

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian mengenai penerapan *Single Page Application* (SPA) bukan untuk pertama kali dilakukan. Dalam penelitian ini, tinjauan pustaka digunakan sebagai alat untuk mencari referensi dan data yang komprehensif untuk menerangi masalah yang diteliti. Berdasarkan tema dan metode pada penelitian, berikut ini merupakan penelitian sebelumnya yang saling berkaitan diantaranya yaitu :

Tabel 2.1 Daftar penelitian terdahulu

No	Judul	<i>Comparing</i>	<i>Contrasting</i>	<i>Criticize</i>	<i>Synthesize</i>	<i>Summarize</i>
1	Analisis Penerapan <i>Single Page Application</i> Menggunakan Teknologi AJAX dan REST API (Studi Kasus: Sistem Informasi Reservasi Wisma Tamu UKSW)", A. Luqman and Azhari, (2017) [10].	Melakukan penelitian terkait dengan pembuatan sistem informasi reservasi tamu sedangkan penulis melakukan penelitian terkait perancangan aplikasi <i>e-commerce</i>	Membahas kekurangan model sistem aplikasi yang lama dan menggantinya yang baru, sedangkan penulis membahas perancangan aplikasi secara menyeluruh	Hanya mengubah arsitektur pada pengembangan aplikasi sebelumnya	Penulis mengembangkan sistem berbasis <i>website</i> dari pengembangan sebelumnya yang berbasis desktop dengan pola sistem yang sama.	Penelitian menunjukkan penggunaan pada bandwidth di sistem yang baru memiliki 70% lebih efektif dibanding pada sistem yang dibuat sebelumnya dimana jauh lebih baik dari pada aplikasi yang digunakan dalam penelitian sebelumnya, yang menghasilkan hasil yang relatif cepat dan ringan.

No	Judul	<i>Comparing</i>	<i>Contrasting</i>	<i>Criticize</i>	<i>Synthesize</i>	<i>Summarize</i>
2	Perancangan <i>Single Page Application</i> Menggunakan Flask-restful Dan Mithril.js (Studi Kasus: <i>Website Directory Travel Haji Dan Umroh</i>), P. D. A. N. Pengabdian, P. Rahmawaty, (2017) [22].	Melakukan penelitian dengan menerapkan menggunakan Flask-restful dan Mithril.js pada pengembangan aplikasi, sedangkan penulis menggunakan <i>library</i> ReactJS	Membahas mengenai aplikasi web biro perjalanan ibadah umrah dan haji yang memiliki waktu respon yang cepat dan interaktif	Belum ada penilaian kinerja aplikasi seperti transfer data, waktu proses, dan loading time. Selain	Penulis mengembangkan aplikasi <i>website</i> dengan sebuah konsep penyegaran pada sebagian halaman atau komponen <i>website</i> yang independen dan menggunakan arsitektur RESTFul-API	Penelitian ini menunjukkan perancangan <i>website</i> guna mengatasi adanya biro perjalanan swasta yang tidak terdaftar secara resmi dan melanggar aturan pemerintah karena belum ada informasi yang lengkap untuk mendata biro perjalanan swasta resmi
3	Pengembangan <i>Single Page Application</i> Pada Sistem Informasi Akademik, P. L. Lokapitasari Belluano, (2018) [23].	Melakukan penelitian dengan menerapkan menggunakan VueJS pada pengembangan aplikasi, sedangkan penulis menggunakan <i>library</i> ReactJS	Membahas mengenai perancangan sistem informasi akademik untuk kegiatan administrasi akademik secara online.	Belum menerapkan teknologi <i>internet of things</i> (IOT) untuk pemrosesan data	Penulis mengembangkan aplikasi <i>website</i> dengan menetapkan <i>library</i> VueJS dan menggunakan PostgreSQL pada penyimpanan data	Penelitian ini menunjukkan perancangan <i>website</i> sistem informasi akademik untuk sistem pengelola data sehingga memudahkan pengguna dalam kegiatan administrasi akademik secara <i>online</i>
4	Penggunaan Aplikasi <i>Content Management System</i> (CMS) Untuk	Melakukan penelitian dengan membuat <i>website</i> dengan	Membahas mengenai warung atau kios di kabupaten	Penggunaan arsitektur pada penelitian ini masih	Penulis mengembangkan aplikasi <i>website</i> dengan menggunakan	Penelitian menunjukkan dengan menerapkan CMS pada <i>e-commerce</i> dapat terealisasi pada

