

# Analisis Perbandingan Bahasa Pemrograman PHP Laravel dengan PHP Native pada Pengembangan Website

Robby Yuli Endra <sup>1\*</sup>, Yuthsi Aprilinda <sup>1</sup>, Yanuarius Yanu Dharmawan <sup>2</sup>, Wahyu Ramadhan <sup>1</sup>

<sup>1</sup> Program Studi Informatika, <sup>2</sup> Program Studi Bahasa Inggris

Universitas Bandar Lampung

Lampung, Indonesia

[robby.yuliendra@ubl.ac.id](mailto:robby.yuliendra@ubl.ac.id), [yuthsi.aprilinda@ubl.ac.id](mailto:yuthsi.aprilinda@ubl.ac.id), [yanu@ubl.ac.id](mailto:yanu@ubl.ac.id), [wahyu.15421037@student.ubl.ac.id](mailto:wahyu.15421037@student.ubl.ac.id)

**ABSTRACT** – To design a website, many website developers use the PHP programming language. Currently the concept that is widely used in the development of an application is to use an object-oriented method approach or OOP. However, the design of the website is still long and requires a large amount of money if it is done from the beginning. For that the right solution to design a website effectively and efficiently is to use a PHP framework. The purpose of this study is to analyze the comparison of a web that implements the Laravel framework with a web that implements native PHP based on lines of code, program code efficiency, speed, folder and url structure, architecture, and web security. Each website is built using the MySQL database system, PHP7 as the native language, Laravel as the PHP framework and Apache as the web server.

**Keywords:** Laravel, PHP, Native, Comparative Analysis, Framework

**ABSTRAK** – Untuk merancang sebuah website banyak developer website menggunakan bahasa pemrograman PHP. Saat ini konsep yang banyak digunakan dalam pengembangan sebuah aplikasi yaitu menggunakan pendekatan metode berorientasi objek atau OOP. Namun perancangan website tersebut masih termasuk lama dan memakan biaya yang cukup besar apabila dilakukan dari awal. Untuk itu solusi yang tepat untuk merancang sebuah website efektif dan efisien adalah menggunakan sebuah framework PHP. Tujuan penelitian ini bermaksud untuk menganalisis perbandingan web yang mengimplementasikan framework Laravel dengan web yang mengimplementasikan PHP native berdasarkan baris kode, efisiensi kode program, kecepatan, struktur folder dan URL, arsitektur, dan keamanan web. Masing-masing web yang dibangun menggunakan sistem basis data MySQL, PHP7 sebagai native language, Laravel sebagai framework PHP dan Apache sebagai web server.

**Kata Kunci:** Laravel, PHP, Native, Analisis Perbandingan, Framework

## 1. PENDAHULUAN

Website adalah kumpulan halaman web yang dijalankan menggunakan browser dan internet. Website berada dalam domain atau subdomain yang sering disebut dengan WWW atau World Wide Web. Sebuah website dibuat dengan bahasa pemrograman HTML (*Hyper Text Markup Language*) yang diakses melalui protokol di internet [1]. Selain menggunakan bahasa pemrograman HTML, website dapat dikembangkan dengan bahasa pemrograman dinamis, salah satunya adalah bahasa pemrograman PHP (*Hypertext Preprocessor*) yang merupakan bahasa pemrograman *open-source server side* [2]. *Server Side* adalah *script* yang dimasukkan untuk diproses oleh dan diproses di server dan PHP memiliki keunggulan bersifat *open-source*, yaitu pengguna bebas memodifikasi dan mengembangkan aplikasi atau sistem sesuai keinginan.

Sebelum ada kerangka kerja atau *framework* dalam bahasa pemrograman, masih banyak yang menggunakan PHP Native. PHP Native adalah bahasa pemrograman atau instruksi yang dibuat tanpa campur tangan pengembang lain untuk prosesnya, sehingga instruksi atau kode program tersebut hanya diketahui oleh pembuat PHP Native [3]. Inilah kelemahan atau kekurangan PHP

Native jika tidak ada dokumentasi dari developer maka akan sulit untuk mempelajari PHP Native. Selain itu, kekurangan lainnya adalah mendesain website menggunakan PHP Native membutuhkan waktu yang lama untuk pengerjaan proyek baik skala menengah maupun proyek skala besar. Hal ini dikarenakan ada beberapa fungsi kode yang ada di PHP Native yang harus diulang ke fungsi kode lainnya sehingga menyebabkan *error* atau *bug* yang lebih besar.

