INFOMATEK: Jurnal Informatika, Manajemen dan Teknologi
Volume 27 Nomor 1, Juni 2025

TREN DAN PRAKTIK TERBAIK DALAM PENGEMBANGAN WEB BERBASIS API : KAJIAN LITERATUR TERHADAP FRAMEWORK LARAVEL DAN REACT

Fauzan Prasetyo Eka Putra, Reynal Widya Efendi, Alief Badrit Tamam*, Walid Agel Pramadi

Teknik Informatika, Universitas Madura, Pamekasan, Indonesia

Abstrak: Pengembangan web modern mengalami pergeseran paradigma dengan meningkatnya adopsi arsitektur berbasis API yang memisahkan tanggung jawab antara frontend dan backend. Kajian ini bertujuan untuk menganalisis tren serta praktik terbaik dalam pengembangan aplikasi web berbasis API dengan menggunakan Laravel sebagai framework backend dan React sebagai library frontend. Metode yang digunakan adalah tinjauan literatur sistematis terhadap publikasi akademik dalam tiga tahun terakhir. Hasil kajian menunjukkan bahwa integrasi Laravel dan React menawarkan fleksibilitas tinggi dalam membangun aplikasi yang skalabel, maintainable, dan responsif. Beberapa praktik terbaik yang diidentifikasi meliputi penerapan prinsip RESTful API, pengelolaan state secara efisien di React, serta penggunaan dokumentasi API yang terstruktur. Tantangan utama dalam integrasi keduanya mencakup pengelolaan otentikasi, sinkronisasi data, serta masalah lintas domain (CORS). Selain itu, tren seperti arsitektur headless, SPA, SSR, dan microservices semakin banyak diadopsi dalam konteks pengembangan modular dan berorientasi performa. Kajian ini memberikan wawasan strategis bagi pengembang dan peneliti untuk memahami dinamika dan arah evolusi arsitektur aplikasi web berbasis API menggunakan Laravel dan React.

Kata kunci: Laravel, React, API, Arsitektur Web, Single Page Application

I. PENDAHULUAN

Perkembangan teknologi web mengalami transformasi besar dalam dekade terakhir dengan meningkatnya adopsi arsitektur berbasis API. Arsitektur ini memungkinkan pemisahan antara frontend dan backend, memberikan fleksibilitas tinggi dalam desain, pengembangan, dan pemeliharaan sistem. Laravel, sebagai framework PHP modern, dan React, library frontend dari JavaScript, menjadi kombinasi dominan dalam membangun aplikasi web yang responsif dan scalable (Hasanuddin, Asgar, & Hartono, 2022)(Sinlae, Irwanda, Maulana, & Syahputra, 2024).

Dalam konteks ini, konsep *separation of concerns* memperoleh signifikansi besar. Dengan menjadikan API sebagai antarmuka komunikasi utama, pengembang dapat memanfaatkan keunggulan masing-masing teknologi tanpa harus terikat secara langsung (Khalfallah, 2024). Laravel menyediakan backend yang kokoh dan efisien, sedangkan React menyajikan antarmuka pengguna yang dinamis. arsitektur ini mampu meningkatkan maintainability dan agility pada proyek berskala besar (Herdiyatmoko, 2022).

*) aliefbadrittamam@gmail.com

Diterima: 12 Mei 2025
Direvisi: 2 Juni 2025
Disetujui:28 Juni 2025
DOI: 10.23969/infomatek.v27i1.25122

Di tengah adopsi model API-driven, pendekatan *Single Page Application* (SPA) dan *Headless Architecture* menjadi dua tren utama. SPA memungkinkan pengalaman pengguna yang lebih lancar dengan memuat data secara dinamis tanpa reload halaman, sedangkan headless mengizinkan backend dan frontend berkembang secara independen. Menurut Shah (2024), pendekatan ini menjadi pilihan utama dalam desain sistem enterprise yang kompleks karena mampu mengakomodasi kebutuhan omnichannel (Rueckauer et al., 2022) (Shah, 2024).

Meskipun begitu, praktik terbaik (best practices) dalam penggunaan Laravel dan React dalam konteks API masih terus dieksplorasi. Beberapa studi menekankan pentingnya penerapan standar RESTful API, penggunaan middleware untuk otorisasi, dan penerapan manajemen state yang efisien di React. dokumentasi API yang baik, seperti menggunakan Swagger/OpenAPI, merupakan elemen krusial dalam pengembangan dalam tim.

Tantangan teknis juga muncul, terutama dalam integrasi antara Laravel dan React. Salah satu isu umum adalah masalah *CORS* (Cross-Origin Resource Sharing), serta sinkronisasi antara lifecycle React dengan data asynchronous dari API. Selain itu, optimalisasi performa menjadi perhatian utama ketika mengimplementasikan server-side rendering (SSR) atau static site generation (SSG), seperti yang dicatat oleh Ribeiro (Reid, 2024).

Di sisi lain, transformasi ke arsitektur berbasis microservices mendorong pengembangan sistem yang modular dan mudah di-scale. Laravel mulai diadopsi sebagai micro-backend dalam sistem yang lebih besar, sementara React digunakan dalam konteks *micro frontends*. menyatakan bahwa komposisi arsitektur ini sangat cocok untuk organisasi yang menganut pendekatan DevOps dan Continuous Deployment.

Adopsi pendekatan SSR dan SSG juga menjadi perhatian karena mampu meningkatkan performa awal dan SEO pada aplikasi web. React, melalui framework seperti Next.js, memungkinkan implementasi SSR secara efisien. Sementara Laravel melalui Inertia.js atau Laravel Livewire dapat menawarkan SSR hybrid dengan pendekatan PHP-native. Praktik ini menunjang kebutuhan modern akan *progressive enhancement* dan kecepatan akses (Ardiyanto & Ardhianto, 2024) (H. P. Putra & Sari, 2024).

Dengan mengkaji berbagai publikasi terbaru, penelitian ini bertujuan untuk menyajikan ringkasan komprehensif mengenai tren, tantangan, dan praktik terbaik dalam pengembangan aplikasi berbasis Laravel dan React dalam kerangka arsitektur API modern. Penelitian ini diharapkan dapat memberikan wawasan berharga bagi praktisi dan akademisi yang ingin mengadopsi pendekatan arsitektur yang scalable, fleksibel, dan berorientasi masa depan (Fauzan, 2024) (Prihantari, 2024).

