

BAB II

TINJAUAN PUSTAKA

2.1 Kajian Pustaka

Referensi dari penelitian sebelumnya yang terkait dengan subjek penelitian ini diperlukan untuk memperkuat pengetahuan yang dikumpulkan dari penelitian ini. Berikut ini adalah kumpulan studi terkait. Studi kasus pertama, "Perancangan Sistem Informasi Penyewaan Bus Pariwisata Berbasis Java", berfokus pada PO Bus Pariwisata Buahbatu. Penelitian ini menggunakan metode Waterfall selama proses pengembangan sistem. Penelitian ini bertujuan untuk menggunakan teknologi informasi untuk meningkatkan efisiensi dan efektivitas administrasi PO Bus Pariwisata Buahbatu. Penelitian ini menghasilkan sistem informasi yang memiliki fitur utama seperti laporan data bus, penyewa, dan penyewa. Studi ini menemukan bahwa sistem informasi penyewaan bus pariwisata dapat membantu pengelola mendaftar penyewa bus, memberikan informasi penyewaan yang lengkap dan dapat diakses saat dibutuhkan, dan memudahkan pelanggan melakukan transaksi[7].

Kedua, penelitian yang berjudul "Perancangan Sistem Informasi Penyewaan Bus Pariwisata Pada PO. Rejeki Gemilang Berbasis Web". Penelitian ini membahas bagaimana menggunakan teknologi internet untuk menyelesaikan masalah pembelian.. Rejeki Gemilang menunjukkan bahwa ada kesalahan dalam pengisian data penyewa bus yang disebabkan oleh pencatatan yang masih menggunakan buku. Penelitian ini menggunakan metode Waterfall selama proses pengembangan sistem. Analisis kebutuhan, desain, pengkodean, pengujian unit, dan pendukung adalah tahapan pengembangan penelitian ini. Penelitian ini menghasilkan rancangan yang terdiri dari diagram UML dan interface yang mencakup halaman bus, halaman tujuan, halaman login admin, halaman home admin, dan halaman home user.[2].

Ketiga, penelitian yang berjudul “Aplikasi Pemesanan Tiket Bus Pada PO. Puspa Jaya Berbasis Android”. Penelitian ini membahas mengenai pembangunan aplikasi pemesanan tiket bus secara online yang berbasis android pada PO. Puspa Jaya untuk mengatasi permasalahan yang terjadi seperti adanya antrian yang menumpuk pada outlet dikarenakan pemesanan tiket masih dilakukan secara manual. Selama proses pengembangan, penelitian ini menggunakan Metode *Agile* dengan jenis *Extreme Programming*. Metode ini dipilih karena sangat baik untuk pengembangan perangkat lunak berskala kecil dan cepat. Penelitian ini menghasilkan sebuah aplikasi pemesanan tiket bus secara online.

Keempat, penelitian yang berjudul “Implementasi Metode *Agile* Untuk Perancangan Sistem Informasi Administrasi Akademik”. Penelitian ini membahas mengenai penerapan metode *Agile* pada proses pengembangan sistem informasi administrasi akademik di Universitas Janabadra. Pada penelitian ini, Sistem informasi administrasi akademik dibutuhkan untuk mengatasi beberapa permasalahan yang terjadi di Biro Administrasi Akademik pada Universitas Janabadra yaitu kesalahan pencatatan dikarenakan proses administrasi masih belum bisa dilakukan secara daring. Penelitian ini menjelaskan bahwa metode *Agile* dipilih karena mampu memperbaiki ketidaksesuaian sistem yang lama dengan kebutuhan saat ini. Identifikasi kebutuhan, pembuatan model kerja atau prototype, pengujian pengguna, identifikasi kebutuhan baru untuk perbaikan sistem, dan operasional sistem adalah tahapan penelitian. Penelitian ini menghasilkan sebuah antarmuka untuk sistem informasi administrasi akademik[15].

