

# **ANALISIS PERFORMA FRAMEWORK FRONTEND JAVASCRIPT BAGI PROGRAMMER**

Axel Christopher<sup>1</sup>, Sherly Anjelina<sup>2</sup>, Lina<sup>3</sup>

<sup>1,2,3</sup> Program Studi Teknik Informasi, Fakultas Ilmu Komputer, Universitas Esa Unggul Jakarta  
Jl. Arjuna Utara No. 9 Kb. Jeruk, Jaarta Barat, 11510  
noviandi@esaunggul.ac.id

## **Abstract**

*With the rapid development of JavaScript lately, several Javascript Frameworks especially Frontend, such as Angular, React JS, and VueJS have become popular. So, in this study, a comprehensive performance comparison test was carried out on the three Javascript Frontend Frameworks, as well as by providing recommendation levels. The test is carried out by measuring the metrics FCP, SI, LCP, TTI, TBT, and CLS. There is one point of performance testing of the Frontend Javascript Framework that will be tested, in the form of the performance of each Frontend Javascript Framework in managing the Simple Dashboard, where the smaller the value generated, the better the performance. Comparison testing was carried out twice on the three frameworks, then recorded and calculated the average value. The test results show that React gets the smallest average value on the FCP metric of 0.5s, the SI metric of 0.95s, the LCP metric of 1.1s, the TTI metric of 1.1s, the TBT metric of 15ms, and the CLS metric of 0s. Armed with the test results that have been carried out in this performance comparison research, it can be concluded that React has the best performance and is highly recommended for use by developers, especially Frontend Developers.*

**Keywords :** *frontend, kerangka kerja, javascript frontend, performance, programmer, website.*

## **Abstrak**

Dengan berkembang pesatnya JavaScript belakangan ini, membuat beberapa Framework Javascript khususnya Frontend seperti Angular, React JS, dan VueJS menjadi populer. Maka, pada penelitian ini dilakukan pengujian perbandingan performa secara menyeluruh pada ketiga Framework Javascript Frontend tersebut, serta dengan pemberian tingkatan rekomendasi. Pengujian dilakukan dengan mengukur metrik FCP, SI, LCP, TTI, TBT, dan CLS. Terdapat satu poin pengujian performa Framework Javascript Frontend yang akan diujikan, berupa performa masing-masing Framework Javascript Frontend dalam mengurus Simple Dashboard, dimana semakin kecil nilai yang dihasilkan, semakin baik performanya. Pengujian perbandingan dilakukan dua kali terhadap ketiga Framework Javascript Frontend, lalu berikutnya dirangkum dan dijumlahkan nilai rerata. Hasil pengujian menyimpulkan React mendapatkan nilai rerata terkecil pada metrik FCP sebesar 0.5s, metrik SI sebesar 0.95s, metrik LCP sebesar 1.1s, metrik TTI sebesar 1.1s, metrik TBT sebesar 15ms, dan metrik CLS sebesar 0s. Berbekal hasil pengujian yang telah dilakukan pada penelitian komparasi performa ini, dapat disimpulkan bahwa React mempunyai performa unggul dan sangat direkomendasikan agar digunakan oleh para pengembang khususnya Frontend Developer.

**Kata Kunci :** *frontend, kerangka kerja, javascript frontend, performa, programmer, website*

## **Pendahuluan**

Dewasa ini, Teknologi Informasi meningkat drastis melalui masa ke masa. Perkembangan Teknologi Informasi diikuti kemajuan teknologi dalam pembuatan sebuah aplikasi serta layanan berbasis situs web. Beberapa teknologi tersebut ialah JavaScript. JavaScript ialah *higher-order programming language* yang pertama kali diperkenalkan pada tahun 1995 untuk lingkungan pengembangan aplikasi berbasis situs web dinamis di sisi client, yang memungkinkan para pengembang dapat membuat tampilan situs web menjadi menarik (Dicoding, 2020). Dewasa ini Javascript ini mendapati kenaikan kemampuan yang sangat laju, yaitu: dukungan pada konsep *Object Oriented Programming, functional, structural, procedural programming, event-driven, prototyping*, dan kemudahan dalam pembuatan sebuah aplikasi di sisi *server* dan *client* (Elliott, 2020). React JS

menjadi salah satu contoh kemudahan bagi pengembang situs web berbasis Javascript di sisi client berupa kecepatan rendering tinggi, sehingga performa pengaksesan data menjadi lebih cepat. Selain itu juga memberikan kemudahan pada para pengembang dalam membuat tampilan dengan bantuan package lainnya (Nasution & Iswari, 2021).

