

ANALISIS PERBANDINGAN FRAMEWORK LARAVEL DENGAN FRAMEWORK CODEIGNITER PADA APLIKASI DI- AGNOSA PENYAKIT UNTUK ANAK MENGGUNAKAN METODE FORWARD CHAINING

Yaya Suharya, S.Kom., M.T¹, Nurul Imamah, S.T., M.T², Galih Rxy Hakiki³

Program Studi Teknik Infomatika

Fakultas Teknologi Informasi

Universitas Bale Bandung

galihrexyhakiki@gmail.com

Abstract

PHP is a popular language and has several frameworks. Among them are the Laravel framework and the Codeigniter framework which are the most widely used framework for the PHP programming language. To compare the two frameworks, the writer will make a disease diagnosis application for children under 5 years using the forward chaining method which will be used to compare the two frameworks in terms of performance (request per second, time per request and execute time), the size of the data generated (resource), and how to access the database.

Keywords: Frameworks, Laravel, Codeigniter, Expert System, Forward Chaining.

Abstrak

PHP merupakan salah satu bahasa yang populer dan memiliki beberapa framework. Diantaranya ada framework Laravel dan framework Codeigniter yang merupakan framework paling banyak digunakan untuk bahasa pemrograman PHP. Untuk membandingkan kedua framework tersebut penulis akan membuat aplikasi diagnosa penyakit untuk anak dibawah 5 tahun menggunakan metode forward chaining yang akan digunakan untuk membandingkan kedua framework tersebut dari segi performa (request per second, time per request dan execute time), ukuran data yang dihasilkan (resource), dan cara akses database.

Kata kunci: Framework, Laravel, Codeigniter, Sistem Pakar, Forward Chaining.

1. PENDAHULUAN

1.1 Latar Belakang

Dengan perkembangan zaman yang semakin modern, penyebaran informasi menggunakan media yang lebih efektif dan efisien. Awalnya semua berita, info, event dan lain lain hanya dapat diperoleh dari buku, koran, radio dan televisi. Namun sejak munculnya internet, semua lapisan masyarakat dapat memperoleh informasi yang cepat, akurat dan relevan dalam sebuah halaman web yang dapat diakses secara real time dari gadget maupun komputer. Perusahaan atau instansi saat ini dalam memberikan berbagai macam informasi menggunakan media perantara yaitu website. Pembuatan sebuah website mengalami banyak perkembangan. Perkembangan tersebut terbukti dengan banyaknya pilihan framework yang dapat digunakan untuk membangun sebuah website. Framework adalah sebuah kerangka kerja dari bahasa pemrograman dasar yang telah dikembangkan dan dipermudah penggunaannya agar suatu website dapat diselesaikan dalam waktu yang relatif singkat. Framework yang digunakan untuk membangun sebuah website yang sedang

populer saat ini seperti CodeIgniter dan Laravel.

CodeIgniter merupakan framework yang diklaim memiliki eksekusi tercepat dibandingkan dengan framework lainnya. CodeIgniter bersifat open source dan menggunakan model berbasis MVC (Model View Controller), yang merupakan konsep modern framework yang digunakan saat ini. Framework CodeIgniter memiliki beberapa kelebihan diantaranya gratis, berukuran kecil dan cepat, dokumentasi, menggunakan konsep MVC, portability (Raelda Rispadina Sitio, 2013).

Laravel adalah salah satu dari sekian banyak framework PHP yang tersedia. Laravel dibuat oleh Taylor Otwell sejak tahun 2011. Framework ini mengaku “clean and classy”, dengan kode yang lebih singkat, mudah dimengerti, dan ekspresif. Kelebihan Laravel adalah ekspresif, simpel, tersedia composer, open source, kompatibel dengan PHP 5.3 keatas, dokumentasi yang lengkap, prinsip model view controller (Asli Khatul Khasanah, 2015).

Aplikasi diagnosa untuk anak merupakan aplikasi sistem pakar untuk mendiagnosa seorang anak terkena

suatu penyakit. Aplikasi ini ditujukan untuk anak 5 tahun kebawah karena pada umur tersebut rawan bagi anak terkena penyakit. Aplikasi juga dibuat semata hanya untuk menganalisis perbandingan framework. Penulis akan membandingkan Framework CodeIgniter dan Framework Laravel dari segi performa (*request per second*, *time per request*, dan *execute time*), segi ukuran, dan cara akses database.

1.2 Rumusan Masalah

Dari latar belakang yang telah dijabarkan diatas, maka dapat dirumuskan suatu rumusan masalah adalah sebagai berikut:

- 1) Bagaimana mendapatkan hasil perbandingan Framework Laravel dan Framework Codeigniter dari segi kecepatan/*performa*?
- 2) Bagaimana mendapatkan hasil perbandingan Framework Laravel dan Framework Codeigniter dari segi ukuran/*resource*?
- 3) Bagaimana mendapatkan hasil perbandingan Framework Laravel dan Framework Codeigniter dari segi cara akses *database*?

- 4) Bagaimana pembuatan aplikasi diagnosa penyakit pada anak dengan menggunakan Framework Laravel dan Framework Codeigniter?

1.3 Pembatasan Masalah

Untuk menjelaskan permasalahan agar pembahasan ini tidak terlalu jauh dari kajian masalah yang penulis paparkan. maka penulis membatasi masalah pada hal-hal berikut ini:

- 1) Aplikasi yang digunakan untuk penelitian ini adalah aplikasi diagnosa penyakit pada anak umur 5 ke bawah.
- 2) Perbandingan framework Laravel dan framework Codeigniter dari segi performa, ukuran dan jumlah penggunaan memory.
- 3) Performa diambil dari pengukuran *request per second*, *time per second* dan *execute time*.
- 4) *Request per second* dan *time per request* diambil dengan tool Apache Benchmark (ab).
- 5) *Execute time* diambil dengan menggunakan fungsi yang sudah tersedia pada PHP. Pengujian dilakukan dengan

menambahkan fungsi pada awal skrip kode dan akhir skrip kode. Maka akan didapat waktu *execute time* dari tiap fitur yang tersedia.

