

BAB II

TINJAUAN PUSTAKA

2.1 Studi Sebelumnya

Terdapat beberapa penelitian yang dijadikan sebagai rujukan dalam penelitian ini. Seluruh penelitian tersebut dibahas dan juga diringkas yang disajikan pada Tabel 2.1. Pada penelitian yang sudah dilakukan oleh R.F. Syafariani, dkk., membahas pengembangan aplikasi untuk pemesanan perawatan bayi[25]. Penelitian tersebut bertujuan untuk mengembangkan aplikasi yang dapat digunakan untuk menemukan dan memesan layanan perawat bayi di Kota Bandung. Aplikasi dikembangkan menggunakan model pengembangan perangkat lunak *Prototyping*. *Prototyping* dimodelkan mulai dari tahapan indentifikasi kebutuhan sistem, pembuatan prototipe, pengujian, evaluasi, dan implementasi. Melalui langkah tersebut, dihasilkan aplikasi terintegrasi yakni aplikasi *Go-Baby* berbasis web yang digunakan oleh penyedia layanan perawatan bayi dan aplikasi *Go-Baby* berbasis *mobile* yang digunakan oleh pelanggan untuk menemukan dan memesan layanan perawatan bayi.

Penelitian berikutnya dilakukan oleh A.D. Mirza, dkk., membahas tentang pengembangan sistem informasi manajemen akademik sekolah di SMA Negeri Mojoagung[26]. Tujuan penelitian tersebut adalah mengembangkan aplikasi sistem informasi yang digunakan oleh tenaga pendidik untuk mengelola kegiatan akademik sekolah di SMA Negeri Mojoagung. Selain itu, secara eksplisit disebutkan tujuan penelitian tersebut adalah mengetahui hasil analisis kebutuhan sistem dan hasil pengujian terhadap sistem yang dibangun. Penelitian tersebut menghasilkan aplikasi sistem informasi yang dapat digunakan dalam mengelola kegiatan akademik di SMA Negeri Mojoagung. Selain itu, penelitian tersebut juga melakukan dokumen analisis kebutuhan sistem, uji *Whitebox* dan *Blackbox* yang menunjukkan hasil valid, uji kompatibilitas dan aksesibilitas yang menunjukkan hasil

dapat diakses dari berbagai peramban, serta uji *User Acceptance Test* (UAT) yang menunjukkan sistem dapat diterima oleh pengguna.

Pada penelitian yang dilakukan oleh F.F. Nursaid, dkk., pada tahun 2020, membahas terkait pengembangan aplikasi pengelolaan persediaan barang pada Toko Uda Fajri[27]. Penelitian tersebut bertujuan dalam pengembangan aplikasi yang akan digunakan dalam pengembangan aplikasi untuk mengelola persediaan barang di Toko Uda Fajri. Pengembangan aplikasi akan menggunakan model pengembangan perangkat lunak *Prototyping*. Dari penelitian yang dijalankan, diperoleh hasil berupa aplikasi berbasis web yang dikembangkan menggunakan kerangka kerja *React Js*. Tidak hanya berbasis web, pengembangan aplikasi berbasis *mobile* juga dikembangkan menggunakan kerangka kerja *React Native*. Selain itu, melalui penelitian tersebut juga berhasil menguji fungsionalitas aplikasi menggunakan pengujian *Blackbox* dan *Whitebox* yang menunjukkan hasil semua fungsi lolos uji. Pengujian UAT juga menjadi hasil penelitian tersebut yang menunjukkan bahwa aplikasi dapat diterima oleh pengguna.

Penelitian oleh H.N. Nouvadila, dkk., pada tahun 2023 membahas mengenai pengembangan *front-end* aplikasi berbasis web untuk jual beli token atau mata uang dalam permainan daring (*game online*)[28]. Fokus penelitian tersebut adalah untuk mengembangkan aplikasi *front-end* yang dapat digunakan untuk melakukan transaksi produk *game online*. Aplikasi dikembangkan menggunakan model pengembangan perangkat lunak *Prototyping*. Melalui pengembangan tersebut, berhasil dikembangkan aplikasi *front-end* berbasis web bernama *CashLess* untuk melakukan transaksi produk *game online*. Pengujian fungsionalitas juga dilakukan terhadap aplikasi yang dikembangkan menggunakan metode *Blackbox* yang menghasilkan 13 dari 14 yang diuji berhasil lolos.

