

# Pengembangan *Front-end* Manajemen Fasilitas Kesehatan Terintegrasi dengan *Vue.JS*

Sukron Nurtado

Program Studi Teknik Informatika  
Fakultas Teknologi Industri Universitas Islam Indonesia  
D.I.Yogyakarta, Indonesia  
15523253@students.uii.ac.id

Hanson Prihantoro Putro

Program Studi Teknik Informatika  
Fakultas Teknologi Industri Universitas Islam Indonesia  
D.I.Yogyakarta, Indonesia  
hanson@uui.ac.id

*Abstract*—Saat ini keberadaan fasilitas pelayanan kesehatan sudah semakin banyak di lingkungan masyarakat dengan fasilitas yang beragam untuk setiap fasilitas pelayanan kesehatannya, akan tetapi setiap masyarakat sulit untuk tahu akan keberadaan fasilitas pelayanan kesehatan yang ada di lingkungannya dan juga fasilitas apa saja yang disediakan oleh fasilitas pelayanan kesehatan karena kurangnya informasi yang dibutuhkan masyarakat. Jika masyarakat tahu akan keberadaan fasilitas pelayanan kesehatan yang ada di lingkungannya, maka fungsi fasilitas pelayanan kesehatan pun akan lebih maksimal dan lebih maju dalam memberikan pelayanan kesehatan kepada masyarakat sekitarnya. Maka ditawarkan solusi untuk mengembangkan tampilan *front-end* fasilitas pelayanan kesehatan terintegrasi yang menggabungkan setiap fasilitas pelayanan kesehatan ke dalam satu tampilan dengan tujuan agar memudahkan dan membantu masyarakat untuk dapat mencari dan melihat fasilitas pelayanan kesehatan yang berada di sekitarnya. Saat ini dalam mengembangkan suatu tampilan yang rapih dan interaktif sangatlah rumit untuk dilakukan karena belum adanya struktur untuk menempatkan kode *script* yang akan membuat tampilan menjadi lebih interaktif, maka penggunaan *framework* sangat dibutuhkan untuk membuat tampilan lebih interaktif dan memudahkan pengembang dalam membuat tampilan karena sudah disediakan struktur folder untuk memasukan *script-script* yang membuat tampilan web menjadi lebih menarik. Akan tetapi, pemilihan *framework* yang tepat untuk digunakan dalam pengembangan tampilan sangatlah sulit karena sekarang sudah banyak *framework front-end* yang muncul. Menurut hasil riset yang telah dilakukan oleh *stateofjs* pada tahun 2018, *framework front-end* terbaik adalah *Vue.js* dan *Angular*. *Vue.js* dan *Angular* merupakan kriteria terfavorit dalam hal kemudahan pembelajaran. Maka dilakukan pengembangan tampilan *front-end* fasilitas pelayanan kesehatan terintegrasi menggunakan *framework* *Vue.js* dengan menggunakan metode *waterfall*. Pengembangannya dengan tahapan penelitian seperti pengumpulan data, perancangan, implementasi kode, dan pengujian pada *framework*. Tampilan ini terdiri dari beberapa fitur utama yaitu: mencari dokter, mencari fasilitas kesehatan, melihat artikel, melihat antrian, booking, menambahkan data user. Pengujian yang dilakukan pada penelitian ini yaitu pengujian *Performance Times* dan *Size Metrics* dengan membandingkan dua projek sama yang dikembangkan oleh dua *framework* yang berbeda yaitu *Vue.js* dan *Angular* yang dikenal memiliki fitur *powerful*, pengujian dilakukan untuk melihat sejauh mana performa *Vue.js*

dibandingkan dengan *Angular*, dan hasilnya untuk *performed times*, waktu yang didapatkan oleh *Vue.js* lebih cepat dua sampai tiga kali lipat dibandingkan dengan waktu yang didapatkan oleh *Angular*. Hasil pengujian *Size Metrics* menunjukkan bahwa *Vue.js* lebih ringan untuk ukuran *project file* yang digunakan dibandingkan *Angular* yang cukup besar hampir dua kali lipat dari *Vue.js*.

*Kata kunci*— *fasilitas kesehatan, front-end, waterfall, blackbox, website, Vue.js, Angular, Performed times.*

## I. PENDAHULUAN

Klinik merupakan salah satu dari fasilitas pelayanan kesehatan yang menyelenggarakan dan menyediakan pelayanan medis dasar dan atau spesialisik, diselenggarakan oleh lebih dari satu jenis tenaga kesehatan dan dipimpin oleh seorang tenaga medis [1]. Keberadaan fasilitas kesehatan sangatlah penting untuk masyarakat karena memberikan fasilitas dan pelayanan kesehatan untuk masyarakat.