II. METODOLOGI

Penelitian dalam artikel ini menggunakan **jenis penelitian kualitatif** dengan pendekatan **studi literatur (literature review)** (Kurniawan, Agoestanto, & Wijayanti, 2023). Penelitian ini tidak melibatkan eksperimen langsung terhadap subjek, melainkan mengkaji, menelaah, dan menganalisis berbagai sumber pustaka yang relevan untuk membangun kerangka pemahaman teoritis terhadap topik yang diteliti. Pendekatan literatur ini bertujuan untuk mengevaluasi dan mensintesis berbagai hasil penelitian terdahulu agar dapat membentuk dasar konseptual yang kuat untuk

mendukung tujuan penelitian (Mitra & Taufik, 2023).

Sumber data yang digunakan dalam penelitian ini adalah **data sekunder** yang diperoleh dari berbagai dokumen ilmiah, jurnal nasional dan internasional, buku akademik, serta hasil penelitian sebelumnya. Penulis menyaring dan memilih sumber-sumber literatur yang relevan dengan topik pembahasan guna menjaga validitas dan keterandalan informasi yang digunakan. Kriteria pemilihan sumber meliputi relevansi topik, reputasi penerbit, serta rentang waktu publikasi, yang difokuskan pada karya ilmiah dalam tiga tahun terakhir untuk memastikan bahwa data yang digunakan bersifat mutakhir (Abidin, Mukhlis, & Zagladi, 2023).

Teknik pengumpulan data dalam studi literatur ini dilakukan melalui **penelusuran sistematis pada basis data ilmiah**, seperti Google Scholar, ResearchGate, dan jurnal terakreditasi nasional maupun internasional. Proses pengumpulan dilakukan dengan menggunakan kata kunci tertentu yang sesuai dengan topik penelitian. Setelah dilakukan penelusuran, data dikumpulkan, diseleksi, dan disimpan untuk kemudian dianalisis. Validasi sumber dilakukan dengan mengevaluasi kredibilitas, relevansi, dan kontribusi sumber terhadap topik kajian(Hasibuan et al, 2023) (Dendi, 2023).

Prosedur analisis data dilakukan melalui **teknik analisis konten (content analysis)**, yang bertujuan untuk mengidentifikasi pola, tema, serta argumen utama dari setiap referensi yang dikaji. Data yang dikumpulkan dikategorikan berdasarkan topik bahasan, dianalisis keterkaitannya, serta disintesis untuk membentuk pemahaman yang integratif terhadap isu penelitian. Pendekatan analisis ini sangat tepat untuk penelitian berbasis literatur karena mampu menggali kedalaman makna dari berbagai sumber secara komprehensif (SITI, 2024).

Karena studi ini tidak melibatkan partisipasi langsung dari individu, maka tidak terdapat subjek atau sampel manusia yang diteliti secara empiris. Namun, dalam konteks pemilihan dokumen sebagai unit analisis, digunakan **purposive sampling**, yaitu pemilihan sumber berdasarkan tujuan tertentu, yaitu relevansi terhadap masalah penelitian. Teknik ini umum digunakan dalam studi literatur karena mampu memastikan bahwa hanya sumber yang bermutu dan sesuai konteks yang dianalisis secara mendalam (Yuniar & Wolor, 2023).

Dengan metode ini, penelitian berhasil mengintegrasikan berbagai hasil dan perspektif sebelumnya untuk mendukung kesimpulan yang valid dan dapat diandalkan. Studi literatur memungkinkan peneliti untuk membangun dasar teoritik yang kuat dan menyusun kerangka analisis yang komprehensif, terutama dalam konteks kajian konseptual atau eksploratif. Validitas diperoleh melalui seleksi sumber yang ketat, sementara reliabilitas dijaga melalui prosedur analisis yang sistematis dan berulang (Hilman, 2018).

III. HASIL DAN PEMBAHASAN

3.1 Tren Penggunaan Laravel dan React dalam Arsitektur Berbasis API

Framework Laravel secara konsisten menempati posisi penting dalam pengembangan backend API modern. Studi literatur menunjukkan bahwa Laravel digunakan luas untuk membangun RESTful API dan implementasi arsitektur headless, memisahkan antara backend dan frontend melalui antarmuka API (Novita, 2024). Di sisi frontend, React menjadi pilihan utama untuk membangun SPA (Single Page Applications)

karena fleksibilitas dan kemampuannya dalam menangani data asinkron melalui API (Prihantari, 2024). Pola arsitektur “separation of concerns” telah menjadi tren dominan, memungkinkan pengembangan yang modular, skalabel, dan dapat dideploy secara independen. Selain menjadi pilihan populer, Laravel dan React sering dipasangkan dalam pengembangan aplikasi modern karena keduanya mendukung prinsip arsitektur berbasis layanan. Laravel menyediakan fitur seperti Laravel Sanctum, Passport, dan API Resources yang menyederhanakan otorisasi dan serialisasi data API. Sementara itu, React sering dikombinasikan dengan pustaka seperti Axios atau React Query untuk mengelola permintaan HTTP dan state asinkron (Iswari, 2022) (Kristianto, 2022).

Tren adopsi ini didukung oleh meningkatnya kebutuhan akan aplikasi interaktif dan real-time yang terpisah antara client-side dan server-side, yang memungkinkan deployment pipeline dan scaling yang lebih efisien. Arsitektur ini juga memudahkan pengembangan tim lintas platform (backend dan frontend) secara paralel.

Tabel 1. Perbandingan Fitur Laravel dan React untuk Arsitektur API

Aspek	Laravel (Backend)	React (Frontend)
Arsitektur	RESTful API, Headless Backend	SPA (Single Page Application), Component-Based
Komunikasi Data	API Resources, Eloquent, Middleware	Axios, React Query, Fetch API
Autentikasi & Keamanan	Laravel Sanctum / Passport	JWT Handling, Token Storage
Skalabilitas	Modular melalui Service Provider & Route Groups	Reusable Components, Virtual DOM
Dokumentasi API	Laravel Swagger / Scribe	Integrasi mudah dengan dokumentasi berbasis JSON
Dukungan Ekosistem	Composer Packages, Laravel Ecosystem	NPM Packages, React Ecosystem

Tabel di atas menunjukkan bagaimana Laravel dan React saling melengkapi dalam membangun arsitektur API yang efisien dan modern. Laravel menyediakan fondasi backend yang kuat melalui fitur seperti API Resources, Eloquent ORM, serta sistem otentikasi seperti Sanctum dan Passport yang mendukung keamanan data dalam komunikasi client-server(Branco, Aversa, & Venticinque, 2023). Dukungan dokumentasi API seperti Scribe atau Swagger juga mempermudah integrasi antar tim dan validasi endpoint. Kemampuan Laravel dalam memfasilitasi pengembangan modular melalui service provider dan route groups menjadikannya ideal untuk sistem berskala besar yang membutuhkan skalabilitas tinggi (Pradana, 2022).

