

Analisis Perbandingan Struktur dan Performa *Framework Codeigniter dan Laravel* dalam Pengembangan *Web Application*

Wini Muthia Kansha¹, Saherih², Muchlis³

Abstract—PHP programming language is considered as one of the most used scripting languages in web application development. Programming trends have started to develop from using native to using frameworks. Professionals widely use programming using a framework because it has a standard framework that is common and clear to help developers build applications quickly. Codeigniter and Laravel are examples of PHP frameworks that are quite popular among web developers. A comparative analysis of the two frameworks was carried out from various perspectives, both in terms of framework structure and research experiments by making simple CRUD applications to measure the performance of the two websites. This comparison in terms of framework structure is intended so that it can be taken into consideration by web developers in choosing a framework that suits their needs in building web applications. From a simple experiment using Google Lighthouse, in terms of performance, Laravel 8 scored 91.7 and CodeIgniter 4 scored 88.9. Thus Laravel 8 is slightly superior to CodeIgniter 4 in terms of performance, there is no significant comparison. So that the thing that can be taken into consideration for choosing a framework is in terms of framework structure.

Intisari—Bahasa pemrograman PHP dianggap sebagai salah satu bahasa *scripting* yang paling banyak digunakan dalam pengembangan aplikasi web. Tren pemrograman sudah berkembang dari penggunaan *native* hingga menggunakan *framework*. Pemrograman menggunakan *framework* banyak digunakan di kalangan profesional karena *framework* memiliki kerangka kerja standar yang umum dan jelas sehingga dapat membantu *developer* dalam membangun aplikasi dengan lebih cepat. Codeigniter dan Laravel merupakan contoh *framework* PHP yang cukup populer di kalangan *web developer*. Dilakukan analisis perbandingan kedua *framework* ini dari berbagai sudut pandang, baik dari segi struktur *framework* dan percobaan penelitian dengan membuat aplikasi CRUD sederhana untuk mengukur *performance* kedua *website* tersebut. Perbandingan dari segi struktur *framework* ini ditujukan agar dapat dijadikan bahan pertimbangan *web developer* dalam memilih *framework* yang sesuai kebutuhan dalam membangun aplikasi web. Dari percobaan sederhana menggunakan Google Lighthouse di dapat dalam segi *performance* Laravel 8 dengan skor 91,7 dan Codeigniter 4 dengan skor 88,9. Dengan demikian Laravel 8 sedikit lebih unggul dari Codeigniter 4 dalam hal performa, tidak ada perbandingan yang signifikan. Sehingga hal yang bisa dijadikan bahan pertimbangan untuk pemilihan *framework* adalah dari segi struktur *framework*.

Kata Kunci—Codeigniter, *Framework*, Laravel, PHP, *Web Application*.

I. PENDAHULUAN

Teknologi *website* saat ini mengalami perkembangan yang cukup pesat. *Website* saat ini bukan hanya menjadi media

informasi yang bersifat membaca atau umumnya bersifat statis, tetapi telah berkembang menjadi media yang lebih dinamis dan interaktif. *Website* adalah sebuah halaman *web* yang terhubung dimana data online berada di server yang sama dan berisi berbagai kumpulan data yang dapat disediakan untuk individu, kelompok, dan organisasi dikutip dari Referensi [1].

Bahkan sebagai salah satu sarana komunikasi yang paling populer di kalangan masyarakat, Penggunaan *website* telah mengalami berbagai perubahan dengan faktor biaya yang sangat murah dan kemudahan penggunaan serta efisiensi karena tersedia dalam waktu 24 jam [1].

Bahasa pemrograman PHP dianggap sebagai salah satu bahasa *scripting* yang paling banyak digunakan dalam pengembangan aplikasi *web* karena menawarkan fleksibilitas yang tinggi, mudah digunakan dan mudah dipelajari [2]. PHP adalah *server-side scripting* language yang digunakan untuk membuat halaman *web* yang dinamis dan interaktif. Selama pengembangan dengan PHP sederhana, logika bisnis dicampur dengan *database query* dan tag presentasi. Karena campuran mode pengembangan ini, pemeliharaan dan skalabilitas aplikasi menjadi sulit. PHP telah membawa kerangka kerja (*framework*) pengembangan yang berbeda untuk mengatasi masalah ini. *Framework* PHP membantu pengembangan membangun aplikasi *web* lebih cepat dan lebih mudah dengan menyediakan model kerangka kerja dasar, serta satu set lengkap API, *library*, dan *extension*, dan membantu pengembang menjadi lebih produktif dengan mengurangi kode berulang dalam proyek [2].