Saat ini keberadaan fasilitas pelayanan kesehatan sudah semakin banyak di lingkungan masyarakat dengan fasilitas yang beragam untuk setiap pelayanan fasilitas kesehatannya, akan tetapi setiap masyarakat belum tentu tahu akan keberadaan fasilitas pelayanan kesehatan yang ada di lingkungannya dan juga fasilitas pelayanan apa saja yang disediakan oleh fasilitas pelayanan kesehatan tersebut. Jika masyarakat tahu akan keberadaan fasilitas pelayanan kesehatan yang ada di lingkungannya, maka fasilitas pelayanan kesehatan pun akan lebih maksimal dan lebih maju dalam memberikan pelayanan kesehatan kepada masyarakat sekitarnya.

Maka ditawarkan solusi untuk mengembangkan tampilan *front-end* fasilitas pelayanan kesehatan terintegrasi yang menggabungkan setiap fasilitas pelayanan kesehatan ke dalam satu tampilan dengan sistem *MedUp* yang terus diperbarui dan dikembangkan menyesuaikan dengan sistem fasilitas kesehatan yang sudah ada maupun belum ada dengan tujuan agar memudahkan dan membantu masyarakat untuk dapat mencari dan melihat fasilitas pelayanan kesehatan apa saja yang berada di sekitarnya beserta fasilitas-fasilitas dan juga dokter-dokter yang ada pada setiap fasilitas pelayanan kesehatan. Pada tampilan *website* juga ditawarkan pelayanan untuk melakukan

booking secara online dengan tujuan efektivitas waktu pasien agar tidak perlu datang lebih awal ke fasilitas pelayanan kesehatan tertentu untuk melakukan pengobatan.

Dalam pengembangan tampilan, penggunaan *framework* dibutuhkan untuk membuat tampilan terintegrasi fasilitas kesehatan. *Framework* didefinisikan sebagai 'struktur' aplikasi, yang di dalamnya pengembang dapat menyesuaikan fitur dan karakteristik kode, *framework* memungkinkan untuk 'mengambil kendali' dengan memanggil metode dan fungsi sesuai dengan bagaimana kode telah ditulis [2].

Menurut hasil riset yang telah dilakukan oleh stateofjs pada tahun 2018, daftar *framework* front-end terbaik adalah Vue.js, Angular, React.js, Ember.js, dan Meteor.js dengan keunggulan yang berbeda-beda untuk setiap *framework*nya [3]. Vue.js dipilih untuk mengembangkan tampilan fasilitas pelayanan kesehatan integrasi ini karena aspek dalam kemudahan pembelajaran sangat tinggi dan disukai oleh pengguna menurut hasil riset yang dilakukan stateofjs. Akan tetapi terdapat permasalahan mengenai pemilihan Vue.js apakah *framework* ini optimal dibandingkan dengan *framework* lainnya sehingga menjadi pilihan yang tepat atau hanya memiliki keunggulan dalam kemudahan pembelajaran tetapi performanya buruk.

Maka ditawarkan juga solusi untuk dapat menganalisis serta membandingkan antara *framework* Vue.js dengan *framework* Angular8 yaitu versi terbaru dari Angular yang memiliki aspek fitur yang powerful menurut hasil riset dengan tujuan untuk membuktikan penggunaan Vue.js tidak hanya mudah dalam pembelajarannya saja tetapi juga kuat untuk performanya. Perbandingan yang dilakukan dibagi menjadi lima jenis berdasarkan yaitu *download results*, *performed time*, *size metrics*, *most popularity*, dan *build times* dengan tujuan untuk membandingkan sejauh mana performa Vue.js dengan *framework* Angular.

## II. KAJIAN PUSTAKA

### A. *Supporting Web Development Decisions by Comparing Three Major Javascript Frameworks: Angular, React and Vue.js*

Pada penelitian ini dilakukan oleh Eric Wohlgethan, eric melakukan analisis *framework javascript* berdasarkan data yang ada pada internet, analisis tersebut dibagi menjadi beberapa bagian, seperti perbandingan *download* terbanyak dan perbandingan *framework* yang disukai pada setiap *framework*. Hasilnya *Vue.js* menempati posisi kedua setelah *React* dan *Angular* diposisi terakhir *Vue.js* [4].

