i>website</i> yang menenkanakan informasi tentang objek wisata Nagekeo. | <i>Location Based Service</i> (LBS). | Aplikasi ini telah diuji dan wisatawan dapat dengan mudah dan cepat mendapatkan informasi tentang objek-objek wisata, kuliner, penginapan di sekitar Lokasi wisata yang ada di kabupaten Nagekeo. |
| 3 | Nur Fuad Azizi, Indra Lukmana Sadri, Aristyo Hadikusuma. Judul artikel "Rancang Bangun <i>Front-end</i> pada Aplikasi <i>Website</i> "Houset" dengan Menggunakan <i>User Centered Design</i> "[5]. Tahun 2023. | Belum adanya sebuah <i>website</i> yang digunakan <i>user</i> sebagai media memberikan bayangan visual mengenai rekomendasi desain interior dan furnitur yang disesuaikan dengan kebutuhan <i>user</i> . | <i>User Centered Design</i> (UCD). | Hasil dari perancangan sistem yang telah dibuat adalah rancangan halaman antarmuka aplikasi <i>website</i> Houset. |

| No | Profil Artikel | Permasalahan | Metode | Hasil |
|----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 4 | Oky Septian. Judul artikel “Analisa Perancangan Sistem Informasi Pemesanan Tiket Museum <i>Online</i> Di DKI Jakarta Berbasis <i>Web</i> ”[4]. Tahun 2019. | Belum sistem informasi pemesanan tiket museum <i>online</i> dan informasi seputar museum. | <i>Waterfall</i> | Pengunjung dapat melakukan pemesanan tiket melalui <i>website</i> dan dapat mengetahui informasi seputar museum, memudahkan laporan penjualan tiket dan laporan pengunjung untuk pengelola. |
| 5 | Agus Salim, Rafly Yusrizal Amrie. Judul artikel “Perancangan <i>FrontEnd</i> Aplikasi Pemandu Pariwisata Menggunakan <i>Framework React.js</i> Di Provinsi Jawa Barat”[10]. Tahun 2021. | Belum adanya <i>website</i> informasi publik dan Akomodasi Pariwisata. | <i>Prototipe</i> | Hasil dari penelitian ini, penggunaan <i>website</i> berjalan dengan baik karena informasi dan promosi yang di sampaikan kepada masyarakat sangat mudah digunakan dan informasi tentang pariwisata bisa diakses kapanpun dan dimanapun. |

2.2 Landasan Teori

2.2.1 Sistem

Sistem yakni rangkaian prosedur yang berkaitan dan berkolaborasi untuk mencapai tujuan tertentu[11].

2.2.2 Pemesanan

Pemesanan merujuk pada langkah-langkah, tindakan, atau cara dalam melakukan pesanan. Menurut KBBI, pemesanan merupakan tindakan, atau cara untuk meminta sesuatu (tempat, barang, dan sebagainya) kepada pihak lain. Secara global, pemesanan merupakan tindakan pembelian barang atau jasa sebelum barang atau jasa tersebut diterima[11].

2.2.3 Curug Pletuk

Curug Pletuk, terletak di Dukuh Pletuk, Pesangkalan, Kec. Pagedongan, Kab. Banjarnegara, Jawa Tengah, KPH atau Kesatuan Pemangkuan Hutan Kedu Selatan memiliki objek wisata dengan pemandangan alam yang memukau. Curug Pletuk diresmikan oleh PERHUTANI Kedu Selatan pada tanggal 30 April 2021. Menurut Administratur KPH Kedu Selatan, Curug Pletuk berada di petak 9e dan 9f RPH Singomerto, BKPH Banjarnegara[3].

2.2.4 *Front-end*

Front-end adalah bagian dari sistem yang menyediakan layar kepada pengguna. Bertanggung jawab untuk mengembangkan komponen visual dalam sistem dan bertanggung jawab untuk menampilkan antarmuka. *Front-end* mengacu pada pengembang yang bekerja di bagian desain tata letak situs *web* atau aplikasi, dimana pengembang akan mengerjakan antarmuka pengguna untuk membuat desain terlihat lebih menarik[12].