- 6) Cara akses *database* dilakukan dengan membandingkan dan menganalisis akses *database* untuk tata cara pengaksesan tabel dalam operasi *Create, Read, Update, Delete* (CRUD). Cara pengaksesan *database* di Laravel menggunakan Eloquent ORM dan cara pengaksesan *database* di CodeIgniter menggunakan Query Builder.
- 7) Aplikasi yang dibuat hanya berfokus menganalisis perbandingan framework dan tidak akan dihosting pada *server online*.

1.4 Tujuan

Adapun tujuan yang ingin dicapai dalam penelitian ini, adalah:

- 1) Menghasilkan analisis dari perbandingan Framework Laravel dan Framework CodeIgniter dari segi performa.

- 2) Menghasilkan analisis dari perbandingan Framework Laravel dan Framework CodeIgniter dari segi ukuran.
- 3) Menghasilkan analisis dari perbandingan Framework Laravel dan Framework CodeIgniter dari segi cara akses database.

1.5 Metodologi Penelitian

1.5.1 Kerangka Kerja Penelitian

Untuk membantu dalam menyelesaikan penyusunan penelitian, maka perlu adanya susunan kerangka kerja yang jelas. Kerangka kerja ini merupakan langkah-langkah yang akan dilakukan dalam penyelesaian masalah yang akan dibahas, dengan tahapan-tahapan sebagai berikut:

1. Observasi Awal

Pada observasi awal penulis melakukan pengamatan melalui internet dan menentukan masalah yang selanjutnya ditentukan tujuan serta pemecahan masalahnya.

2. Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data dengan metode studi

pustaka untuk mendapatkan data yang dibutuhkan penulis

3. Pengembangan Sistem

Pada tahap ini dilakukan proses pengembangan sistem dengan menggunakan metode SDLC (*System Development Life Cycle*) dengan metode *Waterfall* untuk pembuatan aplikasinya.

4. Pengujian

Pada tahap ini dilakukan pengujian perbandingan antara framework Laravel dan framework codeigniter.

5. Pembuatan Laporan

Pada tahap ini dilakukan proses pembuatan laporan yang disusun berdasarkan hasil penelitian.

1.5.2 Metode Pengumpulan Data

Dalam melakukan penelitian ini, penulis melakukan beberapa metode pengumpulan data dan referensi yang dibutuhkan dalam penelitian ini. Adapun beberapa cara tersebut adalah sebagai berikut:

1. Wawancara

Pengumpulan data dengan mengadakan wawancara atau tanya jawab baik secara *offline* atau *online* kepada beberapa orang yang telah menggunakan framework Laravel dan Codeigniter.

2. Studi Pustaka

Mengumpulkan data dengan cara membaca referensi dari buku-buku dan jurnal-jurnal yang berhubungan dengan topik penelitian.

1.5.3 Metode Pengembangan

Metode yang digunakan penulis dalam pengembangan aplikasi adalah metode Waterfall yang memiliki tahapan seperti berikut:

1. Analysis

Pada tahapan ini dilakukan analisis kebutuhan aplikasi dan perancangan aplikasi. Analisis dan perancangan dilakukan untuk menentukan struktur database yang akan digunakan untuk penyimpanan data pada aplikasi.

2. Desain

Pada tahapan ini dilakukan pengumpulan data untuk

kebutuhan aplikasi. Data yang dikumpulkan adalah referensi desain aplikasi, gambar ilustrasi, skema warna, pembuatan logo dan jenis font.

3. *Coding*

Pada tahapan ini dilakukan penulisan skrip kode untuk pembuatan 2 aplikasi dengan menggunakan Laravel dan Codeigniter.

4. *Testing*

Pada tahapan ini dilakukan pengujian apakah aplikasi berjalan dengan baik sesuai dengan fungsi yang telah ditentukan. Jika masih terdapat fungsi yang belum sesuai maka akan dilakukan perbaikan.

5. *Maintenance*

Pada tahapan ini dilakukan maintenance jika dibutuhkan. Dan aplikasi sudah lulus tes untuk masuk ke proses selanjutnya yaitu proses pengujian untuk membandingkan *framework* Laravel dan Codeigniter.

1.6 Metode Pengujian

1.6.1 Pengujian Performa

Pengujian performa dilakukan dengan membandingkan Framework Laravel dengan Framework Codeigniter dari beberapa faktor antara lain:

- Pengujian Request Per Second (RPS)
- Pengujian Time Per Request
- Pengujian Execute Time (Waktu Eksekusi)

1.6.2 Pengujian Ukuran

Analisis ukuran dilakukan dengan membandingkan dan menganalisis struktur direktori dan besarnya total file yang terdapat pada direktori.

1.6.3 Pengujian Cara Akses Database

Cara akses database dilakukan dengan membandingkan dan menganalisis akses database untuk tata cara pengaksesan tabel dalam operasi Create, Read, Update, Delete (CRUD). Cara pengaksesan database di Laravel menggunakan Eloquent ORM dan cara pengaksesan database di CodeIgniter menggunakan Query Builder.

2. DASAR TEORI

2.1 HTML

HTML merupakan singkatan dari *Hypertext Markup Language* yaitu Bahasa standar web yang dikelola penggunaanya oleh W3C (*World Wide Web Consortium*) berupa tag-tag yang Menyusun setiap elemen dari website. HTML berperan sebagai penyusun struktur halaman website yang menempatkan setiap elemen website sesuai layout yang diinginkan. Yang bisa dilakukan dengan HTML yaitu:

- Mengatur tampilan dari halaman web dan isinya.
- Membuat tabel dalam halaman web.
- Mempublikasikan halaman web secara online.
- Membuat form yang bisa digunakan untuk menangani registrasi dan transaksi via web.
- Menambahkan objek-objek seperti citra, audio, video, animasi, java applet dalam halaman web.
- Menampilkan area gambar (canvas) dibrowser.

2.2 CSS

Cascading Style Sheet (CSS) dapat menyimpat format dan menggunakannya kapan pun dan di mana pun kita inginkan. Seperti *Formatting* dan *Style* dalam membuat dokumen office, maka *style sheets* juga sangat dinamis. Memang menggunakan *style sheet* bukan suatu keharusan dalam membuat web, namun ketika website dengan halaman yang sangat banyak kedepannya akan sulit untuk debugging, perbaikan dan perawatannya.