### B. *Evaluation of Front-end Javascript Frameworks for Master Data Management Application Development*

Pada penelitian ini dilakukan oleh Jaako Voutilainen, peneliti melakukan analisis *framework javascript* berdasarkan data yang ada pada internet, analisis tersebut dibagi menjadi beberapa jenis seperti, *build times* dan *Popularity*. *Vue.js* unggul dalam kemudahan adaptasi dan performanya walaupun jaraknya tidak terlalu curam karena sebagai *framework* terbaru dibandingkan dengan *react* dan *Angular* [5].

### C. *Modern Javascript frameworks: A Survei Study*

Pada penelitian ini dilakukan oleh Sanja Delčev dan Dražen Drašković, peneliti melakukan analisis *framework Angular, Backbone, Ember.js, dan Knockout* berdasarkan data pada internet dan dilakukan survei kepada beberapa orang untuk diambil sebuah keputusan analisis, analisis dibagi menjadi berbagai jenis, yaitu: *download results*, *architecture*, dan *technical possibilities*. Hasilnya *Angular* lebih unggul daripada *backbone, ember.js, dan knockout* [6].

### D. *COMPARISON OF FRONT-END FRAMEWORKS FOR WEB APPLICATIONS DEVELOPMENT*

Pada penelitian ini dilakukan oleh Marin Kaluža, Krešimir Troskot, dan Bernard Vukelić, peneliti melakukan analisis *framework* berdasarkan data pada internet seperti skala popularitas yang paling terkenal dari *javascript framework* yang dibandingkan. Hasilnya *Vue.js* lebih unggul dibandingkan dengan *Angular* dan *react* dengan poin kemenangan sebanyak 14 poin untuk *Vue.js, react* 10 poin, dan *Angular* 9 poin [7].

### E. *Benchmarking Javascript Frameworks*

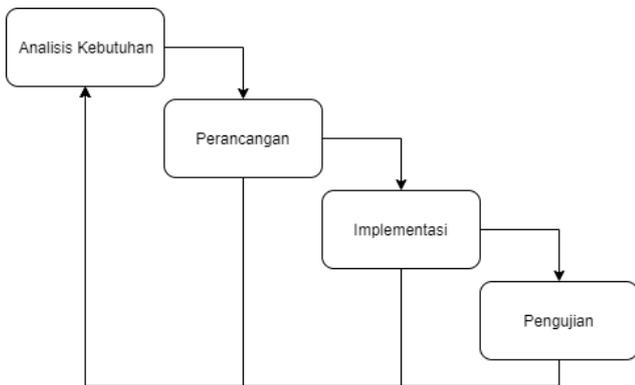
Pada penelitian ini dilakukan oleh Carl Lawrence Mariano, peneliti melakukan analisis *framework* berdasarkan satu *program* yang telah dibuat dengan versi *framework javascript* yang berbeda-beda, analisis tersebut dibagi menjadi beberapa jenis seperti, *line of code* dan *performed time*. *Angular* unggul jika dibandingkan dengan *backbone* [8].

## III. METODOLOGI PENELITIAN

### A. *Metode Waterfall*

Dalam membuat atau mengembangkan suatu perangkat lunak dibutuhkan adanya suatu metode dalam pengembangannya. Menurut Budi, banyak metode yang dapat digunakan dalam mengembangkan perangkat lunak saat ini, empat diantaranya yaitu metode *prototyping*, metode *RAD*, metode *spiral*, dan metode *waterfall* [9].

Pengembangan perangkat lunak pada penelitian tampilan *front-end* manajemen fasilitas kesehatan terintegrasi menggunakan metode *waterfall*. Metode *waterfall* adalah metode untuk pengembangan perangkat lunak dengan pendekatan dengan aktivitas-aktivitas seperti analisis kebutuhan perangkat lunak, perancangan, pembuatan kode atau implementasi kode, dan pengujian [9]. Tahapan aktivitas dari metode *waterfall* menurut Somerville dapat dilihat pada Gambar 3 [10].



