

Kajian Routing pada Framework Laravel 5.0 dari Perspektif Penggunaan

Yusup Miftahuddin, Muhammad Ichwan, Achmad Zaky

¹Program Studi Teknik Informatika, Fakultas Teknologi Industri
Institut Teknologi Nasional - Bandung

Email: yusufm@itenas.ac.id

ABSTRAK

*HTTP routing merupakan suatu cara untuk menangani pemetaan URL dalam aplikasi web. Dari situs laravelbook.com dijelaskan komponen Laravel memiliki penambahan yaitu blok routing, dibandingkan dengan Symfony yang merupakan pendahulunya. Laravel menyediakan pendekatan *painless routing* (dari situs laravel.com) untuk menerapkan routing pada aplikasi web. Laravel pada situs resminya tidak menjelaskan secara mendetail mengenai konsep routing. Hal tersebut yang menjadi fokus penelitian untuk mengkaji syarat, fungsi dan peran, cara kerja, dan apa yang menjadi kelebihan dan kekurangan dengan fitur-fitur routing pada Laravel versi 5.0. Peneliti melakukan pengujian mengukur jumlah langkah pembuatan URI terhadap routing Laravel dan cara konvensional yang menyerupai routing Laravel, hasilnya Laravel memiliki langkah sekali deklarasi dibandingkan secara konvensional. Peneliti tidak menemukan kekurangan dari penggunaan routing pada Laravel tetapi developer harus memahami konsep routing pada Laravel tersebut sebagai konsekuensinya.*

Kata kunci: *Laravel, Laravel 5.0, Routing, Routing Laravel, HTTP Routing.*

ABSTRACT

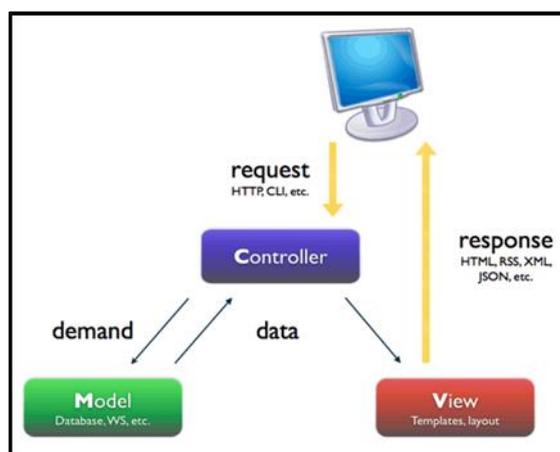
*HTTP routing is a way to handle URL mappings in the web application. From the site of laravelbook.com described laravel have additional routing component, compared with Symfony which is the predecessor. Laravel provide *painless approach routing* (from laravel.com sites) to implement routing in a web application. Laravel on the official website did not elaborate on the concept of routing. This is exactly the focus of research to assess the requirements, functions and role, ways of working, and what the advantages and disadvantages with routing features on laravel version 5.0. Researchers conducted a test to measure the number of steps for making URI against laravel routing and conventional ways that resemble routing laravel, laravel result has once steps than conventional declarations. Researchers did not find the lack of use of the routing on laravel but developers must understand the concept of routing on the laravel as a consequence.*