Kelima, penelitian yang berjudul “Sistem Informasi Kepegawaian Menggunakan Metode *Agile Development* di CV. Angkasa Raya”. Penelitian ini membahas mengenai adanya permasalahan pada pengolahan data pada CV. Angkasa Raya berupa lamanya waktu pengolahan data pegawai berupa gaji dan cuti. Untuk meningkatkan proses pengolahan data pegawai agar lebih efisien, penulis menerapkan teknologi berupa sistem informasi untuk membantu menyelesaikan permasalahan yang terjadi pada CV. Angkasa Raya. Dalam proses pengembangan aplikasi, penulis menggunakan metode *Agile Development* yang terdiri dari

beberapa langkah seperti analisis sistem, perancangan, *development*, *testing*, *deploy* aplikasi, revisi dan evaluasi. Hasil dari penelitian ini adalah aplikasi sistem informasi kepegawaian yang memiliki fitur diantaranya pengolahan data penggajian, pengajuan cuti dan informasi karyawan[16].

Tabel 2. 1 Tabel Penelitian Terdahulu

| NO | Penulis | Tahun | Judul | Metode | Hasil |
|----|---|-------|---|--|---|
| 1 | A.Hekmatiar dan F.Erlangga | 2020 | Perancangan Sistem Informasi Penyewaan Bus Pariwisata Berbasis Java | <i>Waterfall</i> | Menghasilkan sebuah sistem informasi yang terdiri dari fitur utama seperti laporan data penyewa, laporan data penyewaan, dan laporan data bus |
| 2 | W. Chaerul Bachri | 2021 | Perancangan Sistem Informasi Penyewaan Bus Pariwisata Pada PO. Rejeki Gemilang Berbasis Web | <i>Waterfall</i> | Menghasilkan sebuah rancangan berupa diagram <i>UML</i> dan <i>interface</i> yang terdiri dari halaman login admin, halaman home admin, halaman bus, halaman tujuan, halaman login user dan halaman home user |
| 3 | Y.Anestasya, A.Hafiz, H.Setiawan, dan A.Komarudin | 2021 | Aplikasi Pemesanan Tiket Bus Pada PO. Puspa Jaya Berbasis Android | <i>Agile</i> berjenis <i>Extreme Programming</i> | Menghasilkan sebuah aplikasi pemesanan tiket bus secara online yang memiliki fungsi untuk memesan tiket, mencetak tiket, menampilkan laporan konfirmasi dan laporan transaksi |

| | | | | | |
|---|---------------------------------|------|--|-----------------------------------|--|
| 4 | F.Fitriastuti dan T.Krisdiyanto | 2022 | Implementasi Metode <i>Agile</i> Untuk Perancangan Sistem Informasi Administrasi Akademik | <i>Agile Software Development</i> | Menghasilkan sebuah antarmuka sistem informasi administrasi akademik yang berfungsi untuk menangani administrasi seperti pengajuan pindah prodi, pindah kelas dan undur diri |
| 5 | Suhari , A.Faqih, F M Basysyar | 2022 | Sistem Informasi Kepegawaian Menggunakan Metode <i>Agile Development</i> di CV. Angkasa raya | <i>Agile Development</i> | Menghasilkan sebuah aplikasi sistem informasi kepegawaian yang telah diuji menggunakan metode black box yang menunjukkan bahwa aplikasi telah berjalan secara fungsional |

2.2 Landasan Teori

2.2.1 Pengertian Sistem Informasi

Sistem informasi terdiri dari jaringan SPD (Data Processing System) dengan saluran komunikasi yang meningkatkan keuntungannya pada sistem organisasi data. Pengumpulan, pemrosesan, dan distribusi data adalah semua komponen dari proses sistem informasi.[17]. Sistem informasi memiliki elemen – elemen seperti *software*, manusia, prosedur, *hardware*, *database*, komunikasi data dan jaringan komputer[18]. Dengan mempertimbangkan hal-hal di atas, Sistem informasi terdiri dari komponen fisik dan prosedur yang saling terkait yang digunakan untuk mengatur data.

