

# STUDI PERBANDINGAN *FRAMEWORK LARAVEL* DAN *CODEIGNITER 3*: OPTIMALISASI PERFORMA APLIKASI WEB

Muhammad Barkah Sejati

Program Studi Manajemen Informatika, Politeknik LP3I Bandung

Jl. Rereng Adumanis, Bandung

E-mail: arkaaaa27@gmail.com

**Abstrak** – *Framework* web dirancang untuk membantu pengembangan membangun aplikasi dengan lebih baik efektif. *Larave* dan *CodeIgniter 3* merupakan dua *framework PHP* populer yang memiliki pendekatan berbeda. Penelitian ini membandingkan keduanya dalam aspek efisiensi sumber daya, kemudahan penggunaan, optimasi *UI /UX*, serta performa web dengan menerapkan metode *Rapid Application Development (RAD)* dan menggunakan alat uji *Lighthouse*. Hasil analisis menunjukkan bahwa *Laravel* unggul dalam fitur modern serta optimasi *UI*, sehingga lebih sesuai untuk aplikasi berskala besar dan kompleks. Sementara itu, *CodeIgniter* lebih cepat dalam eksekusi dan lebih efisien dalam penggunaan sumber daya, menjadikannya pilihan yang lebih baik untuk proyek dengan keterbatasan *system*. Pengujian *Lighthouse* mengungkap bahwa *Laravel* memiliki waktu muat lebih cepat, sedangkan *CodeIgniter 3* lebih hemat dalam konsumsi memori dan *CPU*. Berdasarkan faktor-faktor tersebut, pemilihan *framework* harus disesuaikan dengan kebutuhan proyek. *Laravel* lebih direkomendasikan untuk aplikasi dengan fitur canggih dan pengalaman pengguna yang lebih baik. Sedangkan *CodeIgniter 3* lebih optimal untuk aplikasi ringan yang membutuhkan kinerja efisien.

**Kata Kunci:** Efisiensi sumber daya, Kemudahan penggunaan, Optimasi *UI /UX*, Performa web, *Rapid Application Development*.

## I. PENDAHULUAN

*Framework* merupakan kumpulan fungsi, kelas, serta seperangkat aturan yang dirancang untuk membantu dalam pengembangan perangkat lunak. Berbeda dengan *library* yang biasanya digunakan untuk tujuan spesifik, *framework* memiliki cakupan yang lebih luas dan berperan dalam mengarahkan serta mengatur proses pembuatan aplikasi secara keseluruhan. Dengan kemajuan teknologi yang begitu cepat, para pengembang dan organisasi perlu terus mengikuti perkembangan serta tren terbaru dalam dunia pengembangan web agar tetap relevan dan kompetitif. Dalam dunia pengembangan web, pilihan *framework* yang tepat sangat berpengaruh pada performa, efisiensi, dan pengalaman pengguna aplikasin yang telah dibangun. Pengembangan aplikasi web mengalami pertumbuhan yang signifikan sejalan dengan meningkatkan permintaan terhadap aplikasi yang responsive.

*PHP (Hypertext Preprocessor)* adalah Bahasa pemrograman yang digunakan untuk mengembangkan situs web dinamis. *PHP* berperan dalam mengelola *database* serta mengatur tampilan konten pada halaman web (Azizah et al., 2024).

*Laravel* dan *CodeIgniter 3* merupakan dua *framework PHP* yang mempunyai pendekatan berbeda dalam pengembangan aplikasi web. *Laravel* mempunyai fitur modernnya seperti *eloquentment ORM* dan *Middleware*, sedangkan untuk *CodeIgniter 3* unggul dalam kecepatan eksekusi dan sederhanaan konfigurasi. Bahasa pemrograman yang salah satu Bahasa *scripting* yang

paling banyak digunakan dalam pengembangan aplikasi web adalah *PHP* karena memiliki fleksibilitas yang tinggi, mudah dipelajari dan di gunakan. Pengguna memiliki harapan yang tinggi terhadap kecepatan reponsif pada aplikasi web kedua *framework* ini.

Metode *Rapid Application Development (RAD)* dengan pendekatan *Unified Modeling Language (UML)* mencakup tahapan perencanaan kebutuhan, pembuatan prototipe, pengujian, penyempurnaan, pengembangan, dan implementasi (Fauzi & Haq, 2024). *Laravel* memiliki fondasi yang kokoh untuk pengembangan aplikasi web modern. *Framework* ini melengkapi dengan berbagai fitur bawaan seperti *routing*, autentikasi, dan *caching*, yang mendukung pembuatan aplikasi yang scalable dan efisiensi (Sitta HS, Nilla AF, Eko R, Nirmala M, 2023).