2.2.5 Pariwisata

Pariwisata adalah suatu ekspedisi dalam bentuk perjalanan untuk tujuan liburan dan bersantai. Pada sektor pariwisata dapat menjadi salah satu industri pendapatan suatu daerah, Terutama dalam sektor jasa, yang meliputi layanan terkait industri pariwisata seperti hotel, restoran, transportasi, perbankan, keamanan, dan sebagainya. Oleh karena itu, perkembangan industri pariwisata menjadi salah satu rencana yang diadopsi

baik oleh pemerintah maupun organisasi non-pemerintah untuk meningkatkan perdagangan dengan menjual produk atau layanan serta memperkenalkan daerah tertentu sebagai destinasi wisata[1].

2.2.6 Website

Website adalah sebuah aplikasi yang memuat dokumen multimedia (teks, gambar, audio, animasi, video) menggunakan *Hypertext Transfer Protocol* (HTTP) dan diakses melalui perangkat lunak bernama *browser*[13]. Karakteristik utamanya meliputi keterkaitan, domain sebagai alamat *web*, serta server penyimpanan data besar. Situs *web* dapat diakses dalam mode *localhost*, yang memungkinkan pengembangan dan modifikasi tanpa koneksi internet. Bahasa pemrograman dasarnya adalah HTML (*Hypertext Markup Language*) dan CSS (*Cascading Style Sheet*)[14].

2.2.7 Perl Hypertext Preprocessor (PHP)

PHP merupakan bahasa pemrograman yang digunakan untuk membuat dan mengembangkan situs *web*, bekerja sama dengan HTML[15]. PHP merupakan bahasa skrip yang dijalankan di sisi *server* yang dapat memproses kode PHP dari kode *web* dengan ekstensi *.php*, yang menghasilkan tampilan situs *web* dinamis di sisi klien (*browser*). Dengan menggunakan skrip PHP, halaman HTML dapat berfungsi sebagai sistem pelengkap seperti portal *web*, *e-learning*, *e-library*, dan lainnya[16].

2.2.8 Laravel

Laravel, sebuah kerangka kerja untuk pengembangan *website* berbasis situs *web* dengan pola MVC, dirancang untuk mengurangi biaya pembuatan dan pemeliharaan perangkat lunak sambil meningkatkan kualitasnya. Dengan sintaksis yang bersih dan efisien, tujuan utamanya adalah meningkatkan produktivitas dan mengurangi waktu pengembangan. Ini adalah proyek sumber terbuka yang mengikuti prinsip MVC dan menggunakan PHP sebagai bahasa pemrogramannya[17].

Laravel menawarkan sejumlah fitur utama, seperti sistem *routing* yang fleksibel, integrasi manajemen otentikasi, dan kemampuan migrasi basis data yang memudahkan pengelolaan skema basis data. Salah satu keunggulan

utama *Laravel* adalah Eloquent, ORM bawaan yang memungkinkan pengembang berinteraksi dengan basis data menggunakan sintaksis PHP yang jelas dan mudah dipahami. *Blade*, sebagai mesin templat *Laravel*, juga mempermudah pengembangan tampilan dengan sintaksis yang mudah dipahami. *Framework* ini juga menyediakan alat-alat untuk pengelolaan antrian, sistem notifikasi, dan caching, semua dirancang untuk meningkatkan kecepatan dan mengoptimalkan kinerja aplikasi *web*. Dengan ekosistem paket tambahan yang kuat dan dukungan terus-menerus dari komunitas pengembang yang aktif, *Laravel* ialah pilihan yang sangat tepat bagi siapa saja yang ingin mengembangkan aplikasi *web* dengan cepat dan efisien[18].

2.2.9 Hypertext Markup Language (HTML)

HTML merupakan bahasa yang digunakan untuk membuat halaman *web* dan aplikasi *web*. HTML diinterpretasikan oleh *browser* agar halaman dapat ditampilkan dan muncul pada posisinya. HTML digunakan untuk membangun struktur dasar halaman *web* dan memberikan dasar awal untuk membangun kerangka yang lebih terstruktur sebelum memasuki tahap desain dan fungsionalitas[19].