Hal inilah yang menyebabkan pengembangan website membutuhkan *framework* atau kerangka kerja. *Framework* adalah kumpulan *library* yang memiliki kemampuan dan fungsi yang berbeda satu sama lain yang dikembangkan oleh developer yang tergabung dalam komunitas *developer framework*. Pada *framework* terdapat fungsi yang dapat mengatasi permasalahan yang ada pada PHP Native yaitu mengatasi aktivitas atau pekerjaan yang berulang, sehingga pekerjaan yang dilakukan lebih mudah dan cepat.

Banyak *framework* bahasa pemrograman terkenal seperti CodeIgniter (CI) telah diteliti [4] untuk sistem Aeroponik dan Laravel. Dalam analisis penelitian ini menggunakan analisis *framework* Laravel. Laravel merupakan *framework* yang dikembangkan oleh Taylor Otwell pada bulan Juni 2011 yang memiliki banyak

pengguna hingga saat ini [5]. Pada *framework* Laravel terdapat fungsi-fungsi kode yang disediakan di *library* kemudian di *install* ke dalam Laravel. Keuntungan umum menggunakan Laravel adalah penyebaran komunitas yang besar berdampak pada penemuan banyak *library* yang berbeda, beberapa peneliti menemukan bahwa *library* di Laravel dapat cukup besar sehingga mereka dapat menyelesaikan proyek pengembangan web dari skala rendah hingga menengah.

Sebagaimana dijelaskan sebelumnya mengenai kelebihan dan kekurangan masing-masing bahasa pemrograman PHP, baik *native* maupun *framework*, penelitian ini akan membuat perbandingan antara PHP Native dan PHP Framework Laravel. Tujuan dari penelitian ini adalah untuk menguji dan mengkaji kedua bahasa pemrograman PHP tersebut untuk mendapatkan kelebihan dan kekurangan masing-masing dalam pengembangan website. Salah satu manfaat yang diperoleh dari penelitian ini adalah memberikan masukan kepada pengembang website pemula untuk menentukan bahasa pemrograman PHP yang akan digunakan sesuai dengan kelebihan dan kekurangan dari hasil penelitian, memberikan rekomendasi penggunaan bahasa pemrograman bagi pengembang pemula untuk bekerja pada proyek tingkat rendah, menengah dan besar dengan kerangka kerja yang direkomendasikan dalam penelitian ini.

## **2. DASAR TEORI**

Pada penelitian terdahulu yang membahas tentang perbandingan performa web service dengan menggunakan *framework* Laravel, Django dan Ruby on Rails. Permasalahan yang timbul pada penelitian ini adalah semakin banyaknya pengguna aplikasi mobile sehingga dibutuhkan web service yang dinamis. Web service yang dinamis dapat memiliki keuntungan yaitu waktu respon yang sangat singkat. Hasil dari penelitian ini adalah *framework* Django memiliki respon waktu yang cepat, penggunaan memori dan CPU yang sedikit [6]. Pada penelitian yang lain membahas tentang analisis *performance* antara *framework* Code Igniter dan *framework* Laravel. Hasil dari penelitian ini menyebutkan dari sisi *performance framework* Code Igniter lebih baik dibandingkan dengan *framework* Laravel. Untuk waktu CodeIgniter lebih rendah nilainya sebesar 150,5 ms dibandingkan Laravel 254.5 ms dan nilai *error* tertinggi yang didapat oleh Laravel sebesar 79.7 [7]. Pada penelitian yang lain yang berjudul Analisa *Model View Controller (MVC)* pada Bahasa PHP penelitian ini melakukan perancangan menggunakan bahasa pemrograman PHP dengan konsep *Model View Controller (MVC)*. Penelitian ini melakukan pengujian metode MVC pada bahasa pemrograman dan melakukan pengujian menggunakan *Developer Tools* pada Chrome yang berjalan pada server Apache dengan menggunakan Wampp. Berdasarkan hasil penelitian tersebut, metode MVC yang digunakan pada bahasa pemrograman PHP berjalan dengan baik tanpa ada kendala *down* ataupun *error* pada Apache Server [8]. Pada penelitian yang pernah kami lakukan yang berjudul *E-*

*Report* Berbasis Web Menggunakan Metode *Model View Controller* untuk Mengetahui Peningkatan Perkembangan Prestasi Anak Didik. Pada penelitian ini peneliti merancang website *e-report* dengan menggunakan *Framework* Codeigniter yang memiliki konsep MVC yaitu Model, *View* dan *Controller* dengan mengembangkan website dengan konsep tersebut untuk pembuatan website tidak membutuhkan waktu yang lama [9].