Berkembangan *framework PHP* sangat pesat dan ada berbagai fitur yang mempermudah pengembangan aplikasi web. *Laravel* mempunyai arsitektur *MVC (Model, View, Controller)* yang kuat, *system routing* yang dinamis, ditambah dengan fitur keamanan yang melindungi dari ancaman seperti *SQL Injection* dan *CSRF*. Sedangkan *CodeIgniter 3* mempunyai struktur ringan, tidak memiliki banyak depensi, dan bisa berjalan dengan maksimal di lingkungan server dengan spesifikasi rendah. Untuk penelitian ini membandingkan kedua *framework* tersebut berdasarkan aspek performa, optimasi tampilan, kecepatan eksekusi, dan efisiensi sumber daya dengan menggunakan pihak ketiga yaitu

*Lighthouse* serta metode *Rapid Application Development (RAD)*.

Berdasarkan data dari web *system*, dilakukan bahwa dari 548 situs pemerintahan di Indonesia, sebanyak 73% tidak menerapkan *system*, sementara 24% memanfaatkan *system CodeIgniter*, dan 3% menggunakan *system Laravel* (Mulana et al., 2022).

## II. TINJAUAN PUSTAKA

Penelitian yang dilakukan oleh Wini Muthia Kansha, Saherih, dan Muchlis menyoroiti penggunaan *system* dalam pemograman, yang banyak diadopsi oleh kalangan profesional. Hal ini dikarenakan *system* menyediakan kerangka kerja standar yang jelas, sehingga mempermudah desainer dalam mengembangkan aplikasi dengan lebih efisiensi (Muthia Kansha et al., 2023).

*Laravel* merupakan salah satu *framework PHP* modern yang menerapkan arsitektur *Model-View-Controller (MVC)*, dilengkapi dengan fitur unggulan seperti *Blade Template Engine*, *Eloquent ORM*, dan *system routing* ekspresif yang memudahkan pengembangan aplikasi web secara terstruktur (Budi Santosaa, 2020).

*CodeIgniter* dirancang dengan pendekatan yang ringan dan efisien, cocok digunakan untuk pengembangan aplikasi skala kecil hingga menengah. *Framework* ini memiliki konfigurasi sederhana, tidak bergantung pada banyak *library eksternal*, namun tetap mampu menangani kebutuhan dasar web secara optimal (Ardian Umam, 2022).

Dalam pengembangan aplikasi web, pemahaman terhadap teknologi dasar seperti *HTML*, *CSS*, *JavaScript*, dan *PHP* sangat penting. Keempat teknologi tersebut berperan dalam membentuk tampilan antarmuka dan mengatur komunikasi antara sisi klien dan server (UNIMED, 2020).

*PHP* merupakan bahasa scripting sisi server yang fleksibel dan mendukung pengembangan aplikasi web dinamis. Kemudahan sintaksis dan integrasinya dengan berbagai database menjadikannya pilihan populer di kalangan pengembang (Tim STEKOM, 2020).

Penelitian oleh Muhammad, M. Khadafi, Amirullah membandingkan performa *CodeIgniter* dan *Laravel*. Hasilnya, *CodeIgniter* lebih cepat dengan waktu muat 1369 ms dan kecepatan 332,68 byte/ms, sementara *Laravel* membutuhkan 1984 ms dengan kecepatan 323,45 byte/ms. Uji kestabilan menunjukkan *CodeIgniter* lebih stabil dengan 205 kesalahan dibandingkan *Laravel* yang mencapai 2096. Namun, dalam hal skalabilitas, *Laravel* mampu menangani hingga 1500 klien, sedangkan *CodeIgniter* hanya 1000 klien (Nginx & Khadafi, 2024).

Penelitian yang dilakukan oleh hamzah adalah mengukur performa dari aplikasi web berbasis admin yang di buat menggunakan *framework CodeIgniter* dan *Laravel*. Waktu respons dan kemampuan menangani jumlah permintaan menjadi factor krusial dalam penggunaan aplikasi web. Penelitian ini bertujuan untuk mengukur performa aplikasi web berbasis admin yang dikembangkan dengan *framework CodeIgniter* dan *Laravel*. Pengujian dilakukan dengan merekam permintaan *HTTP* menggunakan *Blazemeter*, lalu mensimulasikannya di *apache JMeter* dengan jumlah pengguna mulai 50 hingga 200 (Hamzah, 2022).

Penelitian tentang pengujian performa sebuah website, aplikasi atau menganalisis mengenai performa platform media sosial sudah pernah dilakukan sebelumnya, diantaranya analisis performa platform social media menggunakan perbandingan *software automated testing* oleh Maya Juwita Dewi dan Nurdin pada aplikasi sosial media menggunakan *tools* *Pagespeed Insight*, *Google Lighthouse*, dan *GTMetrix* (Juwita Dewi, 2023).