**Gambar 1** Metode Waterfall

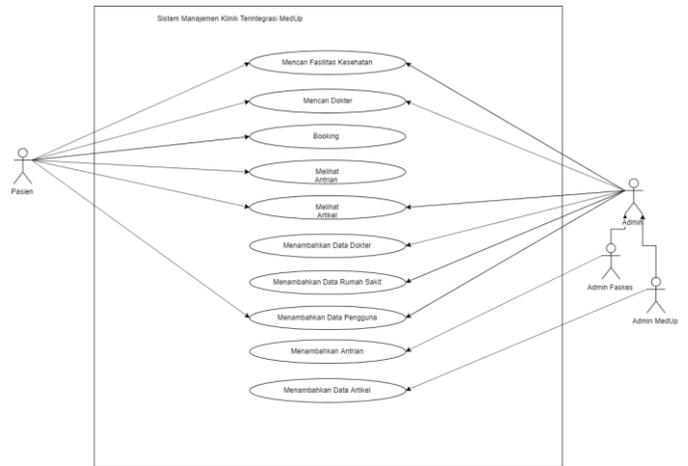
Berdasarkan Gambar 1 menurut Sommerville metode *waterfall* memiliki tahapan-tahapan sebagai berikut [10]:

- a. Analisis Kebutuhan  
Tahapan ini merupakan tahapan awal dalam metode *waterfall*. Tahap ini menganalisis semua kebutuhan-kebutuhan dalam mengembangkan perangkat lunak seperti kebutuhan pengguna dan kebutuhan sistem.
- b. Perancangan  
Tahapan perancangan ini merupakan tahap untuk menggambarkan kebutuhan-kebutuhan sistem yang telah dilakukan dengan membentuk arsitektur secara keseluruhan. Arsitektur sistem dapat berupa penggambaran menggunakan *use case diagram* dan *activity diagram*.
- c. Implementasi  
Hasil dari perancangan perangkat lunak yang telah dibuat akan direalisasikan sebagai serangkaian *program* atau unit *program*.
- d. Pengujian  
Pengujian digunakan untuk memastikan apakah program yang sudah dibuat sesuai atau belum dengan kebutuhan awal perangkat lunak.

#### IV. PERANCANGAN

##### A. Design Use Case

*Use Case diagram* merupakan diagram yang berfungsi untuk menggambarkan siapa saja stakeholder yang terlibat dalam penggunaan tampilan sistem klinik terintegrasi ini, serta apa saja yang bisa dilakukan pada setiap role stakeholder yang ada pada tampilan sistem klinik terintegrasi. *Use case diagram* merupakan bagian dari *Unified Modeling Language (UML)* yaitu *tool* yang digunakan untuk melakukan suatu proses abstraksi terhadap sistem yang berbasis pada objek. Desain *Use case* dapat dilihat pada Gambar 2.



**Gambar 2** Desain Use Case

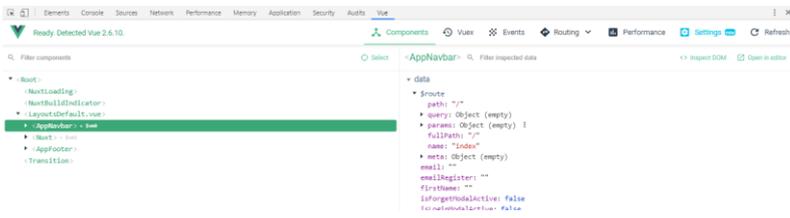
Pada Gambar merupakan gambaran *use case* dari sistem informasi klinik terintegrasi, aktor dari *use case diagram* terdiri dari 3 aktor yaitu pasien, Admin MedUp, dan Admin Klinik. Semua Usecase harus login terlebih dahulu kecuali Usecase mencari rumah sakit, mencari dokter, dan Lihat Artikel. Untuk aktivitas *edit* dan *delete* tetap ada namun tidak ditampilkan.

##### B. Proses Bisnis

Tampilan pada website fasilitas kesehatan tidak akan terlepas dari alur bisnisnya, alur bisnis adalah serangkaian kegiatan yang bekerja dalam lingkungan organisasi dan teknis yang mencapai tujuan bisnis [11]. Alur bisnis berfungsi sebagai gambaran secara umum bagaimana tampilan sistem akan bekerja. Alur bisnis juga berfungsi untuk menerjemahkan alur-alur yang ada pada sistem secara detail agar lebih mudah untuk dipahami orang luar.