Di sisi lain, React unggul dalam membangun antarmuka pengguna yang interaktif dan responsif. Dengan pendekatan component-based dan pengelolaan data asinkron menggunakan Axios atau React Query, React mampu menghadirkan pengalaman SPA

(Single Page Application) yang seamless. Kombinasi ini memperkuat prinsip “separation of concerns” yang memungkinkan pengembangan frontend dan backend secara paralel, meningkatkan efisiensi tim lintas fungsi(Iswari, 2022). Adopsi Laravel dan React secara bersamaan tidak hanya mencerminkan tren teknologi, tetapi juga merespons kebutuhan aplikasi masa kini yang menuntut kecepatan, modularitas, dan kemudahan integrasi lintas platform (Bismoputro, Huda, & Brata, 2024).

3.2 Praktik Terbaik dalam Pembuatan API dengan Laravel

Pengembangan API dengan Laravel telah menjadi pilihan utama dalam berbagai proyek perangkat lunak karena kemampuannya menyediakan arsitektur yang terstruktur, efisien, dan mudah dipelihara. Sejumlah penelitian dan publikasi ilmiah dalam tiga tahun terakhir menyoroti praktik terbaik berikut ini:

3.3 Arsitektur Modular dan Penggunaan MVC

Laravel menerapkan pola arsitektur Model-View-Controller (MVC) yang sangat membantu dalam memisahkan logika bisnis, tampilan, dan data. Hal ini membuat kode lebih terstruktur, mudah dikembangkan, dan dipelihara. Studi perbandingan antara Laravel dan PHP Native menunjukkan bahwa Laravel unggul dalam efisiensi kode, struktur routing URL, dan model arsitektur proyek. Penggunaan Eloquent ORM pada Laravel juga memudahkan interaksi dengan database tanpa perlu menulis SQL secara manual, sehingga proses pengembangan REST API menjadi lebih modular dan scalable (Ariyanto, Farhan, Rachmad, & Puspitasari, 2024).

3.4 Dokumentasi dan Iterasi Pengembangan

Setiap tahapan pengembangan API, mulai dari analisis kebutuhan, desain sistem, pembangunan, hingga pengujian, perlu didokumentasikan dengan baik. Dokumentasi API sangat penting untuk pihak ketiga yang akan memanfaatkan layanan tersebut. Studi pengembangan REST API di Indonesia menekankan pentingnya dokumentasi dan penggunaan metode incremental, di mana setiap iterasi meliputi analisis kebutuhan, desain (class diagram dan ERD), pembangunan endpoint, serta pengujian menggunakan alat seperti Postman untuk memastikan API sesuai dengan kebutuhan fungsional (Filiana, Rini, Prabawati, & Samat, 2022).

3.5 Otentikasi dan Middleware

Keamanan API merupakan aspek krusial. Laravel menyediakan dukungan otentikasi token-based melalui middleware dan sistem otorisasi yang terintegrasi. Penelitian internasional menyoroti pentingnya penerapan authentication API dan middleware untuk menjaga keamanan data serta memastikan hanya pengguna yang berhak dapat mengakses sumber daya tertentu. Framework seperti Laravel juga menyediakan kemudahan dalam implementasi token-based authentication, sehingga API lebih aman dan terjaga integritasnya.

3.6 Pengujian dan Evaluasi API

Pengujian API secara menyeluruh sangat dianjurkan untuk memastikan tidak terdapat bug atau error pada setiap proses CRUD (Create, Read, Update, Delete). Pengujian dilakukan menggunakan alat seperti Postman, yang mampu mengirimkan HTTP request ke endpoint API dan memverifikasi respons yang diterima. Studi empiris menunjukkan bahwa

dengan struktur routing dan controller yang baik di Laravel, proses pengujian menjadi lebih mudah dan sistematis (Ariyanto et al., 2024).

3.7 Migrasi ke Arsitektur Microservices

Dalam kasus sistem yang kompleks dan berkembang, migrasi dari arsitektur monolitik ke microservices menjadi praktik terbaik. Setiap komponen sistem dipecah menjadi layanan-layanan kecil yang saling berkomunikasi melalui REST API, sehingga pengembangan, pemeliharaan, dan scaling menjadi lebih mudah. Studi kasus di Indonesia menunjukkan bahwa pendekatan ini, dikombinasikan dengan Domain Driven Design, memudahkan pengembang baru dalam memahami dan mengembangkan system (Maharana & Acharya, 2024).

3.8 Praktik Terbaik Konsumsi API di React

Konsumsi API di React memainkan peran yang sangat vital dalam pengembangan aplikasi berbasis web. Interaksi antara *frontend* dan *backend* yang efisien tidak hanya bergantung pada alat yang digunakan untuk mengirimkan permintaan, tetapi juga pada bagaimana data tersebut dikelola dan disajikan kepada pengguna. Oleh karena itu, terdapat beberapa praktik terbaik yang perlu diperhatikan untuk memastikan aplikasi React dapat mengelola API secara efektif dan responsif (R. A. Putra, 2024).

3.9 Pemilihan Alat HTTP yang Tepat

Hal pertama yang perlu diperhatikan adalah pemilihan alat untuk melakukan permintaan HTTP. React menyediakan dua pilihan utama untuk konsumsi API: **Fetch API** dan **Axios**. Fetch API adalah fitur bawaan JavaScript yang memungkinkan pengambilan data dari server, namun ia memiliki beberapa keterbatasan. Misalnya, Fetch API membutuhkan penanganan error secara manual dan tidak memiliki dukungan untuk pengaturan konfigurasi global, seperti *base URL* atau pengaturan header otorisasi (Yuliadarnita, et al., 2023).