2.2.2 Pengertian Website

Salah satu layanan Internet yang memungkinkan pengguna terhubung satu sama lain dan menampilkan halaman web yang mereka cari adalah World Wide Web (WWW), kadang-kadang disebut sebagai "web". Dengan menggunakan browser web seperti Chrome, Firefox, dan Internet Explorer, pengguna dapat melihat halaman web yang mereka cari.[19]. Situs web, dalam arti lain, adalah kumpulan banyak halaman web yang mencakup aset pendukung seperti file video, foto, dan file digital lainnya yang disimpan di server web yang terhubung ke internet[20].

Website dapat bersifat dinamis ataupun statis. Website dinamis merupakan website yang terdiri dari konten yang sudah termanajemen, atau dengan kata lain end-user dapat sewaktu-waktu meng-update isi konten tanpa perlu merubah *syntax* pada codingan website[21]. Namun, website statis adalah yang memiliki konten yang jarang berubah, seperti website profil perusahaan[21].

2.2.3 *Framework Laravel*

Untuk menggunakan kerangka kerja, programmer harus memahami aturan yang ada. Kerangka kerja adalah kumpulan komponen, atau komponen program, yang memiliki tujuan tertentu untuk melakukan tugas tertentu, sehingga kode dapat ditulis secara efektif dan efisien.[22]. Laravel adalah framework open-source yang menggunakan bahasa pemrograman PHP untuk membuat aplikasi berpola MVC (Model, View, dan Controller). Routing berfungsi sebagai jembatan antara permintaan pengontrol dan pengguna[23]. MVC adalah pendekatan untuk proses pengembangan software yang memisahkan database, kode interface, dan logika pemrograman. [22].

2.2.4 *Hypertext Preprocessor (PHP)*

PHP adalah bahasa komputer yang banyak digunakan dalam proses konstruksi dan pengembangan situs web berbasis HTML. PHP, juga dikenal sebagai Preprocessor Hypertext, adalah bahasa yang digunakan untuk halaman HTML yang dijalankan di server. Sintax dan instruksi diproses sepenuhnya di server tetapi dimasukkan ke dalam teks HTML konvensional sehingga skrip tidak terlihat bagi pengguna. PHP dirancang untuk berinteraksi dengan server database dan membuat akses database lebih mudah saat membangun halaman HTML[24]. Sistem operasi PHP berjalan dengan browser dan server. PHP melalui browser meminta halaman web melalui server. Server merespon dengan mengunggah file yang diminta[25].

Menurut tim EMS, salah satu keuntungan dari PHP adalah dapat dengan mudah terhubung ke database. PHP dapat secara natif menangani sejumlah database tanpa perlu konektor, seperti bahasa pemrograman Java, membuatnya sangat dapat beradaptasi dalam menangani banyak jenis database; yang paling umum digunakan dalam PHP adalah MySQL. Untuk menghubungkan PHP ke database, Anda hanya perlu mengetahui nama database, lokasi, nama pengguna, dan kata sandi[26].

2.2.5 Database

Database adalah sekumpulan data yang tersimpan pada perangkat komputer untuk dapat dikelola seperti diakses, diperbaharui dan diorganisir dengan menggunakan *Database Manajemen System (DBMS)*. Salah satu model yang paling terkenal pada database yaitu relation data model yang memungkinkan sebuah data akan disimpan pada tabel yang memiliki value dan berelasi satu sama lain. Salah satu keunggulan database ialah kemudahan dalam proses identifikasi data dimana *database* akan menyajikan data yang diminta bersamaan dengan data – data yang memiliki keterkaitan melalui *database managemen system*[27]. Dari segi konsep database merupakan sekumpulan data yang membentuk sebuah file yang memiliki keterkaitan satu sama lain dengan aturan tertentu dan membentuk sebuah data atau informasi baru. Relasi pada tabel di database biasanya ditandai dengan kunci (key) dari setiap file[28].