Pada tampilan sistem fasilitas kesehatan terintegrasi dengan studi kasus MedUp, tampilan sistem memiliki dua alur bisnis yaitu booking online dan lihat antrian. Pada alur proses bisnis, booking online dilakukan oleh pasien dengan cara pasien melakukan pencarian nama dokter atau spesialis di dalam field cari nama dokter, setelah itu pasien memilih dokter berdasarkan kata kunci yang sudah dimasukkan ke dalam field pencarian, setelah itu pasien memilih rumah sakit atau klinik untuk dapat membuat janji, jika sudah memilih klinik dan masuk ke dalam fitur buat janji, pasien memilih jadwal yang sudah disediakan sistem MedUp dan menuliskan keluhannya. Jika sudah pasien harus mengkonfirmasi booking untuk menjadi antrian dan mendapat nomor antriannya. Cara kedua pasien melakukan pencarian nama rumah sakit atau klinik di dalam field cari nama rumah sakit. Setelah itu pasien memilih rumah sakit berdasarkan pilihan-pilihan rumah sakit yang tersedia berdasarkan kata kunci yang sudah dimasukkan sebelumnya. Setelah itu pasien memilih dokter yang tersedia dari rumah sakit tersebut. Setelah memilih dokter, selanjutnya memilih jadwal dokter tersebut dan menuliskan keluhan yang dialami pasien. Jika sudah, pasien harus mengkonfirmasi booking dan antrian sama seperti cara pertama. Selanjutnya adalah lihat antrian, pada alur bisnis ini pasien bisa melihat daftar penjanjian yang telah dilakukan pada halaman profil pasien, akan ada tiga status berbeda untuk jenis perjanjian yang ada pada daftar penjanjian yaitu BOOKED, CHECKIN,

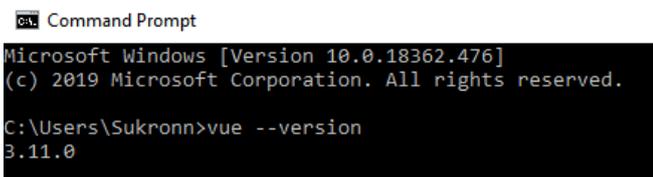




**Gambar 6** Tampilan *Vue Devtools*

Setelah selesai mempersiapkan hal – hal yang dibutuhkan untuk menginstall vue.js, maka kita bisa langsung menginstall vue.js. ada beberapa cara untuk menginstall vue js seperti yang dijelaskan pada <https://v1.vuejs.org/guide/installation.html>. Akan tetapi, yang akan dijelaskan pada penelitian kali ini adalah cara install dengan cara *CLI* atau *Command Line Interface*.

*Vue.js* menyediakan *CLI* resmi yang berisi paket untuk alur kerja frontend modern. Cara menginstall vue menggunakan *CLI* dengan cara membuka command prompt atau terminal yang sudah ada *Node.JS* dan *NPM* didalamnya lalu ketik “*npm install –g @vue-cli*”, “*npm install*” berarti klien baris perintah dari *npm* untuk menginstall suatu framework atau library, “*-g*” yang berarti global yaitu menginstall suatu framework atau library secara menyeluruh, dan “*vue-cli*” yang berarti *CLI* resmi dari *Vue* untuk menginstall *Vue.js*. Lalu tunggu beberapa menit sampai install vue selesai, dan cek apakah vue sudah terpasang pada *cli* dengan mengetik “*vue –version*”. Berikut contoh gambar pengecekan versi *Vue.js*:



**Gambar 7** Cek *Vue Version*

Setelah melakukan install, hal selanjutnya yang harus diketahui adalah cara untuk membuat project atau product dengan menggunakan *vue.js*. Cara membuat project atau product menggunakan *vue.js* dengan mengetikkan “*vue create nama\_project*” pada terminal yang sudah terinstall *vue.js* dan ikuti proses instalasi sampai selesai.

### B. Hasil Implementasi

Pada pengembangannya, tampilan website medup digunakan untuk user pasien sesuai dengan *usecase* seperti mencari dokter, mencari fasilitas kesehatan, booking, lihat antrian, menambah data pengguna, dan melihat artikel.

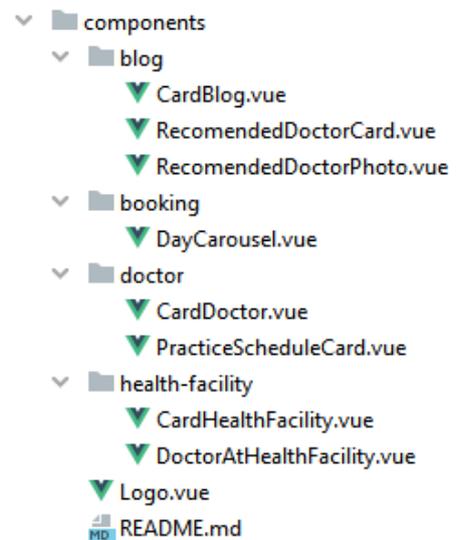
Sebelumnya, akan ditampilkan struktur file dan component code yang sudah dibuat pada tampilan website MedUp dan fokus pembahasan peneliti hanya pada setiap *usecase* yang sudah dianalisis sebelumnya. Struktur file dapat dilihat pada Gambar 8 dan Component File dapat dilihat pada Gambar 9.