Style sheet dapat membuat efek-efek tertentu untuk konten web. Misalnya bagian header tabel pada web selalu ber-*font* Verdana, ukuran 16 dan berwarna hitam. Maka dapat didefinisikan *style* tersebut untuk tag <th>. Contoh lain misalnya tulisan pada suatu web bertipe Verdana dan ukuran 12, maka definisikan *style* tersebut pada tag <p>.

Cascading Style Sheet (CSS) sudah didukung oleh hampir semua web browser karena CSS telah distandarkan oleh World Wide Web Consortium (W3C). Jadi ini pilihan tepat bagi anda untuk memformat halaman web agar terlihat cantik di manapun

web tersebut dibuka. Ada 4 cara memasang kode CSS ke dalam kode HTML / halaman web, yaitu:

- *Inline style sheet* (Memasukkan kode CSS langsung pada tag HTML)
- *Internal Style Sheet* (*Embed* atau memasang kode CSS ke dalam bagian <head>)
- Me-link ke *external CSS*
- *Import CSS file*

2.3 PHP

Pada awalnya PHP merupakan kependekan dari *Personal Home Page* (Situs Personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari web.

Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilisannya kode sumber ini menjadi sumber terbuka, maka banyak pemrogram yang tertarik untuk ikut mengembangkan PHP.

Pada November 1997, dirilis PHP/FI 2.0. Pada rilis 2.0 ini, *interpreter* PHP sudah diimplementasikan dalam program C. Di dalam rilisan ini juga ikut disertakan modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan.

Kemudian pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang *interpreter* PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada juni 1998, perusahaan tersebut merilis *interpreter* baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0 dan singkatan PHP diubah menjadi akronim berulang **PHP: Hypertext Preprocessing**.

Pada pertengahan tahun 1999, Zend merilis *interpreter* PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari *interpreter* PHP mengalami perubahan

besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan Bahasa pemrograman ke arah paradigma berorientasi objek.

2.4 MYSQL

MySQL adalah salah satu aplikasi DBMS yang sudah sangat banyak digunakan oleh para pemrogram aplikasi web. Contoh DBMS lainnya adalah PostgreSQL, (*freeware*), SQL, Server, MS Access dari Microsoft, DB2 dari IBM, Oracle dari Oracle Corp, Dbase, FoxPro, dsb.

Kelebihan dari MySQL adalah gratis, handal, selalu di-update dan banyak forum yang memfasilitasi para pengguna jika memiliki kendala. MySQL juga menjadi DBMS yang sering dibundling dengan web server sehingga proses instalasinya lebih mudah.

2.5 Web Browser

Web browser secara umum adalah suatu perangkat lunak atau software yang digunakan untuk mencari informasi atau mengakses situs-situs yang ada di internet. Perangkat ini akan lebih memudahkan pengguna dalam

mengakses data atau mencari referensi yang dibutuhkan. Ada berbagai macam perangkat web browser yang kini digunakan seperti Mozilla Firefox, Google Chrome, Opera, Safari, Internet Explorer dan lain sebagainya.

Awalnya, web browser berorientasi pada teks dan belum dapat menampilkan gambar. Namun, web browser sekarang tidak hanya menampilkan gambar dan teks saja, tetapi juga memutar file multimedia seperti video dan suara. Web browser juga dapat mengirim dan menerima email, mengelola HTML, sebagai input dan menjadikan halaman web sebagai hasil output yang informative.

Tujuan utama dari web browser adalah untuk membawa sumber informasi kepada pengguna. Proses ini dimulai ketika pengguna memasukan sebuah Uniform Resource Identifier (URI), misalnya, <https://thidiweb.com> ke dalam browser. Sumber yang telah diambil web browser akan ditampilkan. HTML ditampilkan ke mesin tata letak browser, dan akan diubah dari markup ke dokumen interaktif. Selain dari HTML, browser umumnya bisa menampilkan setiap jenis konten yang menjadi bagian dari suatu halaman web.

2.6 XAMPP

Web server adalah tempat menyimpan aplikasi web kemudian mengaksesnya melalui Internet. Setiap perubahan, kecil maupun besar, Dibutuhkannya web server ini adalah karena untuk *server side script* seperti PHP, pemeriksaan baru akan tampil jika menggunakan web server. Itulah bedanya dengan *client side script* seperti HTML, CSS dan Javascript yang cukup dengan browser script dapat diketahui apakah sudah sesuai dengan keinginan atau belum

Web server Apache sudah sangat populer di Internet sejak April 1996. Kelebihan lainnya adalah The Apache Software Foundation sangat tinggi komitmennya untuk terus mengembangkan web server Apache sehingga keterjaminannya untuk senantiasa kompatibel dengan teknologi web terkini sangat tinggi. Dan yang terakhir, begitu banyak forum-forum untuk bertanya jika terdapat masalah dalam penggunaan web server Apache.

2.7 Framework

Framework atau kerangka kerja pengembangan aplikasi adalah suatu standar yang harus diikuti untuk

melakukan pengembangan aplikasi oleh pemrogram. Standar ini mengatur banyak hak, mulai dari nama file, direktori, dan cara memprogramnya. Framework memberikan kerangka program, kumpulan library dan fungsi yang bisa langsung digunakan, serta aturan untuk menggunakannya.

Pengembangan aplikasi diminta untuk mengikuti aturan yang telah ditetapkan oleh framework, agar bisa menghasilkan sebuah aplikasi atau modul dengan cepat dengan menggunakan standar, File apa saja yang harus disiapkan dan isinya harus seperti apa adalah aturan yang telah ditetapkan oleh framework.

Framework pengembang aplikasi merupakan salah satu solusi untuk menjawab tuntutan agar bisa dengan cepat menyelesaikan pembuatan atau pengembangan aplikasi masa kini. Mengapa? Karena sekarang pengembangan aplikasi dituntut untuk bisa segera memberikan hasil dari aplikasi yang diinginkan oleh pengguna atau pemberi pekerjaan.