Framework adalah sebuah kumpulan fungsi, *class*, dan aturan-aturan. Tidak seperti dengan *library* yang sifatnya hanya untuk tujuan tertentu saja, *framework* mengatur bagaimana aplikasi dibuat. *Framework* memungkinkan kita membangun aplikasi dengan lebih cepat karena kita sebagai *developer* akan lebih fokus pada pokok permasalahan, sedangkan untuk hal-hal penunjang lainnya seperti koneksi ke *database*, GUI, *form validation*, dan *security* umumnya telah disediakan oleh *framework* [3].

Ada beberapa *framework* PHP seperti Symfony, CodeIgniter, Laravel, Cake PHP, Yii, dll. Namun penelitian ini dikhususkan untuk studi literatur serta analisa dan pengkajian terkait perbedaan penggunaan *framework* Codeigniter dan Laravel.

Laravel adalah *framework* PHP yang dirilis sejak Juni 2011 dan dirancang untuk mengembangkan aplikasi *website* dengan cara yang lebih terstruktur dan lebih rapi. Sedangkan Codeigniter adalah sebuah *framework* PHP yang dirilis pada Februari 2006 yang merupakan *web application network* yang bersifat *open source* yang digunakan untuk membangun aplikasi PHP yang dinamis.

Analisis perbedaan kedua *framework* PHP tersebut dilihat dari berbagai aspek, antara lain struktur *framework*, performa, dll. Kedua *framework* tersebut dibandingkan karena Codeigniter dan Laravel merupakan *framework* PHP yang populer digunakan khususnya di Indonesia, Tujuan dari perbandingan ini adalah untuk menganalisis perbedaan antara kedua *framework* tersebut sehingga dapat dijadikan bahan pertimbangan untuk memilih *framework* PHP mana yang lebih tepat dalam membangun aplikasi web yang tentunya sesuai dengan kebutuhan sistem dan membantu *web developer* mengembangkan proyek dengan lebih cepat.

II. KAJIAN LITERATUR

A. Website

Website adalah sebuah kumpulan dari halaman-halaman yang bisa menyajikan sebuah informasi tertentu berdasarkan niche dan jenisnya, website bisa di akses oleh siapa saja dan kapan saja melalui jaringan internet. website mempunyai isi dokumen multimedia (teks, gambar, suara, animasi, *video*) di dalam website menggunakan protokol HTTP (*hypertext transfer protocol*) dan untuk bisa mengaksesnya menggunakan perangkat lunak yang disebut *browser* [4].

B. Bahasa Pemrograman PHP

PHP (*Hypertext Preprocessor*) adalah sebuah bahasa pemrograman yang bisa ditanamkan atau disisipkan ke dalam HTML (*Hyper Text Markup Language*). Bahasa pemrograman PHP berjalan pada *server side scripting* dan bersifat *open source*. Penggunaan pada bahasa PHP bisa digunakan untuk membangun aplikasi website dinamis atau bisa juga untuk membuat program lain. *Server-side scripting* sendiri yaitu suatu pemrograman yang penggunaannya berada di sisi *server*. Seluruh proses di dalam bahasa pemrograman PHP akan diproses oleh *server*. Proses dilakukan di *server* tetapi hasilnya akan ditampilkan di *browser* [5].

C. MVC

Menurut Sidik di kutip dari Referensi [6], *Model View Controller (MVC)* adalah sebuah teknik aplikasi pemrograman yang sangat populer pada saat ini, dan mengharapkan pemrogram secara disiplin untuk bisa membagi program menjadi tiga bagian : *model*, *view* dan *controller*.

- 1) *Model*: Bagian dari aplikasi yang dapat digunakan untuk mengimplementasikan logika untuk domain data aplikasi.
- 2) *View*: Sebuah komponen yang dapat menampilkan antarmuka untuk pengguna (*user interface*) aplikasi.
- 3) *Controller*: Sebuah komponen yang dapat digunakan untuk menangani interaksi pengguna, bekerja sama dengan *model*, dan memilih *view* mana yang digunakan untuk merender data.

Menurut Rosa dan Shalahuddin dalam Referensi [6], konsep *Model View Controller (MVC)* mempunyai tujuan agar sebuah aplikasi bisa mudah dipelihara oleh orang-orang di dalam tim pengembangan yang berbeda spesifikasi pekerjaan, misalnya *database administrator (DBA)* untuk bisa mengurus masalah basis data, blok *controller* untuk *programmer*, dan blok *view* untuk desainer antarmuka (*interface designer*).

