

BAB II

LANDASAN TEORI

2.1 Fasilitas Kesehatan

Menurut peraturan menteri kesehatan Republik Indonesia nomor 47 tahun 2016 bahwa fasilitas kesehatan merupakan suatu alat dan/atau tempat yang digunakan untuk menyelenggarakan upaya pelayanan kesehatan, baik promotif, preventif, kuratif maupun rehabilitatif yang dilakukan oleh pemerintah, pemerintah daerah, dan/atau masyarakat (Pemerintah Republik Indonesia, 2016).

Jenis fasilitas kesehatan terdiri atas (Pemerintah Republik Indonesia, 2016):

- a. Tempat praktik mandiri tenaga kesehatan.
- b. Pusat kesehatan masyarakat.
- c. Klinik.
- d. Rumah sakit.
- e. Apotek.
- f. Unit transfusi darah.
- g. Laboratorium kesehatan.
- h. Optik.
- i. Fasilitas pelayanan kedokteran untuk kepentingan hukum dan
- j. Fasilitas pelayanan kesehatan tradisional.

2.2 Sistem Informasi

Menurut Henry C Lucas yang diterjemahkan oleh Jugianto H.M, Sistem Informasi adalah suatu kegiatan yang terdiri dari prosedur–prosedur yang diorganisasikan. Ketika prosedur–prosedur itu dieksekusi, sistem informasi akan menyediakan informasi untuk mendukung pengambilan keputusan dan pengendalian di dalam organisasi (Astuti, 2011).

Menurut Sutabri, sistem informasi adalah suatu sistem di dalam organisasi yang menyatukan kebutuhan pengolahan transaksi harian dengan kegiatan strategi untuk dapat menyediakan laporan–laporan yang diperlukan kepada pihak luar (Sutabri, 2012).

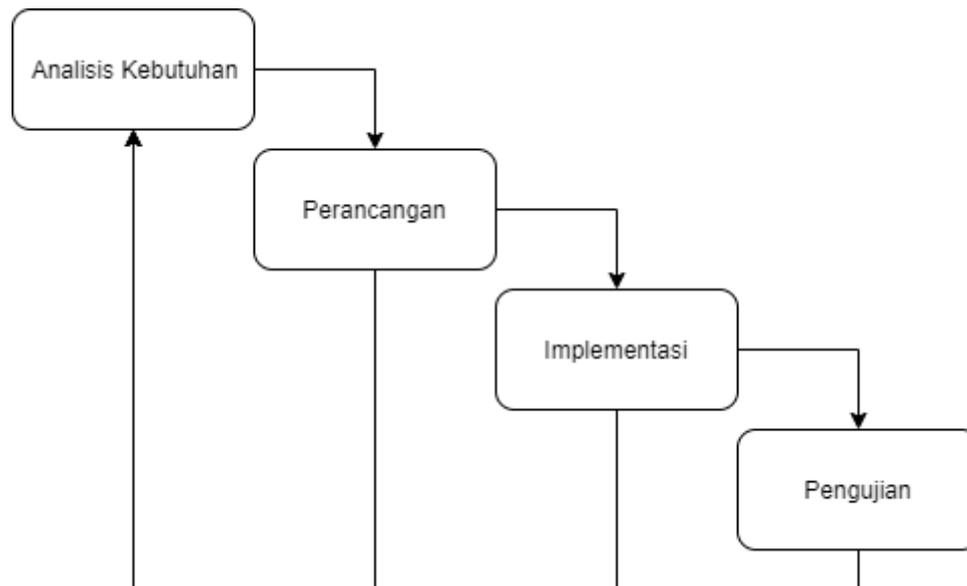
2.3 Sistem Informasi Kesehatan

Sistem informasi kesehatan merupakan salah satu bagian penting yang tidak dapat dipisahkan dari sistem kesehatan di dalam suatu negara. Kemajuan atau kemunduran sistem informasi kesehatan dapat mempengaruhi sistem pemerintahan kesehatan suatu negara. Jika suatu sistem dapat terkonsep dan terstruktur dengan baik. Maka akan menghasilkan *output* yang baik juga. Sistem informasi kesehatan merupakan salah satu bentuk pokok Sistem Kesehatan Nasional (SKN) yang dipergunakan sebagai dasar dan acuan dalam penyusunan berbagai kebijakan, pedoman dan arahan penyelenggaraan pembangunan kesehatan serta pembangunan berwawasan kesehatan (Kompasiana, 2015).