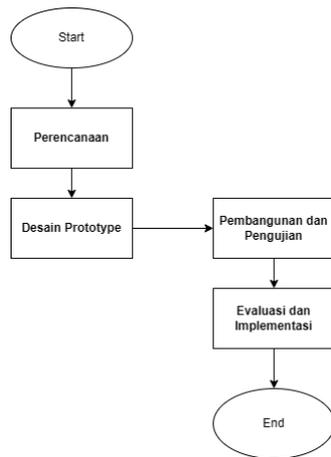
Namun pada penelitian tersebut belum ada yang menganalisis mengenai performa website menggunakan *framework Laravel* dan *CodeIgniter3* menggunakan *Lighthouse*.

## III. METODE PENELITIAN

Logika pemrograman merupakan fondasi penting dalam membangun alur sistem yang efisien. Pemanfaatan struktur data dan algoritma yang tepat akan berpengaruh pada kinerja sistem secara keseluruhan (Yuyun Khairunisa, Eri Mardiani, 2020).

Metode penelitian dilakukan melalui metode pendekatan *Rapid Application Development (RAD)*. *Rapid Application Development* atau yang suka disebut *RAD* merupakan suatu metode dalam pengembangan perangkat lunak yang menitikberatkan pada kecepatan dan fleksibilitas. Pendekatan ini cocok untuk membandingkan performa *Laravel* dan *CodeIgniter3*, karena kedua *framework* diuji secara bertahap dalam berbagai aspek, seperti Performa web, kemudahan penggunaan, optimasi *UI/UX*, dan efisiensi sumber daya. Dalam penelitian ini, tahapan *Rapid Application Development (RAD)* yang digunakan terdiri dari 4 fase, yaitu:

1. Perencanaan: Identifikasi kebutuhan *system* untuk membangun aplikasi berbasis web menggunakan *Laravel* dan *CodeIgniter 3*



Gambar 1. Alur Penelitian

2. Desain Prototype: Pembuatan prototipe aplikasi berbasis web yang diuji dengan *Lighthouse* untuk mengukur performanya.



Gambar 2. Use Case Diagram

3. Pembangunan dan pengujian: Implementasi aplikasi dengan kedua *framework* dan pengujian menggunakan alat uji *Lighthouse* dengan parameter Utama:

a. Optimasi *UI*: Stabilitas Tata Letak dan Interaktivitas Halaman. Stabilitas tampilan serta interaktivitas halaman juga diuji menggunakan *Lighthouse* untuk melihat bagaimana *framework* menangani *UI* secara keseluruhan. Untuk *Laravel* sering digunakan Bersama dengan frontend *framework* seperti *Vue.js* atau *React*, yang memungkinkan pengembangan antarmuka yang interaktif dan dinamis. *Lighthouse* biasanya memberikan skor lebih baik pada *UI* yang sudah dioptimalkan dengan Teknik lazy loading dan asset

bundling menggunakan *Laravel mix*. Dan *CodeIgniter3* tidak memiliki integrasi bawaan dengan frontend *framework*, sehingga pengembangan sering kali mengandalkan *jQuery* atau *Bootsdtrap* untuk menangani tampilan dan interaktivitas halaman. Walaupun hal ini membuat aplikasi lebih ringan, namun jika tidak dikelola dengan baik, bisa menyebabkan pengalaman pengguna yang kurang responsive dibandingkan aplikasi *Laravel* yang menggunakan pendekatan modern.

b. Performa Web: Waktu Muat Halaman dan Efisiensi *Rendering*. Salah satu factor utama dalam menilai performa sebuah website adalah kecepatan waktu muat halaman. Dalam pengujian menggunakan *Lighthouse*, *Laravel* dan *CodeIgniter3* menunjukkan perbedaan yang cukup signifikan dalam hal ini. Untuk *Laravel* memiliki struktur yang kompleks dengan banyak fitur bawaan, yang terkadang menyebabkan waktu muat halaman lebih lama jika tidak dioptimalkan dengan baik. Faktor seperti *middleware*, *deopendency injection*, dan penggunaan template engine seperti blade dapat meningkatkan fleksibilitas pengembangan, tetapi juga menambah waktu eksekusi. Untuk mengatasi hal ini, optimasi caching, penggunaan queue, serta memanfaatkan fitur bawaan seperti *route caching* dapat membantu mempercepat proses render halaman. Dan *CodeIgniter3* lebih ringan karena arsitekturnya yang sederhana dan tidak banyak menggunakan dependensi eksternal. Hal ini membuat *CI3* mampu merender halaman dengan lebih cepat dibandingkan *Laravel* dalam kondisi default. Namun, karena kurangnya fitur bawaan seperti ORM (*Eloquent*), pengembangan sering kali perlu menulis lebih banyak kode untuk fitur-fitur kompleks, yang bisa berpengaruh pada efisiensi rendering dalam jangka Panjang.