Kepopuleran-an Javascript dalam lingkungan para pengembang situs web, diikuti dengan adanya segudang jenis Framework Javascript. Framework Javascript adalah kerangka kerja yang siap digunakan dan sistematis pada Javascript dengan bantuan sekumpulan library, agar dapat mempermudah para pengembang dalam membuat situs web berbasis JavaScript, seperti membuat fungsi, prosedur, serta fitur lainnya memakai bantuan library yang telah ada sehingga tidak harus untuk menuliskan kembali sumber kode-nya (Pano et al., 2018).

Framework Javascript memiliki dua tipe, yaitu: Framework Javascript Frontend dan Framework Javascript Backend. Framework Javascript Frontend menghadirkan dukungan pada pengembangan tampilan antarmuka situs web yang interaktif, menarik agar mempermudah pengguna dalam mengakses situs web serta meningkatkan pengalaman user, sebaliknya Framework Javascript Backend lebih mengarah pada sistem internal seperti dalam bentuk bantuan terhadap pemrosesan data atau processing data, parsing data atau parse data, suplai data atau supply data, penanganan request atau request handle, routing, dan validasi data atau validation data (Arhandi, 2016). Framework Javascript Frontend digunakan untuk membuat situs web menjadi interaktif (Duckett, 2014), menggunakan Bootstrap untuk sistem informasi berbasis situs web (Effendy & Nuqoba, 2016), ataupun memakai preferensi Framework Javascript Frontend misalnya Angular, React JS, dan Vue JS yang dikombinasikan dengan HTML 5 ketika membuat aplikasi dan layanan E-Business berbasis situs web dan mobile (Xing et al., 2019). Penetapan Framework Javascript Frontend diperlukan karena akan mempengaruhi kinerja dan struktur proyek situs web berbasis Frontend, dalam hal pembuatan tampilan antarmuka Frontend. Terdapat tiga buah teratas Framework Javascript Frontend yang paling sering digunakan saat ini oleh beberapa besar para pengembang, yaitu; React JS, Angular, Vue JS di mana ketiga buah Framework Javascript Frontend ini berbasiskan Javascript, yang menggunakan konsep Single Page Application, sebuah konsep aplikasi situs web yang memiliki satu halaman daripada memuat seluruh halamannya (Verma, 2021)(Pitaliya, 2021).

Terkait dengan ketiga Framework Javascript Frontend yang akan dites pada penelitian ini, sebelumnya sudah pernah ada penelitian yang melakukan pengujian komparasi pada ketiga framework ini berdasarkan dua poin penelitian, yaitu: 1) bidang performa, 2) bidang ukuran dimana hasil penelitian menunjukkan bahwa Vue JS memiliki performa kecepatan yang baik dan ukuran paling kecil dibanding keduanya (InVerita, 2020). Namun penelitian tersebut belum menampilkan penilaian performa secara menyeluruh.

Berikutnya, telah dilakukan penelitian oleh (Alexander, 2015) untuk menguji komparasi performa tujuh Framework Javascript Frontend Angular, Aurelia, Backbone, Ember, Knockout, Mithril, dan Vue dalam membuat, menghapus, dan memperbarui elemen HTML. Hasil penelitian tersebut menunjukkan bahwa Vue JS mempunyai performa tercepat dalam membuat, menghapus, dan memperbarui elemen. Namun penelitian tersebut belum memfokuskan pada performa dari Framework Javascript Frontend berbasis Javascript/Typescript.

Pengujian komparasi performa ketiga Framework Javascript Frontend juga telah dilakukan oleh (Martin, 2022) dalam hal menangkap seluruh perubahan pada DOM (Document Object Model), dimana hasil penelitian tersebut mengatakan bahwa Virtual DOM yang digunakan oleh React JS dan Vue JS sangat berguna untuk aplikasi yang rumit. Hasil penelitian ini menampilkan kepada bagaimana cara para framework mengatasi DOM, namun belum mencakup konsep dan fungsional masing-masing framework.