D. Framework

Framework adalah sebuah *software* yang mempunyai susunan kerangka kerja yang bisa digunakan oleh developer untuk mengembangkan sebuah aplikasi berbasis *web*, *mobile*, maupun *desktop*. *Framework* berisi perintah dan sebuah fungsi dasar yang umum dan bisa digunakan untuk mengembangkan sebuah software aplikasi dan diharapkan aplikasi bisa dikembangkan dengan lebih cepat serta tersusun dan terstruktur dengan sangat rapi. *Framework* mempunyai konsep yang menyediakan sebuah susunan fungsional generik atau suatu susunan fungsi yang umum dan terdapat pada suatu aplikasi. *Programmer* bisa menggunakan susunan fungsi tersebut pada aplikasi yang sedang dikembangkan. *Programmer* juga bisa mengubah susunan fungsi secara selektif menggunakan kode tertentu.

Sedangkan dikutip dari Referensi [7], menurut Basuki *framework* adalah paket yang berisi fungsi-fungsi yang dapat digunakan dalam pembuatan aplikasi. Terdapat beberapa fungsi bawaan yang ada dalam sebuah *framework*, diantaranya: *paging*, email, kalender, tanggal, bahasa, *upload file*, validasi *form*, tabel, *session*, manipulasi gambar, text, string, *captcha*, enkripsi, proteksi terhadap XSS, *security* dan lain-lain.

E. Composer

Menurut Arya Febian dari Referensi [8], *Composer* adalah sebuah *tools dependency manager* khusus PHP yang mempunyai fungsionalitas seperti Gem (Ruby) atau Maven (Java). Suatu *library* dapat diinstall di *Composer* dan *Composer* akan secara otomatis bisa menginstall ataupun meng-update *library* yang sangat dibutuhkan.

Menurut Nafies Luthfi dari Referensi [8], yang dimaksud *dependency manager* dalam *composer* adalah bahwa *composer* akan mengelola “ketergantungan” antara *library-library* PHP (yang disebut *package*), yang dibuat oleh seseorang untuk digunakan orang lain.

F. Framework Laravel

Laravel adalah sebuah *framework* PHP yang menekankan pada kesederhanaan dan fleksibilitas pada desainnya. Laravel diluncurkan dibawah lisensi MIT dengan menggunakan Github sebagai tempat berbagi kode. Sama halnya dengan *framework* PHP lainnya, Laravel dikembangkan dengan basis MVC (*Model View Controller*). Laravel dilengkapi *command line tool* yang bernama “*Artisan*” yang dapat digunakan untuk *packaging bundle* dan instalasi *bundle* [4].

G. Framework CodeIgniter

CodeIgniter adalah sebuah *web application framework* yang bisa digunakan untuk mengembangkan aplikasi PHP dinamis yang dibangun menggunakan konsep *Model View Controller development pattern*. CodeIgniter mempunyai berbagai macam *library* yang bisa memudahkan dalam pengembangan dan termasuk *framework* yang tercepat dibandingkan dengan *framework* lainnya [9].

III. METODE PENELITIAN

A. Analisis dan Pengumpulan Data

Penelitian dilakukan dengan mengimplementasi metode deskriptif kualitatif, dimana pengumpulan data dilakukan dengan cara studi literatur menelusuri berbagai sumber yang berkaitan dengan pembahasan *framework* codeigniter dan *framework* laravel. Terlebih pada pembahasan codeigniter versi 4 dan laravel versi 8. Teknik analisis dilakukan dengan pemaparan secara deskriptif atas hasil komparasi. Selain itu dilakukan uji coba sederhana terhadap dua aplikasi yang masing-masing dibuat menggunakan *framework* codeigniter dan laravel.

B. Perancangan dan Pengujian

Komparasi perbandingan hasil studi literatur dari kedua *framework* tersebut disusun berdasarkan berbagai aspek seperti: komparasi struktur, *database*, *programming paradigm*, *routing*, *built in modules*, *https support*, *support RestAPI*, *template engine*, *authentication*, *learning curve* and *community*, *package*. Selain itu dilakukan pembuatan dua aplikasi web sederhana dengan fitur *Create*, *Read*, *Update*, *Delete* (CRUD) menggunakan *framework* Codeigniter 4 dan Laravel 8. Kedua aplikasi web tersebut dilakukan pengujian menggunakan Google Lighthouse untuk mengukurnya dari sisi performa. Pengujian dilakukan secara berulang sebanyak sepuluh kali untuk masing-masing *framework*.

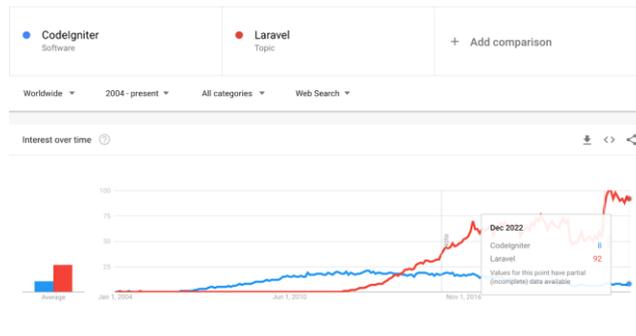
C. Hasil dan Kesimpulan

Setelah dilakukan studi literatur dan pengujian terhadap aplikasi sederhana yang dibuat menggunakan *framework* codeigniter dan laravel, maka selanjutnya didapatkan poin-poin perbandingan atau komparasi dari kedua *framework* PHP tersebut. Hasil komparasi tersebut dapat dijadikan pertimbangan untuk memilih *framework* PHP mana yang sesuai untuk pengembangan aplikasi *web* yang sesuai dengan kebutuhan pengguna.