2.8 Framework Laravel

Laravel adalah sebuah framework PHP yang dirilis dibawah lisensi MIT, dibangun dengan konsep MVC (model view controller). Laravel adalah pengembangan website berbasis MVP yang ditulis dalam PHP yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, dan untuk meningkatkan pengalaman bekerja dengan aplikasi dengan menyediakan sintaks yang ekspresif, jelas dan menghemat waktu.

MVC adalah sebuah pendekatan perangkat lunak yang memisahkan aplikasi logika dari presentasi. MVC memisahkan aplikasi berdasarkan komponen- komponen aplikasi, seperti : manipulasi data, controller, dan user interface.

- 1) Model, Model mewakili struktur data. Biasanya model berisi fungsi-fungsi yang membantu seseorang dalam pengelolaan basis data seperti memasukkan data ke basis data, pembaruan data dan lain-lain.
- 2) View, View adalah bagian yang mengatur tampilan ke pengguna.

Bisa dikatakan berupa halaman web.

- 3) Controller, Controller merupakan bagian yang menjembatani model dan view.

Beberapa fitur yang terdapat di Laravel :

- Bundles, yaitu sebuah fitur dengan sistem pengemasan modular dan tersedia beragam di aplikasi.
- Eloquent ORM, merupakan penerapan PHP lanjutan menyediakan metode internal dari pola “active record” yang mengatasi masalah pada hubungan objek database.
- Application Logic, merupakan bagian dari aplikasi, menggunakan controller atau bagian Route.
- Reverse Routing, mendefinisikan relasi atau hubungan antara Link dan Route.
- Restful controllers, memisahkan logika dalam melayani HTTP GET and POST.
- Class Auto Loading, menyediakan loading otomatis untuk class PHP.

- View Composer, adalah kode unit logikal yang dapat dieksekusi ketika view sedang loading.
- IoC Container, memungkinkan obyek baru dihasilkan dengan pembalikan controller.
- Migration, menyediakan sistem kontrol untuk skema database.
- Unit Testing, banyak tes untuk mendeteksi dan mencegah regresi.
- Automatic Pagination, menyederhanakan tugas dari penerapan halaman.

2.9 Framework Codeigniter

Codeigniter (CI) adalah framework pengembangan aplikasi (Application Development Framework) dengan menggunakan PHP, suatu kerangka pembuatan program dengan menggunakan PHP. Pengembang dapat langsung menghasilkan program dengan cepat, dengan mengikuti kerangka kerja untuk membuat yang telah disiapkan oleh framework CI ini. Dengan menggunakan framework, kita tidak perlu membuat program dari awal, tetapi kita sudah diberikan library fungsi-fungsi yang sudah diorganisasi untuk dapat

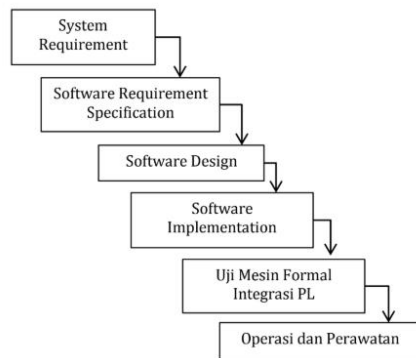
membuat suatu program dengan cepat. Kita hanya perlu memanggil fungsi-fungsi yang sudah ada untuk memproses data, kemudian memanggil fungsi untuk menampilkannya.

Codeigniter (CI) adalah framework pengembangan aplikasi (Application Development Framework) dengan menggunakan PHP, suatu kerangka untuk bekerja atau membuat program dengan menggunakan PHP yang lebih sistematis. Pemrogram tidak perlu membuat program dari awal (From scratch), karena CI menyediakan sekumpulan library dari fungsi yang banyak, yang diperlukan untuk menyelesaikan pekerjaan yang umum, dengan menggunakan antarmuka dan struktur logika yang sederhana untuk mengakses librarinya. Pemrogram dapat memfokuskan dari pada kode yang harus dibuat untuk menyelesaikan suatu pekerjaan.

2.10 Forward Chaining

Forward Chaining merupakan suatu penalaran yang dimulai dari fakta untuk mendapatkan kesimpulan (conclusion) dari fakta tersebut. Forward chaining bisa dikatakan sebagai

strategi inference yang bermula dari sejumlah fakta yang diketahui. Pencarian dilakukan dengan menggunakan rules yang premisnya cocok dengan fakta yang diketahui tersebut untuk memperoleh fakta baru dan melanjutkan proses hingga goal dicapai atau hingga sudah tidak ada rules lagi yang premisnya cocok dengan fakta yang diketahui maupun fakta yang diperoleh.



Gambar 2. 1 Struktur SDLC

Forward chaining bisa disebut juga runut maju atau pencarian yang dimotori data (data driven search). Jadi pencarian dimulai dari premis-premis atau informasi masukan (if) dahulu kemudian menuju konklusi atau derived information (then). Forward Chaining berarti menggunakan himpunan aturan kondisi-aksi. Dalam metode ini, data digunakan untuk menentukan aturan mana yang akan

dijalankan atau dengan menambahkan data ke memori kerja untuk diproses agar ditemukan suatu hasil.

Forward Chaining digunakan jika :

1. Banyak aturan berbeda yang dapat memberikan kesimpulan yang sama.
2. Banyak cara untuk mendapatkan sedikit konklusi.
3. Benar-benar sudah mendapatkan berbagai fakta, dan ingin mendapatkan konklusi dari fakta-fakta tersebut.

2.11 Waterfall

Waterfall merupakan model yang membangun perangkat lunak berdasarkan Daur Hidup Perangkat Lunak (SDLC), yaitu model yang mempunyai struktur yang dimulai dari perencanaan, analisis, desain dan implementasi, hingga tahap pengembangan dalam waterfall mempunyai struktur model pengembangan yang disebut dengan *linier dan sequential*.

Waterfall merupakan salah satu metode dalam SDLC yang mempunyai ciri khas pengerjaan yaitu setiap fase dalam waterfall harus diselesaikan terlebih dahulu sebelum melanjutkan ke fase selanjutnya. Artinya fokus terhadap masing-masing fase dapat dilakukan maksimal karena jarang adanya pengerjaan yang sifatnya parallel walaupun dapat saja terjadi paralelisme dalam waterfall.