Sebaliknya, **Axios** menawarkan berbagai fitur tambahan yang lebih fleksibel. Axios menyediakan sintaks yang lebih ringkas, mendukung *interceptors* untuk menangani permintaan sebelum dikirim atau respons setelah diterima, serta memungkinkan konfigurasi global untuk hal-hal seperti *base URL*, otentikasi, dan pengaturan *headers*. Dengan keunggulan-keunggulan ini, Axios banyak dipilih dalam proyek aplikasi yang lebih besar dan kompleks, karena memungkinkan pengelolaan permintaan dan respons data yang lebih terstruktur dan mudah dipelihara (Ramai, Facciorusso, et al., 2022) (Rahmadhani et al., 2024).

3.10 Manajemen Status Data yang Terorganisir

Mengelola status data yang diambil dari API adalah bagian penting dalam pengembangan aplikasi React. Status data ini harus dikelola dengan cara yang efisien agar aplikasi tetap responsif dan dapat menangani data secara optimal. Di React, ada beberapa pendekatan yang umum digunakan untuk manajemen status (Hesti Syafitri, Hamka, & Yusuf, 2024).

React Context sering digunakan untuk aplikasi dengan skala kecil hingga menengah, di mana status global seperti informasi pengguna atau preferensi dapat dibagikan di seluruh komponen aplikasi tanpa membebani kinerja. Namun, untuk aplikasi yang lebih kompleks, **Redux Toolkit** menjadi pilihan populer. Redux Toolkit memberikan struktur yang lebih jelas untuk manajemen status dengan memisahkan logika bisnis dan antarmuka pengguna. Redux menawarkan kontrol penuh terhadap alur data dan memastikan bahwa status aplikasi selalu

dapat diprediksi (Kadriu & Bilalli, n.d.) (Kumar, 2024).

Selain itu, **React Query** adalah pustaka yang dirancang khusus untuk manajemen *server state* atau status data yang berasal dari server. React Query tidak hanya menyederhanakan proses pengambilan data dari API, tetapi juga menangani hal-hal seperti caching, pengambilan ulang data otomatis saat halaman difokuskan kembali, dan sinkronisasi data real-time. Pendekatan ini mengurangi kebutuhan kode boilerplate dan meningkatkan kinerja aplikasi secara keseluruhan, terutama untuk aplikasi yang mengandalkan data eksternal dalam jumlah besar.

3.11 Penanganan Status Respons API yang Jelas

Salah satu elemen yang tak kalah penting adalah penanganan status respons dari API. Pengalaman pengguna dapat sangat terpengaruh oleh bagaimana aplikasi memberikan feedback terkait permintaan data. Oleh karena itu, sangat penting untuk memberikan indikator yang jelas tentang status permintaan, apakah itu *loading*, *error*, atau *success*. Sebagai contoh, menampilkan spinner atau progress bar saat data sedang diproses memberikan indikasi visual yang jelas bahwa aplikasi sedang bekerja.

Teknik lain yang semakin populer adalah **optimistic UI**, di mana aplikasi menampilkan hasil perubahan data seolah-olah perubahan tersebut berhasil dilakukan, bahkan sebelum server mengonfirmasi. Teknik ini sangat berguna untuk meningkatkan persepsi kecepatan aplikasi, terutama dalam aplikasi dengan latensi tinggi seperti aplikasi e-commerce atau aplikasi berbasis *real-time*. Dengan menggunakan teknik ini, pengguna tidak merasa aplikasi lambat meskipun data yang diminta masih dalam proses pengambilan.

3.12 Pengelolaan Konfigurasi dengan .env

Selain aspek teknis di atas, penting juga untuk mengelola variabel konfigurasi yang berkaitan dengan API secara terpisah dari kode sumber. Salah satu cara yang paling umum adalah dengan menggunakan file **.env**, yang memungkinkan pengelolaan variabel lingkungan dengan cara yang terstruktur. File ini digunakan untuk menyimpan informasi sensitif seperti *base URL* API, token otorisasi, dan parameter konfigurasi lainnya. Sebagai contoh:

```
REACT_APP_API_BASE_URL=https://api.example.com
REACT_APP_API_KEY=abcdef123456
```

Dengan pendekatan ini, pengembang dapat dengan mudah mengganti konfigurasi aplikasi saat berpindah antara berbagai lingkungan (misalnya *development*, *staging*, dan *production*) tanpa perlu mengubah kode program. Selain itu, penggunaan file .env juga meningkatkan keamanan aplikasi karena informasi sensitif tidak disertakan langsung dalam kode yang dapat diakses publik.

Secara keseluruhan, konsumsi API di React melibatkan lebih dari sekadar melakukan permintaan data dari server. Ini mencakup pemilihan alat yang tepat, pengelolaan status data secara efisien, serta penerapan teknik untuk meningkatkan pengalaman pengguna, seperti penanganan status respons yang jelas dan *optimistic UI*. Selain itu, pengelolaan konfigurasi dengan file .env memberikan fleksibilitas dan keamanan yang diperlukan dalam pengembangan aplikasi yang skalabel dan mudah dikelola.

Dengan mematuhi praktik terbaik ini, pengembang dapat membangun aplikasi

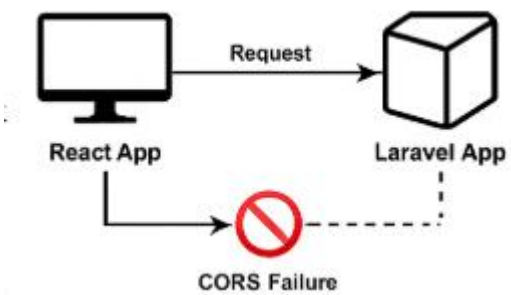
React yang tidak hanya efisien dan responsif, tetapi juga aman dan mudah dipelihara dalam jangka panjang.

Dengan gaya ini, materi lebih mengalir dalam bentuk teks yang mudah diikuti, namun tetap memuat elemen-elemen yang terstruktur sehingga pembaca tidak bingung.