2.2.6 Hypertext Markup Language (HTML)

HTML merupakan sebuah standar bahasa dalam penyajian konten pada halaman di website. *HTML* memiliki beberapa fungsi diantaranya dapat mendesain konten dari tampilan yang ada pada halaman website, mampu membuat sebuah tabel pada halaman *website*, melakukan publikasi secara online pada sebuah halaman website, mampu membuat wadah untuk melakukan *input*, *registrasi* dan transaksi seperti sebuah form pada halaman website serta dapat menyajikan gambar pada halaman *website*. Perintah pada *HTML* biasa disebut dengan *Tag HTML*. Tag berguna sebagai penentu tampilan yang berasal dari dokumen *HTML*. *Tag HTML* memiliki fungsi untuk menandakan bahwa konten pada berkas merupakan sebuah dokumen[29].

2.2.7 Cascading Style Sheet (CSS)





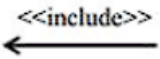
Cascading Style Sheet atau disingkat *CSS* sering digunakan untuk membuat sebuah web karena kemudahannya dalam menyeragamkan format

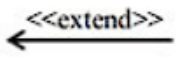
elemen – elemen yang memiliki secara cepat. Penempatan *file CSS* dapat dilakukan diluar atau terpisah dengan dokumen *HTML* dan perubahan yang terjadi pada *file CSS* dapat mempengaruhi semua dokumen *HTML* yang memiliki relasi dengan file tersebut. Oleh karena itu *CSS* dapat mempermudah perubahan yang terjadi pada website yang memiliki banyak halaman[30].

2.2.8 Use Case Diagram

Use case menceritakan bagaimana sebuah sistem digunakan oleh pengguna dan menjelaskan fungsinya. Urutan langkah: Setiap scenario menggambarkan urutan kejadian yang terjadi antara pengguna dan sistem. Setiap urutan dimulai oleh individu, sistem yang berbeda, perangkat lunak, atau urutan waktu[31].

Tabel 2. 2 Simbol *Use Case Diagram*





| Simbol | Keterangan |
|---|---|
|  | Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i> |
|  | <i>Use Case</i> : Abstraksi dan interaksi antara sistem dan aktor |
|  | <i>Association</i> : Abstraksi penghubung antara aktor dengan <i>use case</i> |
|  | Generalisasi : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i> |
|  | Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya. |


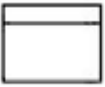
| | |
|---|---|
|  | <p>Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainya jika suatu kondisi terpenuhi</p> |
|---|---|

2.2.9 Activity Diagram

Activity Diagram, bagian penting dari UML, menunjukkan elemen dinamis sistem. Aktivitas diagram dapat dengan mudah digunakan untuk menjelaskan logika prosedural, proses bisnis, dan aliran kerja suatu perusahaan. *Activity diagram* memiliki tujuan yang sama seperti *flowchart*, tetapi mereka bisa mendukung perilaku paralel sementara *flowchart* tidak bisa. Dengan menunjukkan aliran pesan yang mengalir dari satu aktifitas ke aktifitas lainnya, *activity diagram* dirancang dengan tujuan untuk mengidentifikasi tingkah laku dinamis dari sistem[31].

Tabel 2. 3 Simbol *Activity Diagram*

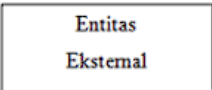
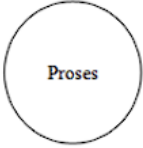
| Simbol | Nama | Keterangan |
|---|---------------------------------|--|
|  | Status Awal | Sebuah diagram aktivitas memiliki status awal |
|  | Aktivitas | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja |
|  | Percabangan/ <i>Decision</i> | Percabangan dimana ada pilihan aktivitas yang lebih dari satu |
|  | Penggabungan / <i>Join</i> | Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu |

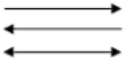
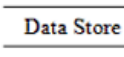
| | | |
|---|--------------|--|
|  | Status Akhir | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir |
|  | Swimlane | Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi |

2.2.10 Data Flow Diagram

Diagram Alir Data, juga dikenal sebagai *Data Flow Diagram* (DFD), adalah alat untuk pembuatan model yang menunjukkan sistem sebagai suatu jaringan proses fungsional yang terhubung satu sama lain melalui alur data, baik secara komputersasi maupun manual. *Data Flow Diagram* memiliki dua tingkat perancangan atau penggambaran sistem: tingkat 0 menunjukkan sistem secara keseluruhan, yang kemudian dipecah menjadi bagian-bagian yang lebih kecil. Hasil pemecahan pada tingkat 0 ini akan menjadi proses DFD tingkat 1, yang digunakan untuk menjelaskan proses dan aliran data yang terlihat dalam tingkat 0[32].