Tahapan dari metode waterfall adalah sebagai berikut:

a. Requirement Analysis

Seluruh kebutuhan *software* harus bisa didapatkan dalam fase ini, termasuk di dalamnya kegunaan *software* yang diharapkan pengguna dan batasan *software*. Informasi ini biasanya dapat diperoleh melalui wawancara, survey atau diskusi. Informasi tersebut dianalisis untuk mendapatkan dokumentasi kebutuhan pengguna untuk digunakan pada tahap selanjutnya.

b. System Design

Tahap ini dilakukan sebelum melakukan *coding*. Tahap ini bertujuan untuk memberikan

gambaran apa yang seharusnya dikerjakan dan bagaimana tampilannya. Tahap ini membantu dalam memspesifikasikan kebutuhan *hardware* dan sistem serta mendefinisikan arsitektur sistem secara keseluruhan.

c. Implementation

Dalam tahapan ini dilakukan pemrograman. Pembuatan *software* dipecah menjadi modul-modul kecil yang nantinya akan digabungkan dalam tahap berikutnya. Selain itu dalam tahap ini juga dilakukan pemeriksaan terhadap modul yang dibuat, apakah sudah memenuhi fungsi yang diinginkan atau belum.

d. Integration & Testing

Di tahap ini dilakukan penggabungan modul-modul yang sudah dibuat dan dilakukan pengujian ini dilakukan untuk mengetahui apakah *software* yang dibuat telah sesuai dengan desainnya dan masih terdapat kesalahan atau tidak.

e. Operation & Maintenance

Ini merupakan tahap terakhir dalam model waterfall. Software yang sudah dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaikan implementasi unit sistem dan peningkatan jasa sistem sebagai kebutuhan baru.

2.12 Sistem Pakar

Sistem pakar sistem computer yang ditujukan untuk meniru semua aspek (*emulates*) kemampuan pengambilan keputusan (*decision making*) seorang pakar. Sistem pakar memanfaatkan secara maksimal pengetahuan khusus selaknya seorang pakar untuk memecahkan masalah.

Pakar atau ahli (*expert*) didefinisikan sebagai yang memiliki pengetahuan atau keahlian khusus yang tidak dimiliki oleh kebanyakan orang. Seorang pakar dapat memecahkan masalah yang tidak mampu dipecahkan kebanyakan orang. Dengan kata lain, dapat memecahkan suatu masalah dengan lebih efisien namun bukan berarti lebih murah.

Pengetahuan yang dimuat ke dalam sistem pakar dapat berasal dari seorang pakar atau pun pengetahuan yang berasal dari buku, jurnal, majalah, dan dokumentasi yang dipublikasikan lainnya, serta orang yang memiliki pengetahuan meskipun bukan ahli. Istilah sistem pakar (*expert system*), sering disinonimkan dengan sistem berbasis pengetahuan (*knowledge-based system*) atau sistem pakar berbasis pengetahuan (*knowledge-based expert system*).

Gambar di bawah ini mengilustrasikan konsep dasar sistem pakar berbasis pengetahuan (*knowledge-based expert system*). *User* memberikan informasi atau fakta kepada sistem dan menerima respon berupa saran ahli (*advice/expertise*). Secara internal, sistem terdiri dari dua komponen utama yaitu pengetahuan (*knowledge-based*), berisi pengetahuan yang akan digunakan oleh komponen lainnya yaitu mesin inferensi (*inference engine*) untuk menghasilkan kesimpulan sebagai respon terhadap kueri yang dilakukan *user*.

2.13 Requests Per Second

Requests Per Second, RPS, atau r / s adalah ukuran skalabilitas yang mencirikan throughput yang ditangani oleh suatu sistem.

Menggunakan RPS ketika menentukan ukuran dan skala pengujian beban adalah umum ketika menguji API, dengan cara yang sama bahwa jumlah pengunjung dan pengguna virtual (*virtual user*) bersamaan adalah konsep dan metrik yang paling umum ketika berbicara tentang situs web atau aplikasi.

Dalam versi LoadImpact saat ini, ukuran tes beban ditentukan oleh VU. Mengubah skala atau ukuran tes beban dari VU ke RPS tergantung pada sejumlah faktor seperti waktu respons, kompleksitas pemrosesan skrip, dan faktor konkurensi VU. Saat menguji titik akhir API tunggal, biasanya lebih logis untuk berpikir dalam hal permintaan per detik daripada pengguna secara bersamaan. Untuk alasan ini, kami telah mengembangkan skrip sampel yang dapat digunakan untuk memaksimalkan efisiensi Pengguna Virtual saat menguji titik akhir API. Kami memanfaatkan kemampuan Pengguna

Virtual untuk dapat membuka banyak koneksi secara paralel.

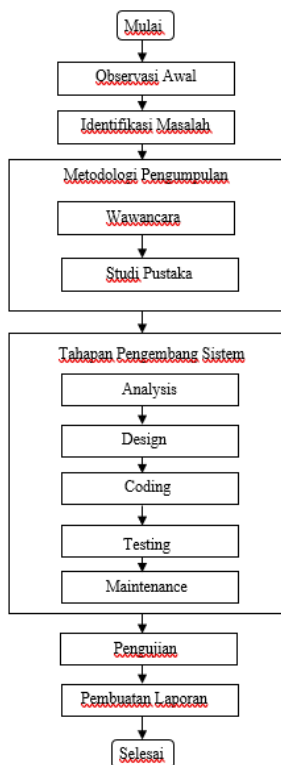
2.14 Execute Time

Execute time adalah waktu yang diperlukan untuk mengeksekusi skrip kode. Dengan mengetahui berapa waktu yang dibutuhkan maka developer dapat mengetahui skrip code yang ditulisnya cepat atau lambat. Sehingga dapat dibandingkan dan diganti dengan skrip yang lain yang lebih cepat dalam proses eksekusinya.

3 METODOLOGI PENELITIAN

3.1 Kerangka Pikir

Dalam penelitian ini, penulis melakukan tahapan sesuai dengan rencana kegiatan yang telah dibuat meliputi metodologi pengumpulan data, pembuatan aplikasi dan pengujian framework



Gambar 3. 1 Kerangka Pikir

3.2 Deskripsi Teori

3.2.1 Observasi Awal

Observasi awal penulis melakukan pengamatan melalui internet. Penulis mengamati terkait perbedaan suatu Framework dengan Framework yang lainnya.

