

Perbandingan Pengembangan Front End Menggunakan Blade Template dan Vue Js

Christian Chastro^{#1}, Erico Darmawan H., S.Kom., M.T.^{#2}

[#]Program Studi Teknik Informatika, Universitas Kristen Maranatha
Jl. Surya Sumantri No.65, Sukawarna, Kec. Sukajadi, Bandung

¹christian.chastro.christianc@gmail.com

²erico.dh@it.maranatha.edu

Abstract — Front-end is one of the most important parts of a website, because front-end is a part of website that will be seen directly by the user. When developing front-end, can using framework to simplify the process of coding. Because there's many existing frameworks, this research will use Laravel framework, this framework provides several options for developing front-end website using Blade Template, and also using Vue Js framework. Based on these choices, this research will examine the performance of websites using Blade Template, and Vue Js, and the features that can be used for develop a website. To test performance will use Chrome DevTools Audits panel, this panel is default from Google Chrome browser. Because testing will be performed on local server, this panel can be used to measure the performance from website. After test the performance, it can be concluded website using Blade Template performance is faster than Vue Js. However, Vue Js provides more convenience in developing complex website applications. Blade Template can be used to develop website with fast performance but the website is simple. Vue Js can be used to create complex websites easily but their performance is slow.

Keywords— blade template, framework comparison, front-end, laravel, vue js.

I. PENDAHULUAN

Front-end merupakan sebuah tampilan dari suatu website yang akan dilihat secara langsung oleh pengguna website. Tampilan sebuah *website* merupakan salah satu komponen penilaian bagus atau tidaknya sebuah *website*, selain dari fungsionalitasnya. Terdapat banyak *framework* yang tersedia untuk membantu *front-end developer* untuk mempermudah *developer* dalam mengembangkan suatu aplikasi atau *website*.

Membandingkan *framework* tersebut dibutuhkan untuk mengetahui *framework* yang cocok untuk digunakan pada saat membangun sebuah *website*. Karena banyaknya jumlah *framework*, penelitian ini membandingkan sebuah *framework* PHP yaitu Laravel, dan *framework* JavaScript yaitu Vue Js. Dalam perbandingan pengembangan *front-end* akan menggunakan *templating engine* dari Laravel yaitu Blade Template. Untuk membandingkan *framework* tersebut akan membuat beberapa macam jenis halaman *website* mulai dari yang sederhana hingga *website* yang kompleks. Perbandingan yang akan dilakukan berupa perbandingan performa, perbandingan kecepatan *script*, perbandingan *routing*, dan perbandingan tampilan. Hasil dari penelitian ini, dapat digunakan sebagai pertimbangan *framework* yang sesuai dengan aplikasi atau *website* yang akan dikembangkan.

Pada penelitian ini terdapat beberapa referensi yang digunakan dalam menentukan hal yang dibandingkan dan dalam melakukan perbandingan. Hal yang akan dibandingkan adalah performa dan tampilan yang dapat dihasilkan dari penggunaan *framework*, karena pemilihan aplikasi yang tepat sesuai dengan kebutuhan menjadi hal yang harus sangat diperhatikan [1]. Terlebih aplikasi yang dipilih akan mempengaruhi dalam pengembangan sistem yang akan dibuat serta bagian *front-end* pada sistem. Pengujian pada *framework* Laravel dilakukan karena sistem yang dihasilkan memiliki performa cepat, dan *response time* yang cepat jika dibandingkan dengan *framework* CodeIgniter, pengujian pada *framework* Vue Js dilakukan karena sistem yang dihasilkan memiliki *request response time* yang cepat jika dibandingkan dengan Vanilla Js [3].

Dalam melakukan analisis pengujian pastilah memiliki Skenario pengujian yang dilakukan pada penelitian tersebut [4]. Alat yang digunakan untuk melakukan pengujian performa dari sistem yang akan dibuat menggunakan Chrome DevTools [3]. Hasil dari perbandingan performa tersebut tersimpan kedalam grafik perbandingan, dimana grafik tersebut dapat terlihat perbandingan dari performa masing – masing *framework* [5].

Berdasarkan referensi yang ada, maka penelitian ini akan membandingkan pengembangan *front-end* menggunakan Blade Template yang ada pada *framework* Laravel dengan *framework* Vue Js. Dimana kedua *framework* tersebut memiliki keunggulan pada *response time* [2], [3]. Salah satu pengujian merupakan performa dari masing – masing *framework* yang

menggunakan Chrome DevTools Audit Panel. Hasil dari perbandingan tersebut akan ditampilkan dalam bentuk grafik perbandingan setiap parameter pengujian performa yang telah dilakukan. Selain itu penelitian ini juga akan membandingkan waktu *scripting* pada Blade Template dan Vue Js untuk mengetahui kecepatan dari perhitungan yang dapat dilakukan oleh setiap *framework*. Setelah itu, data yang didapatkan dari pengujian tersebut akan dianalisis kelebihan dan kekurangan dari masing-masing *framework*.

II. DASAR TEORI

A. Pengertian Front-End

Front-End merupakan antarmuka pengguna yang menyajikan fungsi atau kegunaan dari sebuah *website* serta berinteraksi secara langsung dengan pengguna dan juga *front-end* berkaitan dengan pengalaman dari pengguna ketika berinteraksi dengan *website*. Pengembang tampilan *website* atau pada bagian *front-end* disebut sebagai *Front-End Developer*. *Front-End* dibangun dengan menggunakan beberapa komponen yaitu HTML, CSS, dan JavaScript [6].

B. Hypertext Markup Language (HTML)

Di dalam dunia pemrograman berbasis *website* atau sering disebut sebagai *Web Programming*, HTML merupakan bahasa *formatting* yang digunakan sebagai dasar dalam pembuatan sebuah halaman *website*. *File* HTML yang sudah dibuat disimpan dengan ekstensi *.html* (dot html). Sebuah *file* HTML dapat di-*running* atau dijalankan dengan menggunakan *browser* (Google Chrome, Mozilla Firefox, Opera, Internet Explorer, Microsoft Edge, dan lain-lain) [7].