2.4 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 *protoyping*, metode *RAD*, metode *spiral*, dan metode *waterfall* (Budi, 2016).

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 (Budi, 2016). Tahapan aktivitas dari metode *waterfall* menurut Sommerville dapat dilihat pada Gambar 2.1 (Sommerville, 2009).



Gambar 2.1 Tahapan Metode *Waterfall*

Berdasarkan Gambar 2.1. menurut Sommerville metode *waterfall* memiliki tahapan-tahapan sebagai berikut (Sommerville, 2009):

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.

2.5 Framework Javascript

Framework javascript adalah *framework* aplikasi yang ditulis dalam bahasa pemrograman *javascript* dan dalam arah kontrolnya berbeda dengan *javascript library* seperti:

- a. *Library* menawarkan fungsi untuk dapat dipanggil oleh kode induknya, sedangkan *framework* mendefinisikan keseluruhan desain aplikasi.
- b. Pengembang tidak memanggil *framework*, tetapi *framework* yang memanggil dan menggunakan kode dengan cara tertentu.

Bahasa pemrograman yang paling populer untuk *browser* saat ini adalah *javascript*. *Framework javascript* telah dikembangkan untuk memfasilitasi pekerjaan *programmer web*. Bagi *programmer*, sangat penting untuk memilih *framework* yang lebih sesuai dengan kebutuhan mereka. Ada beberapa faktor yang dapat mempengaruhi pilihan *programmer web* seperti perawatan dan komunitas pendukung (Gizas, Christodoulou, & Papatheodorou, 2016).

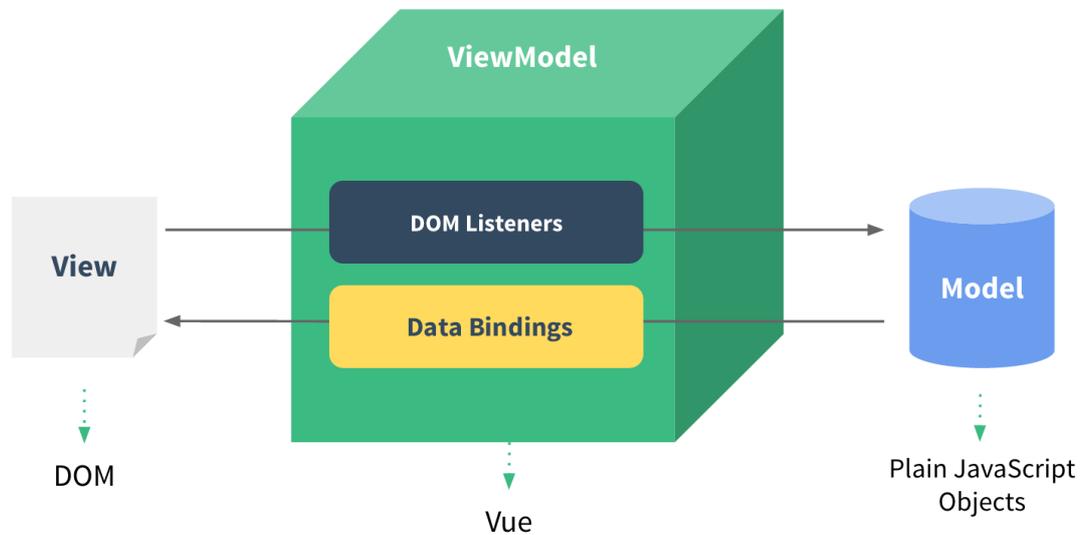
Beberapa *framework javascript* mengikuti paradigma *model-view-controller* yang dirancang untuk memisahkan aplikasi *web* ke dalam unit spesifik untuk meningkatkan kualitas dari pemeliharaan dan pemrogramannya. Contoh *framework* tersebut seperti: *Vue.js* dan *Ember.js* (Yudana, 2018).

