

BAB 2

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian terdahulu pertama dilakukan oleh Muhammad Irfan Hafidzi (2023) dengan judul *"Perancangan Sistem Layanan Ketenagakerjaan Berbasis User Experience Menggunakan Metode Design Thinking (Studi Kasus: SIAPkerja Kemnaker RI)"*. Penelitian ini menerapkan metode Design Thinking untuk merancang sistem layanan ketenagakerjaan berbasis kebutuhan pengguna dan menghasilkan prototipe dengan nilai System Usability Scale (SUS) sebesar 79 yang termasuk kategori *Excellent*.

Penelitian kedua dilakukan oleh Rizal Prakoso (2023) dengan judul *"Perancangan Antarmuka Sistem Informasi Inventory Dengan Metode Design Thinking Pada BPJS Ketenagakerjaan Cabang Yogyakarta"*. Penelitian ini menghasilkan rancangan antarmuka sistem informasi inventory yang lebih terstruktur dan sesuai dengan kebutuhan pengguna melalui tahapan Design Thinking.

Penelitian ketiga dilakukan oleh Muhammad Aidil Fitri Syah dkk. (2023) dengan judul *"Penerapan Metode Design Thinking Dalam Rancangan Design Interface Aplikasi Himbauan Tirta Musi Palembang"*. Hasil penelitian menunjukkan bahwa metode Design Thinking mampu menghasilkan rancangan antarmuka yang lebih sesuai dengan kebutuhan pengguna dan meningkatkan kualitas desain aplikasi.

Penelitian keempat dilakukan oleh Kurniawan Gelar Sentiaji dkk. (2023) dengan judul *"Perancangan User Interface dan User Experience Pada Aplikasi Pelatihan Prakerja.com Menggunakan Metode Design Thinking"*. Penelitian ini menghasilkan rancangan UI/UX aplikasi berbasis web yang lebih mudah digunakan dan sesuai dengan kebutuhan pengguna.

Penelitian kelima dilakukan oleh Dodi Mualfah dkk. (2023) dengan judul *"Analisis dan Perancangan User Interface dan User Experience untuk Sistem Asuransi Karyawan Berbasis Web Menggunakan Metode Design Thinking"*. Penelitian ini bertujuan merancang antarmuka pengguna (UI) dan pengalaman pengguna (UX) pada aplikasi asuransi karyawan berbasis web menggunakan

metode Design Thinking melalui tahapan empathize, define, ideate, prototype, dan testing. Pada tahap prototype dilakukan perancangan user flow, wireframe, design system, dan desain antarmuka (UI Design), sedangkan tahap testing dilakukan menggunakan platform Maze dan metode System Usability Scale (SUS). Hasil penelitian menunjukkan bahwa rancangan antarmuka memperoleh skor SUS sebesar 77,58 dengan kategori Good, yang menunjukkan bahwa desain yang dihasilkan mudah dipahami dan dapat digunakan dengan baik oleh pengguna.

Berdasarkan penelitian terdahulu tersebut, dapat disimpulkan bahwa metode Design Thinking banyak digunakan dalam perancangan UI/UX sistem berbasis web untuk menghasilkan antarmuka yang lebih terstruktur, mudah digunakan, dan sesuai dengan kebutuhan pengguna. Namun, sebagian besar penelitian masih berfokus pada tahap perancangan atau prototipe dan belum mengimplementasikan hasil desain ke dalam front-end berbasis React JS, sehingga menjadi celah yang diangkat pada penelitian ini.

Tabel 2.1 Gap penelitian

Kategori	Muhammad Irfan Hafidzi (2023)	Rizal Prakoso (2023)	Muhammad Aidil Fitri Syah dkk. (2023)	Kurniawan Gelar Sentiaji dkk. (2023)	Dodi Mualfah dkk. (2023)
Fokus	Perancangan UI/UX sistem layanan ketenagakerjaan	Perancangan UI/UX sistem inventory BPJS Ketenagakerjaan	Perancangan UI aplikasi berbasis web	Perancangan UI/UX aplikasi pelatihan berbasis web	Perancangan UI/UX sistem asuransi karyawan berbasis web
Metode	Design Thinking	Design Thinking	Design Thinking	Design Thinking	Design Thinking
Objek	Sistem layanan ketenagakerjaan	Sistem inventory BPJS Ketenagakerjaan	Aplikasi berbasis web	Aplikasi pelatihan berbasis web	Sistem asuransi karyawan berbasis web
Hasil	Prototype UI/UX dengan nilai SUS 79 (Excellent).	Rancangan UI/UX sesuai kebutuhan pengguna.	Prototype UI sesuai kebutuhan pengguna.	Rancangan UI/UX aplikasi berbasis web.	Prototype UI/UX dengan SUS 77,58 (Good).