C. Cascading Style Sheets (CSS)

CSS merupakan bahasa pemrograman yang digunakan untuk membuat web *design*, karena CSS memiliki sifat “*Style Sheet Language*”. Pada HTML CSS memiliki penanda yang berupa *id*, dan *class* yang terletak pada *tag* pembuka HTML. Selain menggunakan penanda, CSS juga dapat langsung dituliskan pada setiap *tag* pembuka HTML dengan menggunakan atribut *style*. CSS dapat mengubah seluruh tampilan dari halaman *website*, dimulai dari warna *background*, jenis *font*, ukuran *font*, warna *font*, ukuran *layout*, warna elemen serta CSS juga dapat mengubah sebuah halaman menjadi *responsive*, dan masih banyak lagi kegunaan yang dapat dilakukan oleh bahasa pemrograman CSS dalam hal mempercantik sebuah tampilan pada halaman *website*.

D. JavaScript

Javascript merupakan bahasa pemrograman yang awalnya dirancang untuk berjalan di atas *browser* atau *client-side*, akan tetapi karena perkembangan zaman Javascript dapat dijalankan pada *computer server* atau *server-side* [8].

E. Hypertext Preprocessor (PHP)

PHP merupakan salah satu bahasa pemrograman yang berjalan pada sisi *server* atau sering di sebut dengan *server-side*. PHP hanya bisa berjalan pada sisi *server* karena untuk menjalankan *file* dengan ekstensi *.php* (dot php) haruslah menggunakan *web server*. Pada *front-end* PHP dapat memproses HTML bisa dilihat dari arti kepanjangan dari PHP itu sendiri yaitu pemrosesan *hypertext*, dimana *hypertext* sendiri merupakan HTML [9]. PHP juga merupakan salah satu dari bahasa pemrograman *script* atau disebut juga sebagai *Scripting Language* karena PHP dapat disisipkan ke dalam dokumen HTML dengan cara menggunakan *tag* PHP, dimana *tag* PHP ini merupakan penanda kode PHP. *Tag* PHP ini dibuka dengan karakter “<?php” dan ditutup dengan karakter “?”>” diantara kedua *tag* tersebut yang merupakan bahasa pemrograman PHP.

F. Bootstrap

Bootstrap merupakan *library framework* CSS dimana Bootstrap dibuat khusus untuk pengembangan *front-end* pada *website*. Bootstrap merupakan salah satu dari *framework front-end* karena terdiri dari HTML, CSS, dan Javascript. Maka dari itu Bootstrap sudah menyediakan kerangka kerja yang terdiri dari komponen-komponen dimana komponen-komponen tersebut dapat digunakan dengan cara menambahkan *class* pada suatu *tag* HTML [10].

G. Framework

Framework memiliki fungsi untuk memudahkan para *developer* mengembangkan ataupun membuat aplikasi yang akan dibuat, karena *framework* merupakan sebuah kerangka kerja dimana berisikan perintah dan fungsi dasar yang umum diharapkan aplikasi yang sedang dibuat dapat dibangun dengan lebih cepat serta tersusun dan terstruktur. *Framework* juga memiliki komponen-komponen pemrograman yang sudah disediakan dan siap untuk digunakan oleh *developer* yang menggunakan *framework* tersebut [11],[12].

- 1) *Framework Laravel*: Laravel merupakan salah satu *framework* PHP, *framework* Laravel ini dibangun dengan menggunakan konsep MVC (Model View Controller) [13]. Pada *framework* Laravel dilengkapi dengan *command line tool* dengan nama Artisan, *artisan* bisa digunakan untuk membantu *developer* dalam melakukan *packaging bundle* dan *instansi bundle* melalui *command prompt* [14]. Blade Template merupakan *template engine* yang sudah disediakan oleh Laravel. *Template engine* merupakan suatu metode yang biasanya digunakan untuk mempersingkat penulisan kode program.
- 2) *Framework Vue Js*: Vue JS merupakan salah satu *framework* JavaScript yang biasanya digunakan untuk membangun tampilan antarmuka pengguna (*user interface*) dari suatu *website*. Vue Js menyediakan beberapa fitur-fitur yang dapat digunakan untuk membantu *developer* untuk mengembangkan aplikasi *website* seperti fitur *Virtual DOM*, *Component Base*, *Template*, *Routing*, dan *State Management* [15].

H. Chrome DevTools

Chrome DevTools merupakan satu set alat yang dapat digunakan untuk mengintegrasikan, dan melakukan *debug* pada *website* yang sedang dibuat. DevTools merupakan bawaan dari *browser* Google Chrome. Panel yang digunakan pada penelitian ini adalah panel Audits yang dapat digunakan untuk melakukan pengujian performa pada lokal *website*. Pengujian performa menggunakan panel Audits akan diukur dengan matriks performa sebagai berikut ini:

- 1) *First Contentful Paint*: *First Contentful Paint* merupakan waktu pertama kali *browser* membuat konten seperti teks, gambar, kanvas, dan komponen lain sebagainya pada layar.
- 2) *First Meaningful Paint*: *First Meaningful Paint* mengukur waktu ketika konten utama pada *website* dapat terlihat pada layar.
- 3) *First CPU Idle*: *First CPU Idle* menandai poin pertama halaman sebuah *website* dapat menanggapi *input* dari pengguna.
- 4) *Time to Interactive*: *Time to Interactive* adalah sejumlah waktu yang diperlukan sebuah *website* dapat interaktif dengan sepenuhnya.
- 5) *Speed Index*: *Speed Index* menampilkan seberapa cepat semua konten sebuah halaman dapat terlihat pada layar.
- 6) *Max Potential First Input Delay*: *Max Potential First Input Delay* merupakan seberapa tinggi durasi dari potensi keterlambatan *input* yang dapat dialami oleh pengguna dalam millidetik, dari tugas yang paling panjang.

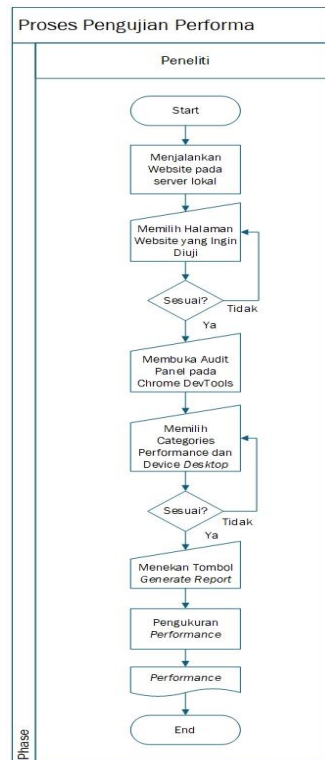
III. PERANCANGAN SISTEM

A. Perancangan Sistem

Pada Perancangan Sistem merupakan tahap perancangan sistem yang akan dibuat sebelum melakukan analisis performa. Dalam perancangan ini sistem yang dibuat akan mendukung ketika dilakukan pengukuran performa dengan membangun sebuah sistem yang memiliki arsitektur MVC untuk *framework* Laravel, dan arsitektur MVVM untuk *framework* Vue Js. Sistem yang akan dibuat meliputi sistem *login*, pengajuan, persetujuan pengajuan, dan penilaian.