2.5.1 *Vue.js*

Vue.js adalah adalah *The Progressive Javascript Framework* yang digunakan untuk membangun tampilan pengguna pada *web* menjadi lebih interaktif. Pertama kali dibuat oleh Evan You dan masih dikembangkan dan dikelola oleh Evan beserta tim inti yang berasal dari berbagai perusahaan seperti Netlify dan Netguru (Riesdiansyah, 2018).

Menurut Sigit, *Vue.js* adalah suatu *library javascript* yang digunakan untuk membangun *interface* sebuah *website* menjadi lebih interaktif, *library* dan *Vue.js* difokuskan hanya pada *view layer* dan mudah untuk diimplementasi dan diintegrasikan dengan *library* lain ataupun dengan projek yang ada sebelumnya (Sigit, 2017).

Inti dari *Vue.js* adalah reaktif *data-binding* sistemnya yang membuatnya menjadi lebih mudah untuk menjaga data dan *Document Object Model (DOM)* agar tetap sinkron. *Vue.js* juga menggunakan konsep *data-drive view* yaitu menggunakan sintaks khusus dalam *template HTML* dengan tujuan untuk “mengikat” *Document Object Model (DOM)* ke data yang mendasarinya. Ketika *binding* telah dibuat, *Document Object Model (DOM)* akan menyinkronkan langsung dengan data. Setelah data disinkronkan, ketika ada perubahan data maka *Document Object Model (DOM)* akan otomatis menyesuaikan pembaruan data tersebut. Hasilnya, lebih besar logika aplikasi yang langsung memanipulasi data dibandingkan dengan mengubah *Document Object Model (DOM)* secara manual. Hal ini juga membuat *programmer* untuk lebih mudah menulis, memahami, dan memperbaiki kodenya.



Gambar 2.2 *Reactive Data Binding (Vue.js, 2019)*

2.5.2 *Angular*

Angular adalah *framework* aplikasi *web open source* yang berbasis *typescript*. *Angular* dikelola oleh Google dan komunitas dari individu pengembang dan juga perusahaan – perusahaan. *Angular* merupakan penulisan ulang secara lengkap dari tim yang sama membangun *Angular.js*.

Versi terbaru dari *Angular* adalah *Angular8* atau *Angular* versi 8 yang dirilis pada tanggal 28 Mei 2019 (*Angular, 2019*). Versi terbaru *Angular* dikabarkan lebih canggih dibandingkan dengan versi sebelum–sebelumnya dan pada *Angular 8* terdapat fitur baru yang tidak kalah canggih yang akan dihargai oleh para pengembang di banyak tingkatan, inti *framework*, *Angular material library*, dan *Command Line Interface (CLI)*.

Pada arsitekturnya, *Angular* dibagi menjadi beberapa bagian, yaitu:

a. *Komponen*

Komponen adalah blok penyusun dari sebuah *user interface*. *Komponen* juga yang mengendalikan *path* pada *screen* atau yang disebut *view* dan setiap *komponen* diarahkan ke sebuah *template*.

b. *Modul*

Angular pada dasarnya bersifat modular dan *Angular* memiliki sistem modularitasnya sendiri yang disebut *NgModule*, setiap aplikasi *Angular* pasti

memiliki minimal satu kelas dengan *NgModule* didalamnya yang biasa disebut *AppModule*.

c. *Template*

Seperti yang dijelaskan pada poin komponen, setiap komponen dilokasikan ke dalam *template*, *template* juga yang memberitahu *Angular* bagaimana caranya untuk me-render komponen ke dalam *screen* atau *view*.

d. *Metadata*

Setiap bagian pada *Angular* seperti *component*, *directive*, *module* atau *service* adalah kelas dasar *typescript*, tetapi *metadata* memberitahu *Angular* bagaimana cara memproses dari kelas tersebut.