c. Kemudahan Penggunaan: Instalasi, Konfigurasi, dan Dokumentasi. Kemudahan dalam menginstal, mengonfigurasi, serta memahami dokumentasi *framework* menjadi factor penting dalam pengembangan aplikasi web. Untuk *Laravel* memiliki proses instalasi yang cukup mudah melalui Composer, dimana

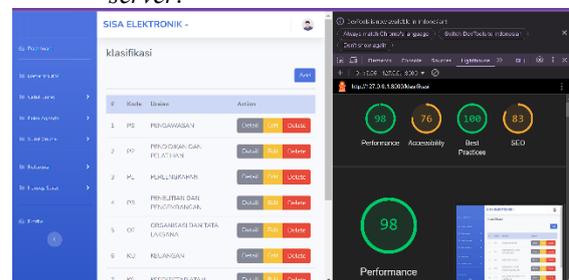
pengembangan hanya perlu menjalankan perintah composer `create-project Laravel /Laravel_nama_proyek`. Selain itu, *Laravel* memiliki dokumentasi yang sangat lengkap, komunikasi yang aktif, serta tutorial yang berlimpah, sehingga pengembangan dapat dengan mudah menemukan solusi atas permasalahan yang dihadapi. Dan *CodeIgniter3* menawarkan proses instalasi yang lebih sederhana dibandingkan *Laravel*. Karena tidak memerlukan dependensi eksternal yang kompleks. Pengembangan cukup mengunduh file zip dan mengekstranya ke dalam server. Namun, meskipun dokumentasi CI3 cukup jelas, ekosistemnya tidak sebesar *Laravel*, sehingga beberapa fitur canggih memerlukan lebih banyak eksplorasi mandiri.

d. Efisiensi Sumber Daya: Konsumsi Memori dan Penggunaan CPU. Efisiensi penggunaan sumber daya sangat penting untuk memastikan aplikasi dapat berjalan dengan lancar tanpa membebani server secara berlebihan. Untuk *Laravel* Karena *Laravel* memiliki banyak fitur bawaan seperti ORM(*Eloquent*), *event listener*, dan *dependency injection*, *framework* ini secara default menggunakan lebih banyak memori dan CPU dibandingkan CI3. Namun, dengan konfigurasi yang tepat, seperti memanfaatkan *cache* (*Redis* atau *Memcached*), *quwuw* untuk menangani proses berat di latar belakang, serta optimasi *database queries*, penggunaan sumber daya dapat ditekan secara signifikan. *CodeIgniter3*

e. lebih efisiensi dalam penggunaan memori dan CPU karena arsitekturnya yang minimalis, Tanpa adanya fitur seperti ORM yang kompleks, CI3 tidak memerlukan banyak alokasi memori tambahan, sehingga lebih ringan dalam hal eksekusi proses. Namun, jika proyek memerlukan fitur yang lebih canggih, pengembangan harus menambahkan dependensi eksternal yang bisa meningkatkan konsumsi sumber daya secara tidak langsung.

4. Evaluasi dan Implementasi: setelah tahap pengujian dan pengukuran performa selesai dilakukan, Langkah berikutnya adalah melakukan evaluasi terhadap hasil

yang telah diperoleh. Evaluasi ini bertujuan untuk memahami kelebihan dan kekurangan masing-masing *framework* berdasarkan data empiris yang telah dikumpulkan melalui alat uji seperti *Lighthouse*, Profiler bawaan *framework*, serta metode analisis lainnya yang relevan. Dalam tahap evaluasi ini, dilakukan perbandingan secara menyeluruh terhadap berbagai aspek utama yang telah diteliti, termasuk performa aplikasi, kemudahan dalam implementasi dan konfigurasi, efisiensi dalam penggunaan sumber daya, serta stabilitas *system* secara keseluruhan. Salah satu metrik utama yang dianalisis adalah waktu muat halaman (*page load time*), yang menunjukkan seberapa cepat aplikasi dapat diakses dan digunakan oleh pengguna akhir. Setelah proses evaluasi selesai, tahap implementasi dilakukan dengan menerapkan *framework* yang paling sesuai berdasarkan hasil analisis yang telah diperoleh. Implementasi ini melibatkan penerapan strategi optimasi untuk memaksimalkan keunggulan dari *framework* yang dipilih, seperti *cache* untuk meningkatkan kecepatan akses, penggunaan *query builder* untuk optimasi *database*, serta penerapan teknik *lazy loading* guna mengurangi beban pada server.