Terakhir, penelitian dilakukan oleh D. Ramdhany, dkk., pada tahun 2021. Penelitian tersebut membahas pengembangan *front-end* aplikasi yang bernama Sistem Informasi Geospasial Patroli Terpadu Kebakaran Hutan dan Lahan

(SIMADU)[29]. Tujuan penelitian tersebut adalah mengembangkan aplikasi *front-end* untuk mengelola kegiatan terkait patroli kebakaran hutan lahan di wilayah Sumatra dan mengintegrasikannya dengan *back-end* yang sudah berjalan pada server. Aplikasi dikembangkan menggunakan model pengembangan perangkat lunak *Prototyping*. Melalui pengembangan tersebut, berhasil dibuat sebuah aplikasi *front-end* berbasis web bernama SIMADU yang terintegrasi dengan aplikasi *back-end* yang sudah berjalan.



Tabel 2.1 Ringkasan Studi Sebelumnya

No	Peneliti	Judul Penelitian	Tahun	Tujuan Penelitian	Metode Penelitian	Hasil yang Diperoleh
1	R.F. Syafariani, dkk.	<i>Application of Backend and Frontend Systems on Go-Baby Application in Bandung City</i>	2019	Mengembangkan aplikasi <i>Go-Baby</i> untuk memudahkan pengguna dalam memasarkan, menemukan, dan memesan layanan perawatan bayi.	<i>Prototyping</i>	<p>Penelitian menghasilkan aplikasi <i>Go-Baby</i> berarsitektur <i>Microservice</i> yang digunakan untuk memudahkan pengguna dalam menemukan dan memesan layanan perawatan bayi di Kota Bandung dengan spesifikasi sebagai berikut.</p> <ul style="list-style-type: none"> • Aplikasi berbasis web yang digunakan oleh penyedia layanan perawatan bayi • Aplikasi berbasis <i>mobile</i> yang digunakan oleh pengguna dalam menemukan dan memesan layanan perawatan bayi
2	A.D. Mirza, dkk.	Pengembangan Sistem Informasi Manajemen Akademik Sekolah Menggunakan Metode <i>Prototyping</i> (Studi pada	2023	Secara eksplisit disebutkan yakni mengetahui hasil analisis kebutuhan, hasil perancangan dan	<i>Prototyping</i>	<ul style="list-style-type: none"> • Dokumen analisis kebutuhan sistem dan perancangan sistem • Aplikasi Sistem Informasi Manajemen Akademik

No	Peneliti	Judul Penelitian	Tahun	Tujuan Penelitian	Metode Penelitian	Hasil yang Diperoleh
		Sekolah SMA Negeri Mojoagung)		implementasi, serta hasil pengujian pada sistem informasi manajemen akademik sekolah.		<p>Sekolah berbasis web yang digunakan di SMA Negeri Mojoagung</p> <ul style="list-style-type: none"> • Pengujian aplikasi menggunakan UAT yang menunjukkan hasil dapat diterima oleh pengguna • Hasil pengujian aplikasi menggunakan <i>Blackbox</i> dan <i>Whitebox</i> yang disebutkan valid • Pengujian kompatibilitas dan aksesibilitas yang menunjukkan hasil dapat digunakan di berbagai peramban
3	F.F. Nursaid, dkk.	Pengembangan Sistem Informasi Pengelolaan Persediaan Barang dengan <i>React Js</i> dan <i>React Naive</i> Menggunakan <i>Prototype</i>	2020	Pengembangan sistem informasi pengelolaan persediaan barang pada Toko Uda Fajri	<i>Prototyping</i>	<ul style="list-style-type: none"> • Aplikasi Sistem Informasi Pengelolaan Persediaan Barang berbasis web untuk Toko Uda Fajri • Aplikasi Sistem Informasi Pengelolaan Persediaan

No	Peneliti	Judul Penelitian	Tahun	Tujuan Penelitian	Metode Penelitian	Hasil yang Diperoleh
		(Studi Kasus: Toko Uda Fajri)				Barang berbasis <i>mobile</i> untuk Toko Uda Fajri <ul style="list-style-type: none"> • Pengujian aplikasi menggunakan UAT yang menghasilkan nilai 80% yang dikategorikan dapat diterima oleh pengguna • Pengujian <i>Blackbox</i> dan <i>Whitebox</i> yang menunjukkan hasil yang valid untuk setiap fungsi yang diuji
4	H.N. Nouvadila, dkk.	Pengembangan <i>Frontend</i> pada <i>Website CahsLess: Top Up Game Berbasis Online</i>	2023	Penelitian bertujuan mengembangkan aplikasi <i>front-end</i> yang digunakan dalam melakukan transaksi token/mata uang atau produk <i>game online</i> .	<i>Prototyping</i>	<ul style="list-style-type: none"> • Aplikasi <i>front-end CashLess</i> yang digunakan untuk melakukan transaksi produk <i>game online</i>. • Pengujian aplikasi menggunakan UAT yang menghasilkan 13 dari 14 tes dapat diterima oleh pengguna