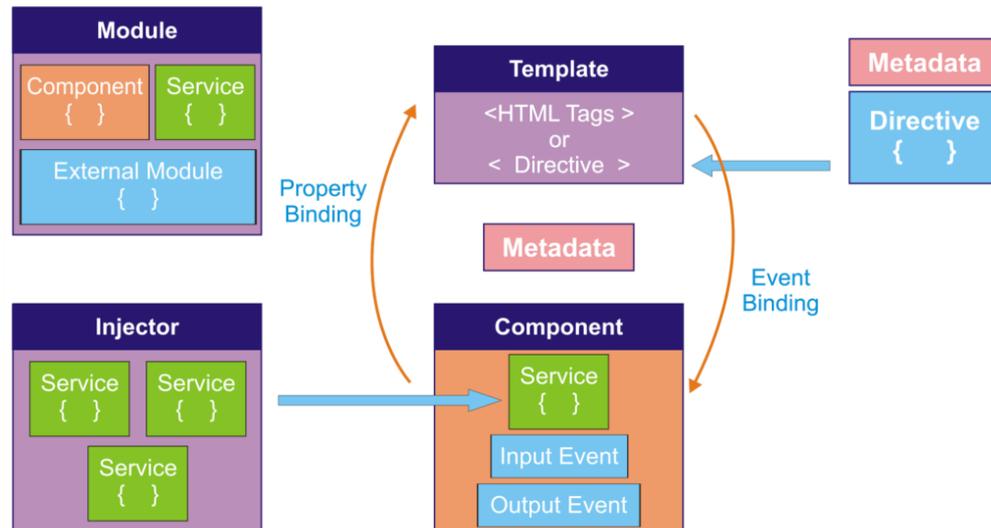
e. *Directive*

Directive memberikan instruksi untuk mengubah *Document Object Model (DOM)* kepada *Angular* ketika membuat sebuah *template*. *Angular* menyediakan dua jenis *directives*, yaitu:

- *Directive Structural* yaitu *directive* yang mengubah suatu struktur *template Document Object Model (DOM)* yang disediakan *Angular*. Contoh *directive structural* pada *Angular* adalah **Ngfor* dan **NgSwitch*.
- *Attribute Structural* yaitu atribut pembaruan yang memperbarui atribut kontrol pada *Hypertext Markup Language (HTML)* tertentu. Contoh atribut struktural pada *Angular* adalah [*NgClass*].

f. *Injector*

Injector menyimpan daftar dari *services* yang akan digunakan pada sistem, kapan pun suatu komponen membutuhkan suatu *services*, maka *injector* akan memberikan instansi kepada komponen tersebut.



Gambar 2.3 Angular Arsitektur (NgDevelop, 2019)

2.6 MedUp

MedUp berdiri sejak 18 Agustus 2018 merupakan sebuah perusahaan digital yang berfokus pada bidang pelayanan kesehatan. MedUp percaya bahwa kesehatan adalah hak dasar setiap manusia. Untuk itu, MedUp mempunyai visi dan misi untuk memberikan solusi perawatan kesehatan dari hulu ke hilir melalui pemanfaatan teknologi dan digital, guna meningkatkan kualitas kesehatan di Indonesia.

MedUp memiliki suatu layanan aplikasi berbasis *android* untuk mengintegrasikan klinik atau rumah sakit yang bekerja sama kedalam aplikasi tersebut dengan tujuan untuk memudahkan *user* dalam mencari klinik atau rumah sakit dan dokter. MedUp juga memudahkan *user* untuk berobat karena *user* dapat melakukan *booking* secara *online* kepada rumah sakit atau klinik yang akan dikunjungi di dalam aplikasi MedUp.

Dan saat ini *platform* MedUp berbasis *web* sedang dikembangkan dengan tujuan yg hampir sama dengan aplikasi berbasis *android* yang sudah dibuat sebelumnya. *Platform* MedUp terintegrasi secara lengkap untuk memenuhi kebutuhan kesehatan masyarakat Indonesia, mulai dari pencarian dokter, informasi lengkap tentang dokter dan fasilitas kesehatan, buat janji dengan dokter, artikel kesehatan, hingga hospital-community engagement bagi pasien, dokter dan pihak rumah sakit (MedUp, 2019).