B. Skenario Pengujian

Skenario Pengujian yang dilakukan ketika melakukan pengujian performa pada sistem yang sudah dibuat menggunakan Chrome DevTools Audits Panel dapat dilihat pada Gambar 1. Skenario tersebut dijelaskan dengan menggunakan *flowmap*, penggunaan *flowmap* tersebut dapat mempermudah untuk melihat langkah-langkah pengujian yang dilakukan untuk pengumpulan data yang akan digunakan untuk analisis pada penelitian ini.

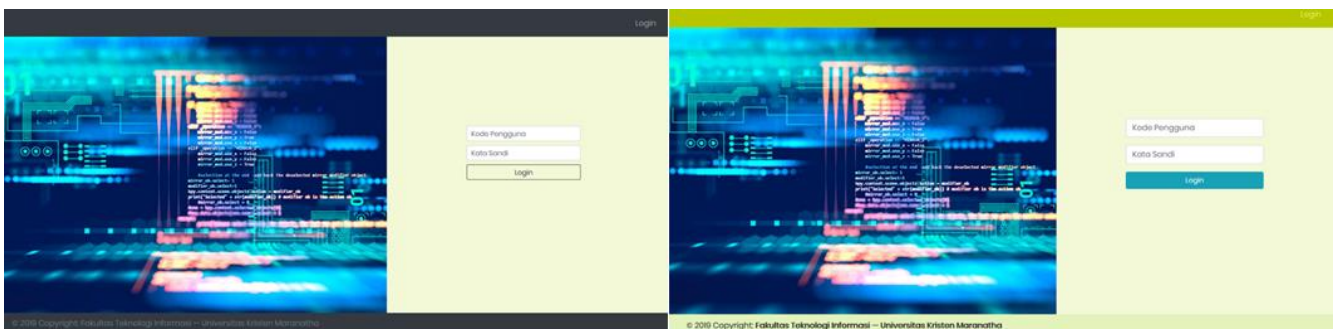


Gambar 1. Flowmap Proses Pengujian Performa

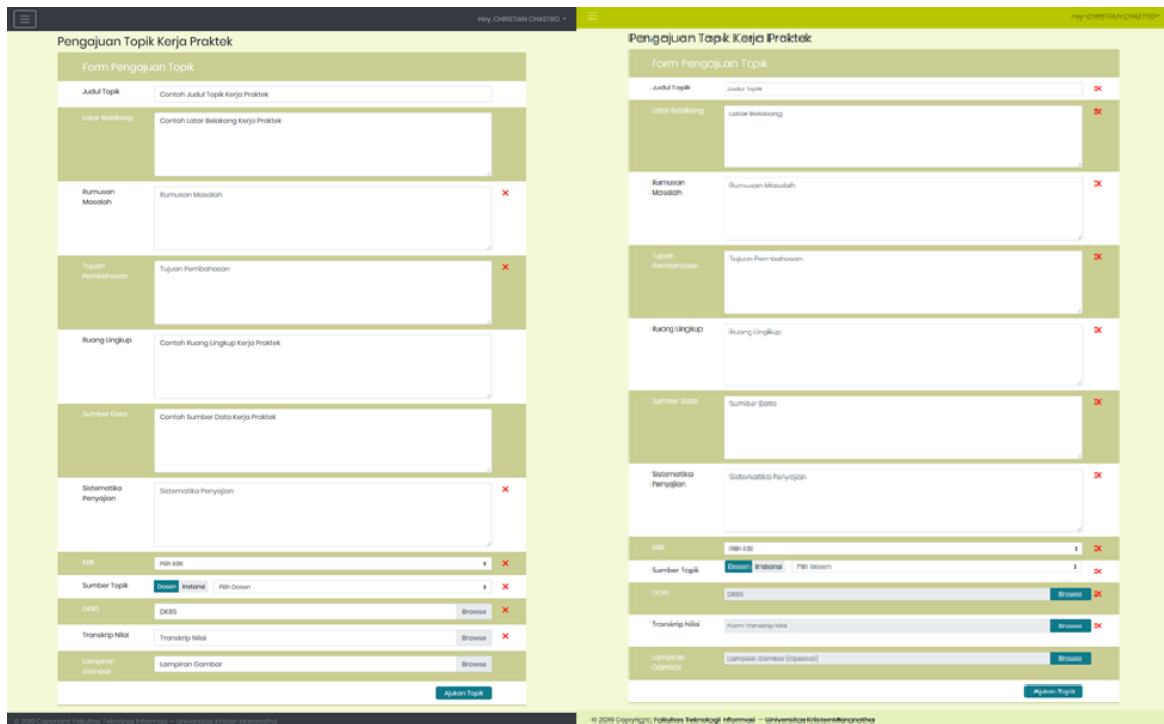
IV. IMPLEMENTASI DAN PENGUJIAN

A. Hasil Aplikasi

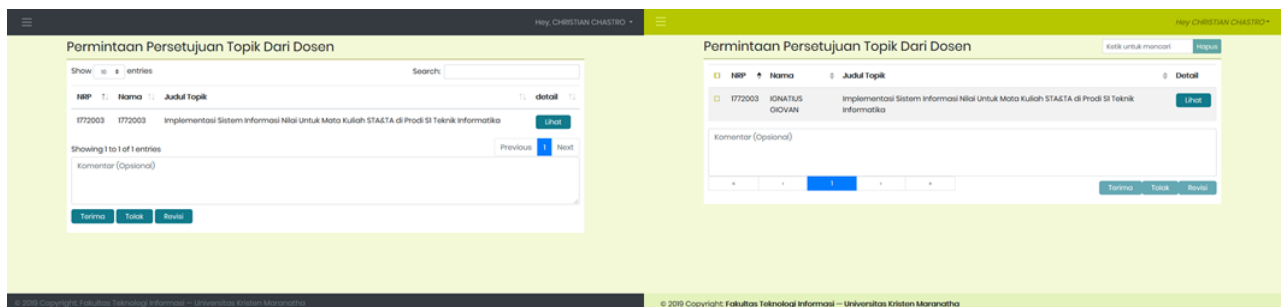