## **3. METODOLOGI**

### **A. Metode Penelitian**

Untuk melakukan analisis pada penelitian ini dan untuk membandingkan bahasa pemrograman PHP Native dan PHP Framework Laravel maka dilakukan metode sebagai berikut: 1) Web yang diimplementasikan akan memiliki spesifikasi yang sama. 2) Parameter yang dibandingkan pada analisis adalah:

- Jumlah Baris Kode; Pada parameter ini penghitungan baris kode berdasarkan fungsi, jadi satu buah fungsi dianggap sebagai satu baris kode.
- Efisiensi Kode Program; Parameter ini akan menguji bagaimana sebuah kode dibuat, dan akan dibandingkan yang mana lebih mudah untuk dimengerti dan dibuat untuk membuat fungsi kode program antara Laravel dengan PHP Native.
- Kecepatan; Penelitian ini akan menguji kecepatan waktu untuk meload dan waktu pengekseskuan proses pada masing-masing web, menggunakan browser Chrome sebagai alat penguji waktu kecepatannya.
- Struktur Folder dan URL; Penelitian ini akan menganalisis struktur dari masing-masing web. Di dalam parameter ini penelitian ini akan menilai bagaimana struktur *folder* dan URL didalamnya.
- Arsitektur; *Framework* Laravel menggunakan konsep/arsitektur MVC dan OOP yang akan dibandingkan dengan PHP native menggunakan arsitektur klasik.
- Keamanan; Pada parameter ini, akan diujikan tingkat keamanan web terhadap pemrosesan data dan pengelolaan data. Pada masing-masing web menggunakan basis data MySQL. Pengujiannya berupa *SQL Injection* sebagai teknik serangan ke masing-masing web.

### **B. Analisis Kebutuhan Sistem**

Spesifikasi *hardware* yang akan digunakan dalam penelitian ini: 1) Minimal Prosesor i3; 2) Memory/RAM 6GB; 3) *Wireless Connection*; 4) Laptop. Sementara *software* yang akan digunakan dalam penelitian ini adalah 1) Sistem Operasi Ubuntu 17.10; 2) *Sublime Text* 3; 4) Lampp; 5) Chrome (*Developer Tools*).

### **C. Diagram Blok**

Diagram Blok merupakan rancangan analisis untuk membandingkan bahasa pemrograman PHP Native dan *Framework* Laravel yang dapat dilihat pada Gambar 1. 1) Tahap pertama yang dilakukan pada penelitian ini adalah

membuat perancang web. Tahap awal ini bertujuan untuk memberikan gambaran bagaimana konsep website yang akan dibuat. 2) Tahap kedua merupakan proses analisis dan membandingkan *Line of Code Program*. Tahap ini peneliti menggunakan aplikasi *Text Editor Sublime* karena didalam fitur Sublime terdapat *line code* yang dapat membantu untuk analisis. 3) Tahap ketiga yaitu menganalisis efisiensi *code* dengan membandingkan antara *code* dengan bahasa pemrograman PHP Native dengan yang menggunakan *framework* Laravel. 4) Tahap keempat penelitian ini dilakukan analisis kecepatan program dengan menggunakan browser Google Chrome dan

*developer tools* yang di *test* secara lokal di laptop. 5) Tahap kelima adalah analisis struktur Folder dan URL dengan membandingkan folder-folder yang terbentuk pada masing-masing bahasa pemrograman. 6) Tahap keenam adalah analisis arsitektur *code*. Pengujian yang dilakukan dengan cara metode yang digunakan salah satunya dengan menggunakan analisis konsep OOP dan MVP pada masing-masing bahasa pemrograman. 7) Tahap ketujuh adalah analisis keamanan web, tahap ini akan diuji dengan beberapa teknik penyerang yaitu salah satunya dengan menggunakan teknik *SQL injection*.



Gambar 1. Diagram Blok Analisis Perbandingan

#### 4. HASIL DAN PEMBAHASAN

##### A. Pengujian dan Hasil

Pada tahap ini, dilakukan pengujian dan analisis terhadap web yang telah dibuat sekaligus melakukan perbandingan dari kedua web yang telah dibuat. Pengujian menggunakan parameter pada Diagram Blok Gambar 1.

##### 1) Uji *Line Code*

Setelah melakukan perancangan sebuah *prototype* dari masing masing web yang dibuat untuk dianalisis. Penelitian ini menggunakan *Sublime Text 3* sebagai *text-editor* untuk merancang masing-masing web. Pada parameter ini akan menghasilkan jumlah baris kode, panjang kode, dan kerumitan kode didalam satu fungsi. Fungsi bandingkan memiliki tujuan/fungsi yang sama. Hasil pengujian beberapa fungsi dapat dilihat pada Gambar 2 dan Gambar 3.

```

if (Input::get('submit')) {
    # code...

    // 1. memanggil objek validasi
    $validation = new Validation();
    // 2. metode check
    $validation = $validation->check(array(
        'npm' => array(
            'required' => true,
            'min'      => 3,
            'max'      => 50,
        ),
        'nama' => array(
            'required' => true,
            'min'      => 3,
        ),
        'status' => array(
            'required' => true,
            'min'      => 3,
        )
    ));
    // 3. lolos ujian
    if ($validation->passed()) {
        $mahasiswa->tambah_mhs(array(
            'npm' => Input::get('npm'),
            'nama' => Input::get('nama'),
            'status' => Input::get('status'),
        ));

        $result = '';
    } else {
        $errors = $validation->errors();
    }
}
?>
<!DOCTYPE html>
  