Berdasarkan Tabel diatas, dapat diketahui bahwa penelitian-penelitian terdahulu telah menerapkan metode Design Thinking dalam perancangan UI/UX. Namun, sebagian besar penelitian masih berfokus pada tahap perancangan atau

prototipe dan belum mengimplementasikan hasil desain ke dalam front-end berbasis React JS. Selain itu, evaluasi yang dilakukan umumnya menggunakan usability testing atau SUS, sedangkan penelitian ini mengombinasikan Cognitive Walkthrough dan A/B Testing. Oleh karena itu, penelitian ini mengembangkan implementasi front-end sistem monitoring klaim JKK berbasis React JS dengan evaluasi menggunakan Cognitive Walkthrough dan A/B Testing serta memanfaatkan Laravel Telescope sebagai tools pendukung implementasi.

2.2 Landasan Teori

Pada bagian ini diuraikan teori-teori yang menjadi landasan dalam penelitian untuk mendukung proses analisis dan perancangan sistem. Teori yang digunakan diperoleh dari berbagai sumber relevan, seperti buku, jurnal ilmiah, dan penelitian terdahulu, yang meliputi konsep sistem informasi, User Interface (UI), User Experience (UX), serta metode pengembangan sistem. Selain itu, dibahas gambaran umum BPJS Ketenagakerjaan Pekanbaru Kota sebagai objek penelitian agar sistem yang dirancang sesuai dengan kebutuhan pengguna dan mendukung proses monitoring klaim JKK.

2.2.1 BPJSTK Pekanbaru Kota

BPJS Ketenagakerjaan (BPJSTK) merupakan lembaga yang memberikan perlindungan kepada tenaga kerja, termasuk melalui program Jaminan Kecelakaan Kerja (JKK). Kantor Cabang BPJSTK Pekanbaru Kota berperan dalam melaksanakan proses pengajuan, pemantauan, dan penyelesaian klaim JKK yang melibatkan petugas BPJSTK, peserta, serta rumah sakit atau klinik mitra.

Dalam pelaksanaannya, proses monitoring klaim masih menghadapi beberapa kendala, seperti pemantauan status klaim yang dilakukan secara manual, belum adanya notifikasi otomatis, kurangnya integrasi data, serta antarmuka pengguna yang belum optimal. Kondisi tersebut menyebabkan proses pencarian informasi, rekapitulasi klaim, dan evaluasi kinerja rumah sakit atau klinik menjadi kurang efisien serta berpotensi menimbulkan kesalahan.

Oleh karena itu, diperlukan sistem informasi berbasis web dengan antarmuka pengguna yang responsif dan mudah digunakan untuk mendukung monitoring status klaim, rekapitulasi data, serta evaluasi kinerja rumah sakit atau klinik. Penerapan sistem yang mempertimbangkan aspek *User Interface* (UI) dan

User Experience (UX) diharapkan dapat meningkatkan efisiensi proses kerja, transparansi, dan kualitas pelayanan di BPJS Ketenagakerjaan Pekanbaru Kota (Putra, 2023).

2.2.2 Sistem Informasi

Sistem informasi merupakan gabungan teknologi, manusia, dan prosedur yang digunakan untuk mengumpulkan, mengolah, dan menyajikan informasi dalam suatu organisasi. Pada penelitian ini, sistem informasi dirancang untuk mengelola data klaim JKK secara terpusat sehingga memudahkan petugas BPJS Ketenagakerjaan Pekanbaru Kota dalam memantau, mengelola, dan mengakses data klaim.

Penerapan sistem informasi mampu mengotomatisasi proses manual sehingga mengurangi kesalahan, mempercepat proses rekapitulasi, pelaporan, dan evaluasi, serta meningkatkan efisiensi kerja. Selain itu, sistem juga mendukung integrasi data dan transparansi proses verifikasi klaim.

Sistem dikembangkan dengan pendekatan yang berpusat pada pengguna (*user-centered*), sehingga memiliki antarmuka yang responsif, mudah digunakan, dan mendukung pemantauan status klaim secara terpusat. Dengan demikian, sistem diharapkan mampu meningkatkan kemudahan akses informasi serta kualitas layanan pengelolaan klaim JKK (Saputri, 2023; Septiara, 2024).