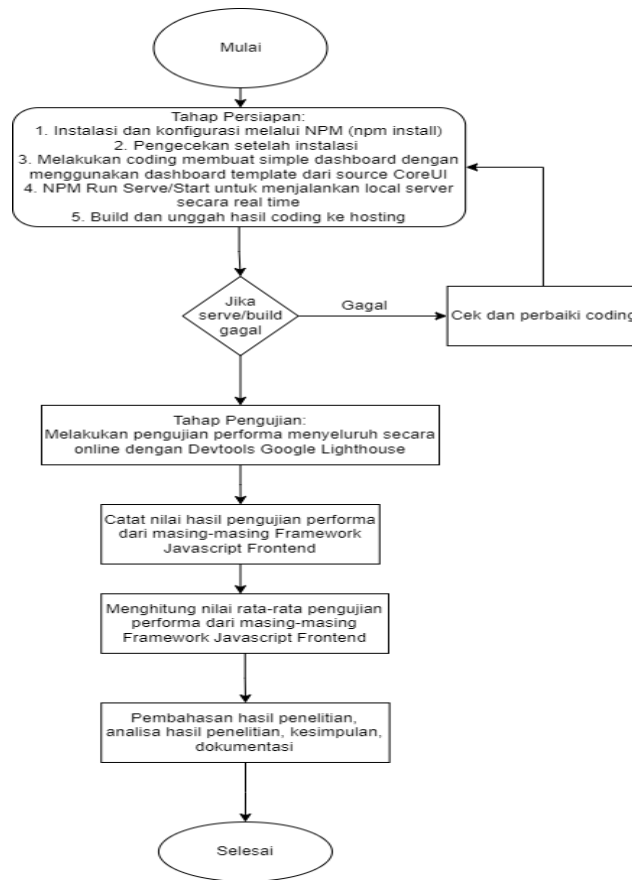
Penelitian lainnya juga dilakukan oleh (Saks, 2019) untuk menguji performa Simple Single Page aplikasi dari masing-masing framework lalu diuji kecepatan loadingnya, dimana hasil penelitian tersebut menunjukkan bahwa framework Vue JS mendapatkan performa tercepat dalam pengujian kecepatan. Vue JS lima kali lebih cepat daripada Angular dan lebih cepat dua kali daripada React JS. React JS mempunyai build terkecil yaitu 587KB, Vue JS memiliki 624KB, dan Angular paling besar adalah 15.7MB. Penelitian dilakukan oleh (Persson, 2020) untuk menguji performa manipulasi DOM antara vanilla Javascript dengan framework Javascript yaitu Angular, React JS, Vue JS menunjukkan bahwa React JS memiliki performa yang terbaik dalam manipulasi DOM dengan teknologi virtual DOM yang dimilikinya. Lalu penelitian yang dilakukan oleh (Wohlgethan, 2018) menunjukkan bahwa framework Vue JS mendapatkan performa terbaik dalam tingkat fleksibel, membuat aplikasi skala besar, dapat bekerja dengan kode HTML, dan populer. Ketiga penelitian ini telah melakukan pengujian terhadap Framework Javascript Frontend, namun belum mencakup penilaian performa secara menyeluruh dan konsep serta fungsional masing-masing framework. Berbekal dari beberapa tinjauan pustaka dan beberapa penelitian yang telah dilakukan sebelumnya dan studi kasus persoalan penetapan Framework Javascript Frontend yang sesuai untuk para pengembang di antara ketiga pilihan yang ada (Angular, Vue JS, React JS), membuat diperlukannya penelitian serta pengujian secara mendetail terkait perbandingan performa menyeluruh ketiga Framework Javascript Frontend tersebut. Black Box Testing adalah metode yang digunakan dalam penelitian ini, untuk menguji performa menyeluruh dengan metrik FCP atau First Contentful Paint, untuk melihat waktu diperlukan untuk menampilkan konten pertama ke pengguna, LCP atau Largest Contentful Paint, untuk melihat waktu diperlukan untuk memperlihatkan satu halaman penuh ke pengguna, CLS atau Cumulative Layout Shift, ialah perubahan elemen gambar di dalam konten, SI atau Speed Index, untuk mengukur seberapa cepat konten ditampilkan pada saat loading halaman, TTI atau Time To Interactive, untuk mengukur seberapa lama sebuah halaman menjadi interaktif penuh, dan TBT atau Total Blocking Time, untuk mengukur jumlah total waktu halaman diblokir untuk merespon input pengguna.