Tabel 2.4 Simbol *Data Flow Diagram*

| Simbol | Keterangan |
|---|---|
|  | Entitas eksternal dapat berupa orang/unit terkait yang berinteraksi dengan sistem tetapi di luar sistem |
|  | Orang/unit yang mempergunakan atau melakukan transformasi data. Komponen fisik tidak diidentifikasi |

| | |
|--|--|
| <p style="text-align: center;">Aliran Data</p>  | <p>Aliran data dengan arah khusus dari sumber ke tujuan</p> |
| <p style="text-align: center;">Data Store</p>  | <p>Penyimpanan data atau tempat data dilihat oleh proses</p> |

2.2.11 Agile Software Development

Agile Software Development adalah metode pendekatan perangkat lunak lebih lanjut dari SDLC (*Software Development Life Cycle*) yang berguna untuk membantu menyediakan fasilitas bagi pengembangan aplikasi berjangka waktu pendek serta meningkatkan keberhasilan perangkat lunak menjadi lebih baik dari metode desain terstruktur[33]. Metode *Agile* merupakan metodologi pengembangan perangkat lunak bersifat *iterative* yang berarti jika terjadi perubahan pada alur pengembangan maka akan dilakukan perulangan dengan tidak menunggu proses terselesaikan lebih dahulu[34]. *Agile Software Development* mempunyai empat nilai inti sebagai berikut :

- 1) Interaksi antar individu lebih utama dibandingkan dengan proses dan alat
 Nilai pertama memiliki makna bahwa pengembangan perangkat lunak akan sulit mencapai kesuksesan ketika tidak adanya kerjasama antara tim pengembangan dengan pemangku kepentingan atau klien. Komunikasi menjadi alat paling penting untuk dapat mencapai pengembangan yang berjalan dengan baik.
- 2) Proses pengembangan perangkat lunak lebih utama dibandingkan dengan dokumentasi. Dokumentasi memang menjadi salah satu alat berupa rincian proses pengembangan yang dapat membantu tim pengembang untuk mengembangkan

sebuah fitur. Namun perangkat lunak yang dapat berjalan sesuai dengan rencana yang telah dibuat jauh lebih utama.

- 3) Kolaborasi kepada klien lebih utama dibandingkan dengan kontrak

Produk yang dibangun oleh tim pengembang terkadang berbeda dengan apa yang telah tertulis pada kontrak. Kolaborasi dengan klien menjadi solusi karena dengan adanya kolaborasi, tim pengembang dapat menyesuaikan pengembangan produk terhadap kebutuhan klien dan apa yang tertulis di kontrak.

- 4) Menanggapi perubahan lebih utama dibandingkan dengan mengikuti rencana

Perubahan pada produk akan terus terjadi jika klien merasa belum terpuaskan dengan hasil pengembangan yang dilakukan oleh tim pengembang. Oleh karena itu tim pengembang dituntut untuk responsive terhadap perubahan yang terjadi[35].

2.2.12 *System Usability Scale (SUS)*

SUS atau *System Usability Scale*, adalah teknik untuk mengukur user-friendliness yang terdiri dari 10 pertanyaan yang dikemas dalam kuesioner. Namun, versi yang diperpanjang dapat dibangun dalam bentuk gambar Pictorial-SUS. Penilaian user-friendliness menggunakan skala likert lima poin dengan tanggapan yang berkisar dari "sangat tidak setuju" hingga "dengan tegas setuju." Skornya berkisar dari 0 hingga 100. Studi usability jenis summative biasanya menggunakan 50 hingga 100 responden untuk setiap grup pengguna. Namun, dalam situasi darurat atau penting, jumlah peserta dapat dikurangi menjadi 30 orang[13]. Tabel 2.2 menggambarkan 10 item pertanyaan dari survei *Usability Scale System*[36].