3.13 Tantangan Umum yang Ditemukan dalam Literatur

Dalam proses integrasi antara React sebagai *frontend framework* dan Laravel sebagai *backend framework*, tantangan teknis yang paling umum dijumpai adalah permasalahan Cross-Origin Resource Sharing (CORS) (Kumaladewi et al, 2023). React, yang secara default berjalan di localhost pada port berbeda dari Laravel, membutuhkan akses ke API lintas domain. Jika header CORS seperti Access-Control-Allow-Origin, Access-Control-Allow-Methods, dan Access-Control-Allow-Headers tidak dikonfigurasi secara eksplisit pada Laravel, maka permintaan HTTP dari React akan diblokir oleh browser. Masalah ini dapat diatasi melalui middleware Laravel atau konfigurasi CORS di file cors.php, namun seringkali terjadi miskonfigurasi yang menyebabkan pengembang kesulitan dalam fase awal integrasi (Astowo, 2024) (Purnomo, 2024).

Dari sisi React, tantangan muncul pada bagaimana aplikasi menangani proses asynchronous. React memanfaatkan hooks seperti useEffect untuk mengeksekusi proses seperti pengambilan data dari API. Namun, tanpa pemahaman yang cukup mengenai *component lifecycle* dan prinsip re-rendering, pengembang pemula rentan mengalami *race condition*. Misalnya, ketika dua proses asynchronous berjalan bersamaan tanpa kontrol yang tepat, maka data yang masuk bisa tidak sinkron dengan status aplikasi. Oleh karena itu, diperlukan strategi pengelolaan state yang jelas, seperti penggunaan useReducer, konteks global, atau pustaka seperti Redux untuk menangani alur data secara lebih stabil (Yuanita, et al, 2022) (Pramadipta, 2024).



Gambar 1. Ilustrasi CORS Failure

sinkronisasi otentikasi menjadi salah satu area yang paling krusial dalam pengembangan fullstack. Laravel menyediakan solusi otentikasi seperti Sanctum dan Passport yang mengandalkan token berbasis HTTP-only cookies atau Bearer Token (Ratino, Astri, & Anggraini, 2023). Namun, dari sisi React, pengembang harus memastikan token disimpan secara aman, misalnya dengan menghindari penyimpanan di localStorage yang rentan terhadap serangan XSS. Tantangan juga muncul ketika mengelola *refresh token*, *session expiration*, dan status login pengguna secara real-time, yang menuntut implementasi logika autentikasi yang terkoordinasi antara dua sistem (Fernandes, 2024) (Ray, 2025).

3.14Tren Arsitektur yang Berkembang

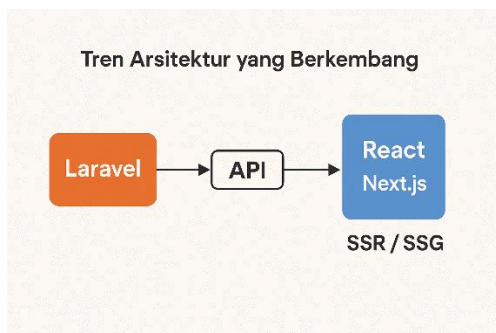
Seiring dengan meningkatnya kebutuhan akan fleksibilitas dan skalabilitas dalam pengembangan aplikasi web modern, arsitektur monolitik mulai tergantikan oleh pendekatan arsitektur yang lebih modular. Salah satu tren yang menonjol adalah pergeseran menuju *headless architecture*, di mana backend dan frontend dipisahkan sepenuhnya. Laravel, dalam konteks ini, tidak lagi digunakan sebagai platform all-in-one, melainkan sebagai service backend yang hanya bertugas menyediakan API berbasis

REST atau GraphQL (Basatha et al., 2024) (Arif, 2024). Frontend dikembangkan secara independen menggunakan React, memungkinkan pengembangan antarmuka pengguna yang lebih dinamis dan dapat dikembangkan secara paralel.

Pendekatan *microservices* juga menjadi perhatian utama dalam arsitektur modern. Alih-alih membangun sistem sebagai satu kesatuan besar, pengembang kini lebih memilih memecah sistem menjadi layanan-layanan kecil yang saling berkomunikasi melalui API. Dengan Laravel yang berperan sebagai salah satu service, sistem menjadi lebih mudah di-maintain, di-deploy secara terpisah, serta di-scale sesuai kebutuhan. Implementasi ini sering kali dikombinasikan dengan penggunaan Docker, orchestrator seperti Kubernetes, dan CI/CD pipeline untuk mendukung proses pengembangan berkelanjutan (Aprilia & Mulianingtyas, n.d.).

Sementara itu, penggunaan GraphQL sebagai alternatif REST API juga mulai populer di kalangan pengembang React. GraphQL memungkinkan klien menentukan data apa saja yang ingin diambil dalam satu query, sehingga lebih efisien dibandingkan REST yang sering mengembalikan data secara berlebih atau bertingkat. Dengan GraphQL, frontend dapat mengoptimalkan pengambilan data tanpa harus bergantung pada struktur response yang rigid, yang sangat sesuai dengan arsitektur headless dan kebutuhan antarmuka yang responsif et al, 2024) (Biruni, Faisol, & Vendyansyah, 2023).

Di ekosistem React, integrasi dengan teknologi seperti Next.js juga turut membentuk tren baru dalam pengembangan frontend. Next.js menyediakan dukungan untuk *Server-Side Rendering* (SSR) dan *Static Site Generation* (SSG), yang sangat ideal untuk kebutuhan performa tinggi dan optimalisasi SEO. Dalam skenario ini, Laravel hanya bertindak sebagai penyedia data, sedangkan proses render halaman dilakukan di server atau saat build time melalui Next.js (Nugroho, 2024). Kombinasi Laravel sebagai API service dan Next.js sebagai rendering engine dianggap sebagai pendekatan arsitektur yang efisien dan modern, yang mampu memberikan pengalaman pengguna yang cepat sekaligus memudahkan pengindeksan oleh mesin pencari (Ardiyanto & Ardhianto, 2024).



Gambar 2. Ilustrasi Tren Aksitektur yang Berkembang

Hasil kajian literatur ini mengindikasikan bahwa integrasi Laravel sebagai kerangka kerja backend dengan React sebagai antarmuka frontend menjadi pendekatan yang semakin luas diadopsi dalam pengembangan aplikasi web berbasis API. Laravel menyediakan arsitektur yang solid untuk pengelolaan logika bisnis, autentikasi, serta pembuatan RESTful maupun GraphQL API secara efisien dan terstruktur(Wardani, Arifianto, Sirojudin, Aziza, & Habibie, 2024) (Faturahman et al., 2022). Di sisi lain, React mendukung pembuatan antarmuka pengguna yang dinamis dan responsif dengan prinsip komponen yang terpisah dan dapat digunakan kembali (reusable). Kombinasi ini mencerminkan adopsi arsitektur modern berbasis *decoupled system* dan *separation of concerns*, yang memungkinkan pengembangan frontend dan backend secara

paralel serta meningkatkan skalabilitas sistem (Pohan, 2024) (Dawis, et al, 2025).