No	Judul	<i>Comparing</i>	<i>Contrasting</i>	<i>Criticize</i>	<i>Synthesize</i>	<i>Summarize</i>
	Pengembangan Bisnis Berbasis <i>E-commerce</i> , B. Huda and B. Priyatna, (2019) [19].	menggunakan <i>Content Management System</i> (CMS), sedangkan penulis menggunakan <i>library React</i> dalam pengembangan aplikasi	Karawang dapat mengaplikasikan penjualan secara <i>online</i>	menggunakan arsitektur tradisional	<i>Content Management System</i> dalam pengembangannya	manajemen produk penjualan dan transaksi pelanggan
5	Rancang Bangun Aplikasi E-Commerce Untuk Usaha Penjualan Helm, T. A. Gelar and B. Budiman, (2019) [20].	Melakukan penelitian dengan membuat <i>website</i> untuk informasi produk helm sebagai sarana dalam menangani pembelian secara online, sedangkan penulis membuat <i>website</i> terkait produk untuk wanita	Membahas mengenai kurang efektifnya bisnis yang dijalankan karena informasi barang yang ditawarkan masih sedikit yang mengharuskan pelanggan mengunjungi untuk membeli produk helm yang secara langsung	Penggunaan arsitektur pada penelitian ini masih menggunakan arsitektur tradisional	Penulis mengembangkan aplikasi <i>website</i> dengan menggunakan <i>Content Management System</i> dalam pengembangannya.	Penelitian menunjukkan hasil pada perangkat lunak dengan peringkat kualitas keseluruhan "Baik" yang dapat dinilai praktis untuk digunakan

No	Judul	<i>Comparing</i>	<i>Contrasting</i>	<i>Criticize</i>	<i>Synthesize</i>	<i>Summarize</i>
6	Penerapan <i>E-Commerce</i> Menggunakan Metode <i>Extreme Programming</i> Pada Umkm Kabupaten Muratara, N. L. G. P. Suwirmayanti, I. K. A. A. Aryanto, (2020) [21].	Melakukan penelitian dengan penerapan <i>e-commerce</i> pada Umkm Kabupaten Muratara dengan dokumentasi perancangan sistem, sedangkan penulis berfokus pada perancangan saat pengembangan aplikasi	Membahas mengenai merancang sistem yang menekankan fleksibilitas setiap proses dan respons dinamis terhadap perubahan yang mungkin timbul dan memungkinkan penyesuaian dilakukan pada penerapan aplikasi	Dokumentasi pada penelitian ini dibuat tidak menggunakan dokumentasi yang formal pada saat pengembangan berlangsung	Penulis mengembangkan aplikasi <i>website</i> dengan pendekatan metode <i>Extreme Programming</i> pada tahap pengembangannya.	Penelitian menunjukkan bahwa dapat dikatakan dokumentasi desain sistem hanya selesai pada tahap eksplorasi dan perencanaan pengembangan
7	Analisis Penerapan <i>Code Splitting Library</i> React pada Aplikasi Penjualan Mebel Berbasis <i>Website</i> , D. Tanudjaja and R. Tanone, (2021) [18].	Melakukan penelitian terkait dengan pembuatan sistem informasi terkait produk mebel, sedangkan penulis membuat <i>website</i> terkait produk untuk wanita	Membahas menurunnya pendapatan penjualan pada toko Mebel Murah akibat terbatasnya akses pembeli	Belum dilakukan unit testing pada aplikasi untuk memastikan kode yang berjalan sesuai dengan rancangan	Penulis mengembangkan sistem berbasis <i>website</i> dengan menerapkan <i>Code Splitting</i> pada <i>library</i> React	Penelitian menunjukkan pengembangan aplikasi <i>website</i> memiliki waktu yang efisien dan memiliki kinerja yang baik berdasarkan pengujian yang dilakukan
8	Perancangan <i>Frontend</i> Aplikasi	Melakukan penelitian dalam	Membahas mengenai	Belum menerapkan	Penulis merancang antarmuka aplikasi	Penelitian ini menunjukkan perancangan antar muka