Pada penelitian kali ini, ada dua nilai yang menjadi definisi masalah berbentuk pertanyaan penelitian, berupa: 1.) Bagaimana langkah-langkah untuk melakukan pengujian komparasi performa menyeluruh dari ketiga Framework Javascript Frontend yang dites? 2.) Framework Javascript Frontend mana yang mempunyai nilai performa dan konsep serta fungsional terbaik yang bisa disarankan kepada para pengembang khususnya frontend? Tujuan dilakukannya penelitian ini dilakukan ialah untuk mendapati timbangan hasil evaluasi performa dari ketiga pilihan Framework Javascript Frontend dari segi kemampuan performa secara menyeluruh dan konsep serta fungsional masing-masing framework, agar bisa menjadi masukan, rujukan, serta timbangan bagi para pengembang khususnya Frontend dalam menentukan Framework Javascript Frontend yang sesuai.

## **Metode Penelitian**

Pada tahap pengujian, dilakukan dua kali pengujian, dengan melakukan pengujian performa menyeluruh secara online melalui Devtools Google Lighthouse. Bilamana pengujian yang dilakukan sukses, lalu hasil setiap pengujian akan dirangkum agar dapat dijumlahkan nilai reratanya. Selepas tahap pengujian selesai dilakukan, maka akan dilakukan kupasan hasil penelitian, simpulan, serta pengarsipan berbentuk publikasi paper. Gambar 1 menunjukkan tahap persiapan serta tahap pengujian. Dalam tahap persiapan, dilakukanlah instalasi masing-masing framework melalui NPM dari NodeJS pada perangkat keras dan perangkat lunak peneliti, melakukan pengecekan struktur project setelah instalasi, melakukan coding membuat simple dashboard dengan menggunakan dashboard template dari source CoreUI, menjalankan server lokal secara realtime dengan perintah NPM Run Serve/Start, serta build dan unggah hasil coding ke hosting. Apabila terdapat kegagalan pada saat serve atau build, dilakukan pengecekan dan perbaiki coding, kemudian melakukan serve atau build ulang. Agar

pengujian dan implementasi dapat berjalan dalam penelitian ini, digunakanlah beberapa perangkat lunak dan keras pendukung dalam penelitian ini. Perangkat keras yang dipakai yaitu: Laptop HP Pavilion Gaming Laptop 15-dk0042 (Intel Core I7-9750H, 16GB RAM, HDD 1TB & SSD 239GB). Sistem Operasi Windows 10 64bit adalah perangkat lunak yang digunakan, NodeJS 14.18.0 serta *NPM* atau *NodeJS Package Manager*. Pengujian akan dilakukan sebanyak dua kali pada ketiga *Framework Javascript Frontend* yang diujikan, dengan menggunakan *Black Box Testing* untuk menguji performa menyeluruh dengan metrik FCP, SI, LCP, TTI, TBT, dan CLS. Megenai evaluasi pengujian yang dilakukan sebanyak dua kali ialah agar mendapatkan nilai akurasi rerata yang baik, juga mempermudah dalam analisa hasil pengujian. Ada satu poin pengujian performa *Framework Javascript Frontend* yang akan diujikan, berupa Performa masing-masing *Framework Javascript Frontend* dalam mengurus *simple dashboard*, dimana semakin kecil nilai yang dihasilkan, maka semakin bagus performanya. Data-data dari hasil pengujian akan diringkas ke dalam tabel, agar berikutnya dapat dijumlahkan nilai rerata. Nilai rerata akan menjadi kriteria penilaian performa masing-masing *Framework Javascript Frontend*. *NPM* digunakan untuk membantu dalam menginstal ketiga *Framework Javascript Frontend* dengan *library* yang tersedia. Konfigurasi serta format yang berisikan *scripts*, *dependencies*, versi dan *library framework* yang dimuat dalam berkas *Javascript Object Notation* bernama *package.json*. Isi dari berkas *package.json* dapt dilihat pada tabel 1:



Gambar 1 Diagram Alir Penelitian

Tabel 1. Config JSON Framework React

```
{
  "name": "my-app",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@testing-library/jest-dom": "^5.16.4",
    "@testing-library/react": "^13.1.1",
    "@testing-library/user-event": "^13.5.0",
    "react": "^18.0.0",
    "react-dom": "^18.0.0",
    "react-scripts": "5.0.1",
    "web-vitals": "^2.1.4"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ]
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  }
}
```