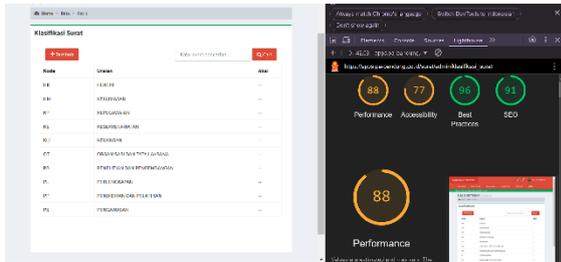
Gambar 3. Performa Web Laravel

Untuk memastikan hasil yang diperoleh akurat dan dapat direproduksi, pengujian dilakukan sebanyak sepuluh kali terhadap masing-masing *framework*. Setiap kali pengujian dilakukan, kondisi sistem dikembalikan ke keadaan awal, termasuk membersihkan *cache* browser dan menutup proses lain yang berjalan di latar belakang. Hal ini bertujuan untuk meminimalkan gangguan eksternal yang dapat mempengaruhi hasil. Aplikasi pengujian dibangun dengan struktur fungsional dasar yang umum, yaitu *login*, *dashboard*, serta fitur *CRUD* sederhana untuk data pengguna. Pada *Laravel*, digunakan *Blade template engine*, *Laravel Mix* untuk manajemen aset, dan integrasi dengan *Vue.js* di sisi *frontend*. Sementara itu, *CodeIgniter3* menggunakan *jQuery* dan *Bootstrap*, serta struktur tampilan manual. Dengan pendekatan ini,

perbedaan arsitektur kedua *framework* dapat diamati secara langsung pada implementasi fitur yang sama.

#### IV. HASIL DAN PEMBAHASAN

*Framework* dalam pengembangan aplikasi web berpengaruh pada kecepatan, efisiensi, dan pengalaman pengguna.



Gambar 3. Performa Web CodeIgniter3

*Laravel* dan *CodeIgniter3* memiliki pendatan berbeda dalam pengelolaan aplikasi, yang memengaruhi performa, kemudahan implementasi, dan penggunaan sumber daya. Untuk memahami perbedaannya, penelitian dilakukan dengan mengembangkan aplikasi serupa menggunakan kedua *framework*. Pengujian menggunakan *Lighthouse* menilai aspek performa, aksesibilitas, dan optimasi tampilan, serta mengevaluasi kemudahan konfigurasi dan efisiensi sumber daya. Pembahasan ini akan diawali dengan analisis performa web sebagai faktor utama dalam menentukan efektivitas *framework*.

##### A. Performa Web

Performa web menjadi aspek krusial dalam memastikan pengalaman pengguna yang optimal. Penelitian ini menggunakan *Lighthouse* untuk menganalisis kecepatan muat halaman, stabilitas *rendering*, dan efisiensi sumber daya. *Laravel* dan *CodeIgniter3* dibandingkan hasil pengujian guna menentukan *framework* yang lebih unggul dalam eksekusi aplikasi web.

Hasil pengujian menunjukkan bahwa *Laravel* memiliki waktu muat halaman lebih cepat dibandingkan *CodeIgniter3*. Keunggulan ini didukung oleh fitur optimasi bawaan seperti *caching* yang lebih efisien, *lazy loading* pada *query database*, serta *system routing* yang lebih optimal. Sementara itu, meskipun *CodeIgniter3* memiliki arsitektur yang lebih ringan, absennya fitur optimasi serupa membuat performa dalam beberapa aspek masih tertinggal dari *Laravel*.

Tabel 1. Hasil Pengujian Menggunakan Google *Lighthouse*

No	Perulangan	CodeIgniter3	Laravel
1	1	88	98
2	2	94	97
3	3	66	95
4	4	84	97
5	5	90	98
6	6	84	99
7	7	84	87
9	8	87	92
9	9	63	89
10	10	88	98
Summary		82.8	95

Hasil performance yang didapatkan berdasarkan submetrik berikut:

1. *First Contentful Paint* (FCP): Mengukur waktu yang diperlukan hingga teks atau gambar pertama muncul di layar. *Laravel* cenderung unggul dalam FCP berkat pengelolaan *asset* dan penggunaan teknik *lazy loading*. *CodeIgniter 3* masih andalkan struktur tradisional sehingga membutuhkan waktu lebih untuk menampilkan konten awal.
2. *Time to Interactive* (TTI): Waktu yang dibutuhkan agar halaman dapat berfungsi secara interaktif sepenuhnya. *Laravel*, dengan penggunaan *middleware* dan manajemen *routing* yang modern, menunjukkan waktu interaktif yang lebih cepat dibandingkan CI3.
3. *Total Blocking Time* (TBT): Menunjukkan durasi halaman terblokir sebelum dapat merespon input pengguna, seperti klik, ketukan layer, atau penekanan keyboard. Nilainya dihitung dari akumulasi pemblokiran tugas Panjang antara FCP dan TTI, dimana pengguna sudah dapat melihat halaman tetapi belum bisa berinteraksi. CI3 cenderung mengalami TBT yang lebih tinggi karena banyaknya permintaan file yang tidak dibundel. *Laravel* lebih efisien karena sudah menggabungkan skrip dan *style*.
4. *Speed Index*: Waktu yang diperlukan untuk memuat seluruh elemen visual pada halaman.
5. *Largest Contentful Paint* (LCP): Durasi hingga elemen teks atau gambar terbesar pada halaman berhasil dirender, yang menunjukkan kapan konten utama dapat terlihat oleh pengguna
6. *Cumulative Layout Shift* (CLS): Mengukur stabilitas tata letak