No	Judul	<i>Comparing</i>	<i>Contrasting</i>	<i>Criticize</i>	<i>Synthesize</i>	<i>Summarize</i>
	Pemandu Pariwisata Menggunakan <i>Framework</i> React.js Di Provinsi Jawa Barat, A. M. Vinka, N. Michele, F. T. Industri, and F. Seni, (2021 [24]).	perancangan antarmuka aplikasi pemandu wisata, sedangkan penulis tidak hanya perancangan antarmuka tetapi juga dengan fungsionalitasnya	perancangan antarmuka aplikasi pemandu wisata di provinsi Jawa Barat dengan menggunakan ReactJS	<i>backend</i> untuk fungsionalitas aplikasi	pemandu wisata dengan menggunakan ReactJS pada proses pengembangan aplikasi	aplikasi pemandu wisata pada provinsi Jawa Barat dengan penyebaran informasi melalui <i>website</i> efektif dan informasi tentang pariwisata dapat diakses di mana saja, kapan saja.

2.2 Dasar Teori

2.2.1 Website

Website adalah sekelompok halaman web yang telah dipublikasikan secara online dan dapat diakses oleh siapa saja dengan memasukkan domain atau URL mereka (*Uniform Resource Locator*) [25]. Situs web semakin banyak digunakan untuk aktivitas selain berbagi informasi, seperti komunikasi langsung, mendengarkan musik, bermain game, dan bahkan kemampuan untuk membuat dan mengedit foto secara langsung. Situs web dibagi menjadi dua kategori berdasarkan sifatnya [26], sebagai berikut:

1. *Website Statis*, merupakan *website* dengan menampilkan konten halaman yang tetap. Data konten yang ditampilkan tidak diambil melalui server database melainkan ditampilkan secara manual yang ditulis secara langsung pada struktur kode.
2. *Website Dinamis*, dimana *website* menampilkan konten halaman yang dapat berubah sewaktu waktu. Konten yang ditampilkan diambil melalui server database yang dibangun pada API yang terhubung dengan aplikasi *website*.

2.2.2 E-Commerce

E-commerce adalah praktik melakukan bisnis secara elektronik melalui penggunaan jaringan komputer, termasuk aktivitas jual beli [19]. *E-commerce* mengacu pada berbagai kegiatan, seperti layanan distribusi, penjualan, pembelian, pemasaran, layanan purna jual, dan layanan produk, yang semuanya dilakukan melalui sistem elektronik, seperti internet atau jaringan komputer lainnya [27].

Karena memberikan banyak kemudahan bagi kedua belah pihak (penjual dan pembeli) dalam melakukan transaksi, walaupun para pihak berada di dua tempat yang berbeda, dan karena transaksi tersebut dilakukan tanpa perlu adanya pertemuan tatap muka, maka adanya e-commerce merupakan keputusan

bisnis yang menjanjikan untuk dilaksanakan saat ini. e-commerce telah membantu dalam tawar-menawar [28]. *E-Commerce* memiliki keuntungan [29], antara lain :