Tabel 2. Config JSON Framework Vue.js

```
{
  "name": "my-project",
```

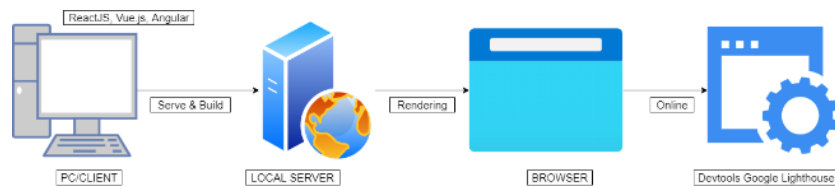
```
"version": "0.1.0",
"private": true,
"scripts": {
  "serve": "vue-cli-service serve",
  "build": "vue-cli-service build"
},
"dependencies": {
  "core-js": "^3.8.3",
  "vue": "^3.2.13"
},
"devDependencies": {
  "@vue/cli-plugin-babel": "~5.0.0",
  "@vue/cli-service": "~5.0.0"
}
}
```

Tabel 3. Config JSON Framework Angular

```
{
  "name": "simple-app",
  "version": "0.0.0",
  "scripts": {
    "ng": "ng",
    "start": "ng serve",
    "build": "ng build",
    "watch": "ng build --watch --configuration development",
    "test": "ng test"
  },
  "private": true,
  "dependencies": {
    "@angular/animations": "~13.1.0",
    "@angular/common": "~13.1.0",
    "@angular/compiler": "~13.1.0",
    "@angular/core": "~13.1.0",
    "@angular/forms": "~13.1.0",
    "@angular/platform-browser": "~13.1.0",
    "@angular/platform-browser-dynamic": "~13.1.0",
    "@angular/router": "~13.1.0",
    "rxjs": "~7.4.0",
    "tslib": "^2.3.0",
    "zone.js": "~0.11.4"
  },
  "devDependencies": {
    "@angular-devkit/build-angular": "~13.1.4",
    "@angular/cli": "~13.1.4",
    "@angular/compiler-cli": "~13.1.0",

```

```
"@types/jasmine": "~3.10.0",
"@types/node": "^12.11.1",
"jasmine-core": "~3.10.0",
"karma": "~6.3.0",
"karma-chrome-launcher": "~3.1.0",
"karma-coverage": "~2.1.0",
"karma-jasmine": "~4.0.0",
"karma-jasmine-html-reporter": "~1.7.0",
"typescript": "~4.5.2"
}
}
```