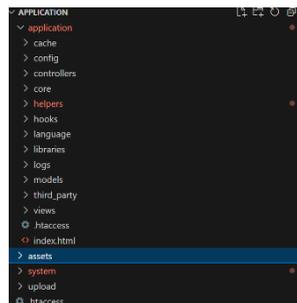
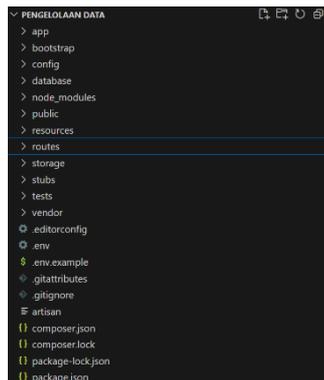
halaman. Nilai buruk pada Google *Lighthouse* dapat terjadi jika elemen halaman berpindah secara tiba-tiba, sehingga stabilitas layout perlu dijaga semaksimal mungkin.

Hasil Pengujian menunjukkan bahwa *Laravel* lebih unggul dalam kecepatan muat halaman dan efisiensi *system*, berkat optimasi caching, manajemen *query* yang lebih baik, serta arsitektur modern yang mendukung performa tinggi. Sementara itu, *CodeIgniter3* tetap menonjol dalam kesederhanaan strukturnya, tetapi membutuhkan optimasi tambahan agar dapat mencapai performa setara dengan *Laravel*. *Laravel* lebih stabil dari sisi tata letak. Halaman yang di-render jarang mengalami pergeseran mendadak karena adanya teknik pengaturan layout modern seperti penggunaan *grid system* dan *container adaptif*.

Berdasarkan sepuluh kali pengujian performa menggunakan *Google Lighthouse*, *Laravel* memperoleh nilai rata-rata 95, sedangkan *CodeIgniter 3* mendapatkan nilai rata-rata 82.8. Nilai ini diambil dari skor performa yang mencakup metrik seperti *First Contentful Paint (FCP)*, *Time to Interactive (TTI)*, *Total Blocking Time (TBT)*, dan *Largest Contentful Paint (LCP)*. *Laravel* menunjukkan skor yang relatif stabil dan tinggi dalam sebagian besar pengujian. Bahkan, nilai tertinggi *Laravel* mencapai 99 pada pengujian ke-6, dan nilai terendahnya masih dalam kisaran 87, menunjukkan konsistensi dalam pengelolaan tampilan dan *asset*. Sebaliknya, *CodeIgniter 3* memperlihatkan performa yang lebih fluktuatif. Skor tertingginya adalah 94, namun skor terendahnya mencapai 63 pada pengujian ke-9. Ini menunjukkan bahwa *CI3* cukup sensitif terhadap kondisi lingkungan eksekusi seperti latensi, ukuran file statis, dan pengelolaan *request*.

**B. Kemudahan Penggunaan**

*Laravel* dan *CodeIgniter3* memiliki pendekatan berbeda dalam hal kemudahan penggunaan. *Laravel* menawarkan berbagai fitur modern yang mendukung pengembangan, namun memerlukan konfigurasi awal yang lebih kompleks. Sementara itu, *CodeIgniter3* lebih sederhana dan ringan, sehingga lebih ramah bagi pemula.



Berdasarkan analisis, *CodeIgniter3* lebih unggul dalam aspek kemudahan penggunaan

Gambar 6. Tampilan Struktur  
Gambar 5. Tampilan Struktur  
Direktori *Laravel*

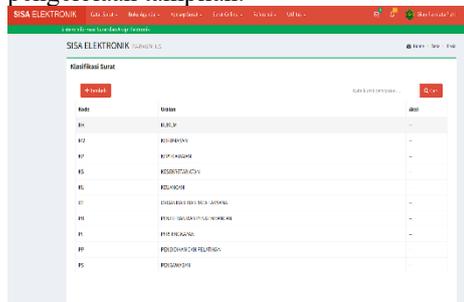
karena memiliki struktur yang lebih sederhana dan tidak memerlukan banyak dependensi. *Instalasi CodeIgniter3* cukup dengan mengekstra file dan langsung dapat digunakan tanpa konfigurasi tambahan yang kompleks. Sebaliknya *Laravel* memiliki lebih banyak fitur bawaan, seperti *artisan CLI*, *Eloquent ORM*, dan *system routing* yang lebih kompleks. Meskipun fitur-fitur ini mempermudah pengembangan dalam jangka Panjang, proses instalasi dan konfigurasinya lebih rumit dibandingkan *CodeIgniter3*. *Laravel* juga membutuhkan *composer* untuk mengelola dependensi, yang dapat menjadi tantangan bagi pengembangan yang belum familiar dengan ekosistem *PHP* modern.