2.2.3 User Interface (UI)

User Interface (UI) adalah bagian dari sistem yang memungkinkan pengguna berinteraksi langsung dengan perangkat lunak atau aplikasi. UI mencakup elemen-elemen visual seperti tombol, menu, *icon*, dan tata letak yang dirancang untuk memudahkan pengguna dalam menjalankan fungsi-fungsi tertentu. Desain UI yang baik bertujuan untuk menciptakan antarmuka yang intuitif dan mudah digunakan, sehingga pengguna dapat berinteraksi dengan sistem tanpa kebingungan.

Menurut Widyasari dan Yustiawan (2020), UI merupakan input dan output yang langsung melibatkan pengguna akhir. Dengan kata lain, UI adalah cara program dan pengguna saling berinteraksi melalui media visual pada perangkat seperti komputer, *smartphone*, atau tablet. Desain UI yang efektif harus

mempertimbangkan aspek estetika dan fungsionalitas agar pengguna merasa nyaman dan efisien saat menggunakan *website*.

Dalam konteks pengembangan sistem informasi, UI berperan penting dalam menentukan keberhasilan suatu aplikasi. Antarmuka yang dirancang dengan baik dapat meningkatkan kepuasan pengguna, mengurangi kesalahan penggunaan, dan mempercepat proses belajar pengguna terhadap sistem tersebut. Oleh karena itu, perancangan UI harus dilakukan dengan cermat dan mempertimbangkan kebutuhan serta preferensi pengguna.

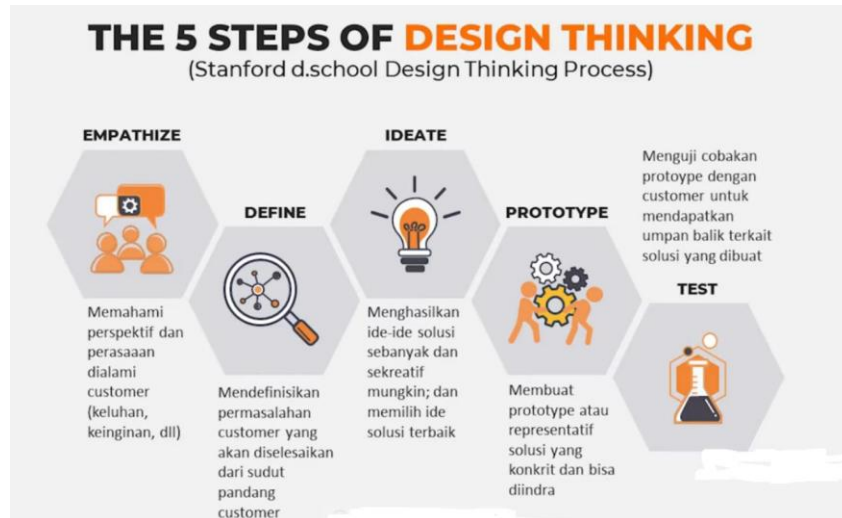
2.2.4 *User Experience (UX)*

User Experience (UX) adalah persepsi dan respon pengguna terhadap penggunaan suatu produk, sistem, atau layanan. UX mencakup seluruh aspek interaksi pengguna, mulai dari kemudahan penggunaan, efisiensi, hingga kepuasan emosional yang dirasakan selama dan setelah menggunakan produk tersebut. Tujuan utama dari UX adalah menciptakan pengalaman yang positif dan bermakna bagi pengguna.

Frank Guo (2012) menyatakan bahwa UX yang baik harus memenuhi empat elemen utama: kegunaan (*usability*), nilai (*value*), kemampuan diadopsi (*adoptability*), dan daya tarik (*desirability*). Kegunaan mengacu pada seberapa mudah pengguna dapat menggunakan produk untuk mencapai tujuan mereka. Nilai berkaitan dengan manfaat yang diperoleh pengguna dari produk tersebut. Kemampuan diadopsi menunjukkan seberapa cepat dan mudah pengguna dapat mulai menggunakan produk. Sedangkan daya tarik mencerminkan seberapa menarik dan menyenangkan produk tersebut bagi pengguna.

Dalam pengembangan sistem informasi, memperhatikan UX sangat penting untuk memastikan bahwa produk yang dikembangkan benar-benar memenuhi kebutuhan dan harapan pengguna. Dengan fokus pada UX, pengembang dapat menciptakan sistem yang tidak hanya fungsional, tetapi juga memberikan pengalaman yang menyenangkan dan memuaskan bagi pengguna. Hal ini dapat meningkatkan loyalitas pengguna dan keberhasilan jangka panjang dari produk tersebut.