2.2.10 Cascading Style Sheets (CSS)

CSS merupakan bahasa yang digunakan untuk menentukan tata letak dan gaya halaman pada sebuah situs *web*. Dengan CSS, kita dapat merapikan jenis font, warna teks, serta latar belakang halaman. CSS bekerja sama dengan bahasa markup seperti HTML dan XML untuk menciptakan situs *web* yang menarik dan fungsional. CSS membantu mengatasi keterbatasan HTML dalam memformat halaman *web*. Jika menggunakan HTML saja untuk membuat *website* dengan banyak halaman, setiap halaman wajib ditulisi tag elemen HTML yang sama. Dengan CSS, cukup menulis kode sekali agar elemen HTML dapat diimplementasikan ke semua halaman. Jika ada perubahan yang perlu dilakukan, cukup dirubah satu kode tersebut. Secara global, CSS berfungsi untuk mengatur tampilan halaman *web* berbasis HTML atau bahasa markup lainnya[20].

2.2.11 *JavaScript*

JavaScript merupakan bahasa pemrograman yang digunakan oleh pengembang untuk membangun situs *web* yang interaktif. Mulai dari memperbarui umpan sosial media serta menampilkan animasi dan peta interaktif, fungsi lain dari *JavaScript* yaitu mampu mengoptimalkan pengalaman pengguna di situs *web*. Sebagai bahasa skrip dari sisi klien, *JavaScript* merupakan salah satu teknologi utama dalam *World Wide Web* atau biasa disebut WWW. Contohnya, ketika menelusuri internet dan melihat fitur gambar, menu *drop-down* yang dapat diklik, atau perubahan warna elemen secara dinamis di halaman *web*, kita sedang menyaksikan hasil kerja *JavaScript*[20].

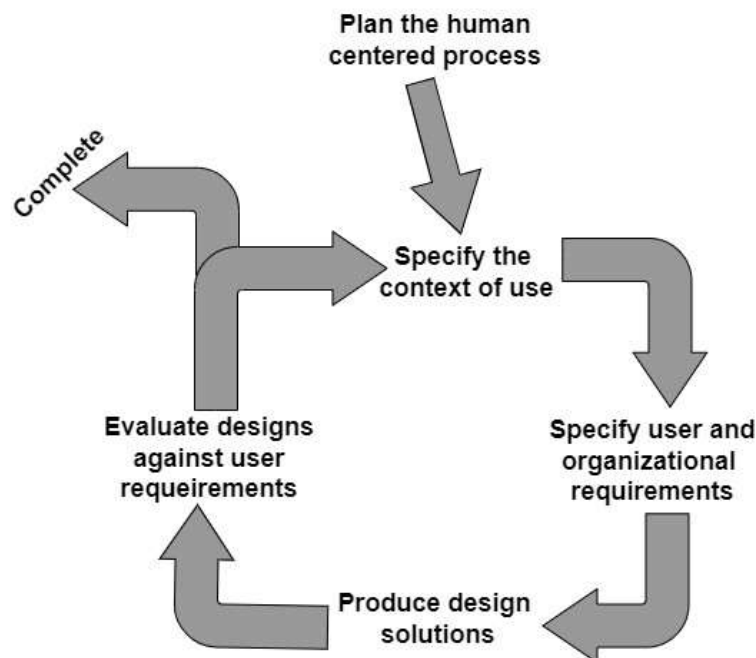
2.2.12 *Bootstrap*

Bootstrap ialah *framework* yang menggabungkan antara HTML, CSS, dan *JavaScript*, berfungsi untuk mempermudah dan mempercepat pembuatan antarmuka (*front-end*) sebuah situs *web* yang responsif. Dengan *Bootstrap*, tidak perlun membuat kode dari awal karena *framework* ini sudah menyediakan kumpulan file CSS dan *JavaScript* yang telah terstruktur dalam kelas-kelas yang dapat langsung digunakan[21]. Kemudahan dan kepraktisan yang ditawarkan oleh *Bootstrap* sangat berguna, terutama penggunaan *web* melalui perangkat *mobile* yang semakin meningkat. Oleh karena itu, pendekatan desain *web* responsif dan ramah *mobile* yang ditawarkan *Bootstrap* menjadi solusi praktis dalam pengembangan *website*[22].

