

# Analisis Perbandingan Framework Front-End Javascript React dan Vue Pada Pengembangan Website

Farro Axza Febsinatra Sofi'ie<sup>1</sup>, Anita Qoiriah<sup>2</sup>

<sup>1,2</sup> Jurusan Teknik Informatika, Fakultas Teknik, Universitas Negeri Surabaya

<sup>1</sup>[farro.19062@mhs.unesa.ac.id](mailto:farro.19062@mhs.unesa.ac.id)

<sup>2</sup>[anitaqoiriah@unesa.ac.id](mailto:anitaqoiriah@unesa.ac.id)

**Abstrak**— Penelitian ini bertujuan untuk melakukan analisis perbandingan performa dan efisiensi antara dua framework front-end JavaScript populer, yaitu React dan Vue, pada pengembangan website. Performa akan diukur berdasarkan waktu yang dibutuhkan untuk melakukan proses tertentu, sementara efisiensi akan dianalisis dari sumber daya yang dibutuhkan oleh masing-masing framework. Penelitian ini akan membantu para pengembang dalam memahami keunggulan dan kelemahan dari masing-masing framework dan memilih yang sesuai dengan kebutuhan proyek. Penelitian ini menggunakan metode eksperimental dengan melakukan pengujian pada kedua framework menggunakan beberapa kasus uji yang telah ditentukan. Waktu yang dibutuhkan untuk merender komponen dan melakukan pembaruan pada tampilan website akan diukur untuk mengevaluasi performa. Selain itu, penggunaan sumber daya seperti penggunaan CPU, memori, dan jaringan akan dianalisis untuk mengukur efisiensi masing-masing framework. Hasil penelitian menunjukkan bahwa Vue memiliki performa yang lebih baik daripada React dalam sebagian besar skenario pengujian. Vue mampu merender komponen dengan lebih cepat dan responsif, serta mengoptimalkan penggunaan sumber daya yang lebih hemat. Selain itu, Vue juga menunjukkan konsistensi performa yang stabil dalam berbagai situasi pengembangan. Berdasarkan hasil penelitian ini, disarankan bagi pengembang untuk mempertimbangkan Vue sebagai pilihan yang kuat untuk pengembangan website yang efisien dan responsif. Meskipun React memiliki keunggulan dalam komponen reusable dan ekosistem yang besar, namun Vue menawarkan performa yang lebih baik dan konsistensi yang stabil.

**Kata Kunci**— Framework, JavaScript, React, Vue, Benchmark, Website, Front-end.

## I. PENDAHULUAN

Pengembangan website merupakan hal yang sangat penting bagi perusahaan, organisasi, dan individu untuk meningkatkan brand awareness dan memberikan layanan yang baik kepada pelanggan. Dalam era teknologi informasi saat ini, website memainkan peran yang sangat besar dalam mempresentasikan identitas dan memberikan informasi kepada audiens. Oleh karena itu, pengembangan website harus dilakukan dengan baik dan menggunakan teknologi terbaru.[1]

Framework front-end adalah salah satu komponen penting dalam pengembangan website yang membantu mempermudah dan mempercepat proses pengembangan.[2] Framework front-end memiliki banyak fitur yang membantu mengatasi masalah yang sering terjadi seperti manajemen kode, performa, dan konsistensi.[3]

React dan Vue adalah dua framework front-end yang sangat populer dan banyak digunakan oleh pengembang website. Kedua framework memiliki fitur dan kelebihan masing-masing, namun sulit untuk membandingkan dan menentukan framework mana yang lebih baik untuk digunakan dalam pengembangan website.[4] Oleh karena itu, perlu adanya analisis dan perbandingan antara kedua framework untuk membantu pengembang membuat keputusan yang informatif.