**Gambar 8** Struktur File

Pada Gambar 8 merupakan struktur file dari tampilan website MedUp, *usecase diagram* pada struktur file terdiri ini terdiri dari enam *usecase* yaitu mencari dokter, mencari fasilitas kesehatan, booking, lihat antrian, dan lihat artikel. Untuk *usecase* booking dan lihat antrian diharuskan untuk login terlebih dahulu kecuali *usecase* mencari rumah sakit, mencari dokter, dan lihat artikel.

Dan selain terdapat struktur file website MedUp, peneliti juga membuat struktur tampilan component yang terhubung ke dalam struktur file, untuk lebih detailnya dapat dilihat pada Gambar 9.



**Gambar 9** Component File

#### 1) Halaman Mencari Fasilitas Kesehatan

Halaman Mencari Fasilitas Kesehatan adalah halaman untuk mencari fasilitas kesehatan berdasarkan nama dan lokasi lalu diarahkan kepada halaman list hasil pencarian yang disesuaikan dengan *usecase* dan *diagram acitivity* agar alur mudah dibaca dan dimengerti. Untuk masuk ke halaman ini

tidak harus login terlebih dahulu jika tujuannya hanya mencari informasi fasilitas kesehatan saja.

Tampilan yang dibuat Pada Gambar 10 adalah tampilan untuk bagian *header*, side filter search, dan template card fasilitas kesehatan. Dan component file *CardHealthFacility.vue* yang terdapat pada folder *health-facility* adalah untuk tampilan untuk bagian dari setiap card yang dipanggil oleh file *index.vue* dan dimasukkan ke dalam bagian template card fasilitas kesehatan. Hasil tampilan halaman mencari fasilitas kesehatan terlihat pada Gambar 10.



**Gambar 10** Tampilan Halaman Mencari Faskes

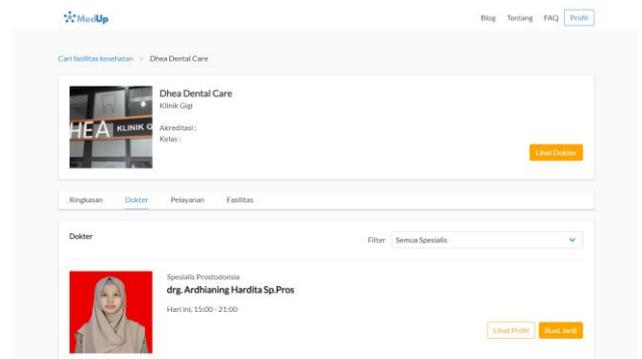
Component Mencari Fasilitas Kesehatan			
No	Use case	File Implementasi	Deskripsi
1.	Mencari Fasilitas Kesehatan	<i>health-facility/ CardHealthFacility.vue</i>	Folder <i>health-facility</i> ditujukan untuk <i>component</i> tampilan fasilitas kesehatan. Pada file <i>CardHealthFacility.vue</i> ditujukan untuk tampilan <i>card</i> pada tampilan <i>list-list</i> fasilitas kesehatan.
2	Melihat Profil Fasilitas Kesehatan	<i>Health-facility/ DoctorAtHealthFacility.vue</i>	Folder <i>health-facility</i> ditujukan untuk <i>component</i> tampilan fasilitas kesehatan. Pada file <i>DoctorAtHealthFacility.vue</i> ditujukan untuk tampilan <i>tab</i> dokter pada tampilan <i>profil</i> fasilitas kesehatan.

**Tabel 1** File dan Component Halaman Mencari Faskes

## 2) Halaman Profile Fasilitas Kesehatan

Halaman Profile Fasilitas Kesehatan adalah halaman yang berisi informasi-informasi tentang fasilitas kesehatan terkait seperti informasi umum atau ringkasan, fasilitas, dokter, dan pelayanan. Halaman profile fasilitas kesehatan juga adalah alur lanjutan dari halaman mencari fasilitas yang disesuaikan dengan *usecase* dan *diagram acitivity* agar alur mudah dibaca dan dimengerti.