IV. HASIL DAN PEMBAHASAN

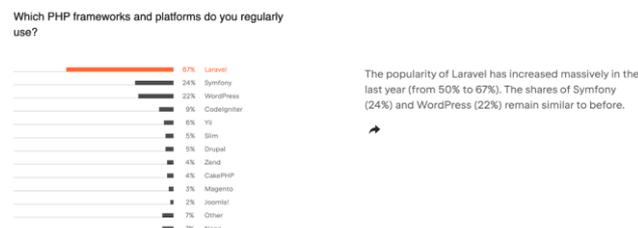
A. Komparasi Struktur

1) *Popularity dan Current Trends*: Komparasi ini diambil dari beberapa sumber kredibel terkait *trend* dan popularitas kedua *framework* PHP tersebut.



Gbr. 1 *Trend dan popularity* Codeigniter dan Laravel berdasarkan Google Trend

Seperti yang di tunjukkan dalam Gbr. 1 jika dilihat dari popularitas dan *trend* yang bersumber dari Google Trends [10]. Pada sepanjang tahun 2022 Laravel menduduki posisi lebih unggul dibandingkan dengan Codeigniter.



Gbr. 2 List *framework* yang paling banyak di gunakan di 2021 berdasarkan survei JetBrains

Sedangkan jika dilihat dari sumber lain, di tahun 2021 dalam situs resmi JetBrains dikatakan bahwa Laravel merupakan *framework* PHP yang paling banyak digunakan dengan persentase sebesar 67%. Codeigniter berada diposisi keempat sebesar 9% [11].

Bersumber dari beberapa referensi diatas maka bisa dilihat bahwa Laravel yang notabennya adalah *framework* PHP baru namun dapat menduduki posisi yang lebih tinggi dari Codeigniter pendahulunya dan bahkan berada di posisi teratas dalam trend di beberapa tahun terakhir.

2) *Database*: Dikutip dari website resmi Laravel, disebutkan bahwa Laravel 8 menyediakan dukungan pihak pertama untuk lima *database*, antara lain: MariaDB, MySQL, PostgreSQL, SQLite, dan SQL Server [12]. Sedangkan dikutip dari website resmi Codeigniter, Codeigniter 4 support dengan DBMS : MySQL, PostgreSQL, SQLite3, MSSQL, Oracle. Selain itu dengan bantuan PDO Driver Codeigniter 4 dapat menggunakan CUBRID, Interbase/Firebird, dan ODBC [13].

3) *Programming Paradigm*: Laravel menggunakan *Component Oriented*, sedangkan Codeigniter menggunakan *Object-Oriented Event Driven Functional*. Dikutip berdasarkan Referensi [14], *Component-Oriented Programming* (COP) atau Pemrograman Berorientasi Komponen memungkinkan program dibuat dari komponen *software pre-built*, yang merupakan blok kode komputer yang dapat digunakan kembali

dan berdiri sendiri. Komponen ini tentunya harus mengikuti standar yang telah ada atau telah ditentukan sebelumnya termasuk koneksi, antarmuka, pembuatan versi, serta penerapan. Sedangkan Codeigniter menggunakan arsitektur *Object Oriented* dengan mengimplementasikan *behaviour event-driven*. *Event-Driven* programming merupakan salah satu programming *paradigm* dimana alur program di tentukan oleh *event* yang merupakan *output* atau tindakan pengguna atau juga bisa berupa pesan dari program lainnya.

4) *Routing*: Dalam situs resmi Laravel mengatakan setidaknya terdapat tiga kategori *routing*, yaitu *route parameters*, *basic routing*, dan *named routes* [12]. Sedangkan pada Codeigniter, sama halnya dengan Laravel *routing* pada Codeigniter memiliki tugas untuk menentukan *controller* serta *method/fungsi* yang bisa dijalankan ketika pengguna aplikasi mengakses pada alamat/url tertentu. Dikutip dari *website* resmi Codeigniter, Codeigniter mempunyai dua jenis *routing*. *Defined Route Routing* dan *Auto Routing*. *Defined Route Routing* memungkinkan pengguna untuk mendefinisikan *routing* secara manual yang mengizinkan fleksibilitas pada URL. Sedangkan *Auto Routing* otomatis merutekan HTTP *request* berdasarkan *conventions* dan *execute* controller yang sesuai. Tidak perlu menentukan rute secara manual, namun secara *default Auto Routing* berstatus *false*, sehingga jika ingin menggunakan fitur ini pengguna harus mengaktifkannya terlebih dahulu [13].