Menurut beberapa jurnal terkait, React dan Vue memiliki perbedaan signifikan dalam hal pendekatan pengembangan dan performa. Salah satu jurnal yang membahas hal ini adalah "Benchmark Comparison of Javascript Frameworks React, Vue, Angular and Svelte" oleh Wenqing Xu pada tahun 2021. Dalam jurnal ini, dijelaskan bahwa pendekatan pengembangan yang digunakan oleh Vue lebih dekat dengan pendekatan pengembangan native, sehingga dapat mempermudah para pengembang yang sudah terbiasa dengan konsep tersebut. Sementara itu, React lebih fokus pada konsep komponen yang memisahkan tampilan dengan logika aplikasi.[5]

Selain perbedaan pendekatan pengembangan, kedua framework ini juga memiliki perbedaan dalam performa. Sebuah jurnal yang membahas hal ini adalah "Comparison of the performance of tools for creating a SPA application interface-React and Vue.js" oleh Krzysztof Boczkowski dan Beata Pańczyk pada tahun 2020. Dalam jurnal ini, dijelaskan bahwa ketiga framework ini memiliki kelebihan dan kekurangan masing-masing dalam hal performa. Vue memiliki performa yang lebih cepat dalam hal rendering, sedangkan React memiliki performa yang lebih cepat dalam hal pengelolaan state aplikasi.[6]

Penggunaan kedua framework ini juga berbeda dalam hal pengelolaan data dan arsitektur aplikasi. Sebuah jurnal yang membahas hal ini adalah "Comparative Analysis on Front-end Frameworks for Web Applications" oleh Rishi Vyas pada tahun 2022. Dalam jurnal ini, dijelaskan bahwa Vue memiliki kemampuan dalam pengelolaan data dengan mudah dan memiliki struktur aplikasi yang sederhana, sedangkan React lebih fleksibel dalam hal pengelolaan state aplikasi dan dapat diintegrasikan dengan arsitektur aplikasi yang kompleks.[7]

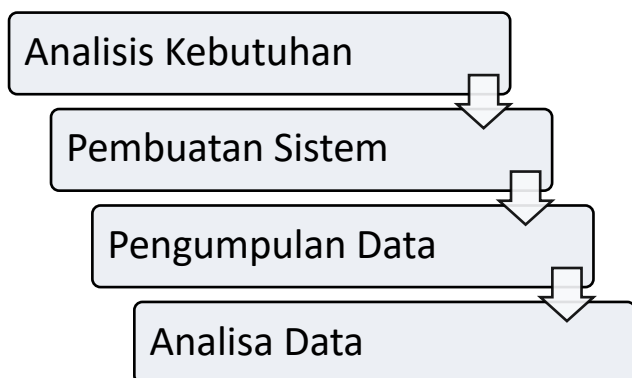
Untuk membantu memahami perbandingan antara React dan Vue, skripsi ini akan menganalisis dan membandingkan kedua framework berdasarkan beberapa kriteria penting seperti performa, fleksibilitas, dan efisiensi. Tujuannya adalah untuk memberikan rekomendasi yang berguna bagi pengembang website dan membantu membuat keputusan yang lebih

informatif tentang pemilihan framework front-end untuk pengembangan website.

Dengan melakukan analisis dan perbandingan antara React dan Vue, skripsi ini akan membantu menjawab pertanyaan penting seperti framework mana yang lebih cocok untuk digunakan dalam pengembangan website tertentu dan apa saja kelebihan dan kekurangan masing-masing framework. Hasil dari skripsi ini akan memberikan informasi yang berguna bagi pengembang website dan membantu meningkatkan kualitas pengembangan website di masa depan.

## II. METODE PENELITIAN

Metode yang diterapkan pada penelitian ini adalah metode kuantitatif. Gbr. 1 berikut merupakan alur tahapan yang dilakukan dalam penelitian ini:



Gbr. 1 Diagram Alur Metode Penelitian

### A. Analisis Kebutuhan

Analisis perbandingan Framework front-end JavaScript React dan Vue pada pengembangan website memerlukan identifikasi kebutuhan yang tepat untuk menentukan detail kebutuhan pada penelitian ini. Penelitian ini akan menggunakan React dan Vue sebagai Framework Front-end JavaScript dan akan dilakukan pada lingkungan pengembangan yang sama.

Perlu diperhatikan bahwa perangkat keras sangat mempengaruhi hasil dari pengujian. Maka hasil dari pengujian yang akan dilakukan adalah hasil pengujian relatif pada perangkat tersebut. Perangkat yang berbeda akan menghasilkan angka yang berbeda. Oleh karena itu, penelitian diuji harus pada perangkat yang sama untuk mendapatkan hasil yang serupa.