```

Gambar 2. Kode Fungsi Simpan Mahasiswa pada PHP Native

```
public function simpan_mhs(Request $request)
{
    $user = $request->user();
    $posts = Mahasiswa::create(['npm'=>$request->npm,
                                'nama'=>$request->nama,
                                'status'=>$request->status]);
    Session::flash('notif', [
        'level'=>"success",
        'message'=>"Data Mahasiswa Berhasil Tersimpan"
    ]);
    return redirect()->route('home');
}
```

Gambar 3. Kode Fungsi Simpan Mahasiswa pada PHP Laravel

### 2) Uji Efisiensi Kode

Pada parameter berikutnya yaitu efisiensi pembuatan sebuah fungsi ataupun kode program. Seperti yang diketahui *framework* merupakan sebuah kerangka kerja yang berfungsi untuk menyediakan berbagai *library* untuk

kepentingan pengembangan sebuah aplikasi. Penelitian ini telah mendapatkan beberapa sampel yang digunakan sebagai bahan perbandingan efisiensi pada pembuatan kode program sebagai pada Gambar 4 dan Gambar 5.

```
<?php
//including the database connection file
require_once 'core/init.php';
if(!Session::exist('username')){
    header('Location: login.php');
}
```

Gambar 4. Kode Cek Login Session pada PHP Native

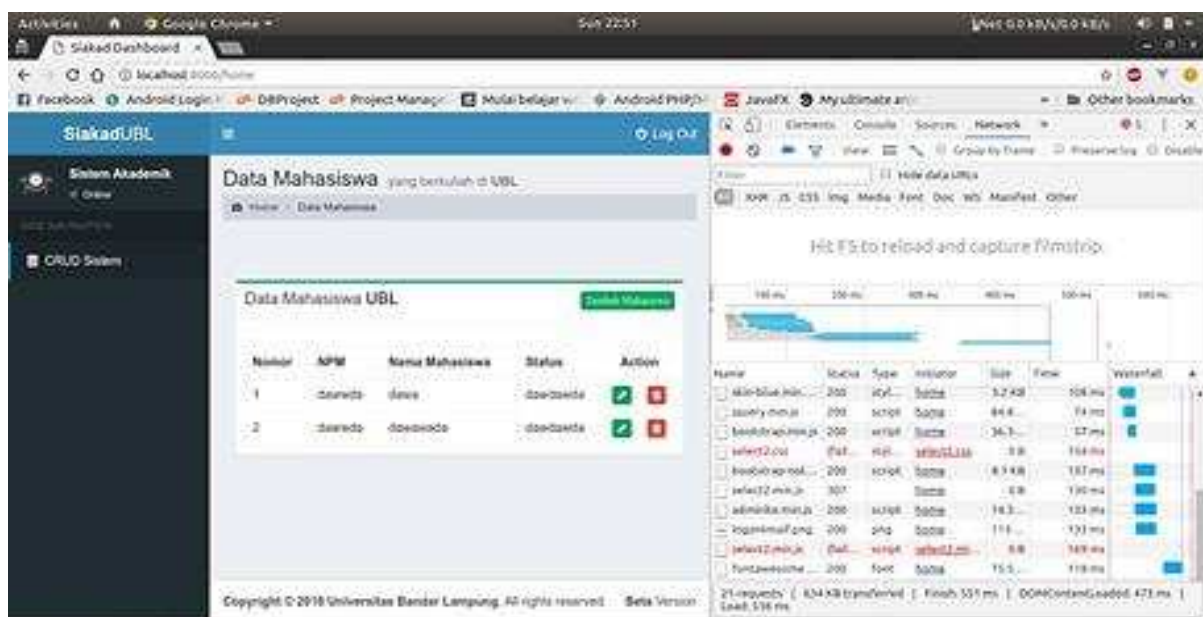
```
Auth::routes();
Route::group(['middleware' => ['auth']], function () {
```

Gambar 5. Middleware Auth untuk Membatasi User yang Tidak Login

Berdasarkan Gambar 4 pada PHP Native kode tersebut harus digunakan berulang-ulang pada sebuah halaman yang memiliki akses login saja. Sedangkan pada Laravel cukup menggunakan sebuah *Middleware Auth* yang ditunjukkan pada Gambar 5 yang dideklarasikan pada *route* Laravel. Jadi pada pengerjaan yang menggunakan Laravel untuk membuat suatu perintah atau fungsi, *developer* tidak perlu untuk membuat dari awal cukup dengan memanggil *class* yang akan digunakan sesuai kebutuhan aplikasi.