5) *Built-in Modules*: Dikutip dari Referensi [15], *Built-in Modules* dalam laravel mengizinkan programmer atau pengembang untuk membantu *project* dalam modul-modul kecil melalui *bundle*, dan dapat juga menggunakan kembali modul di dalam berbagai macam *project* yang berbeda. Sedangkan pada Codeigniter tidak mendukung fitur *Built-in Modules*, disini membutuhkan pengembang untuk membuat sendiri dan mengelolanya menggunakan tambahan *Modular Extension*.

6) *HTTPS Support*: Dikutip dari Referensi [15], HTTPS Support pada Laravel memungkinkan pengembang atau developer untuk mengkonfigurasi HTTPS *Routes* khusus. Pengembang juga dapat membuat URL khusus untuk setiap routes. Laravel menjamin keamanan data yang dikirimkan dengan menambahkan *https://* secara otomatis sebelum URL. Sedangkan pada Codeigniter tidak sepenuhnya mendukung penggunaan HTTPS *Routes*. Pengembang harus mengelola URL *Helper* untuk memastikan bahwa transfer data agar aman dengan pengembangan pats.

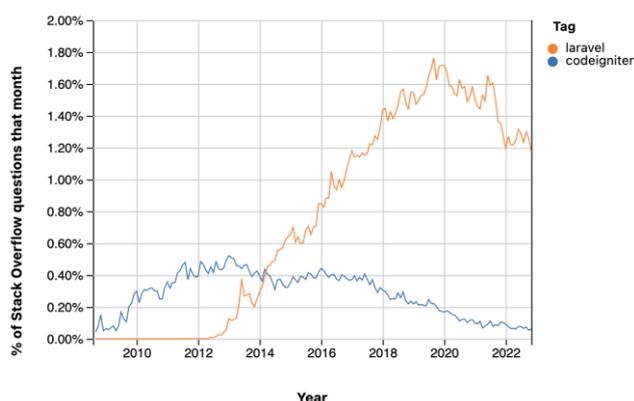
7) *Support RestAPI*: Dalam Laravel, terdapat fitur RESTful *Controllers* yang dapat digunakan oleh para pengembang untuk membangun API REST tanpa harus menambah waktu dan usaha yang lebih. Fitur ini dapat diaktifkan dengan mudah dengan mengatur properti *\$restful* menjadi "true" pada RESTful *Controller*. Sementara itu, pada Codeigniter tidak tersedia fitur khusus untuk pengembangan

API REST. Para pengembang harus menulis kode secara manual untuk membuat API REST dengan desain kustom.

8) *Template Engine*: Laravel hadir dengan *template engine* bawaan yaitu Blade dalam pengelolaan *view*. Blade tidak membatasi pengguna untuk menggunakan kode PHP biasa di *template* pengguna. Faktanya, semua *template* Blade dikompilasi menjadi kode PHP biasa dan di-cache hingga dimodifikasi. File dengan *Blade template* menggunakan ekstensi *.blade.php*. Sedangkan pada Codeigniter tidak mengharuskan untuk menggunakan *template engine* dalam viewnya. Dalam dokumentasi resmi Codeigniter 4 ada juga bawaan seperti *template engine* bernama *Parser*, namun *Parser Class* memproses "script PHP/HTML" yang disimpan di jalur tampilan aplikasi serta skrip ini tidak boleh berisi PHP apa pun. Selain itu pengembang dapat mengintegrasikan secara mandiri *template engine tools* seperti Smarty atau *template engine* yang lainnya.

9) *Authentication*: Laravel menyediakan kelas autentikasi yang memudahkan pengembang untuk mengimplementasikan autentikasi dan otorisasi dalam aplikasi *web*. Tetapi Codeigniter tidak memiliki fitur autentikasi bawaan, jadi pengembang perlu menulis ekstensi Codeigniter khusus untuk menyediakan autentikasi.

10) *Learning Curve dan Community*: Laravel menawarkan beberapa fitur tambahan jadi akan sedikit sulit dipahami oleh pemula [13]. Sedangkan Codeigniter cenderung lebih mudah dipahami oleh pemula, karena struktur yang lebih sederhana dan *easy setup*. Hal tersebut terjadi karena Laravel yang memiliki kerangka kerja yang sedikit sukar di pahami oleh pemula, namun Laravel memiliki sifat "clean architecture" dan "composer friendly" yang memungkinkan *developer* profesional membagi proyek besar kedalam modul kecil sehingga lebih mudah untuk di *maintenance*. Laravel dan Codeigniter adalah *open source tools*. Dilihat pada Desember 2022 Laravel memiliki 71.9 ribu *Github stars* dan 23.2 ribu *forks* pada Github. Sedangkan Codeigniter memiliki 18.2 ribu *Github stars* dan 7.7 ribu *forks*.