Temuan ini konsisten dengan beberapa studi sebelumnya yang menekankan keunggulan Laravel dalam aspek keamanan, dokumentasi API, serta kemudahan integrasi dengan middleware untuk otorisasi. Sementara itu, React terbukti efektif dalam pengelolaan *state*, pemrosesan data asinkron, serta optimalisasi rendering antarmuka melalui pendekatan SPA (Single Page Application). Penelitian ini memperluas temuan-temuan tersebut dengan menggabungkan wawasan terkini, termasuk praktik implementasi *headless architecture*, penerapan *microservices*, serta pemanfaatan teknologi pendukung seperti Axios, React Query, dan Next.js sebagai bagian dari ekosistem modern (Aji, 2023) (Sanjaya, 2025).

Adopsi Laravel dan React secara simultan tidak terlepas dari kebutuhan industri terhadap sistem informasi yang bersifat modular, efisien, dan mudah dipelihara. Laravel mendukung prinsip *convention over configuration* yang mempercepat proses pengembangan, sementara React menawarkan fleksibilitas tinggi dalam membangun antarmuka pengguna yang terintegrasi dengan API secara real-time. Selain itu, pendekatan ini juga mendukung praktik DevOps modern, seperti integrasi dengan Docker dan penerapan CI/CD (Continuous Integration/ Continuous Deployment), yang semakin relevan dalam pengembangan perangkat lunak berskala besar dan berkelanjutan (Zebua, 2024) (Frabroyir, Kom, & Hasanah, n.d.).

Implikasi dari temuan ini menunjukkan bahwa kombinasi Laravel dan React dapat memberikan efisiensi kerja yang signifikan dalam proses pengembangan perangkat lunak, serta mendukung arsitektur sistem yang berorientasi pada layanan (*service-oriented architecture*). Namun demikian, terdapat beberapa tantangan teknis yang perlu diperhatikan, antara lain konfigurasi CORS (Cross-Origin Resource Sharing), manajemen token autentikasi yang aman (seperti JWT), serta pengelolaan status data pada sisi frontend yang kompleks, khususnya dalam skenario komunikasi data asinkron (Shofyan & Isa, 2024) (Astowo, 2024).

Keterbatasan kajian ini terletak pada pendekatannya yang berbasis literatur sekunder, tanpa didukung oleh studi empiris atau eksperimen implementatif. Oleh karena itu, meskipun hasil kajian ini memberikan gambaran komprehensif mengenai praktik terbaik dan tren teknologi, validitasnya masih bergantung pada kualitas dan kedalaman sumber-sumber yang dianalisis. Selain itu, sebagian besar literatur yang digunakan berasal dari konteks pengembangan lokal (Indonesia), sehingga generalisasi hasil ke konteks global perlu dilakukan secara hati-hati.

Secara keseluruhan, kajian ini menyajikan kerangka analisis yang holistik terhadap penggunaan Laravel dan React dalam pengembangan aplikasi berbasis API, serta menegaskan relevansi kombinasi keduanya dalam memenuhi tuntutan arsitektur perangkat lunak modern. Diharapkan bahwa hasil kajian ini dapat menjadi landasan teoretis bagi pengembangan penelitian lanjutan, sekaligus menjadi acuan praktis bagi pengembang dalam membangun sistem informasi yang adaptif, modular, dan berorientasi pada efisiensi.

IV. KESIMPULAN

Hasil studi literatur ini menunjukkan bahwa integrasi antara Laravel sebagai backend dan React sebagai frontend dalam pengembangan aplikasi berbasis API memberikan pendekatan yang efisien dan modular dalam membangun sistem informasi modern. Laravel memberikan kemudahan dalam pengelolaan logika bisnis, keamanan, serta penyusunan arsitektur API yang rapi dan terstruktur. Di sisi lain, React

memungkinkan pengembangan antarmuka pengguna yang interaktif dan responsif dengan manajemen state yang efisien melalui pendekatan komponen. Kombinasi keduanya memberikan kelebihan dalam hal pemisahan tanggung jawab kerja (separation of concerns), kemudahan skalabilitas sistem, dan mendukung integrasi dengan layanan pihak ketiga.

Namun demikian, implementasi gabungan ini juga memiliki beberapa kelemahan, seperti kompleksitas awal dalam konfigurasi, khususnya terkait pengaturan CORS, autentikasi token JWT, dan kebutuhan pemahaman yang mendalam pada kedua teknologi secara bersamaan. Selain itu, komunikasi data antara frontend dan backend yang tidak diatur dengan baik dapat menyebabkan inkonsistensi atau kerentanan dalam sistem. Meskipun demikian, pendekatan ini tetap menawarkan peluang besar untuk pengembangan sistem yang berkelanjutan dan adaptif terhadap kebutuhan masa depan.

Kemungkinan pengembangan selanjutnya meliputi integrasi dengan teknologi pendukung lain seperti GraphQL untuk efisiensi data, serta penggunaan Docker untuk meningkatkan portabilitas dan konsistensi lingkungan pengembangan. Untuk mengatasi keterbatasan dalam penelitian ini yang bersifat literatur, disarankan adanya penelitian lanjutan dalam bentuk implementasi langsung di lingkungan riil, seperti studi kasus proyek perangkat lunak atau eksperimen sistem prototipe. Penelitian tersebut dapat menilai performa, keamanan, kemudahan pemeliharaan, serta tingkat kepuasan pengguna dalam menggunakan sistem berbasis Laravel dan React secara terintegrasi.

DAFTAR PUSTAKA

Abidin, H., Mukhlis, I., & Zagladi, A. N. (2023). Multi-method Approach for Qualitative Research: Literature Review with NVivo 12 PRo Mapping. *Kalam Cendekia: Jurnal Ilmiah*

Aji, B. S. (2023). 2023 Complete Front-End Engineer Career With ReactJS Di PT Marka Kreasi Persada. repository.unisbablitar.ac.id.

Aprilia, A., & Mulianingtyas, O. (n.d.). Orkestrasi Continuous Integration/ Continuous Delivery (Ci/Cd) dan Automated Testing Pada Devops Marketplace *Journal.Uinsi.Ac.Id*.