2.2.13 *User Centered Design (UCD)*

User Centered Design atau biasa disebut UCD adalah konsep yang menitikberatkan pada objek, komunikasi, ruang, antarmuka, dan layanan guna memastikan kebutuhan pengguna terpenuhi. Pendekatan ini dilakukan untuk peningkatan sistem yang saling terhubung dan fokus pada penciptaan sistem yang bermanfaat. *User Centered Design* merupakan model terbaru dalam pengembangan aplikasi *web*, dengan strategi yang melibatkan

tinjauan literatur khusus dan menekankan interaksi berulang antara pengguna dan pengembang dalam domain tertentu[23].



Gambar 2. 1 Tahapan Metode *User Center Design*[14].

Pada gambar 2.1 menampilkan langkah-langkah pengembangan *website* menerapkan Metode *User Centered Design* (UCD), sebagai berikut:

1. *Specify the Context of Use* (Menentukan Konteks Pengguna)

Pada tahapan ini menentukan siapa saja yang akan menggunakan sistem[24].

2. *Specify User and Organisational Requirement* (Menentukan Spesifikasi Kebutuhan Pengguna)

Tahapan ini mengidentifikasi kebutuhan pengguna dengan wawancara dan observasi dapat dilakukan untuk mengumpulkan informasi mengenai persyaratan fungsional dan non-fungsional yang akan diterapkan pada aplikasi[24].

3. *Product Design Solutions* (Pembuatan Solusi Desain)

Tahap ini peneliti membuat desain antarmuka berdasarkan hasil analisis kebutuhan. Langkah ini dilakukan agar pengguna dapat lebih

memahami seperti apa produk dalam bentuk *wireframe*, yang menjelaskan gambaran umum dan pengoperasian sistem yang akan dibuat[24].

4. *Evaluate Design Against User Requirement* (Evaluasi Solusi Desain)

Setelah tahap perancangan desain solusi selesai, hasil desain tersebut diserahkan kepada calon pengguna untuk dievaluasi guna memastikan kesesuaiannya dengan kebutuhan mereka. Jika ternyata belum sesuai, perlu dilakukan perbaikan pada desain sistem berdasarkan masukan dari pengguna tersebut. Proses ini mungkin akan berulang sampai desain memenuhi kebutuhan pengguna[24].

2.2.14 *Unified Modeling Language* (UML)

Unified Modeling Language atau biasa disebut UML adalah salah satu bahasa yang digunakan dalam menciptakan representasi visual dari berbagai aspek dalam suatu sistem, termasuk alur kerja fungsi, urutan *query*, desain *database*, aplikasi, dan struktur sistem yang baru. Biasanya, UML mengadopsi prinsip-prinsip *Object Oriented Programming* (OOP) dan tidak terikat pada bahasa pemrograman spesifik. UML juga berfungsi sebagai panduan lengkap dan rinci, mirip dengan cetak biru, dalam proses membangun dan menerapkan sistem perangkat lunak[1]. Dalam konteks sistem penyusunan perangkat lunak, UML digunakan di beberapa diagram yang penting, diantaranya:

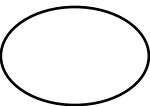

1. *Use Case Diagram*

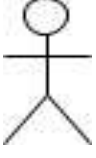
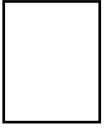


Use Case Diagram adalah representasi interaksi antara aktor dan sistem, menggunakan serangkaian diagram yang menggambarkan detail hubungan tersebut. Dalam membuat *Use Case Diagram* untuk sistem pemesanan tiket, teknik yang diterapkan bertujuan menangkap kebutuhan fungsional dari sistem tersebut. Secara ringkas, *Use Case Diagram* memberikan gambaran lengkap tentang interaksi antara aktor dan sistem atau perangkat lunak yang dikembangkan. Diagram ini menekankan elemen utama yang disebut "*Use Case*" yang mencerminkan aktivitas atau fungsi spesifik yang dapat dilakukan

oleh aktor dalam interaksinya dengan sistem. Selain itu, diagram ini juga menyoroti aktor-aktor yang terlibat dalam interaksi tersebut, yaitu satuan eksternal yang berinteraksi dengan sistem, seperti pengguna, sistem eksternal, atau satuan lainnya yang terlibat dalam penggunaan fungsi sistem. Garis yang menghubungkan aktor dengan *Use Case* menunjukkan keterkaitan antara keduanya.