2.7 Perbandingan Penelitian Sejenis

Berdasarkan hasil penelitian–penelitian dan tampilan *website* yang sudah dikembangkan terdahulu, terdapat persamaan dan perbedaan terhadap penelitian yang dilakukan.

Perbandingan penelitian pun dibagi menjadi dua bagian yaitu perbandingan *website* dan perbandingan *framework*.

2.7.1 Perbandingan Website

Pembuatan tampilan *website* fasilitas kesehatan terintegrasi MedUp selain didasarkan oleh kebutuhan *user*, pembuatan tampilan juga didasarkan pada beberapa *website* terdahulu dan sejenis dengan tujuan *website* MedUp, maka dari itu dilakukan juga perbandingan atas tampilan dan fitur *website* MedUp dengan *website* lain. Berikut beberapa *website* sejenis:

- a. HaloDoc. Pada aplikasi Halodoc Fitur yang disediakan memiliki beberapa kesamaan seperti artikel, cari rumah sakit, cari dokter, buat janji, dan konsultasi dokter (Halodoc, 2019).
- b. ALODOKTER. Pada aplikasi ALODOKTER fitur yang disediakan yaitu artikel, cari dokter, buat janji, dan konsultasi dokter. ALODOKTER hanya fokus kepada dokter saja, tidak dengan rumah sakit atau klinik (Alodokter, 2019).
- c. KelDoc. Pada aplikasi KelDoc fitur yang disediakan yaitu cari dokter, dan buat janji. KelDoc lebih fokus kepada dokter saja, tidak dengan rumah sakit atau klinik dan juga artikel (KelDoc, 2019).
- d. Pliro. Pada aplikasi Pliro fitur yang disediakan memiliki beberapa kesamaan seperti artikel, cari rumah sakit, cari dokter, buat janji (Pliro, 2019).
- e. DokterNet. Pada aplikasi DokterNet fitur yang disediakan memiliki beberapa kesamaan seperti artikel, cari rumah sakit, cari dokter, buat janji (DokterNet, 2019).
- f. MedUp (*Web Tugas Akhir*). Pada aplikasi MedUp fitur yang disediakan memiliki beberapa kesamaan seperti artikel, cari rumah sakit, cari dokter, buat janji, konsultasi dokter, dan *tag recommended* pada artikel.

Beberapa *Website* sejenis diatas dapat dilihat pada Tabel 2.1

Tabel 2.1 Perbandingan *Website* Sejenis

	MedUp	Halodoc(Halodoc, 2019)	ALODOKTER (Alodokter, 2019)	KelDoc(KelDoc, 2019)	Pliro(Pliro, 2019)	DokterNet(DokterNet, 2019)
Aritkel	√	√	√		√	√
Cari Rumah	√	√			√	√

Sakit / Klinik						
Cari Dokter	√	√	√	√	√	√
Buat Janji	√	√	√	√	√	√
Konsultasi Dokter	√	√	√			
Tag dokter recommended pada artikel	√					

2.7.2 Perbandingan *Framework*

Pembuatan tampilan *website* fasilitas kesehatan terintegrasi MedUp selain didasarkan oleh kebutuhan *user*, pembuatan tampilan juga didasarkan pada beberapa *website* terdahulu dan sejenis dengan tujuan pembuatan *website* MedUp. Penggunaan *framework Vue.js* pada fasilitas kesehatan terintegrasi MedUp selain didasarkan dengan kesepakatan antara penulis dengan tim MedUp dan juga hasil riset yang dilakukan oleh *stateofjs* pada tahun 2018 yang menyatakan *Vue.js* menjadi salah satu *framework front-end* terbaik bersama dengan *Angular* dan *react*. *Vue.js* juga menjadi *framework* terbaik dalam kriteria kemudahan pembelajarannya (Stateofjs, 2018), penggunaan *framework Vue.js* juga akan didasarkan pada perbandingan antara *Vue.js* dengan *Angular8* yang dikenal memiliki fitur yang *powerful*.