2.2.5 Metode Design Thinking



Gambar 2.1 Metode *Design Thinking*

Design Thinking merupakan metode yang berorientasi pada pengguna (*user-centered*) dan digunakan sebagai pendekatan berbasis solusi untuk memahami permasalahan serta menghasilkan rancangan yang sesuai dengan kebutuhan pengguna. Metode ini terdiri atas lima tahapan utama, yaitu *empathize*, *define*, *ideate*, *prototype*, dan *testing*. Penerapan Design Thinking dalam perancangan *user interface* (UI) dan *user experience* (UX) dapat membantu pengembang memahami kebutuhan, harapan, serta permasalahan yang dialami pengguna sehingga solusi yang dihasilkan menjadi lebih tepat sasaran.

Pada tahap *empathize*, dilakukan pendekatan kepada pengguna melalui kegiatan observasi, wawancara, atau penyebaran kuesioner untuk memperoleh informasi mengenai kebutuhan dan kendala yang dialami pengguna. Selanjutnya, tahap *define* digunakan untuk mengidentifikasi serta merumuskan permasalahan berdasarkan hasil pengumpulan data. Permasalahan yang telah dirumuskan kemudian menjadi dasar dalam menentukan solusi yang akan dikembangkan.

Tahap *ideate* dilakukan dengan mengumpulkan berbagai ide dan alternatif solusi untuk menyelesaikan permasalahan pengguna. Ide-ide tersebut selanjutnya diprioritaskan berdasarkan kebutuhan dan tingkat kepentingannya. Pada tahap *prototype*, solusi yang telah dipilih diwujudkan dalam bentuk rancangan yang dapat disimulasikan dan diuji, seperti *user flow*, *design system*, dan desain antarmuka. Tahap terakhir, yaitu *testing*, dilakukan dengan melibatkan pengguna secara langsung untuk mengetahui kemudahan penggunaan, mengidentifikasi kekurangan,

serta memperoleh umpan balik terhadap rancangan yang telah dibuat. Hasil pengujian tersebut dapat digunakan sebagai dasar untuk melakukan evaluasi dan penyempurnaan agar sistem yang dikembangkan lebih mudah digunakan serta sesuai dengan kebutuhan pengguna (Mualfah & Rajif, 2023).

2.2.6 Website

Website merupakan kumpulan halaman digital yang saling terhubung dan dapat diakses melalui jaringan internet menggunakan peramban web (*browser*). *Website* dapat menyajikan berbagai jenis informasi, seperti teks, gambar, video, serta elemen interaktif untuk mendukung berbagai kebutuhan pengguna.

Menurut Wijaya dan Hartati (2021), *website* merupakan media digital yang menyediakan informasi atau layanan secara *online* dan dapat diakses kapan saja selama terhubung dengan internet. Selain itu, informasi pada *website* dapat diperbarui sesuai dengan kebutuhan sehingga menjadikannya sebagai media yang fleksibel dalam penyampaian informasi dan layanan.

Dalam pengembangan sistem informasi, *website* banyak digunakan karena dapat diakses tanpa memerlukan instalasi khusus pada perangkat pengguna. *Website* juga dapat dirancang secara responsif agar tampilannya dapat menyesuaikan dengan berbagai perangkat. Oleh karena itu, perancangan tampilan dan struktur *website* perlu diperhatikan agar dapat memberikan kemudahan bagi pengguna dalam mengakses informasi dan menggunakan fungsi sistem.

2.2.7 Figma

Figma merupakan *platform* desain yang digunakan untuk merancang antarmuka pengguna (UI) dan pengalaman pengguna (UX). Figma mendukung pembuatan prototipe interaktif serta kolaborasi secara terpusat antara desainer dan tim pengembang (Wang & Zhang, 2020). Selain itu, Figma dapat diakses melalui *browser* sehingga memudahkan proses desain dan kolaborasi (Hernandez & Marquez, 2021).

Figma juga mendukung proses desain secara iteratif melalui pembuatan dan pengujian prototipe. Desainer dapat melakukan perbaikan berdasarkan umpan balik pengguna atau pemangku kepentingan. Fitur desain responsif dan interaktif membantu dalam mengevaluasi tampilan dan interaksi antarmuka (Lee & Kwon, 2022).