Use Case Diagram memberikan pandangan umum tentang fungsionalitas sistem serta cara aktor berinteraksi dengan sistem tersebut. Ini membantu dalam memahami awal mengenai kebutuhan pengguna, fungsi sistem, dan kemungkinan skenario interaksi yang mungkin terjadi. Dengan menggunakan *Use Case Diagram*, pengembang perangkat lunak dapat membangun sistem yang sesuai dengan kebutuhan pengguna serta memastikan semua aspek fungsionalitas penting telah dipertimbangkan dalam perancangan. Selain itu, diagram ini memfasilitasi komunikasi antara tim pengembang dan pemangku kepentingan untuk memahami kasus penggunaan yang harus diakomodasi oleh sistem. Detail mengenai nama, simbol, dan fungsionalitas terdapat pada tabel 2.2.

Tabel 2. 2 Tabel *Use Case Diagram*


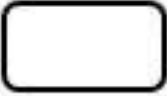
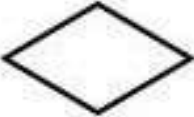


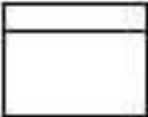
| No. | Simbol | Nama | Deskripsi |
|-----|-------------------------------------------------------------------------------------|-----------------|------------------------------------------------------------------------------------|
| 1. |  | <i>Use case</i> | Mewakili fungsi atau tindakan yang bisa dijalankan oleh sistem. |
| 2. |  | Asosiasi | Asosiasi berguna sebagai menggambarkan hubungan antara aktor dan <i>use case</i> . |

| | | | |
|----|----------------------------------------------------------------------------------------------------------------------|----------------|-----------------------------------------------------------------------------------------------|
| 3. |  | <i>Actors</i> | Merepresentasikan pengguna atau entitas eksternal lain yang berinteraksi dengan sistem. |
| 4. |  | <i>System</i> | Menampilkan Sistem yang sedang berjalan |
| 5. | <i><<include>></i>  | <i>Include</i> | Relationship antara <i>use case</i> yang memasukkan fungsionalitas dari <i>use case</i> lain. |
| 6. | <i><<extend>></i>  | <i>Extend</i> | Hubungan antara <i>use case</i> yang menambahkan fungsionalitas ke <i>use case</i> lain. |

2. Activity Diagram

Activity diagram atau yang disebut dengan Diagram Aktivitas adalah gambaran visual dari proses-proses dalam sebuah sistem. Diagram ini menampilkan susunan proses sistem secara vertikal dan perpanjangan dari *Use Case* dengan penekanan pada rangkaian aktivitas. Alur aktivitas ini dapat berupa rangkaian menu yang ada dalam sistem tersebut[22]. *Activity diagram* membantu tim pengembang atau pihak yang terlibat untuk memahami secara visual bagaimana alur kerja dan interaksi terjadi dalam sistem atau proses yang sedang dikembangkan. Informasi lengkap mengenai nama, simbol, serta fungsinya dapat ditemukan dalam tabel 2.3.

Tabel 2. 3 Tabel *Activity Diagram*

| Simbol | Nama | Keterangan |
|-------------------------------------------------------------------------------------|---------------------|---------------------------------------------------------------------------------------------------------------------------------|
|  | <i>Initial Node</i> | Merupakan awal dari suatu kegiatan.aktivitas. |
|  | Aktivitas | Umumnya dimulai dengan kata kerja, menggambarkan kegiatan yang sedang dilakukan oleh sistem. |
|  | <i>Decision</i> | Menjelaskan terjadinya percabangan atau adanya pilihan aktivitas lebih dari satu. |
|  | Penggabungan | Penggabungan dimana lebih dari satu kegiatan akan digabungkan menjadi satu. |
|  | Status Akhir | Mengakhiri alur sistem. |
|  | <i>Swimlane</i> | Swimlane merujuk pada siapa yang bertanggung jawab dalam melakukan kegiatan dalam suatu diagram atau kegiatan yang berlangsung. |