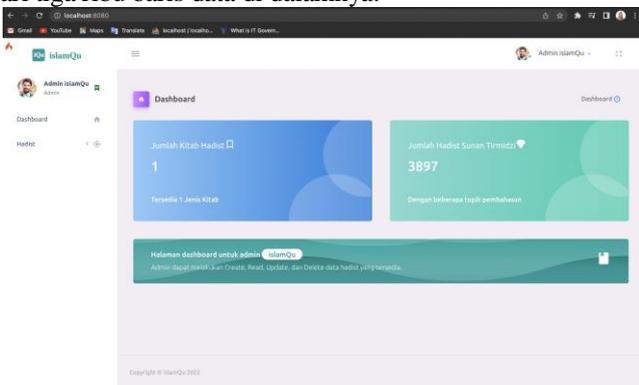
Gbr. 3 Stack Overflow Trend Laravel vs Codeigniter

Pada Gbr. 3 Merupakan grafik yang didapat dari Stack Overflow Trend. Berdasarkan gambar tersebut dapat dilihat bahwa Laravel memiliki indeks pencarian yang lebih tinggi.

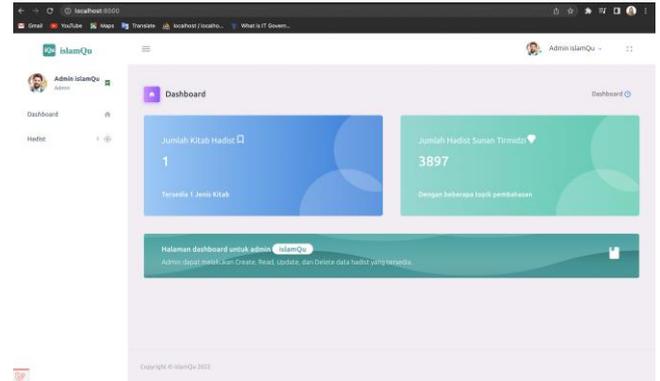
11) *Package*: Laravel khususnya pada versi 8 memiliki beberapa *package* yang dapat digunakan antara lain : Breeze, Cashier (Stripe), Cashier (Paddle), Envoy, Fortify, Homestead, Horizon, Octane, Passport, Jetstream, Sail, Socialite, Sanctum, Scout, Telescope, Dusk, Valet [12]. Sedangkan pada Codeigniter terkhusus untuk versi 4 memiliki beberapa *package* antara lain: Shield, Settings, Cache, DevKit, Coding Standard [13]. *Package* yang tersedia di kedua *framework* tersebut memiliki fitur yang dapat digunakan untuk pengembangan *website*, contohnya *package* yang berkaitan dengan *authentication* dan *authorization*.

B. Komparasi Sistem Aplikasi Web

Aplikasi web dibuat menggunakan Framework PHP Codeigniter 4 dan Framework PHP Laravel 8 yang mengimplementasikan arsitektur MVC didalamnya (dapat dilihat pada Gbr. 4 dan Gbr. 5). Arsitektur MVC (*Model View Controller*) mengutamakan pembagian komponen aplikasi menjadi tiga bagian utama, yaitu *model*, *view*, dan *controller*. *Model* digunakan untuk mengelola data aplikasi, *View* digunakan untuk menampilkan data, dan *controller* digunakan untuk mengontrol aliran data antara *model* dan *view*. Kedua aplikasi ini dibuat serupa yaitu aplikasi yang melayani service CRUD (*Create, Read, Update, Delete*) sederhana dengan lebih dari tiga ribu baris data di dalamnya.



Gbr. 4 Aplikasi web menggunakan *framework* Codeigniter 4



Gbr. 5 Aplikasi web menggunakan *framework* Laravel 8

Pengujian dilakukan menggunakan Google Lighthouse terhadap kedua *website* yang telah dibuat. *Tools open source* yang dibuat oleh Google ini digunakan untuk mengaudit atau mengukur kualitas suatu aplikasi *website*. Pengujian dilakukan sepuluh kali pada masing-masing aplikasi. Terdapat beberapa metrik utama dalam pengukuran menggunakan Google Lighthouse dengan *range* penilaian dari 0 sampai dengan 100, dimana nilai 0-49 bernilai buruk, nilai 50-89 bernilai cukup, dan 90-100 bernilai baik. Pada penelitian kali ini difokuskan pada salah satu metrik utama, yaitu metrik *performance*. Perlu diperhatikan pembuatan kedua aplikasi ini menggunakan template yang sama dan koneksi *database* yang sama. Pada Table 1 dapat dilihat hasil pengujian pada kedua aplikasi dengan iterasi sebanyak masing-masing sepuluh kali pengujian.