3.2.2 Identifikasi Masalah

Penulis melakukan identifikasi masalah mengenai perbandingan Framework dengan Framework lainnya.

3.2.3 Metodologi Pengumpulan Data

Penelitian ini memerlukan data dan informasi untuk mendukung kebenaran dalam pembahasannya. Metodologi pengumpulan yang dilakukan adalah sebagai berikut:

1. Wawancara

Pengumpulan data terkait dengan penelitian ini salah satunya wawancara secara online. Data yang didapatkan adalah masalah terkait pemilihan framework sebagai bahan belajar atau memulai membuat program.

2. Studi Pustaka

Dalam tahap ini penulis mengumpulkan dan membaca referensi yang berasal dari buku-buku dan jurnal-jurnal yang berhubungan dengan topik penelitian.

3.2.4 Pengembangan Sistem

Dalam pengembangan sistem penulis membuat 2 aplikasi. Pertama dengan menggunakan Framework Laravel dan kedua menggunakan Framework Codeigniter. Penulis juga

menggunakan metode Forward Chaining untuk metode Sistem Pendukung Keputusannya.

1. *Analysis*

Pada tahap ini penulis melakukan analisis sistem dan analisis kebutuhan.

2. *Desain*

Perancangan dibuat dengan *Unified Modeling Language* (UML) agar dengan mudah dalam proses pengembangan dan visualisasinya.

3. *Coding*

Aplikasi dibuat dengan menggunakan Framework Laravel dan Framework Codeigniter.

4. *Testing*

Pengujian terhadap aplikasi yang dibuat dengan cara mengecek apakah semua fungsi telah berjalan sesuai dengan yang telah ditentukan.

5. *Maintance*

Apabila aplikasi tersebut sudah layak maka aplikasi tersebut siap untuk digunakan

untuk tahap berikutnya yaitu pengujian Framework.

3.2.5 Pengujian Framework

1. Pengujian Performa

Pengujian performa dilakukan dengan membandingkan Framework Laravel dengan Framework Codeigniter dari beberapa faktor antara lain:

- Pengujian Request Per Second (RPS)
- Pengujian Time Per Request
- Pengujian Execute Time (Waktu Eksekusi)

2. Pengujian Ukuran

Analisis ukuran dilakukan dengan membandingkan dan menganalisis struktur direktori dan besarnya total file yang terdapat pada direktori.

3. Pengujian Cara Akses Database

Cara akses database dilakukan dengan membandingkan dan menganalisis akses database untuk tata cara pengaksesan tabel dalam operasi Create, Read, Update, Delete (CRUD). Cara pengaksesan database di Laravel menggunakan Eloquent ORM dan cara pengaksesan database di CodeIgniter menggunakan Query Builder.

4. ANALISIS DAN PERANCANGAN

4.1 Analisis

4.1.1 Analisis Masalah

1. Bagaimana mendapatkan hasil perbandingan Framework Laravel dan Framework Codeigniter dari segi kecepatan/*performance*?
2. Bagaimana mendapatkan hasil perbandingan Framework Laravel dan Framework Codeigniter

dari segi ukuran/*resource*?

3. Bagaimana mendapatkan hasil perbandingan Framework Laravel dan Framework Codeigniter dari segi cara akses database?
4. Bagaimana pembuatan aplikasi diagnosa dengan menggunakan Framework Laravel dan Framework Codeigniter?

4.1.2 Analisis Kebutuhan

Ada beberapa *software* yang digunakan penulis untuk membuat skripsi diantaranya :

- XAMPP
- Sublime Text
- Command Prompt / Git
- Web Browser
- Apache Benchmark (ab)
- Framework Laravel
- Framework Codeigniter

4.2 Perancangan Aplikasi

4.2.1 Unified Modeling Language

1. Use Case Diagram

Use Case Diagram adalah gambaran interaksi antara pengguna (*user/actor*) dengan sistem informasi yang akan dibuat. *Use case* menggambarkan siapa saja actor yang terlibat dan fungsi apa saja yang dapat digunakan *actor* pada sistem informasi tersebut.

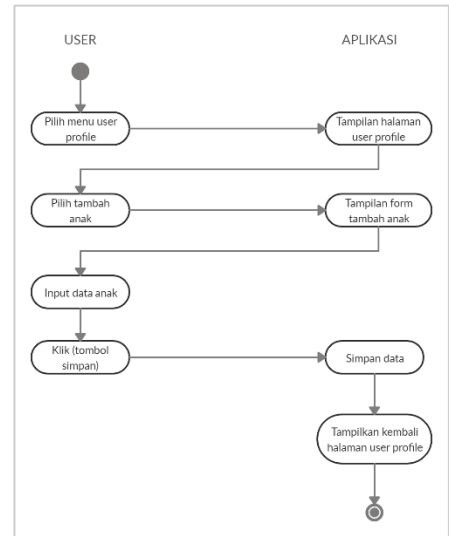


Gambar 4. 1 Use case user

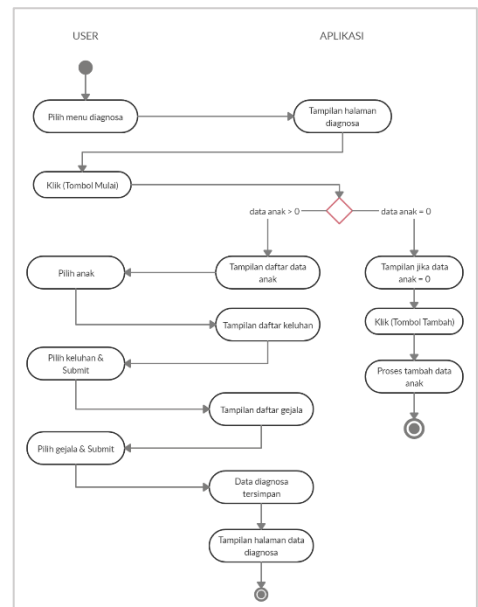
2. Activity Diagram

Activity diagram atau *diagram aktivitas* menggambarkan urutan kegiatan atau urutan aktivitas dari sebuah sistem. Tujuan dibuatnya *activity diagram* adalah untuk memudahkan dalam memahami proses bisnis sistem. Berikut merupakan *activity diagram* untuk aplikasi

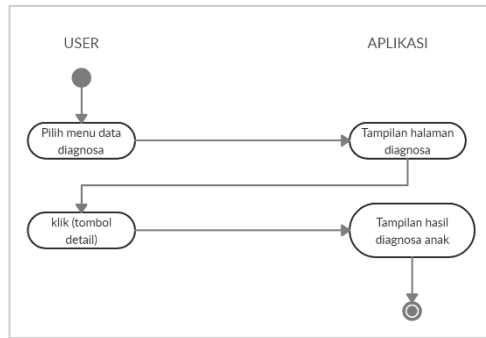
diagnosa untuk anak dibawah 5 tahun.