Setelah melakukan perancangan sistem, pada tahap ini merupakan tahap implementasi dari hasil rancangan system yang sudah dilakukan. Supaya dapat membandingkan dari sisi desain dan fungsionalitas pada Blade Template dan Vue Js, dalam pembuatannya menggunakan Bootstrap dalam pembuatan tampilannya. Halaman *website* yang menggunakan Blade Template ditandai dengan *navigation bar* atau *navbar* berwarna hitam, dan untuk Vue Js ditandai dengan warna hijau. Pada Gambar 2 sampai dengan Gambar 9 merupakan tampilannya.



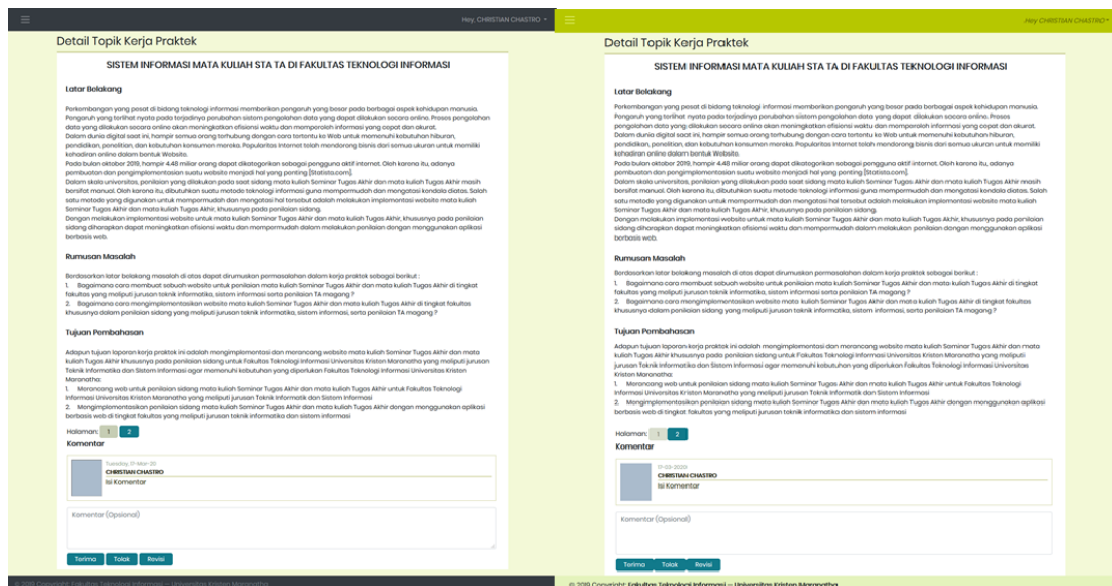
Gambar 2. Perbandingan Halaman Login



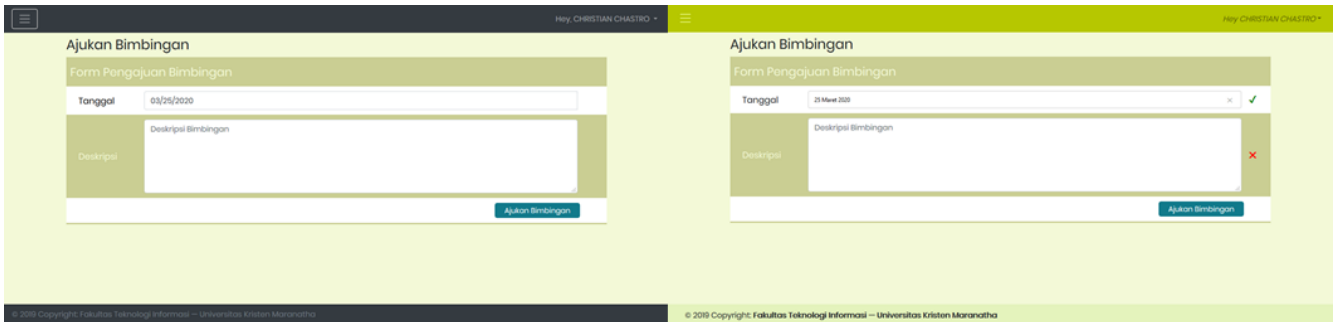
Gambar 3. Perbandingan Halaman Pengajuan Topik



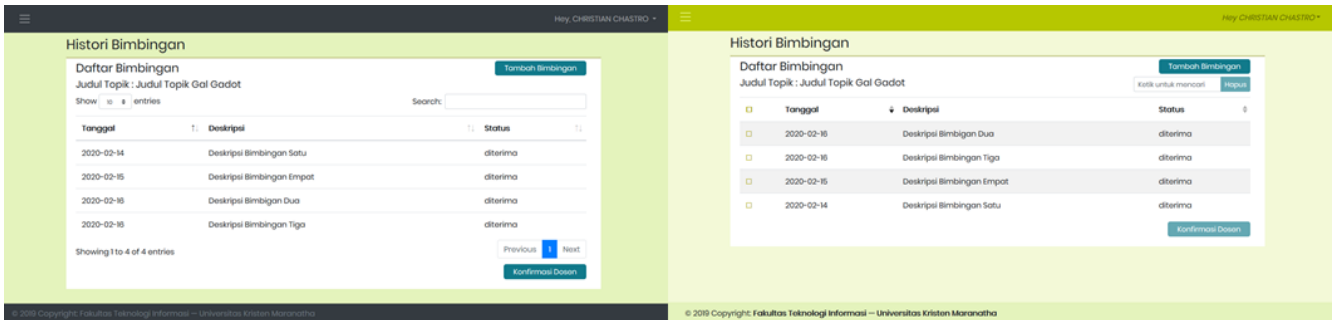
Gambar 4. Perbandingan Halaman Persetujuan Topik