No	Peneliti	Judul Penelitian	Tahun	Tujuan Penelitian	Metode Penelitian	Hasil yang Diperoleh
5	D. Ramdhany, dkk.	Modul <i>Frontend</i> Sistem Informasi Geospasial Patroli Terpadu Kebakaran Hutan dan Lahan	2021	Bertujuan mengembangkan <i>front-end</i> aplikasi sistem informasi untuk mengelola kegiatan patroli kebakaran hutan dan lahan di wilayah Sumatra dan mengintegrasikannya dengan <i>back-end</i> yang sudah berjalan pada server.	<i>Prototyping</i>	Penelitian menghasilkan aplikasi <i>front-end</i> SIMADU yang digunakan untuk memantau sekaligus mengelola kegiatan patroli terkait kebakaran hutan dan lahan di wilayah Sumatra.

2.2 Dasar Teori

2.2.1 Sistem Informasi Manajemen

Sistem informasi manajemen atau *Information Management System* (IMS) adalah sebuah sistem yang dirancang untuk membantu organisasi atau perusahaan dalam mengelola dan mengoptimalkan penggunaan informasi untuk mendukung proses pengambilan keputusan, pengendalian operasi, analisis strategis, dan pencapaian tujuan organisasi[30]. IMS mencakup perangkat lunak, perangkat keras, prosedur, orang, dan data yang bekerja bersama – sama untuk menghasilkan informasi yang diperlukan untuk pengelolaan, perencanaan, pengawasan, serta pengambilan keputusan[31].

2.2.2 Arsitektur Perangkat Lunak *Client-server*

Arsitektur perangkat lunak merupakan gambaran/model dari struktur perangkat lunak itu sendiri. Struktur perangkat lunak mencakup karakteristik arsitektur, prinsip desain, dan ketetapan/keputusan arsitektur, termasuk gaya arsitektur. Gaya arsitektur perangkat lunak secara garis besar dapat dikategorikan menjadi dua, yakni *Monolithic* dan *Distributed*. Aplikasi yang memiliki arsitektur *Monolith* dibangun sebagai satu kesatuan atas semua unit kode (antarmuka, logika bisnis, *database*). Sementara itu, *Distributed* melibatkan distribusi antar komponen aplikasi yang terintegrasi melalui serangkaian protokol akses jarak jauh[32].

Klien-server (*client-server*) merupakan salah satu gaya arsitektur yang dapat dikategorikan sebagai *Distributed*. Arsitektur klien-server memisahkan komponen aplikasi menjadi server dan klien, atau memisahkan fungsi teknis antara *back-end* dan juga *front-end*[32]. Server berperan memberikan layanan terhadap permintaan yang dilontarkan oleh klien. Pada arsitektur klien-server, mengatur logika bisnis dan koneksi dengan *database* diperankan oleh server[33], [34]. Klien sendiri berperan memprentasikan data yang diterima dari server dan mereprentasikan logika bisnis yang terjadi

pada server dalam bentuk tampilan yang dimengerti manusia sebagai pengguna.

Dalam konteks aplikasi berbasis web, server diperankan oleh server web dan klien diperankan oleh peramban (*browser*) yang digunakan pengguna. Interaksi antara server dan klien terjadi dengan kesepakatan antara server dan klien. Ketika kesepakatan terpenuhi, maka server akan menangani permintaan yang dikirimkan klien.

2.2.3 Front-End

Front-End merupakan bagian dari seluruh produk (aplikasi) yang terlihat oleh pengguna akhir. *Front-End* tidak hanya mengacu pada aplikasi berbasis web yang berjalan di peramban. Namun setiap aplikasi yang memiliki antarmuka grafis (*graphical user interface*) ataupun antarmuka baris perintah (*command line interface*) dapat dianggap sebagai aplikasi *front-end*. Dalam konteks aplikasi berbasis web, *front-end* berkaitan erat dengan penggunaan HTML untuk struktur halaman, CSS untuk gaya tata letak halaman, dan Javascript untuk interaktivitas dinamis[23].