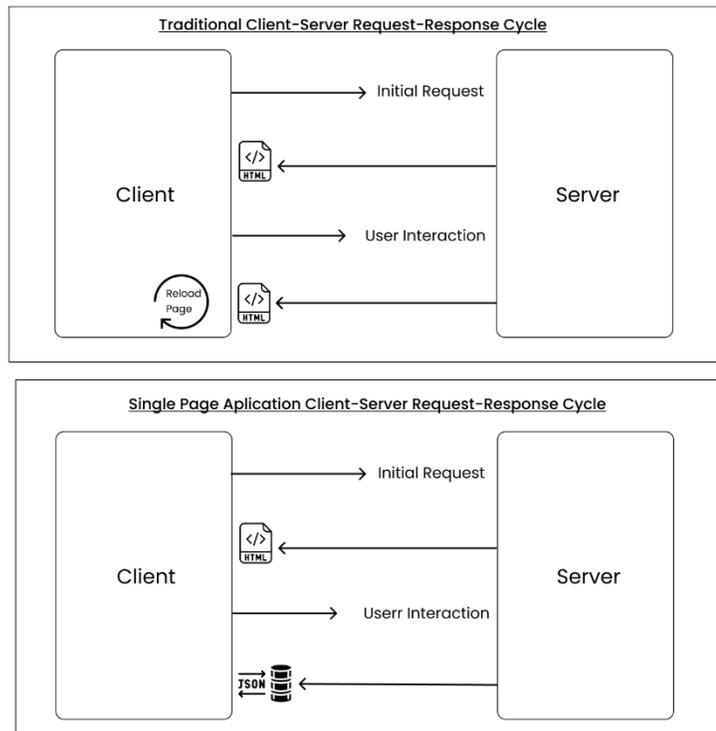
1. Bagi perusahaan
 - a. Memperluas jangkauan pasar
 - b. Biaya operasional berkurang
2. Bagi konsumen
 - a. Menyediakan akses 24 jam bagi konsumen untuk berbelanja atau melakukan transaksi lainnya
 - b. Bahkan jika tidak langsung ke toko, konsumen tetap mudah untuk memilih pilihan barang.
3. Bagi masyarakat
 - a. Memungkinkan orang untuk bekerja dari rumah dan berbelanja tanpa meninggalkan rumah
 - b. memungkinkan berbagai produk ditawarkan dengan harga yang lebih murah

2.2.3 Single Page Application

Single Page Application (SPA) merupakan arsitektur untuk meningkatkan bagaimana pengguna berinteraksi dengan sistem.. Tujuan utama SPA selain menjadi interaktif, adalah untuk mengefisienkan penggunaan data selama proses request dan respon dari web server [22]. Semua komponen yang diperlukan seperti CSS, gambar, *source code*, dan sumber daya lainnya dimuat sekaligus di halaman utama dalam aplikasi SPA serta konten atau komponen terkait dimuat secara dinamis berdasarkan interaksi atau request pengguna [20].

Single Page Application (SPA) hanya menggunakan satu halaman web sebagai tampilan aplikasi sehingga beban kerja server dan *browser* dapat berkurang serta membuat kinerja aplikasi web tampak seperti aplikasi desktop dengan proses rendering semua data yang telah dimasukkan akan diubah menjadi sebuah output [23]. Satu-satunya perbedaan antara aplikasi berbasis SPA dan aplikasi non-SPA adalah bahwa SPA hanya meminta satu halaman

dari server; di luar itu, klien mengelola mekanisme perutean yang dikelola sebelumnya oleh server [30].



Gambar 2.1 *Client-Server Request-Response Cycle*

Pada gambar 2.1 menunjukkan perbedaan antara web tradisional dan web yang menerapkan SPA. Pada *traditional client*, *user interaction* ke server akan dikembalikan kepada *client* dengan menyegarkan halaman dimana menyajikan berkas html dari server yang kemudian ditampilkan kepada pengguna. Sedangkan dengan menggunakan SPA, *client* akan mengelola data dari server yang berformat JSON (*JavaScript Object Notation*) lalu akan dikelola oleh JavaScript untuk ditampilkan kepada pengguna. JavaScript akan mengatur lalu lintas *browser* dan mengatur keluar masuknya data antar komponen aplikasi. Di sisi *client*, SPA dapat memproses render data yang dikelola oleh dirinya sendiri tanpa bantuan dari sisi server. Dengan Teknologi SPA, pekerjaan server tidak lagi seintensif dengan web tradisional, sehingga beban server menjadi

lebih ringan dan mampu untuk menangani lebih banyak *client*, oleh sebab itu server jarang mengalami *down*.