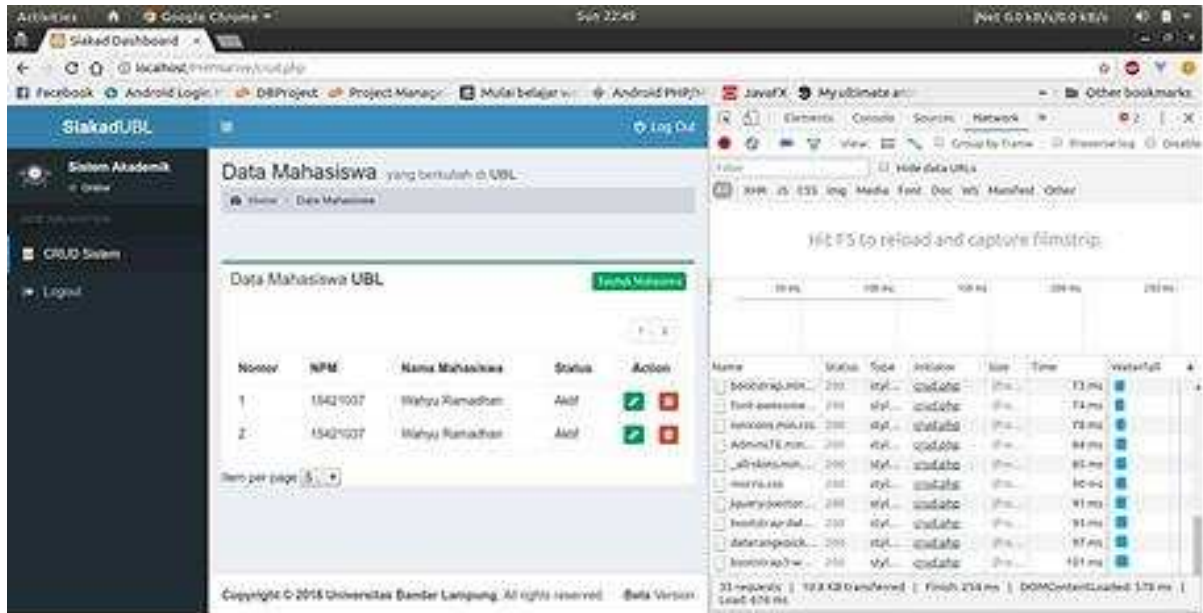
### 3) Uji Kecepatan Web

Pengujian kecepatan web ini dilakukan menggunakan browser Chrome dengan bantuan *Developer Tools*-nya yang pada saat ini masih dieksekusi secara *local* menggunakan Laptop pada penelitian ini. Pengujian yang dilakukan dengan membuka sebuah halaman yang sama dan mengambil data selama lima kali, halaman yang sama yang dimaksud adalah tampilannya dan isi konten halaman seperti pada Gambar 6 dan Gambar 7.



Gambar 6. Uji Kecepatan pada PHP Laravel





Gambar 7. Uji Kecepatan pada Native

Berdasarkan pengujian, didapatkan hasil pengujian kecepatan seperti pada Tabel 1. Dari pengujian ini Laravel lebih unggul pada kecepatan proses *Load* dan *DomContentLoaded* sedangkan untuk memproses halaman hingga finish, PHP Native lebih unggul. Ini dikarenakan page size yang dimiliki oleh Laravel jauh lebih besar

dibandingkan dengan PHP Native dalam page size terdapat konten-konten yang membuatnya menjadi besar seperti CSS, javascript, gambar dan sebagainya, untuk itu pada penggunaan asset javascript dan CSS sebaiknya menggunakan versi minify untuk mengurangi ukurannya.