Gambar 4. 2 Activity diagram tambah data anak



Gambar 4. 3 Activity diagram proses diagnosa



Gambar 4. 4 Activity diagram lihat hasil diagnosa

4.3 Perancangan Pengujian

4.3.1. Perancangan Analisis Performa

1. Pengujian Request Per Second

Pengujian *request per second* ini dilakukan dengan menggunakan Apache Benchmark (ab). Parameter yang dihasilkan dari pengujian yang dilakukan adalah *request per second* (r/s) yang merupakan besarnya request/permintaan yang dapat ditangani dalam waktu 1 detik.

2. Pengujian Time Per Request

Pengujian *time per second* ini dilakukan dengan menggunakan Apache Benchmark (ab). Parameter yang dihasilkan dari pengujian yang dilakukan adalah *time per request* (ms) merupakan waktu yang

dibutuhkan untuk menangani 1 request/permintaan.

3. Pengujian Execute Time

Pengujian *execute time* ini dilakukan dengan menggunakan fungsi `microtime()` pada php. Dengan fungsi ini dapat diketahui berapa waktu yang dibutuhkan untuk menjalankan suatu skrip yang ingin diuji kecepatannya. Dengan begitu dapat dilakukan pengujian pada aplikasi diagnosa anak untuk mengetahui *execute time* (waktu eksekusi) per-fitur pada anak. Fitur tersebut antara lain:

- Registrasi akun
- Login
- Melihat profil dan tambah data anak
- Diagnosa penyakit anak
- Lihat data hasil diagnosa

4.3.1 Perancangan Analisis Ukuran

Analisis ukuran dilakukan dengan membandingkan dan menganalisis struktur direktori dan besarnya total file yang terdapat pada direktori. Kebanyakan framework PHP yang menganut pola MVC (Model-View-Controller) menggunakan skema direktori dengan nama "Model", "View" dan

"Controller" yang seluruhnya dikumpulkan kedalam sebuah direktori utama yang bernama "src" atau "app" / "application" seperti Codeigniter. Sedikit berbeda dengan Codeigniter, direktori dengan nama "View" pada Laravel justru diletakkan di luar direktori "app". Direktori "View" digunakan untuk menyimpan file-file yang berhubungan dengan tampilan aplikasi. Lalu pada direktori "app" baik pada Framework Laravel atau Framework Codeigniter terdapat direktori "Model" yang digunakan untuk menyimpan class PHP yang berhubungan dengan model database. Kemudian direktori "Controller" digunakan untuk menyimpan class PHP yang berhubungan dengan *application logic*.

4.3.3 Perancangan Analisis Cara Akses Database

Cara akses database dilakukan dengan membandingkan dan menganalisis akses database untuk tata cara pengaksesan tabel dalam operasi Create, Read, Update, Delete (CRUD). Cara pengaksesan database di Laravel menggunakan Eloquent ORM dan cara pengaksesan database

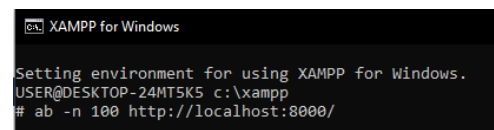
di CodeIgniter menggunakan Query Builder.

5. IMPLEMENTASI DAN PEN-GUJIAN

5.1 Pengujian Segi Performa

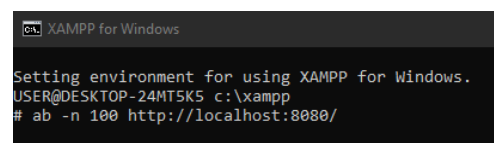
1. Request Per Second

Pengujian dengan menggunakan Apache Benchmark (*ab tool*). Pengujian dilakukan kepada server Laravel (Artisan) dan Codeigniter (Spark) dengan memberikan *request* sebanyak 100 dan pengujian dilakukan sebanyak 10 kali. Dan perintah yang dimasukkan pada *ab tool* adalah sebagai berikut:



```
XAMPP for Windows
Setting environment for using XAMPP for Windows.
USER@DESKTOP-24MT5K5 c:\xampp
# ab -n 100 http://localhost:8080/
```

Gambar 5. 1 Perintah untuk server Laravel



```
XAMPP for Windows
Setting environment for using XAMPP for Windows.
USER@DESKTOP-24MT5K5 c:\xampp
# ab -n 100 http://localhost:8080/
```

Gambar 5. 2 Perintah untuk server Codeigniter

Dan hasil yang akan ditampilkan pada ab tool adalah:

```

Setting environment for using XAMPP for Windows.
USER@DESKTOP-24MTSK5 C:\xampp
# ab -n 100 http://localhost:8080/
This is ApacheBench, Version 2.3 <Revision: 1843412>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient).....done

Server Software:
Server Hostname:      localhost
Server Port:          8080
Document Path:        /
Document Length:      3826 bytes
Concurrency Level:     1
Time taken for tests:   13.313 seconds
Complete requests:     100
Failed requests:        0
Total transferred:     422000 bytes
HTML transferred:      382600 bytes
Requests per second:    7.51 [#/sec] (mean)
Time per request:       133.134 [ms] (mean)
Time per request:       133.134 [ms] (mean, across all concurrent requests)
Transfer rate:          30.95 [Kbytes/sec] received
  