Perbandingan ini dilakukan dengan membandingkan data yang sudah ada di internet dan penelitian sejenis mengenai hal-hal yang dibutuhkan untuk dapat dibandingkan pada setiap *framework* yang dibandingkan dan juga pada *website* yang peneliti buat dengan *framework* yang berbeda. Berikut beberapa studi literatur sejenis mengenai analisis perbandingan *framework*, yaitu:

- a. Eric Wohlgethan dengan judul “*Supporting Web Development Decisions by Comparing Three Major Javascript Frameworks: Angular, React and Vue.js*“. Pada penelitian ini, peneliti 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. Hasilnya *Vue.js* menempati posisi kedua setelah *react* dan *Angular* diposisi terakhir (Wohlgethan, 2018).

- b. Jaakko Voutilainen dengan judul “*Evaluation of Front-end Javascript Frameworks for Master Data Management Application Development*”. Pada penelitian ini, 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* (Voutilainen, 2017).
- c. Sanja Delčev, Dražen Drašković dengan judul “*Modern Javascript frameworks: A Survei Study*”. Pada penelitian ini, peneliti melakukan analisis *framework* berdasarkan data pada internet, analisis dibagi menjadi berbagai jenis, yaitu: *download results*, *architecture*, dan *technical possibilities*. Hasilnya *Angular* lebih unggul daripada *backbone*, *ember.js*, dan *knockout* (Delčev & Drašković, 2018).
- d. Marin Kaluža, Krešimir Troškot, Bernard Vukelić dengan judul “*COMPARISON OF FRONT-END FRAMEWORKS FOR WEB APPLICATIONS DEVELOPMENT*”. Pada penelitian ini, 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 untuk *Vue.js*, *react* 10, dan *Angular* 9 (Troškot, 2018).
- e. Carl Lawrence Mariano dengan judul “*Benchmarking Javascript Frameworks*”. Pada penelitian ini, 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* (Mariano, 2017).
- f. Tugas Akhir. Pada penelitian ini peneliti menggabungkan analisis berdasarkan internet dan penelitian yang telah dilakukan. Peneliti membagi menjadi 5 jenis perbandingan yaitu *download results* yaitu perbandingan jumlah *download Vue.js* dengan *Angular* untuk melihat jumlah pengguna antara *Vue.js* dan *Angular* berdasarkan *website npm*. Selanjutnya *build time* yaitu perbandingan *build times*

berdasarkan uji yang ada pada internet. *Size metrics* yaitu perbandingan ukuran program yang telah dibuat dengan menggunakan *Vue.js* dan *Angular*. *Most Popularity* yaitu perbandingan popularitas dari kedua *framework* berdasarkan hasil survei yang dilakukan oleh *stateofjs* tahun 2018 dan terakhir *performed time* yaitu membandingkan performa dari dua *program* yang sama dengan masing masing menggunakan *framework Vue.js* dan *Angular*.

Beberapa penelitian sejenis di atas dapat dilihat pada Tabel 2.2:

Tabel 2.2 Perbandingan Berdasarkan Penelitian Sejenis

	<i>Download results</i>	<i>Performed time</i>	<i>Size metrics</i>	<i>Most Popularity</i>	<i>Build times</i>
(Wohlgethan, 2018)	<i>Vue.js</i>	-	-	<i>Vue.js</i>	-
(Voutilainen, 2017)	-	-	-	<i>Vue.js</i>	<i>Vue.js</i>
(Delčev & Drašković, 2018)	<i>Angular</i>	-	-	-	-
(Troskot, 2018)	<i>Vue.js</i>	-	-	<i>Vue.js</i>	-
(Mariano, 2017)	-	<i>Angular</i>	<i>Angular</i>	-	-
Tugas Akhir	<i>Vue.js</i>	<i>Vue.js</i>	<i>Vue.js</i>	<i>Vue.js</i>	<i>Vue.js</i>