Figma memungkinkan desain dibagikan dengan mudah kepada anggota tim maupun pemangku kepentingan. Kemudahan berbagi desain dan prototipe membantu mempercepat proses pemberian umpan balik (Gómez & Hernández, 2023). Dalam penelitian ini, Figma digunakan untuk merancang prototipe antarmuka yang menjadi acuan dalam implementasi *front-end*.

2.2.8 FigJam

FigJam merupakan alat kolaborasi visual dari Figma yang digunakan untuk mendukung proses *brainstorming*, perencanaan, dan pengembangan ide. Berbeda dengan Figma yang berfokus pada desain antarmuka, *FigJam* menyediakan papan tulis digital untuk mencatat dan mengorganisasi berbagai ide. Fitur seperti *sticky notes*, *shapes*, dan *connector lines* membantu menyusun ide dan alur kerja secara terstruktur (Wang & Zhang, 2020).

FigJam juga mendukung komunikasi dan kolaborasi antaranggota tim dalam proses perancangan. Ruang kerja dapat diakses secara bersamaan sehingga anggota tim dapat memberikan masukan secara langsung. Fitur komentar dan pengeditan membantu proses diskusi dan perencanaan tanpa harus dilakukan secara langsung di lokasi yang sama (Hernandez & Marquez, 2021).

Dalam penelitian ini, *FigJam* digunakan pada tahap eksplorasi dan pengembangan ide sebelum pembuatan prototipe di Figma. *FigJam* digunakan untuk mendiskusikan dan mengelompokkan ide fitur serta menyusun alur yang dibutuhkan dalam perancangan antarmuka. Proses tersebut membantu mengorganisasi hasil ide sebagai dasar untuk melanjutkan proses perancangan prototipe (Lee & Kwon, 2022).

2.2.9 A/B Testing

A/B Testing merupakan metode evaluasi untuk membandingkan dua versi desain antarmuka guna menentukan desain yang memberikan pengalaman penggunaan lebih baik. Pengguna diberikan Versi A dan Versi B untuk menyelesaikan tugas yang sama. Hasilnya dibandingkan menggunakan indikator seperti *Success Rate*, *Time on Task*, *Misclick Rate*, *Task Failure*, dan *User Preference*.

Menurut Kohavi et al. (2020), *A/B Testing* merupakan eksperimen terkontrol yang digunakan untuk mengevaluasi efektivitas desain atau fitur

berdasarkan perilaku pengguna. Metode ini membantu menentukan alternatif desain yang lebih sesuai berdasarkan data hasil pengujian. Dalam pengujian antarmuka, platform seperti Maze dapat digunakan untuk menguji *prototype* Figma dan memperoleh metrik interaksi pengguna.

Penggunaan Maze dalam *usability testing* dapat menghasilkan data seperti *Success Rate*, *Misclick Rate*, *Completion Time*, dan pola interaksi pengguna. Ranty et al. (2024) menggunakan Maze untuk mengevaluasi *prototype* berdasarkan aspek *learnability*, *efficiency*, *error*, dan *success rate*. Berdasarkan hal tersebut, *A/B Testing* dengan Maze dapat digunakan untuk membandingkan alternatif desain secara objektif sebelum desain diterapkan pada tahap implementasi.

2.2.10 React JS

React JS merupakan pustaka JavaScript yang digunakan untuk membangun antarmuka pengguna yang dinamis dan responsif. React menggunakan konsep *Virtual DOM* untuk membantu mempercepat proses pembaruan tampilan pada halaman web. Dengan konsep tersebut, perubahan tampilan dapat dilakukan secara efisien tanpa memuat ulang seluruh halaman.

React mendukung pengembangan berbasis komponen sehingga antarmuka dapat dibagi menjadi bagian-bagian yang lebih terstruktur. Komponen tersebut dapat digunakan kembali pada bagian aplikasi yang memiliki kebutuhan serupa. Menurut Meta (2023), React memungkinkan pengembang membangun antarmuka pengguna yang interaktif sesuai dengan kebutuhan aplikasi web modern.

React dapat digunakan untuk membangun aplikasi web yang memiliki berbagai interaksi dan menampilkan data secara dinamis. React juga mendukung integrasi data serta pengembangan tampilan yang dapat digunakan pada berbagai perangkat. Dalam penelitian ini, React digunakan untuk mengimplementasikan rancangan *front-end* Sistem Informasi Pengajuan dan Status Bayar Klaim JKK agar dapat mendukung monitoring data, informasi status klaim, dan interaksi pengguna.