Front-End terbagi menjadi dua modul utama, yakni representasi dan logika[24]. Representasi mengacu pada bagaimana konten disajikan kepada pengguna dan bagaimana pengguna berinteraksi dengan dengan konten tersebut. *Front-End* juga harus menyesuaikan responsivitas tampilan agar dapat diakses dari berbagai jenis perangkat dan ukuran. Sementara itu, logika mengacu pada semua yang membuatnya menjadi aplikasi, seperti pengambilan data dari *back-end*, mengubah untuk menampilkannya ke pengguna, menangani permintaan untuk diteruskan, validasi, dan lain – lain.

2.2.4 Application Programming Interface (API)

API merupakan mekanisme yang memberikan landasan bersama bagi perangkat lunak untuk berkomunikasi satu sama lain[25]. Interaksi antar

perangkat lunak dimungkinkan melalui protokol yang mengatur cara komunikasi. Penggunaan API memungkinkan pengembang untuk melakukan monetisasi terhadap aset digital, memperluas nilai produk, dan menghubungkan pengguna dari platform atau perangkat yang berbeda. Selain itu, penggunaan API memungkinkan pihak lain untuk menggunakan layanan (API) untuk membuat aplikasi baru dan mengembangkan bisnis[35].

2.2.5 Representational State Transfer (REST)

REST merupakan salah satu pendekatan arsitektur perangkat lunak yang digunakan dalam pengembangan aplikasi, termasuk yang berbasis web[36]. Salah satu prinsip utama REST adalah *statelessness*, yang berarti setiap permintaan dari klien ke server harus berisi semua informasi yang diperlukan untuk memahami dan memproses permintaan tersebut, tanpa menyimpan status klien antara permintaan[37]. REST juga menggunakan antarmuka yang seragam, dengan penggunaan URI (*Uniform Resource Identifier*) untuk mengidentifikasi sumber daya, metode HTTP (*GET*, *POST*, *DELETE*) untuk berinteraksi dengan sumber daya, dan representasi data yang berorientasi sumber daya (berupa JSON atau XML) untuk mengirim atau menerima data[38], [39].

2.2.6 Client Side Rendering (CSR)

CSR merupakan suatu pendekatan dalam pengembangan perangkat lunak berbasis web yang proses rendering atau pembuatan tampilan halaman terjadi di sisi klien, yakni di peramban[40]. Dalam CSR, halaman HTML awal yang diterima oleh peramban berisi struktur dasar dan referensi ke dokumen *Javascript* dan *CSS* yang diperlukan. Setelah halaman awal dimuat, *Javascript* akan dieksekusi oleh peramban untuk mengambil data tambahan dari server dan membangun tampilan halaman secara dinamis[41]. Dengan demikian CSR dapat memberikan pengalaman pengguna yang interaktif dan responsif. Hal ini dimungkinkan karena banyak

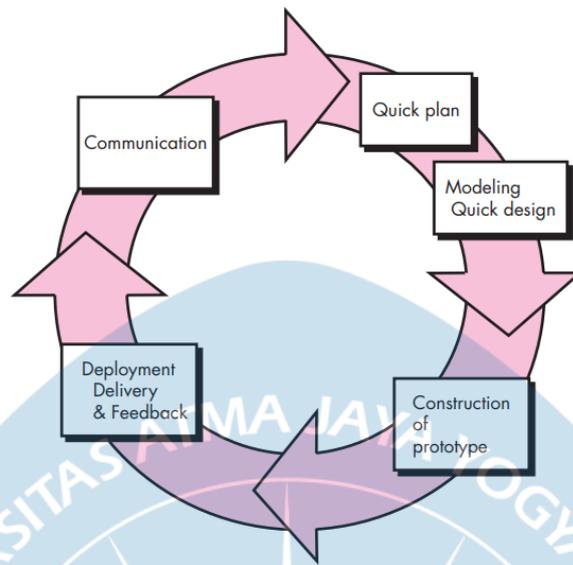
operasi yang dapat dilakukan di sisi klien tanpa perlu memuat ulang seluruh halaman.

2.2.7 *Single Page Application (SPA)*

SPA adalah pendekatan aplikasi web yang hanya memuat satu halaman web atau aplikasi yang hanya mengandung sebuah halaman HTML[42]. Aplikasi yang menggunakan SPA secara dinamis memperbarui konten pada halaman sesuai permintaan tanpa perlu memuat ulang seluruh halaman, sehingga tidak terjadi permintaan HTML baru ke server ketika terjadi interaksi. Hal ini memungkinkan pengalaman pengguna yang lebih responsif dibandingkan dengan pendekatan *Multi Page Application* yang mengharuskan memuat ulang halaman setiap mengakses konten yang berbeda[43]. Penggunaan pendekatan ini bertujuan mempersingkat waktu penyajian data yang sedang diminta, sehingga dapat memberikan pengalaman pengguna yang lebih optimal.