Tabel 1. Hasil Pengujian Kecepatan

Tahap	Request		Page Size		Finish		DomContentLoaded		Load	
	PHP Native	Laravel	PHP Native	Laravel	PHP Native	Laravel	PHP Native	Laravel	PHP Native	Laravel
1	33	26	10.8 KB	634 KB	254 ms	551 ms	578 ms	473 ms	676 ms	536 ms
2	33	26	10.8 KB	634 KB	368 ms	503 ms	588 ms	403 ms	731 ms	477 ms
3	33	26	10.8 KB	634 KB	241 ms	483 ms	557 ms	409 ms	667 ms	460 ms
4	33	26	10.8 KB	634 KB	292 ms	472 ms	488 ms	393 ms	649 ms	449 ms
5	33	26	10.8 KB	634 KB	302 ms	470 ms	513 ms	363 ms	695 ms	448 ms
Hasil	33	26	10.8 KB	634 KB	293,4 ms	495,8 ms	544,8 ms	408,2 ms	683,6 ms	474 ms

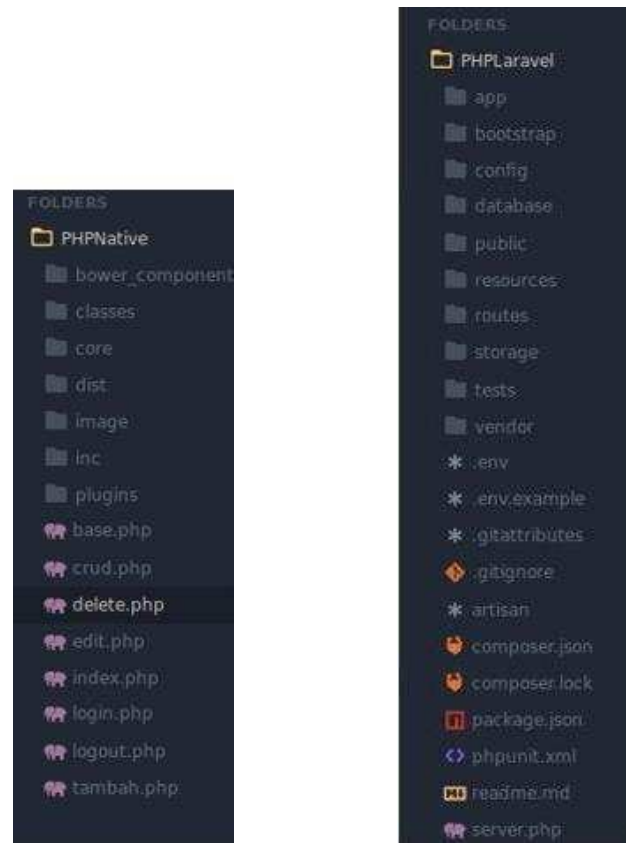
#### 4) Struktur Folder dan URL

Pada parameter ini, penelitian ini menganalisis lalu membandingkan struktur kedua *folder* maupun URL yang dimiliki masing-masing web seperti pada Gambar 8 dan Gambar 9. Didapatkan hasil yaitu, Laravel memiliki *folder* yang terstruktur rapi dan sesuai dengan kebutuhan tempat, namun untuk pengguna/developer awam mungkin bingung terhadap *folder-folder* tersebut dan juga *root folder* pada Laravel tidak boleh diubah nama atau dipindahkan. Apabila salah satu diganti maka terjadi kegagalan untuk *load* halaman yang diinginkan, yang artinya *folder* tersebut bersifat tetap sehingga harus dimengerti terlebih dahulu. Kelebihan lain dari *folder* tetap ini adalah pengembang lain

atau baru dapat bekerja secara tim tanpa harus mengubah-ubah *folder* tersebut karena sudah mengerti isinya. Berbeda dengan PHP Native, pengembang dapat membuat *folder* apa saja sesuai dengan kehendak pengembang. Jadi rapi atau tidaknya *folder* tersebut tidak dapat ditentukan karena berbeda pengembang, maka struktur *folder* akan berubah. Jadi untuk bekerja, tim menggunakan PHP Native mungkin sedikit susah. Ini dikarenakan folder yang dibuat hanya dimengerti oleh pengembang pertama saja. Ukuran proyek Laravel jauh lebih besar dibandingkan dengan PHP Native dikarenakan banyaknya *library* dan *class*.



Gambar 8. Struktur URL pada PHP Native (kiri) dan Struktur URL pada Laravel (kanan)



Gambar 9. Struktur Folder pada PHP Native (kiri) dan Struktur Folder pada Laravel (kanan)

Untuk struktur URL, Laravel menggunakan *routing*, sama halnya seperti *framework* lain sehingga untuk menampilkan halaman perlu sebuah *method* yang dipanggil pada sebuah *controller* dan juga meningkatkan keamanan *folder-folder* didalamnya. Sedangkan pada PHP Native masih menggunakan cara klasik yaitu dengan mengakses *file* yang berekstensi *\*.php* untuk menampilkan sebuah halaman. Cara ini masih dikatakan belum aman karena letak *file* maupun *folder* dapat ditebak dengan mudah. Ini bisa saja dimanfaatkan sebagai celah dari keamanan web.