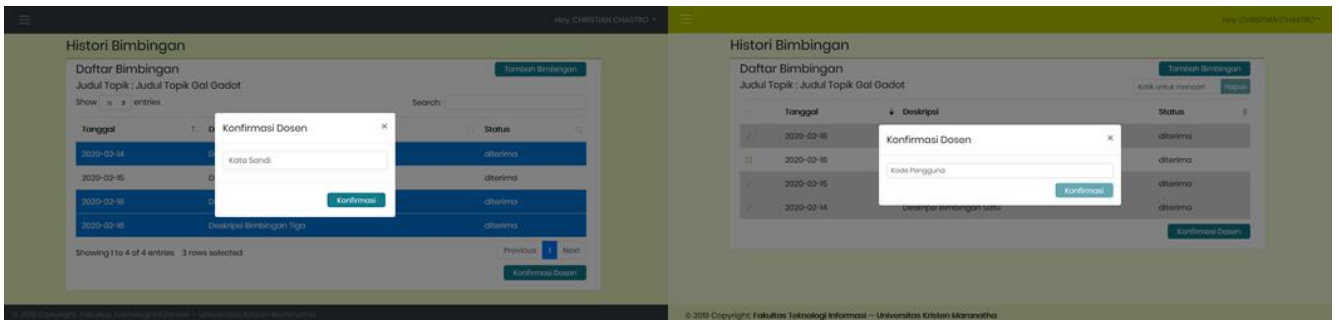
Gambar 5. Perbandingan Halaman Detail Pengajuan Topik



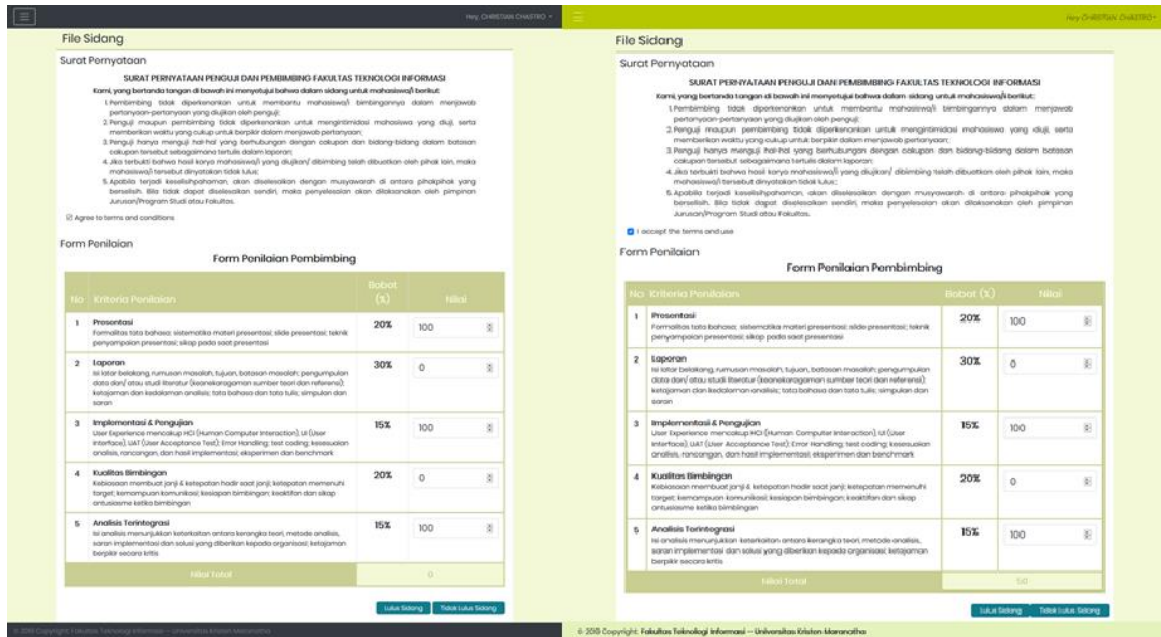
Gambar 6. Perbandingan Halaman Pengajuan Bimbingan



Gambar 7. Perbandingan Halaman Histori Bimbingan



Gambar 8. Perbandingan Selected dan Modal



Gambar 9. Perbandingan Halaman Penilaian Sidang

Pada Gambar 2 merupakan tampilan halaman *login*, dimana pada tampilan terlihat tidak ada perbedaan yang signifikan. Pada Gambar 3 merupakan tampilan halaman pengajuan topik, dimana pada tampilan ini Blade Template memiliki perbedaan yang signifikan dengan Vue Js pada saat melakukan validasi, Pada Blade Template pengguna terlebih dahulu *submit* data baru melakukan validasi akan tetapi pada Vue Js ketika pengguna memasukkan data maka sistem akan langsung melakukan validasi setiap kali pengguna melakukan perubahan data. Pada Gambar 4 merupakan tampilan halaman persetujuan topik, pada halaman ini terdapat tabel, ketika menggunakan Blade Template untuk membuatnya tidak cukup dengan Bootstrap saja, harus menambah *framework* lain, disini digunakan *datatables*, akan tetapi pada Vue Js cukup menggunakan Bootstrap Vue, yang merupakan Bootstrap yang dibuat secara khusus untuk dapat digunakan pada Vue Js.

Pada Gambar 5 merupakan halaman detail pengajuan topik yang sudah dilakukan, pada halaman ini juga tidak terlihat perbedaan yang signifikan. Berikutnya adalah halaman pengajuan bimbingan yang ada pada Gambar 6, pada halaman ini sama dengan Gambar 3 dimana validasi pada Blade Template akan dilakukan setelah pengguna *submit*, sedangkan pada Vue Js validasi dilakukan setiap kali ada perubahan data yang dilakukan oleh pengguna. Pada Gambar 7 merupakan perbandingan halaman histori bimbingan, dimana pada halaman ini memiliki komponen yang sama dengan Gambar 4. Gambar 8 merupakan halaman histori bimbingan pada saat memilih data pada tabel dan tampilan *modal* atau *pop-up*, terlihat perbedaan pada saat memilih data pada tabel jika pada Blade Template menyesuaikan yang sudah disediakan oleh *datatables*, akan tetapi pada Vue Js tampilannya bisa disesuaikan dengan keinginan. Pada Gambar 9 merupakan tampilan halaman penilaian sidang, pada Blade Template nilai total tidak berubah ketika ada perubahan pada nilai, akan tetapi pada Vue Js setiap kali ada perubahan data maka nilai total akan ikut berubah.