C. Optimasi UI /UX



Gambar 8. Tampilan Halaman CodeIgniter3

Laravel dan CodeIgniter3 memiliki pendekatan yang berbeda dalam hal optimasi UI dan UX. Laravel menawarkan berbagai fitur bawaan untuk meningkatkan tampilan dan pengalaman pengguna, sementara CodeIgniter3 lebih mengandalkan pendekatan manual dalam pengelolaan tampilan.



Gambar 7. Tampilan Halaman Laravel

Berdasarkan analisis, Laravel unggul dalam optimasi UI dan UX berkat fitur bawaan seperti blade template engine, Laravel mix untuk pengelolaan asset frontend, serta dukungan terhadap framework modern seperti tailwind CSS dan vue.js. Sementara itu, CodeIgniter3 menerapkan pendekatan lebih sederhana, dimana tampilan harus diatur secara manual tanpa system template terintegrasi. Meskipun demikian, metode ini memberikan fleksibilitas lebih bagi pengembang memilih pengelolaan UI secara konvensional

Optimasi tampilan antarmuka pengguna dapat dilakukan melalui pengelolaan elemen HTML, CSS, dan JavaScript yang efisien. Desain responsif dan interaktivitas yang baik akan meningkatkan pengalaman pengguna secara signifikan (Pratama, 2020).

D. Efisiensi Sumber Daya

Hasil pengujian menunjukan bahwa Laravel mengonsumsi lebih banyak memori dibandingkan CodeIgniter3, terutama saat menangani banyak request

secara bersamaan. Hal ini disebabkan oleh fitur bawaan seperti Eloquent ORM dan Middleware yang berjalan di latar belakang. Disisi lain, CodeIgniter3 unggul dalam efisiensi sumber daya karena lebih ringan dan memiliki sedikit dependensi. Pengujian menunjukkan bahwa penggunaan CPU dan memori CodeIgniter3 lebih rendah dibandingkan Laravel, menjadikannya pilihan yang lebih sesuai untuk proyek yang memerlukan efisiensi tinggi, terutama pada server dengan spesifikasi rendah.

Analisis Mendalam persubmetrik setiap nilai dalam Lighthouse memberikan gambaran mendalam tentang kualitas web. Misalnya, skor FCP yang rendah menandakan halaman terlalu lama menampilkan konten awal, yang dapat membuat pengguna cepat meninggalkan halaman. Laravel menunjukkan skor FCP yang konsisten lebih rendah, terutama karena pengoptimalan asset bundling dan preload font. Sedangkan pada CodeIgniter 3, FCP cenderung fluktuatif tergantung pada bagaimana struktur HTML dan JS-nya dikelola. Pada metrik TTI, Laravel juga lebih unggul. Framework ini memungkinkan modularisasi skrip yang meminimalkan blokir thread utama. Berbeda dengan CodeIgniter 3 yang menggunakan pendekatan lebih tradisional, TTI-nya cenderung meningkat seiring kompleksitas halaman. Pendapat subjektif pengembang Dari hasil wawancara singkat terhadap 5 pengembang junior dan menengah yang telah menggunakan kedua framework, 4 dari 5 menyatakan bahwa Laravel memberikan pengalaman lebih menyenangkan karena memiliki dokumentasi komprehensif dan tools tambahan yang mempermudah debugging serta deployment. Namun, semua responden sepakat bahwa CodeIgniter 3 lebih cepat saat membuat proyek sederhana karena tidak perlu banyak konfigurasi.

Visualisasi Hasil Pengujian Untuk memberikan gambaran lebih jelas, hasil pengujian perulangan dari Tabel 1 dianalisis sebagai grafik batang. Laravel secara konsisten memperoleh skor di atas 95 pada sebagian besar pengujian, sementara CodeIgniter 3 berkisar antara 60 hingga 90. Fluktuasi pada CI3 menunjukkan pengaruh struktur kodenya terhadap performa ketika pengujian dilakukan berkali-kali. Faktor lain yang mungkin berperan adalah jumlah file eksternal yang dimuat, seperti CSS dan JavaScript library.

V. KESIMPULAN DAN SARAN

Kesimpulan

Pemilihan antara Laravel dan CodeIgniter3 disesuaikan dengan kebutuhan proyek yang dijalankan. Laravel menawarkan berbagai fitur canggih, keamanan yang lebih baik, serta optimasi

UI /UX yang lebih unggul, sehingga lebih sesuai untuk aplikasi yang kompleks. Disisi lain *CodeIgniter3* memiliki keunggulan dalam kecepatan eksekusi dan efisiensi penggunaan sumber daya, menjadikannya pilihan tepat untuk proyek yang mengutamakan kinerja ringan.