Ardiyanto, R., & Ardhianto, E. (2024). Analisa Performasi Metode Client Side Rendering, Server Side Rendering, dan Incremental Static Regeneration dalam Proses Website Rendering. *Computer Science (CO-SCIENCE)*. 103.75.24.116.

Arif, A. H. (2024). *Analisis Performa Model Protokol API REST, SOAP, GraphQL dan RPC dalam Simulasi Create, Read, Update, Delete, Sorting, dan Searching Data*. rama.unimal.ac.id.

Ariyanto, Y., Farhan, M., Rachmad, F., & Puspitasari, D. (2024). Laravel Framework and Native PHP : Comparison in the Creation of Rest API. *Matrix: Jurnal Manajemen Teknologi Dan Informatika*, *14*(2), 66–73.

Astowo, U. B. (2024). *Desain Komunikasi dan Keamanan Data Arsitektur Aplikasi Multimasjid*. dspace.uii.ac.id.

Basatha, R., MT, M., Chafid, N., Kom, S., Kom, M., Wihardjo, E., & ... (2024). *Pemrograman Web Dan Aplikasi Mobile*. books.google.com.

Biruni, M. Al, Faisol, A., & Vendyansyah, N. (2023). Penerapan Rest Api Dan Integrasi Midtrans Sebagai Payment Gateway Pada Platform Pelatihan Online. *JATI (Jurnal Mahasiswa ….*

Bismoputro, I., Huda, F. Al, & Brata, A. H. (2024). Pengembangan Single Page Application Berbasis Reactjs Untuk Usaha Percetakan Online (Studi Kasus: Global Grafika). *Jurnal Pengembangan Teknologi*

Branco, D., Aversa, R., & Venticinque, S. (2023). A tool for creation of virtual exhibits presented as IIIF collections by intelligent agents. *International Conference on …*. https://doi.org/10.1007/978-3-031-28694-0_22

Dawis, A. M., Rahmayanti, D., Rachman, T., Impron, A., & ... (2025). Pendekatan Modern Dalam Analisis Dan Desain Teknologi Informasi. researchgate.net.

Dendi, D. A. R. (2023). *Analisis Bibliometrik Publikasi Ilmiah Tentang Pembayaran Bank Syariah Berbasis Data Scopus Periode 2010-2020*. repository.radenintan.ac.id.

Faturahman, E. T., Putra, W. H. N., & ... (2022). Pembangunan Sistem Informasi Pemesanan Jasa Foto berbasis Web menggunakan REST API pada Heroe Photography. *… Teknologi Informasi Dan …*.

Fauzan, A. (2024). *Rancang Bangun Aplikasi Donasi Berbasis Android Dengan Bahasa Pemrograman Kotlin Menggunakan Metode Agile Unified Process* repository.nurulfikri.ac.id.

Fernandes, J. P. (2024). Authentication API-A SSO Authentication and Authorisation Infrastructure for Web.

Filiana, A., Rini, M. N. A., Prabawati, A. G., & Samat, R. A. (2022). Pengembangan Rest Api Untuk Informasi Pasar Tradisional Di Kota Yogyakarta Dengan Metode Incremental. *SINTECH (Science and Information Technology) Journal*, *5*(1), 10–23. https://doi.org/10.31598/sintechjournal.v5i1.1060

Frabroyir, H., Kom, S., & Hasanah, I. (n.d.). Software House: Analisis dan Implementasi Pengembangan Perangkat Lunak Sistem UTBK di PT Aksamedia Mulia Digital. *Repository.Its.Ac.Id*.

Hasanuddin, Asgar, H., & Hartono, B. (2022). Rancang Bangun Rest Api Aplikasi Weshare Sebagai Upaya Mempermudah Pelayanan Donasi Kemanusiaan. *Jurnal Informatika Teknologi Dan Sains*, *4*(1), 8–14. https://doi.org/10.51401/jinteks.v4i1.1474

Hasibuan, N., Zulaikha, S. R., Sari, K. P., & ... (2023). Aksesibilitas Jurnal Elektronik Gale dalam Memenuhi Kebutuhan Informasi Pemustaka di Perpustakaan Politeknik Pembangunan Pertanian Yogyakarta. *Educaniora: Journal of …*.

Herdiyatmoko, H. F. (2022). Desain Sistem Backend Berbasis Rest Api Menggunakan Framework Laravel 7. *Skanika*, *5*(2), 136–144. https://doi.org/10.36080/skanika.v5i2.2947

Hesti Syafitri, Hamka, & Yusuf. (2024). Efektivitas Model Pembelajaran React Dalam Meningkatkan Motivasi Belajar Siswa. *BEGIBUNG: Jurnal Penelitian Multidisiplin*, *2*(1), 346–355. https://doi.org/10.62667/begibung.v2i1.70

Hilman, I. (2018). Penetapan Desa Wirausaha Dan Strategi Pengembangannya. *JIMFE (Jurnal Ilmiah Manajemen Fakultas Ekonomi)*, *3*(2), 28–36. https://doi.org/10.34203/jimfe.v3i2.644

Iswari, S. T. L. (2022). *Implementasi Mongo Db, Express Js, React Js Dan Node Js (Mern) Pada Pengembangan Aplikasi Formulir, Kuis, Dan Survei Online*. dspace.uii.ac.id.

Kadriu, A., & Bilalli, S. (n.d.). Comparison of Context API and Redux Working with Complex State. *Repository.Seeu.Edu.Mk*.

Khalfallah, H. Ben. (2024). HOFA: The Path Toward Clean Architecture. *Crafting Clean Code with JavaScript and React: A …*. https://doi.org/10.1007/979-8-8688-1004-6_4

Kristianto, S. (2022). *Analisis Dan Penyewaan Sistem Informasi Kost-An Di Kota Tangerang*. repositori.buddhidharma.ac.id.

Kumaladewi, N., Refardi, G. N., & ... (2023). Design and Build a Web-Based e-Learning System using ReactJS Framework. *2023 11th International …*.

Kumar, T. (2024). *Fluent React: Build Fast, Performant, and Intuitive Web Applications*. books.google.com.

Kurniawan, M. R., Agoestanto, A., & Wijayanti, K. (2023). Systematic literature review: identifikasi kemampuan berpikir aljabar dan resiliensi matematis pada pembelajaran matematika. *Jurnal Cendekia: Jurnal …*. core.ac.uk.