Keywords: *Laravel, Laravel 5.0, Routing, Laravel's Routing, HTTP Routing.*

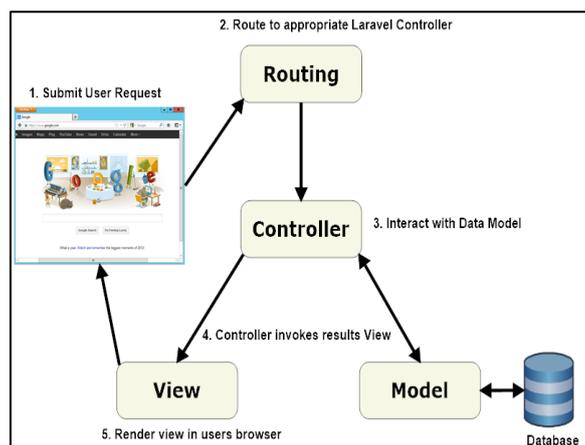
1. PENDAHULUAN

Dalam membangun tradisional *web* diperlukan membuat direktori (atau *virtualdirektory* pada IIS) untuk mengatur peta URL pada *web server*. Hal itu tidak akan menjadi masalah jika direktori-nya bukan merupakan parameter. Pengembangan *web* semacam ini memiliki batasan karena setiap parameter perlu dibuat direktorinya begitupun dalam pengelolaannya, diperlukan suatu sistem yang dapat menutupi celah permasalahan tersebut. Permasalahan ini dapat ditangani dengan *http routing*.

Pada tahun 2015 SitePoint, sebuah perusahaan *training online* melakukan *survey* mengenai *framework* PHP yang digunakan pada tahun 2015, yang mana *framework* Laravel menjadi *framework* PHP yang paling diminati dalam pengerjaan proyek di tempat bekerja maupun proyek pribadi [1]. Dari hasil *survey* tersebut membuat penulis tertarik untuk meneliti lebih dalam lagi Laravel, yang mana versi waktu keluaranya hasil survey adalah 5.0. *Framework* PHP Laravel dibangun dari *framework* PHP Symfony.



Gambar 1. Komponen pada *framework* PHP Symfony (Potencier, 2005)



Gambar 2. Komponen pada *framework* PHP Laravel (Ehsan, 2013)

Gambar 2 menjelaskan bahwa *framework* Laravel memiliki penambahan komponen yaitu blok *routing*. Laravel pada situs resminya <https://laravel.com/docs/5.0/routing> tidak dijelaskan secara rinci mengenai *routing*, pada situs resmi tersebut dijelaskan cara penggunaan dari fitur-fitur *routing* Laravel. Hal tersebut yang menjadi fokus penelitian untuk mengkaji syarat, fungsi dan peran, cara kerja, dan apa yang menjadi kelebihan dan kekurangan dengan *routing* pada *framework* Laravel khususnya versi 5.0.

Penelitian ini diharapkan dapat menjadi referensi dalam mengkaji dan memahami fungsi dan penggunaan *routing* pada *framework* PHP lainnya terutama *framework* PHP Laravel.

2. METODOLOGI PENELITIAN

2.1 Studi Literatur

Literatur yang digunakan adalah yang terkait dengan MVC, *routing* pada MVC, dan *framework* Laravel 5.0. Pembelajaran dilakukan dengan mencari referensi yang diperoleh dari buku dan media online yang diantaranya <https://laravel.com/docs/5.0>.

3. ANALISIS DAN PEMBAHASAN

Aspek yang dikaji pada penelitian ini adalah:

1. Syarat untuk menggunakan *routing* pada Laravel
2. Fungsi atau peran *routing* pada Laravel
3. Bentuk atau parameter yang harus dipenuhi dari *routing* pada Laravel
4. Fitur dari *routing* pada Laravel
5. Cara kerja *routing* pada Laravel

Untuk mengkaji kelima aspek tersebut dibangun sebuah aplikasi. Aplikasi menggunakan *framework* laravel 5.0. Selain dikaji kelima aspek tersebut dengan aplikasi, penulis juga melakukan studi banding antara *routing* dengan menggunakan Laravel dan cara yang menyerupai pembuatan *routing* dengan menggunakan Laravel (secara konvensional). Hal itu dilakukan untuk mengukur jumlah proses dalam membangun *routing* pada Laravel. Studi banding secara konvensional yang dibandingkan ada dua yaitu membangun URI menggunakan skrip dan tanpa menggunakan skrip. Untuk itu juga diperlukan membangun aplikasi pengujiannya.