Tabel 1 HASIL PENGUJIAN MENGGUNAKAN GOOGLE LIGHTHOUSE

Perulangan	Performance	
	Codeigniter 4	Laravel 8
1	92	89
2	87	88
3	90	93
4	87	98
5	89	90
6	90	91
7	86	91
8	89	93
9	91	93
10	88	91
SUMMARY	88,9	91,7

Hasil dari metrik *performance* didapatkan berdasarkan submetrik berikut:

- *First Contentful Paint* (FCP) = Mengukur waktu yang dibutuhkan saat teks atau gambar pertama di *render*.
- *Time to Interactive* (TTI) = Jumlah waktu yang dibutuhkan halaman untuk menjadi interaktif sepenuhnya.
- *Total Blocking Time* (TBT) = Mengukur total waktu halaman diblokir untuk merespons input pengguna, seperti klik mouse, ketuk layar, atau penekanan *keyboard*. Jumlahnya dihitung dengan menambahkan

bagian pemblokiran dari semua tugas panjang antara *First Contentful Paint* (FCP) dan *Time to Interactive* (TTI). Pada momen ini *user* bisa melihat halaman, tapi belum bisa berinteraksi.

- *Speed Index* = Lama waktu yang dibutuhkan untuk memuat seluruh elemen visual pada suatu halaman
- *Largest Contentful Paint* (LCP) = Waktu yang dibutuhkan untuk teks atau gambar terbesar di-render pada halaman. Ini memperkirakan kapan konten utama halaman dapat dilihat oleh pengguna.
- *Cumulative Layout Shift* = Penilaian Google Lighthouse akan berpengaruh buruk jika pergerakan layout di halaman situs tidak stabil dan terdapat elemen-elemen yang bergerak-gerak, oleh karena itu pergerakan layout harus dibatasi serendah mungkin."

Berdasarkan hasil pengujian performa kedua aplikasi menggunakan Google Lighthouse nilai *performance* rata-rata aplikasi menggunakan Laravel 8 sedikit lebih tinggi dibandingkan dengan nilai rata-rata aplikasi yang dibuat menggunakan Codeigniter 4. Jika aplikasi web yang dibangun sangat membutuhkan *performance* yang cepat, Laravel 8 bisa dijadikan pertimbangan. Namun perlu diperhatikan kembali pemilihan *framework* dari segi struktur, sehingga sesuai dengan *developer* dan kebutuhan aplikasi web yang sedang dibangun.

V. KESIMPULAN

Berdasarkan hasil pengujian yang telah dilakukan, maka dapat disimpulkan bahwa :

- Google Trends menyatakan bahwa sepanjang tahun 2022 Laravel menduduki posisi lebih unggul dibandingkan dengan Codeigniter. Sedangkan JetBrains menyatakan bahwa Laravel merupakan *framework* PHP yang paling banyak digunakan dengan persentase sebesar 67% pada tahun 2021. Maka dari itu dapat dilihat bahwa Laravel yang notabennya adalah *framework* PHP baru namun dapat menduduki posisi yang lebih tinggi dari Codeigniter pendahulunya dan bahkan berada di posisi teratas dalam tren di beberapa tahun terakhir.
- Codeigniter dan Laravel juga mempunyai beberapa perbedaan jika dilihat dari struktur, serta fitur yang tersedia. Codeigniter tidak disertai *template engine* pada *view*, sedangkan Laravel mewajibkan untuk menggunakan *template engine* pada *view*. Codeigniter tidak mendukung fitur *Built-in Modules* secara *default* sehingga harus di tambahkan *Modular Extension*, sedangkan Laravel mendukung fitur tersebut. Codeigniter memiliki support *database* yang lebih banyak dibandingkan dengan Laravel. Secara *learning curve* Codeigniter cenderung lebih mudah dipelajari untuk pemula dalam memahami konsep MVC dalam *framework* karena struktur folder yang lebih sederhana.
- Berdasarkan hasil pengujian performa kedua aplikasi menggunakan Google Lighthouse nilai *performance* rata-rata aplikasi menggunakan Laravel 8 sedikit lebih

tinggi dibandingkan dengan nilai rata-rata aplikasi yang dibuat menggunakan Codeigniter 4.