B. Pengujian Performa

Pada tahap Pengujian Performa akan dilakukan dengan menggunakan Chrome DevTools Audits Panel, dimana dengan *tool* ini dapat mengukur performa yang didapatkan dari suatu halaman *website* meskipun *website* tersebut dijalankan pada *server* lokal.

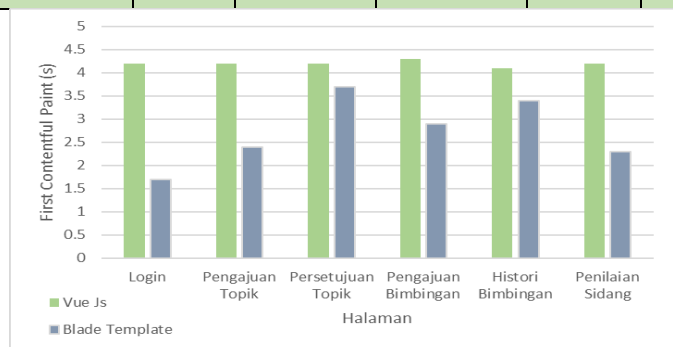
Pada Tabel 1 dan Tabel 2 merupakan data-data yang sudah diperoleh dari pengujian yang sudah dilakukan sebelumnya terhadap system yang sudah dibuat. Tabel 1 merupakan performa dari Blade Template dan Tabel 2 berisi data performa dari Vue Js. Pada Gambar 10 sampai dengan Gambar 16 data performa tersebut akan digambarkan dalam bentuk grafik berikut penjelasannya.

TABEL I
PENGUJIAN PERFORMA BLADE TEMPLATE

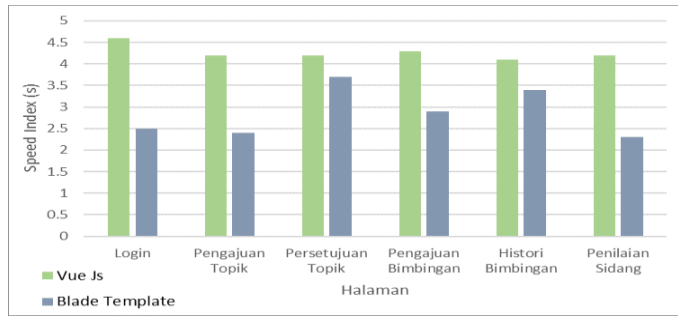
Halaman	First Contentful Paint (s)	Speed Index (s)	Time to Interactive (s)	First Meaningful Paint (s)	First CPU Idle (s)	Max Potential First Input Delay (ms)	Score
Login	1.7	2.5	2.5	1.7	2.5	100	98
Pengajuan Topik	2.4	2.4	2.4	2.4	2.4	30	95
Persetujuan Topik	3.7	3.7	3.9	3.8	3.8	200	79
Pengajuan Bimbingan	2.9	2.9	3.2	2.9	3.1	150	90
Histori Bimbingan	3.4	3.4	3.6	3.4	3.5	130	84
Penilaian Sidang	2.3	2.3	2.3	2.3	2.3	60	97
Rata-rata	2.73	2.87	2.98	2.75	2.93	111.67	90.50

TABEL II
 PENGUJIAN PERFORMA VUE JS

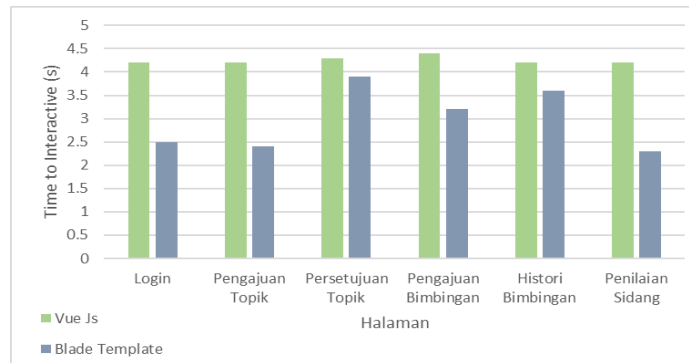
Halaman	First Contentful Paint (s)	Speed Index (s)	Time to Interactive (s)	First Meaningful Paint (s)	First CPU Idle (s)	Max Potential First Input Delay (ms)	Score
Login	4.2	4.6	4.2	4.2	4.2	290	70
Pengajuan Topik	4.2	4.2	4.2	4.2	4.2	290	72
Persetujuan Topik	4.2	4.2	4.3	4.2	4.2	330	72
Pengajuan Bimbingan	4.3	4.3	4.4	4.3	4.3	340	71
Histori Bimbingan	4.1	4.1	4.2	4.1	4.1	250	74
Penilaian Sidang	4.2	4.2	4.2	4.2	4.2	290	73
Rata-rata	4.20	4.27	4.25	4.20	4.20	298.33	72.00



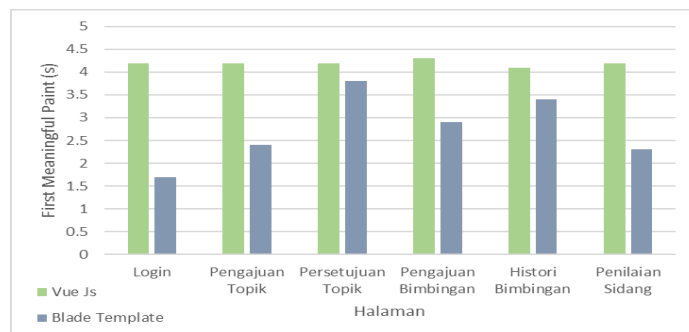
Gambar 10. Grafik First Contentful Paint