File *\_id.vue* yang terdapat pada folder *search-health-facility*. Tampilan yang dibuat Pada Gambar 11 adalah tampilan yang berisi informasi informasi yang dibutuhkan pasien seperti nama, foto, peta, dokter, fasilitas, asuransi, dll yang ada pada profile fasilitas kesehatan. Dan component file *DoctorAtHealthFacility.vue* yang terdapat pada folder *health-facility* adalah untuk tampilan bagian setiap card dokter yang dipanggil oleh file *\_id.vue* dan dimasukkan ke dalam bagian tab *doctor* di dalam profile fasilitas kesehatan. Hasil tampilan halaman profile fasilitas kesehatan terlihat pada Gambar 11.



**Gambar 11** Halaman Profil Faskes

## C. Performance Times

Pada *Performance Times*, terdapat batasan yaitu hanya menampilkan dua usecase saja, yaitu mencari dokter dan mencari rumah sakit. Adapun evaluasi halaman adalah untuk melihat performance test yang dihasilkan dari masing – masing framework. Tools yang digunakan untuk melihat performance website adalah *webpagetest*. [14].

Sebelum melakukan evaluasi, peneliti menghosting website MedUp menggunakan *Vue.js* dan website MedUp menggunakan *Angular* pada *netlify*, *Netlify* adalah platform web developer yang menggandakan produktivitas [15].

Hasil dari hostingan menggunakan *netlify* adalah sebagai berikut:

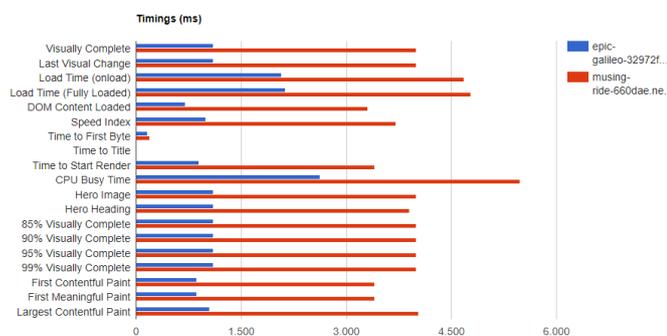
<https://musing-ride-660dae.netlify.com> adalah duplikat website MedUp menggunakan *angular 8*

<https://epic-galileo-32972f.netlify.com> adalah duplikat website MedUp menggunakan *Vue.js*

Berikut evaluasi perbandingan dari tampilan yang sudah di buat pada duplikat website menggunakan *angular 8* dan *vue.js*.

Pada tampilan utama berisi input search dokter atau rumah sakit yang akan diarahkan ke halaman list dokter ataupun fasilitas kesehatan. Selain input search dokter atau fasilitas kesehatan, terdapat informasi-informasi mengenai MedUp. Tampilan akan *visual progress* nya. Berikut hasil dari tampilan utama website:

Untuk hasil dari tampilan utama website menggunakan *vue.js*, *visual progress* (warna biru) untuk dapat menampilkan halaman utama secara penuh adalah 0.9 – 1.6 detik. Sedangkan hasil dari tampilan utama website menggunakan *angular 8*, *visual progress* (warna merah) untuk dapat menampilkan halaman utama secara penuh adalah 3.4 – 4.8 detik. Adapun grafik detail dari *visual progress* dapat dilihat pada Gambar 12.



Gambar 12 Grafik detail visual progress

Tabel 2 Detail Request

Request	Vue	Angular
HTML	2	2
JS	5	6
CSS	5	3
Image	12	12
Font	3	2
Total	27	25

Adapun untuk detail request yang terjadi pada kedua framework dapat dilihat pada Tabel 2. Hampir setiap kategori vue.js jauh lebih cepat dibandingkan angular 8, padahal untuk total request vue.js lebih banyak dibandingkan dengan angular 8.

Dengan demikian dapat disimpulkan untuk visual progress tampilan utama, duplikat website menggunakan vue.js jauh lebih cepat yaitu 0.9–1.6 detik dibandingkan menggunakan angular 8 yaitu 3.4–4.8 detik. Walaupun total untuk request vue lebih banyak yaitu 27 requests dibandingkan dengan angular 8 yaitu 25 requests.