```

Gambar 5. 3 Tampilan hasil tool ab

Didapatlah waktu pengujian dan jumlah *request per second*. Pengujian pun dilakukan sebanyak 10 kali lalu dijumlahkan dan diambil nilai rata-ratanya. Dan setelah pengujian sebanyak 10 kali dilakukan didapatlah data sebagai berikut:

- Laravel

Nomor Tes	Waktu Tes (detik)	Jumlah Request	Request Per Second
1	53,203 s	100	1,88 r/s
2	44,410 s	100	2,25 r/s
3	59,491 s	100	1,68 r/s
4	47,741 s	100	2,09 r/s
5	43,743 s	100	2,29 r/s
6	50,589 s	100	1,98 r/s
7	46,763 s	100	2,14 r/s

8	46,994 s	100	2,13 r/s
9	44,282 s	100	2,26 r/s
10	51,326 s	100	1,95 r/s
Rata-rata	48,854 s	100	2,06 r/s

Tabel 5. 1 Request per second Laravel

- Codeigniter

Nomor Tes	Waktu Tes (detik)	Jumlah Request	Request Per Second
1	13,629 s	100	7,34 r/s
2	15,284 s	100	6,54 r/s
3	14,790 s	100	6,76 r/s
4	13,370 s	100	7,48 r/s
5	21,656 s	100	4,62 r/s
6	13,360 s	100	7,47 r/s
7	12,988 s	100	7,70 r/s
8	17,124 s	100	5,84 r/s
9	13,881 s	100	7,20 r/s
10	14,541 s	100	6,88 r/s
Rata-rata	15,062 s	100	6,78 r/s

Tabel 5. 2 Request per second Codeigniter

6. KESIMPULAN DAN SARAN

6.1 Kesimpulan

Dari pengujian dan pembahasan yang telah dilakukan, penulis dapat menarik kesimpulan analisis yang dibuat sebagai berikut:

1. Performa pada Framework Laravel dan Framework Codeigniter pada analisis *request per second*, *time per request*, dan *execute time* memiliki hasil yang berbeda. Pada analisis *response time*, Framework Codeigniter memiliki nilai yang lebih unggul 4,72 r/s dibanding Framework Laravel. Pada analisis *time per request* Framework Codeigniter memiliki nilai yang lebih unggul 2315 ms dibanding Framework Laravel. Pada analisis *execute time*, baik Framework Laravel dan Framework Codeigniter memiliki nilai yang relative sama.
2. Ukuran pada Framework Laravel lebih besar dibanding Framework Codeigniter. Besarnya file Framework dikarenakan banyak vendor yang menyediakan package-package yang akan sangat membantu ketika aplikasi dibuat

adalah aplikasi yang besar dan lumayan susah dalam pengelolaannya.

3. Cara akses database pada Framework CodeIgniter menggunakan Query Builder, sedangkan pada Framework Laravel menggunakan Eloquent ORM. Dalam penerapannya, semuanya memiliki satu tujuan yaitu agar dapat terkoneksi dengan database dan melakukan CRUD. Keduanya mampu merepresentasikan sebuah tabel pada sebuah model dan dapat membuat operasi CRUD jadi lebih mudah.

6.2 Saran

Dari pengujian yang telah dilakukan, penulis dapat mengumpulkan saran sebagai berikut:

1. Jika ingin membuat program dengan performa yang lebih cepat, dinilai dari *request per second*, *time per request*, dan *execute time*, Framework Codeigniter dapat dijadikan pilihan. Namun, jika ingin membuat program tanpa memikirkan tentang kecepatan, Framework Laravel dapat dijadikan sebuah pilihan.

2. Jika ingin membuat program dengan ukuran lebih kecil, Framework Codeigniter bisa dijadikan sebuah pilihan. Namun, jika ingin membuat program dengan *package* cukup lengkap, Framework Laravel dapat dijadikan sebuah pilihan.
3. Jika ingin membuat program dengan cara akses database yang mempermudah operasi CRUD, baik Framework CodeIgniter maupun Framework Laravel dapat dijadikan pilihan.

DAFTAR PUSTAKA

- Hamid, Muhammad Nur. 2019. Analisis Perbandingan Framework Codeigniter dan Framework Laravel (Studi Kasus Inventaris HMJ TI STMIK AKAKOM YOGYAKARTA). Yogyakarta: STMIK AKAKOM.
- Erinton, Ruli., *Ridha Muldina Negara dan Danu Dwi Sanjoyo*. 2017. Analisis Performasi Framework Codeigniter dan Laravel Menggunakan Web Server Apache. Bandung: Telkom University.
- Yanto, Bagus Fery., Indah Werdingsih dan Endah Purwanti. 2017. Aplikasi Sistem Pakar Diagnosa Penyakit Pada Anak Bawah Lima Tahun Menggunakan Metode Forward Chaining. Surabaya: Universitas Airlangga.
- Hidayatullah, Priyanto dan Jauhari Khairul Kawistara. 2017. Pemrograman WEB. Bandung: Informatika.
- Sidik, Betha. 2018. FRAMEWORK CODEIGNITER 3 Membangun Pemrograman Berbasis Web Dengan Berbagai Kemudahan & Fasilitas Codeigniter 3. Bandung: Informatika.
- Maharani, Meilan Anastasia. 2018. Analisa dan Perancangan Sistem Informasi dengan Codeigniter dan Laravel. Yogyakarta: Lokomedia.
- Rika Rosnelly. 2012. Sistem Pakar: Konsep dan Teori. Yogyakarta: Andi.
- Rohi Abdulloh. 2018. 7 in 1 Pemrograman WEB Untuk Pemula. Jakarta: PT Elex Media Komputindo
- Yurindra. 2017. Software Engineering. Yogyakarta: CV BUDI UTAMA
- Request per second. (2020, Juli 21), Retrieved from <https://support.loadimpact.com/3.0/test-configuration/what-are-requests-per-second-rps/>
- Apache Benchmark (ab). (2020, Juli 21), Retrieved from <https://httpd.apache.org/docs/2.4/programs/ab.html>

Ruli Erinton, Ridha Muldina Negara,
Danu Dwi Sanjoyo. 2017.
ANALISIS PERFORMASI
FRAMEWORK CODEIG-
NITER DAN LARAVEL
MENGUNAKAN WEB
SERVER APACHE. Ban-
dung: Telkom University.