2.2.8 *Metode Pengembangan Prototyping*

Prototyping merupakan pendekatan pengembangan perangkat lunak yang mengutamakan pembuatan prototipe atau model awal dari perangkat lunak yang dibuat. Prototipe digunakan untuk memvalidasi desain, fitur, dan fungsionalitas perangkat lunak yang dikembangkan. Pengembangan ini berfokus pada iterasi cepat dan umpan balik dari pengguna untuk memastikan perangkat lunak yang dihasilkan sesuai dengan kebutuhan dan harapan[44], [45]. Selain itu, *Prototyping* digunakan ketika pengguna belum mampu mendeskripsikan terkait spesifikasi kebutuhan dengan rinci[46], [47]. Terdapat beberapa tahapan dalam metode *Prototyping* yang dimodelkan pada Gambar 2.1[48].



Gambar 2.1 Model Tahapan *Prototyping*

Tahapan – tahapan tersebut secara berturut – turut dijelaskan sebagai berikut.

1) Komunikasi (*Communication*)

Pada tahap ini dilakukan diskusi bersama pemangku kepentingan untuk menentukan beberapa hal mendasar. Melalui pendalaman ditentukan terkait permasalahan klien yang ingin diatasi serta objektif yang ingin dicapai. Dengan demikian, dapat diuraikan prioritas pengembangan, batasan/ruang lingkup, serta kebutuhan fungsional dan nonfungsional.

2) Perencanaan Cepat (*Quick Plan*) dan Pemodelan Desain Cepat (*Modelling Quick Design*)

Pada tahap ini dilakukan perencanaan berdasarkan kebutuhan yang berhasil diidentifikasi sebelumnya. Melalui tahap ini, konsep dari soulsi yang disetujui divisualisasikan sehingga memudahkan tim dan pemangku kepentingan untuk memahaminya. Selain itu, melalui tahap ini memberikan gambaran yang jelas sebagai panduan dalam mengerjakan protipe.

3) Pengembangan Prototipe (*Construction of Prototype*)

Pada tahap ini dilakukan pembuatan prototipe berdasarkan hasil perancangan yang dilakukan pada tahap sebelumnya. Hasil akhir dari tahap ini dapat digunakan oleh pengguna akhir dan dapat berinteraksi untuk memberikan pemahaman lebih baik terkait cara kerja serta fungsionalitas perangkat lunak.

4) Implementasi dan Umpan Balik (*Deployment & Feedback*)

Pada tahap ini melibatkan pemangku kepentingan termasuk pengguna akhir dalam memberikan umpan balik terhadap prototipe yang dicoba. Melalui tahap ini tim dapat melakukan penyesuaian berdasarkan umpan balik yang diberikan. Keseluruhan tahap pada *Prototyping* akan berulang mulai dari tahap awal sampai dicapai kesepakatan bahwa prototipe sudah memenuhi kebutuhan dan harapan klien.

2.2.9 *Unified Modelling Language (UML)*

UML adalah bahasa standar yang digunakan untuk memodelkan, merancang, dan mendokumentasikan sistem perangkat lunak[49]. UML menggunakan notasi grafis untuk menggambarkan struktur, perilaku, dan interaksi antar komponen dalam sistem perangkat lunak. Tujuan utama UML adalah menyediakan cara yang jelas dan konsisten untuk mengkomunikasikan desain sistem dan berkolaborasi antara pengembang dan pemangku kepentingan[50].

Salah satu bentuk diagram UML yang dapat menggambarkan interaksi sistem adalah Diagram *Use case*. Diagram *Use case* membantu memvisualisasikan fungsionalitas sistem dan hubungannya dengan aktor. Adapun komponen utama dalam diagram ini adalah Aktor, *Use case*, Sistem, dan Relasi.

2.2.10 Pengujian *Blackbox*

Pengujian *Blackbox* adalah metode pengujian perangkat lunak tanpa memperhatikan struktur internal atau logika program yang diuji[51], [52]. Pendekatan ini memperlakukan perangkat lunak yang diuji sebagai kotak hitam. Masukan diberikan dan keluaran yang dihasilkan dievaluasi tanpa memperhatikan bagaimana proses pengolahan data dilakukan didalamnya. *Blackbox* hanya fokus pada fungsionalitas dari perangkat lunak, dengan menguji pemenuhan persyaratan spesifikasi yang telah ditentukan dan perilaku yang diharapkan dari sistem[53], [54].