## KESIMPULAN DAN SARAN

### A. Kesimpulan

Setelah dilakukan penelitian, perancangan, implementasi dan pengujian pada tugas akhir ini, maka didapatkan kesimpulan sebagai berikut:

- Frontend website MedUp beserta Dashboard Admin telah dikembangkan dengan mempelajari penelitian sejenis yang telah dilakukan dan metode waterfall. Setelah mempelajari penelitian sejenis, peneliti merangkum konten apa saja yang perlu dimunculkan melalui tahap analisis dan perancangan seperti analisis proses bisnis yang berisi alur proses dari website yang dibuat, analisis kebutuhan dibagi menjadi tiga bagian yaitu kebutuhan fungsional, kebutuhan non-fungsional, dan kebutuhan pengguna, dan lain-lain. Setelah tahap analisis dan perancangan dilakukan, tahapan implementasi code dilakukan berdasarkan perancangan

tampilan website yang sudah dilakukan dan pengujian download results, performed time, size metrics, most popularity, dan build times dilakukan berdasarkan penelitian sejenis dan analisis yang telah dilakukan untuk mengetahui sejauh mana performa Vue.js yang digunakan untuk membuat tampilan dengan Angular8 yang diketahui memiliki fitur yang powerful.

- Telah dilakukan perbandingan performance time website antara Vue.js dan Angular8 dan hasilnya Vue.js lebih baik dibandingkan Angular8 berdasarkan performance times yaitu Vue.js lebih cepat dua sampai tiga kali lipat dari Angular8 dalam memuat suatu tampilan yang sama.

### B. Saran

Berdasarkan implementasi dan pengujian terhadap tampilan website MedUp, website tersebut masih memiliki kekurangan. Sehingga perlu dilakukan pengembangan lebih lanjut dengan beberapa saran dan masukan antara lain :

- Penambahan Fitur Obat, Rekam Medis, dan Konsultasi Online.
- Membuat tampilan menjadi lebih responsif dan menarik di segala ukuran layar.
- Membuat tampilan *review* dan *feedback*
- Melakukan perbandingan dengan framework javascript lain seperti react.js, angular.js, dll.

## REFERENCES

- [1] Kemenkes, "Peraturan Menteri Kesehatan Republik Indonesia Nomor 9 Tahun 2014 Tentang Klinik," vol. Nomor 65, no. 879, hal. 2004–2006, 2014.
- [2] T. Henson, "JavaScript : an analysis of the language and its impact on web development .," 2017.
- [3] Stateofjs, "State of JS 2018," 2018. [Daring]. Tersedia pada: <https://2018.stateofjs.com/>. [Diakses: 28-Nov-2019].
- [4] E. Wohlgethan, "Bachelorarbeit Comparing Three Major JavaScript Frameworks :," 2018.
- [5] J. Voutilainen, "Evaluation of Front-end JavaScript Frameworks for Master Data Management Application Development," no. December, 2017.
- [6] S. Delčev dan D. Drašković, "Modern JavaScript frameworks: A Survey Study," 2018 *Zooming Innov. Consum. Technol. Conf.*, hal. 106–109, 2018.
- [7] K. Troskot, "COMPARISON OF FRONT-END FRAMEWORKS FOR WEB APPLICATIONS DEVELOPMENT 4," vol. 6, no. 1, hal. 261–282, 2018.
- [8] C. L. Mariano, "Benchmarking JavaScript Frameworks Benchmarking JavaScript Frameworks," 2017.

- [9] D. S. Budi, "Analisis Pemilihan Penerapan Proyek Metodologi Pengembangan Rekayasa Perangkat Lunak," vol. 5, no. November, hal. 24–31, 2016.
- [10] Sommerville, *Alur Metode Waterfall*. 2001.
- [11] A. T. Helmi, I. Aknuranda, dan M. C. Saputra, "Analisis Dan Pemodelan Proses Bisnis Menggunakan Business Process Improvement ( BPI ) Pada Lembaga Bimbingan Belajar ( Studi Kasus : Lembaga Bimbingan Belajar Prisma )," vol. 2, no. 10, 2018.
- [12] Node.js, "About Node.js." [Daring]. Tersedia pada: <https://nodejs.org/en/about/>.
- [13] Node.js, "What is npm?," 2011. [Daring]. Tersedia pada: <https://nodejs.org/en/knowledge/getting-started/npm/what-is-npm/>.
- [14] Webpagetest, "Web Page Test," 2019. [Daring]. Tersedia pada: [webpagetest.org](http://webpagetest.org). [Diakses: 28-Nov-2019].
- [15] Netlify, "About Netlify." [Daring]. Tersedia pada: <https://www.netlify.com/about/>.