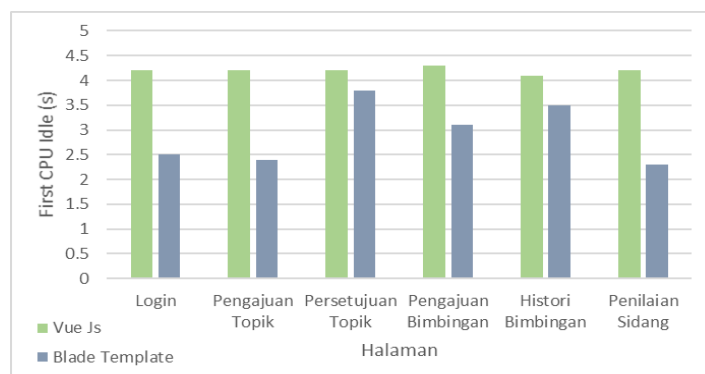
Gambar 11. Grafik Speed Index



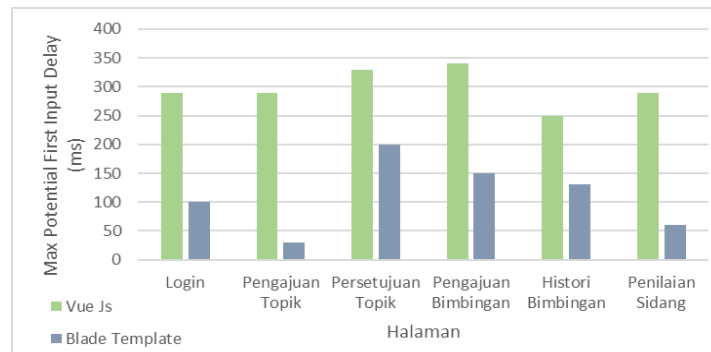
Gambar 12. Grafik Time to Interactive



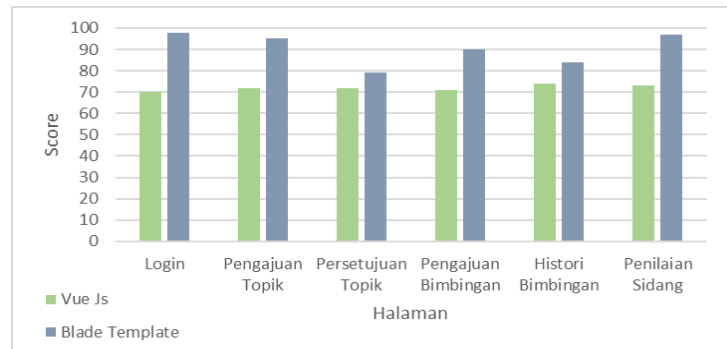
Gambar 13. Grafik First Meaningful Paint



Gambar 14. Grafik First CPU Idle



Gambar 15. Grafik Max Potential First Input Delay



Gambar 16. Grafik Score

Pada Gambar 10 sampai dengan Gambar 16 merupakan grafik performa dari Tabel I dan Tabel II, dimana grafik di group berdasarkan parameter pengujian dari Chrome Devtools. Terlihat pada Gambar 10 sampai dengan Gambar 15, waktu yang dibutuhkan pada Blade Template lebih sedikit jika dibandingkan dengan Vue Js yang memerlukan waktu lebih lama. Pada Gambar 16 merupakan gambar grafik *score* yang bernilai 0-100 sesuai dengan performa yang diperoleh, terlihat jika Blade Template memiliki *score* yang paling tinggi jika dibandingkan dengan Vue Js pada sisi performanya.

C. Pengujian Kecepatan Scripting

Pada pengujian perbandingan kecepatan scripting ini menggunakan ChromeDevtools yaitu *Timeline* Panel, panel ini dapat digunakan untuk mengetahui berapa lama suatu *script* akan berjalan. Pada pengujian ini akan membuat sebuah *website* khusus pada Blade Template dan Vue Js. *Website* yang dibuat adalah perhitungan nilai siswa yang sudah didapatkan dalam kurun waktu 1 semester dengan jumlah siswa 15 orang dan jumlah matapelajaran 100 buah, serta siswa memiliki 16 jenis nilai yaitu 7 nilai tugas, 7 nilai ulangan, 1 nilai UTS, 1 nilai UAS.

Setiap Nilai yang diperoleh setiap siswa akan diacak pada sistem, setelah itu sistem akan menghitung nilai akhir dari siswa dan rata-rata setiap jenis nilai pada setiap matapelajaran. Untuk menghitung rata-rata menggunakan rumus:

$$NA = \left(\frac{NT}{7} \times 0.3 \right) + \left(\frac{NU}{7} \times 0.2 \right) + (UTS \times 0.25) + (UAS \times 0.25)$$

Dimana:

NA: Nilai Akhir yang diperoleh oleh seorang siswa

NT: Total dari nilai tugas yang diperoleh oleh seorang siswa

NU: Total dari nilai ulangan yang diperoleh oleh seorang siswa

UTS: Nilai UTS yang diperoleh oleh seorang siswa

UAS: Nilai UAS yang diperoleh oleh seorang siswa

Pada Gambar 17 merupakan contoh sebuah tabel hasil kalkulasi penilaian siswa yang sudah dibuat dari satu matapelajaran.

Kalkulasi Nilai																
Tugas							Ulangan							UTS	UAS	Nilai Akhir
1	2	3	4	5	6	7	1	2	3	4	5	6	7			
61	54	63	75	87	52	58	70	62	56	87	57	50	79	80	87	74.21
51	70	74	70	98	58	63	63	52	98	71	78	95	92	59	67	67.93
85	57	75	98	67	89	81	93	93	59	59	91	51	75	100	75	82.29
100	69	98	91	50	63	87	65	84	90	72	59	100	67	95	59	77.76
54	69	75	85	66	51	54	94	90	73	94	77	77	86	88	82	78.84
56	56	99	90	100	86	88	97	71	64	82	84	60	53	53	91	75.24
67	52	73	65	76	85	62	97	71	88	81	96	70	82	84	66	74.79
81	80	88	75	52	85	55	55	64	64	72	75	84	71	88	59	72.72
96	87	83	92	75	80	71	81	69	62	70	75	81	88	60	72	73.06
61	78	71	87	83	64	56	54	76	79	54	53	88	82	59	58	64.56
50	63	81	87	77	84	82	56	87	62	78	56	78	99	67	94	77.45
68	85	65	84	86	73	55	55	56	63	58	84	52	87	70	62	68.11
87	82	66	66	62	87	88	70	59	62	78	83	67	96	100	50	75.27
62	53	91	90	74	53	93	67	73	78	87	72	98	71	95	71	79.21
88	53	80	86	94	59	97	96	86	61	60	95	50	59	81	78	78.11
71.13	67.2	78.8	82.73	76.47	71.27	72.67	74.2	72.87	70.6	73.53	75.67	73.4	79.13	78.6	71.4	74.64

Gambar 17. Tabel Hasil Kalkulasi Penilaian

Pada Gambar 17 merupakan hasil dari pengujian kecepatan *scripting* yang sudah dilakukan antara Blade Template dan Vue Js memiliki hasil yang sangat berbeda jauh. Kecepatan *scripting* yang diperlukan oleh Blade Template adalah 355 ms, sedangkan untuk Vue Js kecepatan *scripting* yang diperlukan adalah 66670 ms. Terlihat bahwa Blade Template memiliki kemampuan *scripting* yang jauh lebih cepat dibandingkan dengan Vue Js pada kasus yang sama.