Maharana, K. C., & Acharya, S. (2024). Laravel- A Centralized framework with authentication API. *SSRN Electronic Journal*, *4*(6), 152–156. https://doi.org/10.2139/ssrn.4909110

Mitra, Y., & Taufik, T. (2023). Penerapan Model Discovery Learning (Dl) Dalam Pembelajaran Tematik Terpadu Di Kelas IV Sekolah Dasar (Studi Literatur). *E-Jurnal Inovasi Pembelajaran Sekolah Dasar*.

Novita, I. (2024). Implementasi Model E-Commerce Berbasis Wordpress Pada Bengkel Las Trian Jaya. *Jurnal Mahasiswa Ilmu Komputer*.

Nugroho, R. P. A. (2024). Meningkatkan Performa Frontend dengan Menggunakan Framework Next. Js dalam Pengembangan Website. *Journal of Cyber Health and Computer*.

Pohan, H. M. (2024). Desain Arsitektur Fleksibel: Membangun Lingkungan Yang Responsif Terhadap Perubahan. *WriteBox*.

Pradana, L. (2022). *TA: Rancang Bangun Web Service Api Dan Dokumentasi Rest Api Web Portal Unit Kegiatan Mahasiswa Di Politeknik Negeri Lampung*. repository.polinela.ac.id.

Pramadipta, M. B. (2024). Rancang Bangun Frontend Website Untuk Pemungutan Suara Dengan Menggunakan React.Js. *Jurnal Informatika Dan Teknik Elektro Terapan*, *12*(2). https://doi.org/10.23960/jitet.v12i2.4173

Prihantari, R. A. (2024). *Rancang Bangun Sistem Informasi E-Commerce Berbasis Web Menggunakan Framework Spring Boot Pada Toko Gigha Steel*. repository.nurulfikri.ac.id.

Purnomo, C. A. (2024). Pengembangan Sistem Monitoring Pertanian Dalam Greenhouse Untuk Mendukung Smart Precision Farming 4.0: Modul digilib.uns.ac.id.

Putra, H. P., & Sari, A. P. (2024). Implementasi Server Side Rendering Pada Sistem Travel Berbasis Website. *Prosiding Seminar Nasional Informatika …*.

Putra, R. A. (2024). *Perancangan Aplikasi Layanan Pengaduan Customer Berbasis Web Menggunakan React Js: Studi Kasus Pada PT Inovatif 78*. repository.nurulfikri.ac.id.

Rahmadhani, S., Wildana, D. W., & ... (2024). Penerapan React JS dan Axios untuk Pengembangan Front-end Aplikasi

iCare. *Software …*.

Ramai, D., Facciorusso, A., DeLuca, M., & ... (2022). Adverse events associated with AXIOS stents: Insights from the manufacturer and user facility device experience database. *Endoscopic …*. journals.lww.com.

Ratino, A., Astri, R., & Anggraini, P. (2023). Implementasi Framework Laravel Dalam Pengembangan Aplikasi E-Commerce Untuk Toko Jago Software. *Journal Of Informatics And Busisnes*, *01*, 33–43.

Ray, P. P. (2025). A Survey on Model Context Protocol: Architecture, State-of-the-art, Challenges and Future Directions. *Authorea Preprints*. https://doi.org/10.36227/techrxiv.174495492.22752319

Rayhan, Y., Suhayati, M. S. M., & ... (2024). Comparison Of Rest Api And Graphql In The Strapi Content Management System. *Jurnal Riset Teknik …*.

Reid, R. L. (2024). Engineering the Arts. *Civil Engineering Magazine*, *94*(2), 38–49. https://doi.org/10.1061/ciegag.0001713

Rueckauer, B., Bybee, C., Goettsche, R., Singh, Y., Mishra, J., & Wild, A. (2022). NxTF: An API and Compiler for Deep Spiking Neural Networks on Intel Loihi. *ACM Journal on Emerging Technologies in Computing Systems*, *18*(3). https://doi.org/10.1145/3501770

Sanjaya, I. (2025). *Pengembangan Aplikasi Website Informasi Instansi X untuk Efisiensi Pengambilan Data API dan Desain Antarmuka Responsif dengan Next. js*. repository.its.ac.id.

Shah, H. (2024). Navigating UI Engineering Architectures : A Deep Dive into Modern Web Application Design Navigating Ui Engineering Architectures : A Deep Dive, (December). https://doi.org/10.13140/RG.2.2.30307.77603

Shofyan, S., & Isa, S. M. (2024). Perancangan Dasbor yang Secure Scalable dan Reusable dengan Microservices Case Study Di PT. XYZ. *Jurnal Sosial Teknologi*.

Sinlae, F., Irwanda, E., Maulana, Z., & Syahputra, V. E. (2024). Penggunaan Framework Laravel dalam Membangun Aplikasi Website Berbasis PHP. *Jurnal Siber Multi Disiplin (JSMD)*, *2*(2), 119–132.

SITI, A. (2024). *Analisis Isi Pesan Dakwah Pada Akun Vidgram@ Halimahalaydrus*. repository.radenintan.ac.id.

Wardani, A. K., Arifianto, A. S., Sirojudin, A., Aziza, A. N., & Habibie, A. S. (2024). Implementasi Digital Twin Dengan Komunikasi Data Nirkabel pada Liquid Filling Machine. *JUKI : Jurnal Komputer Dan Informatika*, *6*(1), 46–54. https://doi.org/10.53842/juki.v6i1.457

Yuanita, H. I., Wijayanto, B., & ... (2022). Frontend Development Of Course Scheduling System Integrated Sia At Engineering Faculty University Of Jenderal Soedirman Using Devops Method. *Jurnal Teknik Informatika …*.

Yuliadarnita, Y., Febriansyah, M., Wijaya, A., & ... (2023). Analisis Komparatif Aplikasi Open Source Intelligence Berbasis Website Dengan Tools Osint Command Line Kominfo Bengkulu. *Jurnal Media …*.

Yuniar, N. P., & Wolor, C. W. (2023). Analisis Arsip Pada Era Digital Di PT XYZ. *Bridging Journal of Islamic Digital …*.

Zebua, M. (2024). Penggunaan DevOps untuk Meningkatkan Kecepatan Pengembangan Aplikasi. *Circle Archive*.