Yang dibandingkan pada studi banding dalam penelitian ini meliputi:

1. Membuat URI aplikasi web
2. Membuat URI berparameter
3. Mengalihkan URI
4. Mengelompokkan URI
5. Membuat URI untuk proses CRUD

3.1.Aspek yang dikaji

3.1.1.Syarat untuk menggunakan *routing* pada Laravel

Pada tahap ini adalah mengkaji syarat-syarat yang harus dipenuhi dalam menggunakan *routing*. Pengkodean *routing* secara default berada di *file* `app/Http/routes.php`, yang dimuat oleh kelas `App\Providers\RouteServiceProvider` yang berada pada *file* `App\Providers\RouteServiceProvider.php`. *File* tersebut merupakan *file* untuk *route* yang dijalankan pada saat menggunakan aplikasi *web* pada *browser*. Dari hal tersebut baru dapat dilakukan pendefinisian *route*, untuk mendefinisikannya harus menggunakan kelas *Route*.

3.1.2. Fungsi atau peran *routing* pada laravel

Routing pada Laravel berfungsi untuk menangani semua URL *requested* pada aplikasi *web*. Terdapat dua peran *routing* pada Laravel, yaitu:

1. **Sebagai diri sendiri(*routing*)**, *routing* pada peran ini adalah sebagai penghubung antara *browser* dengan kelas *controller* aplikasi *web*. Pada Gambar 2 ditunjukkan peran *routing* pada Laravel sebagai *routing*. Ketika *user* men-*submit* permintaan pada *web browser* maka akan diterima oleh *routing*, *routing* akan mengeksekusi *route* yang sesuai permintaan URL dan menunjuk *controller* yang mana beserta *action*-nya, dari *controller* tersebut pada *action*-nya kode program akan dijalankan, dimana akan berinteraksi dengan *Model*, *routing* akan menghasilkan keluaran untuk ditampilkan menggunakan *View*, pada tahap terakhir dibuat tampilan ke *browser user*.
2. **Sebagai *controller***, *routing* pada Laravel merupakan bagian dari *controller*, tetapi *routing* bisa dibangun tanpa memerlukan kelas *controller*.

3.1.3. Bentuk atau parameter yang harus dipenuhi dari *routing* pada laravel

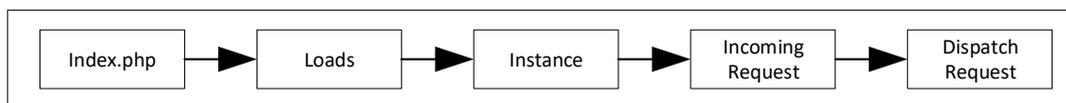
Bentuk route yang paling sederhana terdiri dari URI dan *closure*. URI disini dapat berupa "/" yang artinya nama *host* -nya, atau antara URI satu dengan yang lainnya dipisahkan dengan tanda "/". *Closure* disini terdiri dari nama kelas *controller* dan *action*, dimana *action* merupakan *function* yang didefinisikan dalam nama kelas *controller* tersebut. Cara penulisan atau deklarasinya adalah "NamaKelasController@action"; *Function* langsung pada saat mendeklarasikan *route*.

3.1.4. Fitur dari *routing* pada laravel

Laravel menghadirkan *routing* dengan berbagai fitur, hal ini membuat pengembang aplikasi dapat leluasa dalam membangun aplikasinya. Berikut adalah fitur *routing* dalam Laravel, yaitu:

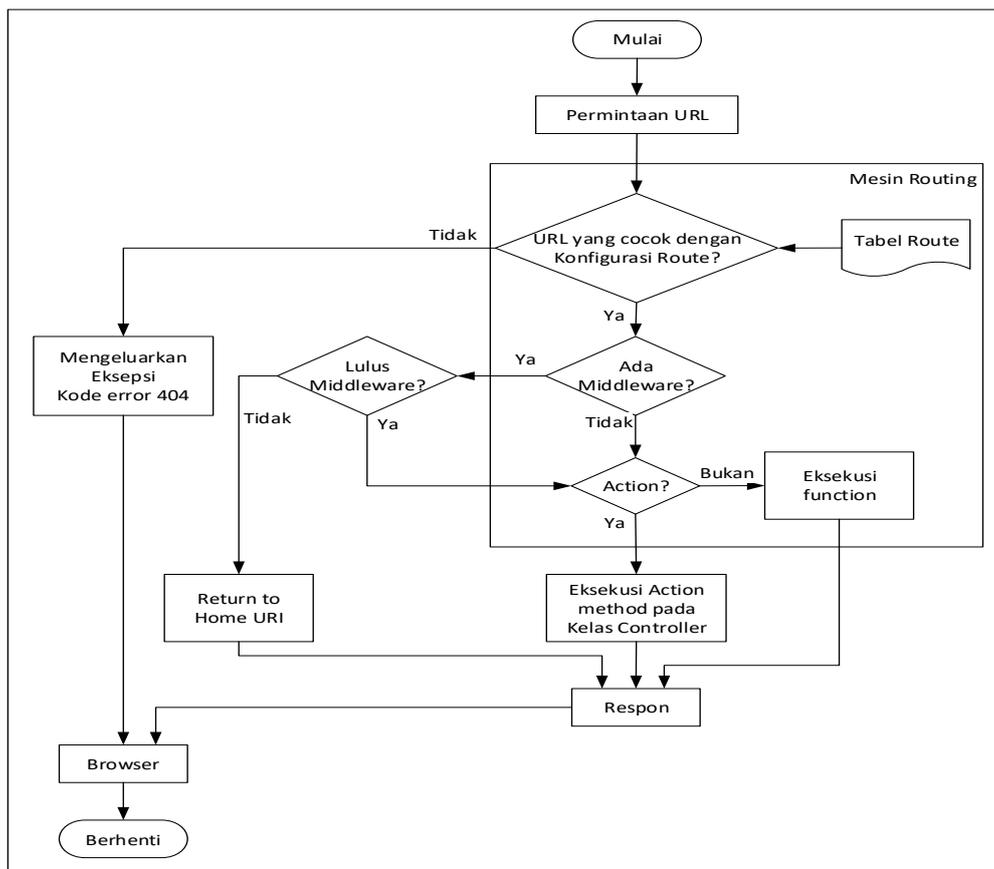
1. Menggunakan *HTTP Verb* yaitu *get*, *post*, *put*, dan *delete*. Selain itu juga dapat merupakan gabungan beberapa verb, atau mewakili semua *verb*.
2. Memiliki satu parameter, atau lebih.
3. Menggantikan *routing* dengan nama yang diinginkan dengan mengaliaskannya.
4. Mengelompokkan *routing* sehingga suatu *route* dapat memiliki banyak *route*. Dapat mengelompokkan berdasarkan *namespace*, atau memberikan hak akses dengan *middleware*.
5. *Route* sebagai *controller* yaitu menggantikan posisi *controller* dimana sintaks program dituliskan pada *route*.
6. Didefinisikan satu kali, untuk banyaknya *action* dari sebuah *controller*, yang dikenal dengan implicit controller.
7. Didefinisikan satu kali, untuk kebutuhan CRUD pada sebuah *controller*.

3.1.5. Cara kerja *routing* pada laravel



Gambar 4. Request lifecycle dari Laravel 5.0("Laravel 5 request lifecycle", 2016)

Tahap terakhir(pada Gambar 4) *dispatch request*(mengirim *request*), Setelah aplikasi telah *bootstrapped* dan semua *service provider* telah terdaftar, *request* akan diserahkan ke *router*(pada Gambar 5) untuk pengiriman(*dispatching*).



Gambar 5. Cara kerja *routing* pada Laravel

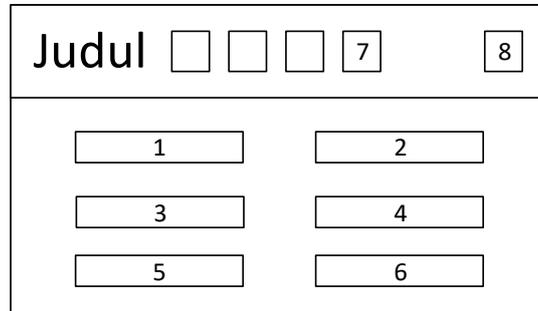
Penjelasan proses alur kerja *routing* pada Laravel 5.0 yaitu sebagai berikut:

1. *Routing* akan menerima permintaan URL dari *browser*
2. *Routing* akan mencari URL yang cocok dengan konfigurasi pada tabel *route*. Jika ditemukan maka ke langkah selanjutnya, jika tidak ditemukan lanjut ke nomor 10.
3. Apakah terdapat *middleware* pada konfigurasi *route*? Jika terdapat *middleware* maka lanjut ke langkah no.4, jika tidak ditemukan *middlewaare* maka ke lanjut ke nomor 6.
4. Uji kelulusan *middleware* pada mesin *middleware*. Jika lulus maka lanjut ke nomor 6, jika tidak maka ke langkah selanjutnya.
5. *Redirect* ke halaman *home*, lalu lanjut ke langkah
6. Cek apakah *closure* berupa *action*. Jika bukan maka lanjut ke nomor 8, Jika ya ke langkah selanjutnya.
7. Eksekusi *actionmethod* -nya pada kelas *controller* berdasarkan konfigurasi *route* tersebut, lalu lanjut ke nomor 9.
8. Eksekusi *function* pada konfigurasi *route* tersebut, kelangkah selanjutnya.
9. Mengeluarkan *response*, lalu lanjut ke nomor 11.
10. Mengeluarkan eksepsi kode *error* 404
11. Menampilkan ke *browser*
12. Berhenti

3.2. Perancangan

3.2.1. Aplikasi Menggunakan *routing* Laravel

Aplikasi ini dibuat untuk membuktikan kajian dari *routing* dengan *framework* Laravel 5.0.



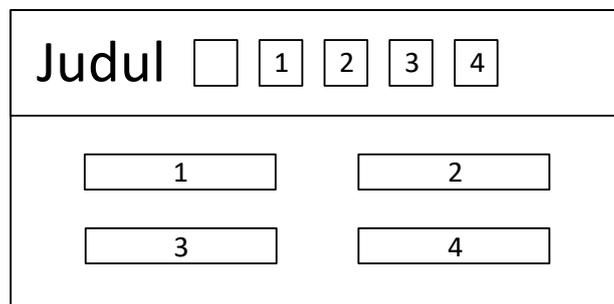
Gambar 6. Skema menu utama pengujian *routing* Laravel

Penjelasan Gambar 6 skema menu utama sebagai berikut:

Tombol 1 berisi pengujian untuk menguji *routing* yang berperan sebagai dirinya sendiri. Tombol 2 berisi pengujian untuk menguji *routing* yang berperan sebagai *controller*. Tombol 3 berisi pengujian untuk menguji *routing* yang dialiaskan. Tombol 4 berisi pengujian untuk menguji *routing* yang dikelompokkan. Tombol 5 berisi pengujian untuk menguji *routingimplicitcontroller*. Tombol 6 berisi pengujian untuk menguji *routingrestfullresourcecontroller*. Tombol 1, 2, 3, 4, 5, dan 6 menampilkan tampilan yang sama yaitu proses CRUD, tetapi terdapat perbedaan dari *methodrouting* –nya. Tombol 7 berisi pengujian untuk menguji *routing* dengan lokasi kelas *controller* ditentukan sendiri. Tombol ini menampilkan tampilan *formlogin*. Tombol 8 berisi pengujian untuk menguji *routing* dengan *middlewareauth*. Tombol ini menampilkan tampilan dengan teks *header* dan paragraf.

3.2.2. Aplikasi konvensional tanpa skrip

Aplikasi ini dibuat untuk membuktikan proses perhitungan langkah pembuatan URI secara konvensional tanpa menggunakan skrip. Maksud dari konvensional tanpa menggunakan skrip adalah pembuatan URI –nya tidak menggunakan skrip PHP ataupun skrip pada *webserver* Apache, tetapi langsung pada pembuatan direktori pada *webserver* untuk setiap URI –nya.