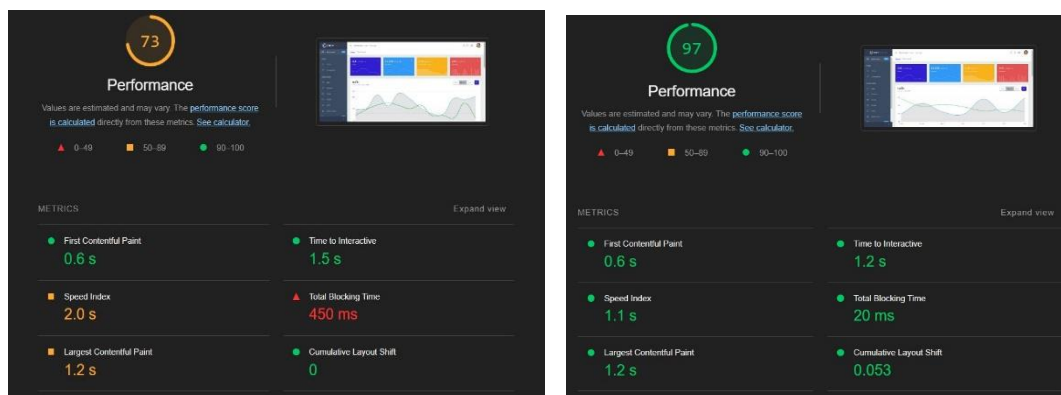


Gambar 2. Desain Skema Pengujian Perbandingan Performa Menyeluruh

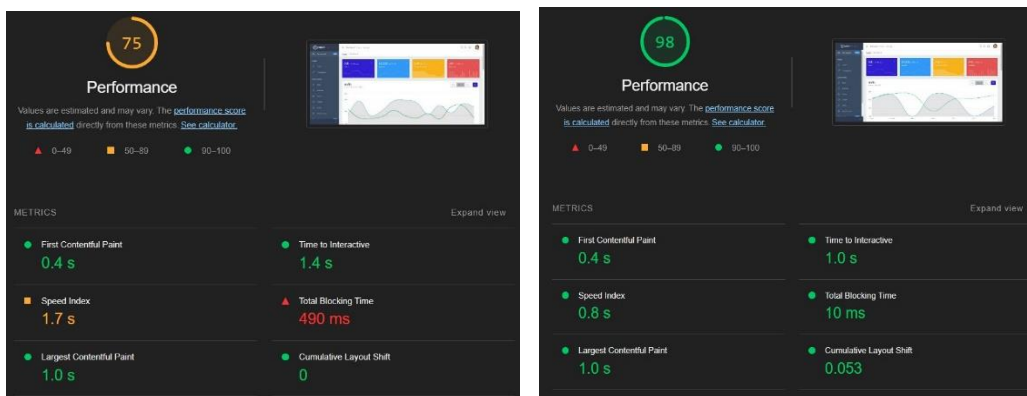
Desain Skema Pengujian Perbandingan Performa Menyeluruh, dilakukan dari tahap pertama, yaitu menginstall masing-masing *Framework* Javascript Frontend terlebih dahulu pada perangkat keras yang digunakan peneliti, lalu melakukan *serve & build* pada *local server* secara realtime, *rendering* hasil kodingan ke browser menggunakan *server* lokal, unggah hasil *coding* ke *hosting* agar dapat diakses secara *online*, serta melakukan pengujian performa secara menyeluruh melalui *Devtools Google Lighthouse*. Hasil pengujiannya kemudian dirangkum, dibahas, serta dianalisa.

### Hasil dan Pembahasan

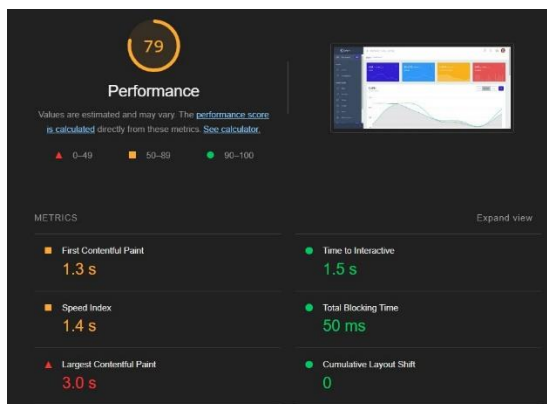
Pengujian perbandingan dilakukan dua kali pada metode *Black Box Testing* secara *online* dengan *Devtools Google Lighthouse* untuk menguji performa menyeluruh dengan metrik FCP, SI, LCP, TTI, TBT, dan CLS. Hasil pengujian performa secara menyeluruh yang usai dilakukan, ditampilkan pada Gambar 3 hingga Gambar 8.



Gambar 3. Pengujian Pertama Simple Dashboard Angular, dan Simple Dashboard React JS



Gambar 4. Pengujian Kedua Simple Dashboard Angular, dan Simple Dashboard React JS



Gambar 8. Pengujian Kedua Simple Dashboard Vue JS

Nilai yang dihasilkan dari masing-masing pengujian, diolah dalam enam tabel performa, yang berikutnya dihitung nilai rata-rata dengan rumus pengujian pertama ditambah pengujian kedua lalu dibagi dua, sehingga mempermudah dalam melakukan analisa hasil pengujian. Dari dua kali pengujian performa secara menyeluruh, nilai pengujian dari masing-masing framework, dirangkum dalam Tabel 4 hingga Tabel 9 berikut:

Tabel 4. Hasil Pengujian Kecepatan Performa Metrik FCP

Pengujian Ke	Angular	React JS	Vue JS
1	0.6s	0.6s	0.8s
2	0.4s	0.4s	1.3s



Tabel 5. Hasil Pengujian Kecepatan Perfroma Metrik SI

Pengujian Ke	Angular	React JS	Vue JS
1	2.0s	1.1s	1.9s
2	1.7s	0.8s	1.4s

Tabel 6. Hasil Pengujian Kecepatan Perfroma Metrik LCP

Pengujian Ke	Angular	React JS	Vue JS
1	1.2s	1.2s	2.9s
2	1.0s	1.0s	3.0s

Tabel 7. Hasil Pengujian Kecepatan Perfroma Metrik TTI

Pengujian Ke	Angular	React JS	Vue JS
1	1.5s	1.2s	1.2s
2	1.4s	1.0s	1.5s

Tabel 8. Hasil Pengujian Kecepatan Perfroma Metrik TBT

Pengujian Ke	Angular	React JS	Vue JS
1	450ms	20ms	50ms
2	490ms	10ms	50ms

Tabel 9. Hasil Pengujian Kecepatan Perfroma Metrik CLS

Pengujian Ke	Angular	React JS	Vue JS
1	0	0	0
2	0	0	0

Berdasarkan data pengujian yang telah dilakukan, *React JS* memiliki performa secara menyeluruh pada metrik FCP, SI, LCP, TTI, TBT, dan CLS. Dimana *React JS* pada metrik LCP mendapatkan performa 0.6s pada pengujian pertama dan mengalami penurunan 0.2% pada pengujian kedua menjadi 0.4s, pada metrik LCP mendapatkan performa 1.2s pada pengujian pertama dan mengalami penurunan 0.2% pada pengujian kedua menjadi 1.0%, pada metrik SI mendapatkan performa 1.1s pada pengujian pertama dan mengalami penurunan 0.3% pada pengujian kedua menjadi 0.8s, pada metrik TTI mendapatkan performa 1.2s dan mengalami penurunan 0.2% pada pengujian kedua menjadi 1.0s, pada metrik TBT mendapatkan performa 20ms dan mengalami penurunan 10%

pada pengujian kedua menjadi 10ms, pada metrik CLS mendapatkan performa 0, dimana semakin kecil nilainya maka semakin baik performanya. *React JS* memiliki nilai rata-rata sebesar 0.5s pada metrik FCP, 0.95s pada metrik SI, 1.1s pada metrik LCP, 1.1s pada metrik TTI, 15ms pada metrik TBT, dan 0 pada metrik CLS.

*VueJS* memiliki performa medioker secara menyeluruh, meskipun sempat mengalami kenaikan sebesar 0.5% pada metrik FCP, dan 0.1% pada metrik LCP, namun *VueJS* memiliki performa yang stabil pada metrik TBT. *VueJS* memiliki nilai rata-rata sebesar 1.05s pada metrik FCP, 1.65s pada metrik SI, 2.95s pada metrik LCP, 0.85s pada metrik TTI, 50ms pada metrik TBT, dan 0 pada metrik CLS. Sedangkan *Angular* memiliki performa tidak baik dalam menangani *Simple Dashboard*, karena mendapatkan response sebesar 450ms dan naik menjadi 490ms pada metrik TBT.

Penelitian yang dilakukan oleh (Radosław Nowacki\*, 2016) menunjukkan bahwa *React JS* mendapatkan penilaian terbaik dalam menangani *Single Page Application* berkat kinerja komponen yang hebat serta *tools* yang disediakan daripada *Angular* dan *Emberjs*. Penelitian juga dilakukan oleh (Kowalczyk & Plechawska-Wójcik, 2016) yang meneliti perbandingan performa antara *Angular* dan *React JS* dalam hal Response Time, Resource Consumption, dan Script Size. Hasil penelitian tersebut menunjukkan bahwa *React JS* sangat baik dalam menangani perubahan jumlah data secara dinamis.

### Tingkat Rekomendasi Framework

Berdasarkan hasil pengujian, pembahasan dan analisa yang dilakukan telah dilakukan, maka diberikan tingkatan rekomendasi pada ketiga *Framework Javascript Frontend* yang diujikan. Tingkatan rekomendasi yang diberikan meliputi tiga tingkatan, yaitu: 1) Sangat Direkomendasikan, 2) Direkomendasikan, 3) Tidak direkomendasikan. Tabel 7 menampilkan tingkat rekomendasi dari ketiga *Framework Javascript Frontend* yang diujikan:

Tabel 7. Tingkat Rekomendasi Framework Javascript Frontend

Framework	Penilaian	Rekomendasi
Angular	Buruk	Tidak Direkomendasikan
React JS	Sangat Baik	Sangat Direkomendasikan
Vue JS	Baik	Direkomendasikan

Berdasarkan Tabel 7, para pengembang dapat memilih *React JS* dalam membuat *Simple Dashboard* karena memiliki performa yang baik secara menyeluruh, juga memiliki struktur folder yang teratur. Lalu penggunaan *Vue JS* adalah opsional dikarenakan memiliki performa cukup dalam menangani *Simple Dashboard*. Para pengembang tidak disarankan untuk menggunakan *Angular* dalam membuat *Simple Dashboard*.

### Kesimpulan

Kesimpulan yang didapat dari penelitian ini berupa bahwa *React JS* memiliki performa baik secara menyeluruh dengan performa di atas 90% berdasarkan *Devtools Google Lighthouse* yang dapat menjadi pertimbangan pilihan untuk para pengembang *Frontend Web Developer* karena memiliki skor rendah, dimana semakin kecil artinya semakin baik. Pemilihan *Framework Javascript Frontend* yang baik dapat membantu pengembangan situs web *Frontend* menjadi lebih baik.