#### 5) Uji Arsitektur

Pengujian pada parameter ini berupa arsitektur pada masing-masing web. Laravel menggunakan arsitektur MVC (*Model, View, and Controller*) dan OOP (*Object Oriented Programming*). Sementara itu PHP Native yang telah dibangun menggunakan OOP saja, tidak menggunakan MVC.

MVC dan OOP pada Laravel sangat membantu *developer* dalam membangun sebuah aplikasi web, penelitian ini dirasakan perbedaan yang cukup signifikan pada saat membangun masing-masing web. Ini

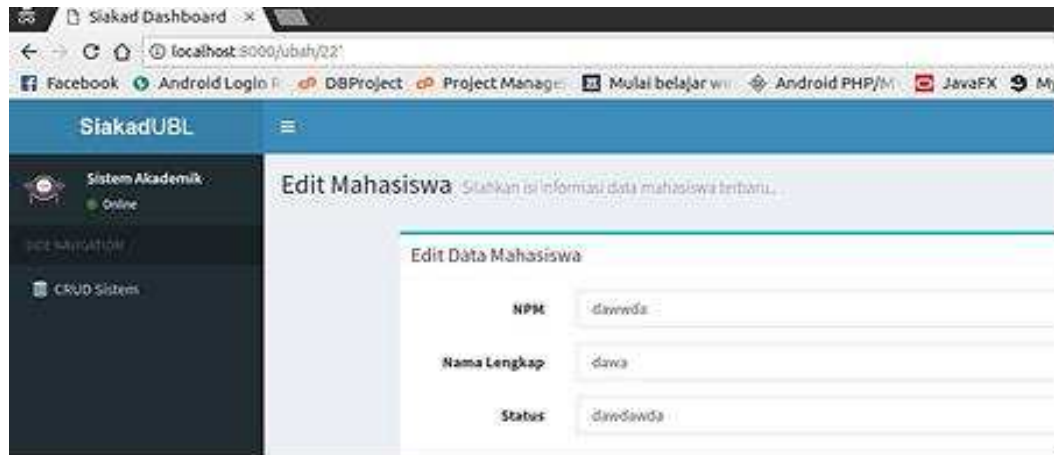
dikarenakan terdapat dokumentasi yang lengkap mengenai arsitektur Laravel dan fungsi-fungsi lainnya pada situs resmi Laravel. Jadi pada proses pengembangan tidak bingung alur dari *view*, *model*, dan *controller* yang memiliki fungsi masing-masing. Sementara itu pada PHP Native, penelitian ini membangun web dengan konsep OOP menggunakan *classes*. *Developer* diharuskan memikirkan sendiri konsep OOP yang berjalan.

#### 6) Uji Keamanan

Pada parameter ini, Laravel memiliki pertahanan dari serangan-serangan dasar yang cukup mumpuni dibandingkan PHP Native. Alasan yang pertama adalah komunitas yang besar yang bersuka rela untuk melakukan *Penetration Test*, jadi pada saat ditemukan *bug* atau celah keamanan maka para *developer* bersama-sama mencari solusi untuk memecahkan masalah tersebut. Laravel menyediakan beberapa fitur yang mencegah serangan dari *SQL Injection*, *xsrif*, *xss*. Berikut contoh *testing SQL Injection* berupa *error-based syntax* MySQL seperti pada Gambar 10 dan Gambar 11.



Gambar 10. *SQL Injection Test* pada PHP Native



Gambar 11. *SQL Injection Test* pada Laravel

Berdasarkan Gambar 10 dan Gambar 11, pada PHP Native terdapat *error SQL* yang bisa dijadikan celah keamanan untuk membobol sistem basis data pada web yang dibangun. Sementara itu pada Laravel tidak terdapat *error* yang dijadikan sebagai celah keamanan. Karena hal tersebut penelitian ini merancang PHP Native menggunakan *escape()*. Fungsi itu pun harus diulang-ulang saat ingin mendeklarasikan sebuah *input* ke dalam *database* atau mengambil data dari *database*. Apabila sebuah *input* tidak menggunakan fungsi *escape()* maka akan menjadi sasaran yang empuk untuk diserang menggunakan *SQL Injection*. Bukan itu saja serangan melalui format ekstensi, Xsrf, dan Xss belum ada pada PHP Native jadi *developer* diharuskan untuk membuat keamanan tersebut.