V. KESIMPULAN

Template Setelah melakukan perancangan, analisis, implementasi, dan pengujian pada aplikasi yang sudah dibuat, dapat diambil kesimpulan bahwa dalam penelitian ini untuk memangun aplikasi sederhana sebaiknya menggunakan Blade Template, akan tetapi jika aplikasi tersebut kompleks dapat menggunakan Vue Js. Selain itu berikut ini merupakan simpulan yang didapatkan selama melakukan penelitian:

1. Pengujian performa menggunakan yaitu Audits panel yang ada pada Chrome DevTools. Panel ini dapat digunakan untuk mengukur performa setiap halaman *website* meskipun, *website* itu tidak di *hosting* pada server *online*. Pada Blade Template diperlukan perintah *php artisan serve*, dan pada Vue Js menggunakan perintah *npm run build* atau *yarn run build* supaya aplikasi yang dibuat dapat diuji performanya menggunakan Audits panel.
2. Kelebihan menggunakan Blade Template adalah performanya yang cepat dapat dilihat dari hasil pengujian yang ada pada Pengujian Performa, akan tetapi Blade Template memiliki kelemahan yaitu semakin kompleks aplikasi tersebut akan membuat performa pada Blade Template ikut menurun.
3. Kelebihan menggunakan Vue Js adalah performanya yang stabil meskipun aplikasi tersebut kompleks dapat dilihat dari hasil pengujian yang ada pada Pengujian Performa, akan tetapi Vue Js memiliki kelemahan yaitu pada aplikasi yang sederhana performa Vue Js kalah jika dibandingkan dengan Blade Template.
4. Kelebihan melakukan kombinasi Blade Template dan Vue Js adalah Blade Template dapat membuat sebuah aplikasi yang kompleks juga, akan tetapi jika dilihat dari hasil Pengujian Performa, performa yang didapatkan menurun drastis.
5. Menggunakan gabungan antara Blade Template dan Vue Js merupakan pilihan terakhir jika sudah terlanjur membuat suatu halaman *website* menggunakan Blade Template dan ingin menggantinya ke Vue Js karena *website* sudah semakin kompleks dan susah untuk ditangani menggunakan Blade Template lagi.
6. Membuat beberapa contoh aplikasi sederhana hingga kompleks yang memanfaatkan fitur-fitur pada *framework*, dengan begitu dapat mengetahui *framework* mana yang lebih mempermudah dalam mengembangkan sebuah aplikasi. Setelah itu melakukan pengujian performa dari setiap aplikasi yang sudah dibuat pada *framework* tersebut. Setelah itu barulah dapat mendapatkan kesimpulan dalam penelitian ini.

DAFTAR PUSTAKA

- [1] W. R. D. Septian, "Analisis Perbandingan Framework PHP Berdasarkan MOOSE CK dan Properti Kualitas Desain Menggunakan Metode Analytic Hierarchy Process (AHP)," Universitas Islam Negeri Syarif Hidayatullah, Jakarta, 2010.

- [2] M. N. Hamid, "Analisis Perbandingan Framework CodeIgniter dan Framework Laravel (Studi Kasus Inventaris HMJ TI STMIK AKAKOM Yogyakarta)," Sekolah Tinggi Manajemen Informatika dan Komputer AKAKOM, Yogyakarta, 2019.
- [3] C. W. Widianarko, "Studi Perbandingan Website View Framework (Vue Js) Dengan Vanilla Js," Sekolah Tinggi Manajemen Informatika dan Komputer AKAKOM, Yogyakarta, 2018.
- [4] R. Erinton, R. M. Negara dan D. D. Sanjoyo, "Analisis Performasi Framework CodeIgniter dan Laravel," *e-Proceeding of Engineering*, vol. 4, no. 3, pp. 3565-3571, 2017.
- [5] D. Wijonarko dan R. F. Aji, "Perbandingan Phonegap dan React Native Sebagai Framework Pengembangan Aplikasi Mobile," *MISI (Jurnal Manajemen Informatika dan Sistem Informasi)*, vol. 1, no. 2, 2018.
- [6] G. Pratama, *How to be a Web Developer (and Designer) and Getting Paid Online*, Bandung, 2015.
- [7] D. A. Hadi, *Ebook Belajar HTML & CSS Dasar*, 2016.
- [8] A. Muhandian, "Pemrograman Javascript: Langkah Awal Belajar Javascript," *Petani Kode*, 20 September 2018. [Online]. Available: <https://www.petanikode.com/javascript-dasar/>. [Diakses 12 Desember 2019].
- [9] A. Pratama, *PHP Uncover Panduan Belajar PHP untuk Pemula*, Duniailkom, 2016.
- [10] A. Pratama, *Bootstrap Uncover Panduan Belajar Bootstrap*, Duniailkom, 2018.
- [11] I. Syafitri, "Pengertian Framework Beserta Fungsi dan Jenis-Jenis Framework," *Nesabamedia*, 3 September 2019. [Online]. Available: <https://www.nesabamedia.com/pengertian-framework/>. [Diakses 12 Desember 2019].
- [12] Feradhita, "Kelebihan Dan Kekurangan dari Penggunaan Framework," *Logique Digital Indonesia*, 23 April 2019. [Online]. Available: <https://www.logique.co.id/blog/2019/04/23/kelebihan-dan-kekurangan-framework/>. [Diakses 12 Desember 2019].
- [13] Mohsai, "Pengertian, Kelebihan dan Sejarah Framework Laravel," *Mohsai*, 19 Januari 2019. [Online]. Available: <https://mohsai.com/pengertian-kelebihan-sejarah-laravel/>. [Diakses 12 Desember 2019].
- [14] Aminudin, *Cara Efektif Belajar Framework Laravel*, Yogyakarta: Lokomedia, 2015.
- [15] H. Mukhlisin dan M. Azamuddin, *Vue.Js The Progressive JavaScript Framework*, Jakarta, 2018.