2.2.11 VS Code

Visual Studio Code (VS Code) merupakan editor kode sumber yang dikembangkan oleh Microsoft dan mendukung berbagai bahasa pemrograman. VS Code menyediakan fitur seperti integrasi Git, *debugging*, IntelliSense, dan berbagai ekstensi untuk mendukung proses pengembangan. Menurut Microsoft (2024), VS

Code banyak digunakan karena fleksibel dan mudah digunakan dalam pengembangan perangkat lunak.

VS Code memiliki antarmuka yang sederhana serta mendukung pengelolaan struktur proyek secara terorganisir. Dukungan ekstensi memungkinkan pengembang menyesuaikan fitur editor dengan kebutuhan pengembangan *front-end*, *back-end*, maupun *full-stack*. Kemampuan tersebut menjadikan VS Code sesuai digunakan dalam pengembangan sistem informasi berbasis web.

VS Code juga menyediakan fitur pendukung seperti Live Share untuk kolaborasi dan Live Server untuk melihat perubahan aplikasi selama proses pengembangan. Fitur-fitur tersebut membantu pengembang dalam melakukan pengembangan dan pengujian aplikasi secara lebih efisien. Dalam penelitian ini, VS Code digunakan sebagai editor kode dalam proses implementasi *front-end* Sistem Informasi Pengajuan dan Status Bayar Klaim JKK.

2.2.12 Cognitive Walkthrough

Metode *Cognitive Walkthrough* merupakan teknik evaluasi *usability* yang berfokus pada kemampuan pengguna dalam memahami dan menyelesaikan tugas pada sistem tanpa pelatihan khusus. Evaluator menelusuri langkah-langkah pengguna untuk menilai apakah tindakan yang harus dilakukan dapat dikenali dan dipahami dengan mudah. Metode ini menekankan aspek *learnability* dan kemudahan navigasi antarmuka (Wharton et al., dalam Nielsen, 1994).

Beberapa penelitian menunjukkan bahwa *Cognitive Walkthrough* dapat digunakan untuk mengidentifikasi permasalahan pada antarmuka. Yuana Putra dan Waspodo (2024) menggunakan metode ini untuk mengevaluasi navigasi, *icon*, dan struktur menu pada *website*. Khalisatifa dan Ibrahim (2025) juga menunjukkan bahwa metode ini dapat membantu mengevaluasi kemudahan pengguna dalam menemukan dan memahami fitur pada suatu sistem.

Lyon et al. (2021) menerapkan *Cognitive Walkthrough* pada sistem digital dalam konteks pelayanan untuk mengidentifikasi hambatan pada alur kerja dan penggunaan fitur. Metode ini relevan untuk mengevaluasi sistem yang memiliki proses administratif karena dapat menilai kemampuan pengguna dalam memahami alur dan menyelesaikan tugas. Dalam penelitian ini, *Cognitive Walkthrough* digunakan untuk mengevaluasi kemudahan pengguna dalam memahami alur,

menemukan fitur, dan menjalankan fungsi pada Sistem Informasi Pengajuan dan Status Bayar Klaim JKK.

2.2.13 *Laravel Telescope*

Laravel Telescope merupakan *tools debugging* yang digunakan untuk memantau aktivitas aplikasi Laravel selama proses pengembangan. Telescope menampilkan informasi seperti HTTP *request*, *response*, log aplikasi, *query database*, dan *exception*. Informasi tersebut membantu pengembang dalam mengidentifikasi kesalahan dan memantau proses aplikasi (Laravel, 2024).

Laravel Telescope menyediakan fitur pemantauan *request*, pencatatan log, pelacakan *exception*, serta informasi waktu respons aplikasi. Fitur tersebut dapat membantu pengembang memeriksa komunikasi antara *front-end* dan *back-end* serta menemukan kesalahan pada proses integrasi data. Informasi yang ditampilkan secara terstruktur juga mempermudah proses *debugging* dan evaluasi aplikasi (Otwell, 2024).

Dalam pengembangan *front-end* berbasis React JS, Laravel Telescope dapat digunakan untuk memantau pertukaran data antara antarmuka dan *back-end*. Pengembang dapat memeriksa *request*, *response*, dan waktu eksekusi untuk memastikan proses integrasi berjalan sesuai fungsi sistem. Dalam penelitian ini, Laravel Telescope digunakan sebagai *tools* pendukung untuk memeriksa proses integrasi antara *front-end* dan *back-end*.