### 5. KESIMPULAN

Setelah melakukan analisis pada penelitian ini, maka disimpulkan bahwa: Jumlah baris kode berorientasi fungsi, website yang dibangun dengan menggunakan *framework* Laravel memiliki lebih banyak baris pada fungsi dibandingkan dengan website yang dibangun dengan PHP Native. Website yang memakai *framework* Laravel memiliki tingkat ke-efisien-an membuat sebuah fungsi kode program yang lumayan tinggi dibandingkan dengan PHP Native. Dikarenakan Laravel telah menyediakan berbagai *library* untuk mengeksekusi program tersebut. Seperti pada penggunaan ORM (*Object Relation Mapping*) untuk pengekseskuan kode program untuk mengelola basis data sehingga pengguna waktu untuk membuat program berkurang dan mudah untuk di-*maintenance*. PHP Native unggul dalam kecepatan dan *page size*, ini dikarenakan konten dan ukuran projek pada Laravel jauh lebih banyak dan besar dibandingkan dengan *native*. Walaupun struktur *folder* pada PHP Native jauh lebih *flexible* daripada Laravel apabila dikerjakan sendiri, tetapi untuk pengerjaan secara tim atau kelompok Laravel lebih unggul, sebab tempat dan struktur *folder* yang akan digunakan sama dibandingkan dengan Laravel. Tetapi pada struktur URL yang digunakan pada Laravel jauh lebih *flexible* dan mudah untuk diubah pada *routing*. Arsitektur yang digunakan Laravel jelas dan terarah, ini dikarenakan Laravel telah menyediakan dokumentasi

pada website resminya. Sedangkan untuk PHP Native untuk harus membuat sendiri dan tidak ada dokumentasinya. Untuk segi keamanan, Laravel menyediakan ketahanan terhadap serangan-serangan dasar keamanan web mulai dari *SQL Injection*, CSRF, dan XSS tanpa perlu dibuat dari awal seperti pada PHP Native.

### 6. DAFTAR PUSTAKA

- [1] Y. Trimarsiah And M. Arafat, "Analisis dan Perancangan Website sebagai Sarana Informasi pada Lembaga Bahasa Kewirausahaan dan Komputer Akmi Baturaja," *J. Ilm. Matrik*, Vol. 19, No. 1, Pp. 1–10, 2017.
- [2] K. Wibowo, "Analisa Konsep Object Oriented Programming pada Bahasa Pemrograman PHP," *J. Khatulistiwa Inform.*, Vol. 3, No. 2, Pp. 151–159, 2015.
- [3] B. Pasaribu and W. Susanti, "Sistem Informasi Pengajuan Rancangan Usulan Penelitian Menggunakan PHP Native dan Bot Telegram," *J. Mhs. Apl. Teknol. Komput. Dan Inf.*, Vol. 3, No. 1, Pp. 29–38, 2021.
- [4] R. Y. Endra, A. Cucus, and M. A. Wulandana S, "Perancangan Aplikasi Berbasis Web pada System Aeroponik untuk Monitoring Nutrisi Menggunakan Framework CodeIgniter," *Explor. J. Sist. Inf. Dan Telemat.*, Vol. 11, No. 1, P. 10, 2020, Doi: 10.36448/Jsit.V11i1.1453.
- [5] D. Mediana, "Rancang Bangun Aplikasi Helpdesk (A-Desk) Berbasis Web Menggunakan Framework Laravel (Studi Kasus di PDAM Surya Sembada Kota Surabaya)," *J. Manaj. Inform.*, Vol. 8, No. 2, Pp. 75–81, 2018, [Online]. Available: <http://Ejournal.Ukrida.Ac.Id/Ojs/Index.Php/Tik/Article/View/1495/1617>.
- [6] D. Saputra And F. R. Aji, "Analisis Perbandingan Performa Web Service Rest Menggunakan Framework Laravel, Django dan Ruby on Rails untuk Akses Data dengan Aplikasi Mobile," *J. Bangkit Indones.*, Vol. 7, No. 2, P. 17, 2018, Doi: 10.52771/Bangkitindonesia.V7i2.90.

- [7] R. Erinton, M. R. Negara, And D. D. Sanjoyo, "Analisis Performasi Framework CodeIgniter dan Laravel Menggunakan Web Server Apache," 2017, vol. 4, no. 3, pp. 3565–3572.
- [8] P. Simanjuntak and A. Kasnady, "Analisis Model View Controller (MVC) pada Bahasa PHP," *Acad. J. - J. ISD*, vol. 1, no. 2, pp. 56–66, 2016.
- [9] R. Y. Endra and D. S. Aprilita, "E-Report Berbasis Web Menggunakan Metode Model View Controller untuk Mengetahui Peningkatan Perkembangan Prestasi Anak Didik," *Explor. – J. Sist. Inf. dan Telemat.*, vol. 9, no. 1, pp. 15–22, 2018.