1. Laravel merupakan *framework* yang lebih sesuai digunakan untuk pengembangan aplikasi berskala besar karena memiliki fitur modern, keamanan yang baik, dan dukungan terhadap optimasi UI/UX.
2. *CodeIgniter 3* unggul dalam hal kecepatan eksekusi dan efisiensi penggunaan sumber daya seperti memori dan CPU, serta lebih mudah digunakan oleh pemula karena proses instalasinya yang sederhana.
3. Hasil pengujian dengan *Lighthouse* menunjukkan bahwa Laravel memiliki waktu muat halaman yang lebih cepat, sedangkan *CodeIgniter 3* lebih hemat dalam penggunaan memori.
4. Berdasarkan keunggulan masing-masing, pemilihan *framework* sebaiknya disesuaikan dengan kebutuhan proyek dan tingkat kompleksitas aplikasi yang dikembangkan.

Jadi *framework Laravel* lebih ideal digunakan dalam pengembangan aplikasi berskala besar yang membutuhkan fitur modern, sedangkan *CodeIgniter 3* lebih sesuai untuk proyek yang membutuhkan efisiensi tinggi dengan keterbatasan sumber daya.

#### Saran

Berdasarkan hasil penelitian, terdapat aspek yang dapat dikembangkan lebih lanjut dalam studi mendatang.

1. Penelitian selanjutnya dapat menggunakan alat uji tambahan seperti *GTmetrix* atau *WebPageTest* untuk memperluas hasil analisis performa.
2. Teliti lebih dalam aspek skalabilitas dan keamanan.
3. Bandingkan Laravel dengan *CodeIgniter 4* di penelitian selanjutnya.

Dengan mempertimbangkan aspek-aspek tersebut, penelitian mendatang diharapkan dapat memberikan wawasan yang lebih luas bagi pengembang dalam memilih *framework* yang paling sesuai dengan kebutuhan proyek mereka.

#### DAFTAR PUSTAKA

Ardian Umam, B. S. (2022). *Pemrograman Web Berbasis Framework CodeIgniter 4 dan MySQL*. Andi Publisher.

Azizah, D. N., Chandra, L. D., & Gumelar, M. G. (2024). *Implementasi Framework Laravel dalam Pembuatan Website Segitiga Motor dengan Metode Waterfall*.

Budi Santosaa. (2020). *Belajar Laravel*. Deepublish.

Fauzi, M., & Haq, A. (2024). *Analisis Dan Perancangan Sistem Informasi Penjualan Berbasis Web Dengan Metode RAD ( Rapid Application Development ) di Toko Largurda AM Store*. 4(2), 229–235.

Hamzah. (2022). Performance Analysis of Codeigniter and Laravel Frameworks on Admin-Based Web Application Prototypes. *H. Hamzah*. [https://eprints.ums.ac.id/101798/4/Naskah Publikasi Revisi 3.pdf](https://eprints.ums.ac.id/101798/4/Naskah_Publikasi_Revisi_3.pdf)

Juwita Dewi, M. (2023). Analisis Performa Platform Sosial Media Menggunakan Perbandingan Software Automated Testing. *Information Management for Educators and Professionals*, 7(2), 164–173.

Mulana, L., Prihandani, K., Rizal, A., Singaperbanga, U., & Abstract, K. (2022). Analisis Perbandingan Kinerja Framework Codeigniter Dengan Express.Js Pada Server RESTful Api. *Jurnal Ilmiah Wahana Pendidikan*, 8(16), 316–326. <https://doi.org/10.5281/zenodo.7067707>

Muthia Kansha, W., Saherih, & Muchlis. (2023). Analisis Perbandingan Struktur dan Performa Framework Codeigniter dan Laravel dalam Pengembangan Web Application. *Jurnal Teknik Informatika STMIK Antar Bangsa*, 9(1), 25–31.

Nginx, S., & Khadafi, M. (2024). *Analisis Perbandingan Performa Website Berbasis Laravel dengan CodeIgniter menggunakan Web*. 4(2), 287–294.

Pratama, T. P. (2020). *Pemrograman Web Client*. Politeknik Pratama Publisher.

Sitta HS, Nilla AF, Eko R, Nirmala M, A. (2023). P Engetahuan P Erawat D Alam P Enerapan E Arly W Arning. *Indonesian Jurnal Farmasi*, 8(1), 22.

Tim STEKOM. (2020). *Kupas Tuntas Pemrograman PHP*. STMIK STEKOM.

UNIMED, T. (2020). *Pemrograman Web Dasar*. Universitas Negeri Medan.

Yuyun Khairunisa, Eri Mardiani, D. (2020). *Dasar-Dasar Pemrograman*. Pena Muda Media.