REFERENSI

- [1] O. W. Purbo, "A Systematic Analysis: Website Development using Codeigniter and Laravel Framework," *Enrichment: Journal of Management*, pp. 1008-1014, 2021.
- [2] M. Laaziri, K. Benmoussa, S. Khouli and M. L. Kerkeb, "A Comparative study of PHP frameworks performance A Comparative study of PHP frameworks performance," in *Procedia Manufacturing*, Tetouan, 2019.
- [3] R. Erinton, R. M. Negara and D. D. Sanjoyo, "ANALISIS PERFORMASI FRAMEWORK CODEIGNITER DAN LARAVEL MENGGUNAKAN WEB SERVER APACHE," in *e-Proceeding of Engineering*, 2017.
- [4] A. Hidayatullah and D. Arius, "ANALISIS DAN PERANCANGAN TOKO ONLINE GERAIHANNA BERBASIS SEO MENGGUNAKAN LARAVEL," STMIK AMIKOM Yogyakarta, Yogyakarta, 2015.
- [5] Y. Yudhanto and H. A. Prasetyo, "Mudah Menguasai Framework Laravel," PT Elex Media Komputindo, Jakarta, 2019.
- [6] I. Solikin, "PERANCANGAN SISTEM INFOMASI PENJUALAN BERBASIS FRAMEWORK MODEL VIEW CONTROLLER (MVC) PADA PT THAMRIN BROTHER CABANG OKI," *Jurnal Media Informatika dan Komputer*, vol. 4, no. 1, pp. 174-191, 2014.
- [7] A. P. Basuki, "Proyek Membangun Website Berbasis PHP dengan Codeigniter," Lokomedia, Yogyakarta, 2014.
- [8] A. Tanari, A. Handoyo and J. Andjarwirawan, "Aplikasi Pencarian Jurnal Ilmiah dengan Term FrequencyInverse Document Frequency," *Jurnal Infra*, vol. 7, no. 1, 2019.
- [9] I. D. Id, Ebook Framework CodeIgniter: Sebuah Panduan dan Best Practice, Jakarta, 2015.
- [10] "Google Trends," [Online]. Available: <https://trends.google.com/trends/explore?geo=ID&q=%2Fm%2F02qgdkj,%2Fm%2F0jwy148>.
- [11] "JetBrains Devecosystem 2021," 2021. [Online]. Available: <https://www.jetbrains.com/lp/devecosystem-2021/php/>.
- [12] "Laravel," Laravel LLC, 2022. [Online]. Available: laravel.com. [Accessed December 2022].
- [13] "Codeigniter," CodeIgniter Foundation, 5 November 2022. [Online]. Available: codeigniter.com. [Accessed December 2022].
- [14] F. A. Budiyanto, N. A. Indartono and R. Hartono, "PENGEMBANGAN FRONT END WEBSITESTIKOM PGRI BANYUWANGI DENGAN PENDEKATAN COMPONENT ORIENTED PROGRAMMING DAN STYLE FLAT DESIGN," *Jurnal Informatika dan Komputer*, vol. 9, no. 2, pp. 105-117, 2019.
- [15] P. Jackson, "CodeIgniter vs Laravel – Difference Between Them," Guru99, 26 November 2022. [Online]. Available: <https://www.guru99.com/laravel-vs-codeigniter.html>. [Accessed December 2022].



Wini Muthia Kansha. Bogor, 08 Oktober 1999. Lulus D3 Teknik Komputer di Institut Pertanian Bogor (IPB *University*) pada Tahun 2020. Di Tahun 2022 sedang menempuh pendidikan tahun terakhir jenjang SI Teknik Informatika di STMIK Antar Bangsa. Dua tahun terakhir bekerja sebagai *software developer*. Pernah mempublikasikan jurnal sebagai penulis pertama dengan judul “Pembuatan Sistem Database Cluster Menggunakan Aplikasi Galera Cluster di Sekolah Vokasi IPB *University*” yang dipublish pada Jurnal Sains Terapan : Wahana Informasi dan Alih Teknologi Pertanian, Sekolah Vokasi IPB *University*.



Saherih. Subang, 17 januari 1994, pada tahun 2018 mengambil pendidikan paket C di PKBM PPI Taiwan atau setara dengan SMA dan lulus pada tahun 2020. Tahun 2022 sedang menempuh pendidikan semester 5 Teknik Informatika di kampus STMIK Antar Bangsa. Selama hampir 11 tahun ini menjadi Pekerja Migran Indonesia di Taiwan sebagai pengurus lansia.



Muchlis. Tangerang, 05 Mei 1986. Gelar Sarjana Komputer didapat pada Jurusan Teknik Informatika di Universitas Pamulang pada tahun 2013. Tahun 2018 lulus program Pasca Sarjana Ilmu Komputer di STMIK Nusa Mandiri. Saat ini menduduki posisi sebagai Kepala Divisi Pengelolaan Sistem Informasi & Teknologi dan Kepala Program Studi Teknik Informatika STMIK Antar Bangsa.