Gambar 7. Skema menu utama pengujian konvensional tanpa skrip

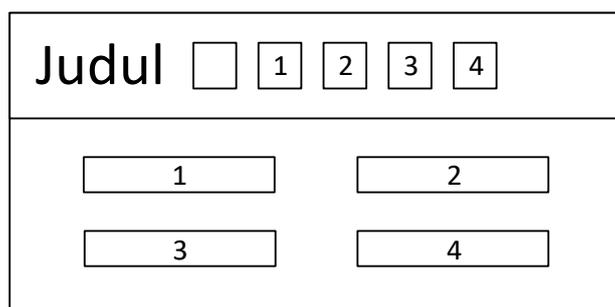
Penjelasan Gambar 7 skema menu utama sebagai berikut:

Halaman skema merupakan halaman pengujian untuk menguji membuat URI aplikasi *web*. Tombol 1 berisi pengujian untuk menguji URI berparamater. Tombol ini menampilkan

tampilan yang membaca parameter yang di *request* dari URL *browser*. Tombol 2 berisi pengujian untuk menguji mengaliaskan URI. Tombol ini menampilkan tampilan-tampilan yang dialiaskan URI –nya. Tombol 3 berisi pengujian untuk menguji mengelompokkan URI. Tombol ini menampilkan tampilan-tampilan yang telah dikelompokkan URI –nya. Tombol 4 berisi pengujian untuk menguji proses CRUD. Tombol ini menampilkan tampilan daftar data, form untuk membuat data baru, melihat salah satu isi data, dan form untuk mengubah data.

3.2.3. Aplikasi konvensional menggunakan skrip

Aplikasi ini dibuat untuk membuktikan proses perhitungan langkah pembuatan URI secara konvensional yang menggunakan skrip. Maksud dari konvensional yang menggunakan skrip adalah pembuatan URI –nya menggunakan skrip pada *webserver* Apache dan skrip PHP, sehingga akan tampak seperti URI dari *routing* Laravel.



Gambar 8. Skema menu utama pengujian konvensional menggunakan skrip

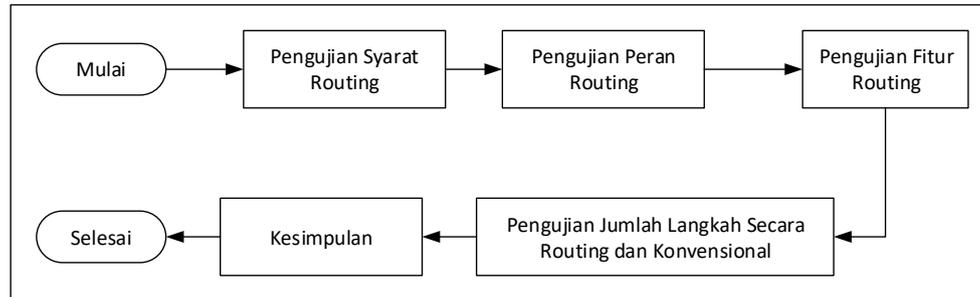
Penjelasan Gambar 8 skema menu utama sebagai berikut:

Halaman skema merupakan halaman pengujian untuk menguji membuat URI aplikasi *web*. Tombol 1 berisi pengujian untuk menguji URI berparamater. Tombol ini menampilkan tampilan yang membaca parameter yang di *request* dari URL *browser*. Tombol 2 berisi pengujian untuk menguji mengaliaskan URI. Tombol ini menampilkan tampilan-tampilan yang dialiaskan URI –nya. Tombol 3 berisi pengujian untuk menguji mengelompokkan URI. Tombol ini menampilkan tampilan-tampilan yang telah dikelompokkan URI –nya. Tombol 4 berisi pengujian untuk menguji proses CRUD. Tombol ini menampilkan tampilan daftar data, form untuk membuat data baru, melihat salah satu isi data, dan form untuk mengubah data.