2.2.15 Rumus Slovin

Rumus Slovin digunakan untuk menghitung jumlah sampel atau responden dari populasi yang jumlahnya sudah diketahui[25]. Penggunaan Rumus Slovin bertujuan untuk menghindari kesalahan dalam ukuran sampel

yang bisa terlalu kecil atau terlalu besar. Di bawah ini adalah persamaan dari Rumus Slovin (2.1) yang digunakan untuk menghitung sampel dalam suatu penelitian[26]:

$$n = \frac{N}{1 + Ne^2} \quad (2.1)$$

Keterangan:

n = ukuran sampel/jumlah responden

N = Jumlah populasi

e = tingkat kesalahan yang ditentukan

Untuk menerapkan rumus ini, pertama-tama perlu ditentukan batas toleransi kesalahan yang dinyatakan dalam bentuk persentase. Ketepatan sampel populasi ditentukan oleh toleransi kesalahan tersebut. Sebagai contoh, penelitian dengan batas kesalahan 10% (taraf signifikansi 10%) berarti memiliki tingkat ketepatan 90% (taraf kepercayaan 90%).

2.2.16 System Usability Scale (SUS)

Usability diukur untuk menilai manfaat produk menggunakan metode *System Usability Scale* (SUS). SUS adalah metode evaluasi yang melibatkan pengguna akhir dalam proses pengujian dan perhitungan yang lebih kompleks. SUS menggunakan kuesioner yang terdiri dari 10 pertanyaan, seperti yang tercantum tabel 2.4. Kuesioner SUS terdiri dari 2 jenis pertanyaan, yaitu pertanyaan bernomor ganjil dan pertanyaan bernomor genap, yang digunakan oleh pengguna untuk mengevaluasi manfaat produk tersebut dapat dilihat pada tabel 2.5[14].

Tabel 2. 4 Daftar Pertanyaan SUS

| No | Pernyataan SUS |
|----|-------------------------------------------------------------------------------------|
| 1 | Saya berpikir akan menggunakan sistem ini lagi. |
| 2 | Saya merasa sistem ini rumit untuk digunakan. |
| 3 | Saya merasa sistem ini mudah digunakan. |
| 4 | Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini. |

| | |
|----|-------------------------------------------------------------------------------------|
| 5 | Saya merasa fitur-fitur pada sistem ini berjalan dengan semestinya. |
| 6 | Saya merasa ada banyak hal yang tidak konsisten (tidak serasi pada sistem ini). |
| 7 | Saya merasa orang lain dapat memahami cara menggunakan sistem ini dengan cepat. |
| 8 | Saya merasa sistem ini sangat membingungkan atau rumit. |
| 9 | Saya merasa tidak ada hambatan dalam menggunakan sistem ini. |
| 10 | Saya perlu membiasakan diri atau belajar banyak hal sebelum menggunakan sistem ini. |

Tabel 2. 5 Jawaban Pernyataan SUS

| Jawaban | Skor Nilai |
|---------------------------|------------|
| STS (Sangat Tidak Setuju) | 1 |
| TS (Tidak Setuju) | 2 |
| N (Netral) | 3 |
| S (Setuju) | 4 |
| SS (Sangat Setuju) | 5 |

Metode SUS dipakai untuk menilai kegunaan produk karena tidak seperti kuesioner lainnya yang validitasnya diuji dengan sampel kecil, SUS menawarkan beberapa kelebihan[14], yaitu:

1. SUS memberikan nilai dari 0 hingga 100, sehingga mudah digunakan.
2. SUS sederhana dan mudah dihitung.
3. SUS tersedia secara gratis, tanpa biaya tambahan.
4. SUS terbukti valid dan realiable, meskipun menggunakan sampel kecil.

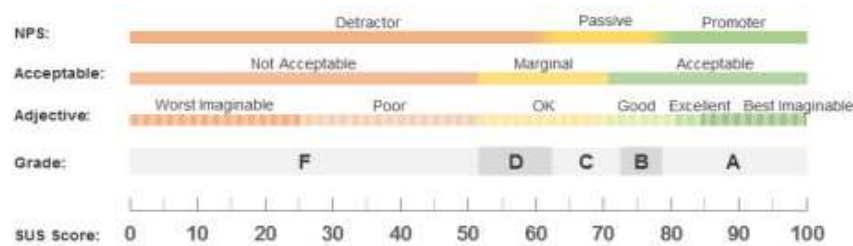
Untuk perhitungan skor, SUS menggunakan aturan-aturan berikut[14]:

1. Untuk pertanyaan bernomor ganjil (1,3,5,7,9) nilai responden dikurangi 1 ($X-1$).
2. Untuk pernyataan bernomor genap (2,4,6,8,10) dihitung dengan nilai 5 dikurang nilai responden ($5-X$).
3. Jumlah nilai responden kemudian kalikan dengan 2.5

4. Hitung rata-rata jawaban pernyataan.

Penilaian SUS bisa dilihat lebih jelas pada rumus perhitungan dilihat pada (2.2) merupakan rumus perhitungan SUS[14].

$$SUS = 2.5 \times [\sum_{n=1}^5 (U_{2n} - 1) + (5 - U_{2n})] \quad (2.2)$$



Gambar 2. 2 Interpretasi SUS score[14].

Tabel menunjukkan lima pendekatan dalam menginterpretasi nilai skor SUS[14], yaitu:

1. *Percentiles Rank* (Persentil)

Persentil diperoleh dari konversi skor mentah menjadi skor SUS.

2. *Grade* (Peringkat)

Skor mentah SUS dikategorikan dalam peringkat A sampai F, dengan A sangat baik dan F sangat buruk

3. *Adjective* (Sifat)

Enam atribut diberi skor SUS mentah. Skor 81 atau lebih dinilai sangat baik, sementara skor 51 dianggap paling rendah.

4. *Acceptable* (Tingkat Penerimaan)

Skor SUS di atas 70 dianggap "dapat diterima," di bawah 50 "tidak dapat diterima," dan antara 50-70 "agak dapat diterima".

5. *Net Promotore Score* (NPS)

Mengukur kepuasan pengguna dan kemungkinan rekomendasi. Kategori Promoter dengan skor 9-10, Pasif dengan skor 7-8, dan Detraktor dengan skor 6 ke bawah.

Tabel 2. 6 Hasil Skor SUS[27].

| Tingkatan | SUS | Kriteria |
|-----------|---------|-----------|
| A | >80,3 | Excellent |
| B | 68-80,3 | Good |
| C | 68 | Ok/Fair |
| D | 51-68 | Poor |
| F | <51 | Worst |

Pada Tabel 2.6 kriteria metode SUS dijelaskan, jika diatas > 80,3 dianggap sangat bagus, sedangkan nilai < 51 dianggap kurang[27].

2.2.17 *Blackbox Testing*

Blackbox testing adalah suatu kegiatan menguji program yang dilakukan dengan mengamati hasil keluaran program yang diimplementasikan. Dengan kata lain, *blackbox testing* merupakan kegiatan untuk menguji fitur dan fungsi tombol yang terdapat dalam program yang dibuat [28]. Metode pengujian *blackbox* bertujuan untuk menunjukkan fungsionalitas atau struktur sistem[7]. Pengujian *blackbox* bertujuan untuk memperlihatkan bahwa input dan output data beroperasi sebagaimana yang diharapkan[29].

2.2.18 Teknik Analisis Deskriptif

Teknik Analisis Deskriptif merupakan metode statistik yang digunakan untuk menggambarkan data dengan menjelaskan dan membuat kesimpulan tentang kelompok data tersebut[30]. Analisis deskriptif dilakukan dengan menginterpretasikan nilai rata-rata dari setiap indikator variabel yang digunakan dalam pengujian fungsionalitas sistem[31]. Perhitungan untuk analisis deskriptif ini mengikuti rumus (2.3), serta menghasilkan evaluasi kelayakan dengan tabel yang disediakan sebagai berikut[32]:

$$\text{Persentasi Kelayakan} = \frac{\text{Skor Observasi}}{\text{Skor yang Diharapkan}} \times 100\% \quad (2.3)$$

Tabel 2. 7 Presentase Kelayakan[32].

| Presentase | Tingkat Kelayakan |
|-------------------|--------------------------|
| 80% - 100% | Sangat Layak |
| 61% - 80% | Layak |
| 41% - 60% | Cukup |
| 21% - 40% | Tidak Layak |
| $\leq 20\%$ | Sangat Tidak Layak |