## Daftar Pustaka

- Alexander, S. (2015). *Speed Performance Comparison of JavaScript MVC Frameworks*. <http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A998701&dswid=2851>
- Arhandi, P. P. (2016). Pengembangan Sistem Informasi Perijinan Tenaga Kesehatan Dengan Menggunakan Metode Back End Dan Front End. *Teknologi Informasi*, 7(1), 39–48.
- Dicoding. (2020). Apa Itu JavaScript? Fungsi dan Contohnya. In *www.dicoding.com*. <https://www.dicoding.com/blog/apa-itu-javascript-fungsi-dan-contohnya/>
- Duckett, J. (2014). *JavaScript and JQuery: Interactive Front-End Web Development*. Wiley & Son.
- Effendy, F., & Nuqoba, B. (2016). Penerapan Framework Bootsrap Dalam Pembangunan Sistem Informasi Pengangkutan Dan Penjadwalan Pegawai (Studi Kasus: Rumah Sakit Bersalin Buah Delima Sidoarjo). *Informatika Mulawarman: Jurnal Ilmiah Ilmu Komputer*, 11(1), 9. <https://doi.org/10.30872/jim.v11i1.197>
- Elliott, E. (2020). Top JavaScript Frameworks and Tech Trends for 2021 | by Eric Elliott | JavaScript Scene | Medium. In *medium.com*. <https://medium.com/javascript-scene/top-javascript-frameworks-and-topics-to-learn-in-2020-and-the-new-decade-ced6e9d812f9>
- InVerita. (2020). *Vue vs React vs Angular: What Framework Would You Choose?* | by inVerita | The Startup | Medium. <https://medium.com/swlh/vue-vs-react-vs-angular-what-framework-would-you-choose-5d77a3680b0d>
- Kowalczyk, K., & Plechawska-Wójcik, M. (2016). AngularJS and ReactJS libraries - performance analysis. *Journal of Computer Sciences Institute*, 2, 2016–2027.
- Martin, S. (2022). Angular vs React vs Vue.js: Which is the Best Choice for 2022? In *medium.com*. <https://javascript.plainenglish.io/angular-vs-react-vs-vue-js-which-is-the-best-choice-for-2022-5ef83f2257ab>
- Nasution, & Iswari, L. (2021). Penerapan React JS Pada Pengembangan FrontEnd Aplikasi Startup Ubaform. *Automata*, 2(2).
- Pano, A., Graziotin, D., & Abrahamsson, P. (2018). Factors and actors leading to the adoption of a JavaScript framework. *Empirical Software Engineering*, 23(6), 3503–3534. <https://doi.org/10.1007/s10664-018-9613-x>
- Persson, M. (2020). *JavaScript DOM Manipulation Performance: Comparing Vanilla JavaScript and Leading JavaScript Front-end Frameworks*. *Independen*, 40. <http://bth.diva-portal.org/smash/get/diva2:1436661/FULLTEXT01.pdf%0Ahttp://urn.kb.se/resolve?urn=urn:nbn:se:bth-19531>
- Pitaliya, S. (2021). List of Best Frontend Frameworks to Consider in 2021 | Radixweb. In *medium.com*. <https://medium.com/radixweb/a-comprehensive-list-of-best-frontend-frameworks-to-consider-in-2021-8616ba9f3143>
- Radosław Nowacki\*, M. P.-W. (2016). *Analiza porównawcza narzędzi do budowania aplikacji Single Comparative analysis of tools dedicated to building Single Page Applications –*. 2, 98–103.
- Saks, E. (2019). JavaScript Frameworks: Angular vs React vs Vue. *University of Applied Sciences*, 42.
- Verma, H. (2021). Top 5 Frontend Frameworks to Learn in 2022 | by Harendra Verma | JavaScript in Plain English. In *medium.com*. <https://javascript.plainenglish.io/top-5-frontend-frameworks-to-learn-in-2022-57b76a6508dd>
- Wohlgethan, E. (2018). *Supporting Web Development Decisions by Comparing Three Major JavaScript Frameworks: Angular, React and Vue.js*. [https://reposit.haw-hamburg.de/bitstream/20.500.12738/8417/1/BA\\_Wohlgethan\\_2176410.pdf](https://reposit.haw-hamburg.de/bitstream/20.500.12738/8417/1/BA_Wohlgethan_2176410.pdf)
- Xing, Y. K., Huang, J. P., & Lai, Y. Y. (2019). Research and analysis of the front-end frameworks and libraries in e-business development. *ACM International Conference Proceeding Series*, 68–72. <https://doi.org/10.1145/3313991.3314021>