3.3. Pengujian

Pada tahap ini perlu diketahui kerangka pengujian terlebih dahulu, yaitu:

1. **Pengujian syarat *routing***, pengujian ini bertujuan untuk mengetahui akibat-akibat yang terjadi jika terdapat perubahan dari syarat dalam menghadirkan dan berjalannya *routing*.
2. **Pengujian peran *routing***, pengujian ini dilakukan untuk mengetahui pengaruh yang dihasilkan dari perbedaan peran *routing*.
3. **Pengujian fitur *routing***, pengujian ini dilakukan pada beberapa fitur *routing* meliputi fitur alias *routing*, pengelompokkan *routing*, dan sekali definisi *routing*.
4. **Pengujian jumlah langkah secara *routing* dan konvensional**, pengujian ini dilakukan untuk mengukur jumlah proses atau langkah pembuatan URI terhadap *routing* Laravel dan cara konvensional yang menyerupai *routing* Laravel.



Gambar 5. Cara kerja *routing* pada Laravel

Pengujian routing Laravel dengan cara menggunakan aplikasi yang dibuat menggunakan Laravel. Sedangkan pengujian secara konvensional menggunakan aplikasi konvensional tanpa skrip (untuk menguji proses perhitungan konvensional tanpa skrip) dan aplikasi konvensional menggunakan skrip (untuk menguji proses perhitungan konvensional yang menggunakan skrip).

Tabel 1. Hasil perbandingan jumlah langkah secara routing Laravel dan konvensional

Yang Dibandingkan	Secara Konvensional		Routing Laravel (dalam langkah)
	Tanpa Skrip (dalam langkah)	Dengan Skrip (dalam langkah)	
Membuat URI aplikasi <i>web</i>	2	2	1
Membuat URI berparameter	2	2	1
Mengalihkan URI	2	2	1
Mengelompokkan URI – membuat URI induk	1	1	1
Mengelompokkan URI – membuat URI anggota	2	2	1
Membuat URI untuk proses CRUD	8	8	1

4. KESIMPULAN

Berdasarkan tabel 1 perbandingan jumlah langkah dalam membuat URI terhadap *routing* Laravel dan cara konvensional, bahwa *routing* pada Laravel 5.0 memiliki langkah sekali definisi jika dibandingkan tanpa *routing* pada saat membuat URI aplikasi *web*, membuat URI berparameter, mengalihkan URI, mengelompokkan URI, dan membuat URI untuk proses CRUD. Untuk proses CRUD, Laravel menyediakan *Restful Routing* yang membuat *developer* mempersingkat langkahnya dalam membangun CRUD dengan satu baris deklarasi. *Routing* pada Laravel dapat berperan sebagai *controller*, sehingga menggantikan fungsi kelas *controller*. *Routing* pada Laravel 5.0 terdapat fitur untuk keamanan ataupun mem-*filterroute*, dengan menggunakan fitur *HTTP Middleware*. Laravel menyediakan fitur *implicitcontroller* yang membuat cukup sekali deklarasi *route* –nya untuk berbagai macam *action* dalam *controller*. Peneliti tidak menemukan kekurangan dari penggunaan *routing* pada Laravel. Ada konsekuensi yang harus ditempuh yaitu *developer* harus memahami konsep *routing* pada Laravel tersebut.

DAFTAR RUJUKAN

- [1] B. Skvorc, "The Best PHP Framework for 2015: SitePoint Survey Results," 2015. [Online]. Available: <http://www.sitepoint.com/best-php-framework-2015-sitepoint-survey-results/>. [Diakses 15 April 2015].
- [2] F. Potencier, "Day 4: The Controller and the View," 2005. [Online]. Available: http://symfony.com/legacy/doc/jobeeet/1_2/en/04?orm=Propel. [Diakses 26 April 2015].
- [3] T. Otwell, "HTTP Routing - Laravel - The PHP Framework For Web Artisans," 2015. [Online]. Available: <https://laravel.com/docs/5.0/routing>. [Diakses 26 April 2015].
- [4] M. Ehsan, "Architecture of Laravel Applications - Laravel Book," 2013. [Online]. Available: <http://laravelbook.com/laravel-architecture/>. [Diakses 26 April 2015].
- [5] "Laravel 5 request lifecycle," 2016. [Online]. Available: <http://www.phpcodehub.co.in/article/laravel-5-request-lifecycle-aid-10.html>. [Diakses 2016 Juli 2016].