2.2.4 ReactJs

React merupakan sebuah kerangka kerja *open source* untuk mengembangkan *Single Page Application* (SPA) dan aplikasi mobile yang memanfaatkan *library* JavaScript. React dikembangkan untuk memfasilitasi pembangunan dari komponen antarmuka yang interaktif, *stateful*, dan mudah digunakan ulang [24]. React unggul dalam menghasilkan antarmuka pengguna yang rumit dengan cepat [31]. Penggunaan React memungkinkan pembuatan komponen yang dapat digunakan kembali, menghasilkan kode yang lebih terstruktur, mudah dirawat, dan dapat mempercepat proses pengembangan.

Sebelum adanya ReactJs, developer merasa kesulitan dalam pengembangan *user interface* (UI) dimana hanya menggunakan JavaScript murni atau menggunakan pendahulunya yang tidak terlalu berfokus pada pengembangan *user interface* (UI) seperti JQuery yang memakan waktu lama pada pengembangan dan berpotensi adanya *bug* dan *error* [32]. Komponen React dapat dibuat dengan dua cara yaitu memanfaatkan fungsi atau kelas. Karena penulisan kode menjadi lebih efisien, pengembang disarankan untuk menggunakan fungsi. React memiliki fungsi *code splitting* di dimana bekerja dengan membagi file yang akan diproses untuk tampilan *browser*. Hanya komponen situs web yang saat ini dibutuhkan oleh pengguna yang akan dimuat dengan lambat berkat *code splitting*. *Code Splitting* merupakan salah satu metode untuk meningkatkan fungsionalitas situs web.

2.2.5 Rest API

Application Programming Interface (API) adalah format komunikasi dan bahasa yang digunakan didalam aplikasi agar terhubung dengan sistem lainnya. REST adalah cara komunikasi *Hypertext Transfer Protocol* (HTTP) antara dua sistem yang mencakup membaca halaman web tertentu yang berisi

file XML atau JSON [18].

REST API umumnya digunakan untuk web service berorientasi resource dalam praktiknya. Tujuan mengarahkan resource adalah untuk menciptakan orientasi yang menyediakan resource sebagai layanan dari pada kumpulan aktivitas yang menggunakannya. Karena kemudahan akses dan hasil data yang diberikan dalam format JSON, bentuk web service dengan *style* REST sangat ideal untuk dimanfaatkan sebagai backend untuk sebuah aplikasi [33].

2.2.6 HTTPS

Protokol situs web yang disebut *HyperText Transfer Protocol* (HTTP) menawarkan instruksi untuk komunikasi jaringan, terutama untuk komunikasi antara jaringan komputer klien dan server web, sedangkan *HyperText Transfer Protocol Secure* (HTTPS) merupakan versi lanjutan dari HTTP yang menawarkan manfaat keamanan dengan memanfaatkan *Transmission Control Protocol* (TCP) sehingga data yang dikirim dan diterima melalui port 443 oleh pengguna atau *browser* web akan dienkripsi oleh *Secure Socket Layer* (SSL) atau *Transport Layer Security* (TLS) untuk menggagalkan kehilangan informasi selama proses transmisi [34].

Situs web yang mendistribusikan data atau informasi sensitif milik pengguna dan administrator situs web menggunakan HTTPS sebagai solusi keamanan. Kehadiran indikator simbol kunci di *browser*, serta URL "https://:" di bilah alamat *browser*, dapat mengidentifikasi situs web yang menggunakan layanan HTTPS. Protokol HTTPS menggunakan *Secure Sockets Layer* (SSL), yaitu lapisan yang berada di belakang lapisan aplikasi HTTP. Karena klien terhubung ke server, menegosiasikan koneksi SSL, dan kemudian mentransmisikan HTTP melalui SSL, penerapan HTTPS sangat mudah [35].