Tabel 2. 5 Tabel Pertanyaan *SUS Questionnaire*

| No | Pertanyaan | <i>Strongly Disagree</i> | | | | <i>Strongly Agree</i> |
|----|--|--------------------------|---|---|---|-----------------------|
| 1 | Saya pikir saya ingin menggunakan web ini | 1 | 2 | 3 | 4 | 5 |
| 2 | Saya menemukan web ini tidak perlu rumit | 1 | 2 | 3 | 4 | 5 |
| 3 | Saya pikir web ini mudah digunakan | 1 | 2 | 3 | 4 | 5 |
| 4 | Saya pikir saya perlu bantuan orang teknis dalam menggunakan web ini | 1 | 2 | 3 | 4 | 5 |
| 5 | Saya menemukan berbagai fungsi web ini terintegrasi dengan baik | 1 | 2 | 3 | 4 | 5 |
| 6 | Saya berpikir terdapat banyak ketidak konsistenan dalam web ini | 1 | 2 | 3 | 4 | 5 |
| 7 | Saya membayangkan bahwa kebanyakan orang akan belajar dengan mudah dalam mempelajari web ini | 1 | 2 | 3 | 4 | 5 |
| 8 | Saya menemukan bahwa web ini tidak praktis | 1 | 2 | 3 | 4 | 5 |
| 9 | Saya merasa sangat percaya diri saat menggunakan web ini | 1 | 2 | 3 | 4 | 5 |
| 10 | Saya perlu banyak belajar sebelum menggunakan web ini | 1 | 2 | 3 | 4 | 5 |

Dalam perhitungan skor pada metode *SUS*, ada beberapa aturan yang perlu diperhatikan, antara lain :

- 1) Pada pertanyaan yang bernomor ganjil, skor pada setiap pertanyaan yang didapat dari pengguna dikurang 1.
- 2) Pada pertanyaan yang bernomor genap, skor yang didapat dari nilai 5 dikurangi dengan skor pertanyaan yang didapat dari pengguna.
- 3) Skor *SUS* dihasilkan dari penjumlahan skor setiap pertanyaan kemudian dikali dengan 2,5

Rumus perhitungan skor *SUS* adalah sebagai berikut :

Skor *SUS* =

$$(R1 - 1) + (5 - R2) + (R3 - 1) + (5 - R4) + (R5 - 1) + (5 - R6) + (R7 - 1) + (5 - R8) + (R9 - 1) + (5 - R10) \times 2.5$$

Keterangan : $R = \text{Responden}$

Langkah selanjutnya adalah mencari rata – rata dari masing – masing skor *SUS* yang didapatkan dibagi dengan jumlah responden.

Berikut merupakan rumus perhitungan rata – rata skor *SUS* :

$$\bar{R} = \frac{\sum R}{n} \quad (2.1)$$

Keterangan : $\bar{R} = \text{skor rata rata}$

$\sum R = \text{jumlah skor } SUS$

$n = \text{jumlah responden}$

Ranking *SUS* dibagi menjadi lima tingkat, yaitu A, B, C, D, dan F. Setiap tingkat memiliki pilihan rating Excellent, Good, Ok, Poor, dan Awful. Di sini detailnya :

Tabel 2. 6 Pembobotan Skor *SUS Questionnaire*

| Skor | <i>Letter Grade</i> | <i>Adjective Rating</i> |
|-------------|----------------------------|--------------------------------|
| > 80.3 | A | <i>Excellent</i> |
| 68 - 80.3 | B | <i>Good</i> |
| 68 | C | <i>OK</i> |
| 51 - 67 | D | <i>Poor</i> |
| < 51 | F | <i>Awful</i> |