2.2.7 Extreme Programming

Extreme programming (XP) adalah bagian dari metode Agile untuk pengembangan perangkat lunak, yang dirancang untuk memenuhi tuntutan pengembangan tertentu dan didasarkan pada *Software Development Life Cycle* (SDLC) [36]. *Extreme Programming* (XP), sering dikenal sebagai pendekatan *technical how to*, adalah seperangkat konsep dan metodologi pengembangan perangkat lunak yang berguna yang membantu tim teknis membuat perangkat lunak dengan cepat dan efektif [37]. *Extreme Programming* (XP) telah menciptakan beberapa praktik terbaik untuk penulisan perangkat lunak selama beberapa tahun terakhir yang sangat berlaku untuk pembuatan situs web [38].

Implementasi metode *Extreme Programming* (XP) dalam pengembangan aplikasi untuk *e-commerce* menunjukkan bahwa teknik pendekatan ini pada setiap tahapan dapat mempercepat pengembangan serta perubahan yang diinginkan oleh pengguna tanpa ada batasan dan mampu membantu menghasilkan perangkat lunak yang secara fungsional sesuai dengan kebutuhan [39]. Keuntungan dari metodologi *Extreme Programming* (XP) adalah bahwa desain perangkat lunak tidak akan berubah tetapi hanya sistem yang sedang dibangun yang akan melakukannya. Tidak adanya analisis kebutuhan tambahan pada aplikasi dipandang sebagai bukti bahwa aplikasi telah selesai dalam teknik ini meskipun tidak ada dokumentasi formal yang digunakan untuk alat ukur dan menilai apakah suatu aplikasi telah dikembangkan atau tidak [40].

2.2.7 Black Box Testing

Blackbox Testing adalah jenis pengujian yang terutama berkaitan dengan persyaratan fungsi yang dihasilkan, termasuk struktur data, akses basis data, masalah GUI, dan kelemahan kinerja tertentu [21]. Ketika sebuah fitur aplikasi telah selesai, pengujian *black box* menjadi tahap pertama dari pengujian proses kerja aplikasi.

Pengujian *black box* digunakan untuk mendeteksi berbagai masalah, termasuk fungsional, antarmuka, struktur data, fungsional, deklarasi, dan kesalahan terminasi [41]. Pengujian *black box* juga cepat hal ini disebabkan karena pengujian hanya melihat tampilan luar dan fungsionalitas aplikasi saja tanpa mengetahui proses internal bekerja. Metode ini memiliki keuntungan karena tidak mengharuskan pengguna untuk memahami bahasa pemrograman sistem, dan pengujian dilakukan dari sudut pandang pengguna.

2.2.7 Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan metode pembuatan sistem yang menggunakan bahasa dinamis sebagai sarana untuk mengeksplorasi dan mendokumentasikan spesifikasi sistem [42]. Dalam lingkungan perusahaan, UML merupakan salah satu standar linguistik yang sering digunakan untuk menentukan kebutuhan, melakukan analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek [43]. Berikut ini merupakan beberapa diagram UML yang biasa digunakan untuk pengembangan sebuah system, diantaranya yaitu :

- a. *Use Case Diagram* : Merupakan Diagram kelas adalah representasi dari struktur sistem dalam hal kelas-kelas yang akan digunakan untuk membangun sistem.
- b. *Activity Diagram* : Merupakan representasi visual dari aktivitas atau proses sistem yang ada dalam program.
- c. *Sequence Diagram* : Merupakan Rentang hidup objek dan pesan yang dikirim dan diterima antar objek digunakan untuk menggambarkan bagaimana objek berperilaku dalam kasus penggunaan.