Dibawah ini disebutkan beberapa kebutuhan yang dapat mendukung penelitian ini agar dapat berjalan sesuai tujuan, diantaranya:

#### 1) Kebutuhan perangkat keras (Hardware)

Perangkat keras yang digunakan dalam penelitian guna mewujudkan tujuan penelitian yaitu perangkat laptop sebagai uji coba dengan spesifikasi berikut :

- Prosesor : AMD Ryzen 5 3550H
- RAM : 16.00 GB

- Penyimpanan : SSD 512 GB
  - Sistem Operasi : Windows 11 Enterprise 64-bit
- #### 2) Kebutuhan perangkat lunak (Software)

Perangkat lunak berfungsi untuk pengoperasian sistem pada penelitian ini. Pada penelitian ini digunakan adalah sebagai berikut :

- Google Chrome 114.0.5735.199 (32-bit)
- Visual Studio Code 1.79.2 (System Setup)
- Vue.js 3.2.47
- React.js 18.2.0
- js-Framework-benchmark for Chrome 114
- Lighthouse
- Node.js 16.19.0 (LTS)
- NPM 8.19.3
- IBM SPSS Release 27 0.1.0 (64-bit edition)

### B. Pembuatan Sistem

Sistem dibuat pada lingkungan pengujian yang sama berdasarkan kebutuhan penelitian. Setelah lingkungan pengujian disiapkan, selanjutnya adalah melakukan implementasi dan pengujian pada masing-masing kasus pengujian. Pada tahap ini, penulis harus mencatat waktu yang diperlukan untuk setiap operasi dan membandingkan performa kinerja antara React dan Vue. Library js-framework-benchmark digunakan untuk membantu pencatatan nilai-nilai yang dihasilkan oleh setiap pengujian.

### C. Pengumpulan Data

Data dikumpulkan melalui uji coba pada dua framework dengan mengikuti prosedur benchmarking dan pengujian fungsional. Data akan dikumpulkan melalui pengukuran waktu, memori yang digunakan, dan skor pengujian fungsional.

Benchmark akan dilakukan dengan aplikasi sederhana yang akan mengoperasikan komponen UI dengan data random dan menghitung durasi yang dibutuhkan untuk setiap operasi termasuk juga rendering time. Berikut adalah prosedur yang akan digunakan dalam proses benchmarking.

Performa:

1. Membuat 1.000, 10.000, 20.000 baris setelah halaman dimuat
2. Mengganti semua 1.000, 10.000, 20.000 baris pada elemen dengan baris yang baru
3. Memperbarui teks setiap baris kelipatan 10 pada elemen dengan 1.000, 10.000, 20.000 baris
4. Memilih baris sebagai respons terhadap klik pada baris tersebut
5. Mengganti posisi dua baris pada elemen dengan 1.000, 10.000, 20.000 baris
6. Menghapus satu baris pada komponen tabel
7. Menambahkan 1.000, 10.000, 20.000 baris pada elemen yang sudah memiliki 1.000 baris
8. Membersihkan semua baris (1.000, 10.000, 20.000)

Efisiensi:

9. Penggunaan memori setelah halaman dimuat

10. Penggunaan memori setelah menambahkan 1.000, 10.000, 20.000 baris
11. Penggunaan memori setelah update baris dengan 1.000, 10.000, 20.000 baris
12. Penggunaan memori setelah membuat dan membersihkan 1.000, 10.000, 20.000 baris

Startup:

13. Lighthouse metric TimeToConsistentlyInteractive: waktu yang dibutuhkan untuk interaksi yang konsisten pada halaman setelah startup
14. Lighthouse metric ScriptBootUpTime: total waktu yang dibutuhkan untuk parsing/compile/evaluate semua script pada halaman saat startup
15. Lighthouse metric MainThreadWorkCost: total waktu yang dibutuhkan untuk melakukan pekerjaan pada thread utama, termasuk layout dan lainnya saat startup
16. Lighthouse metric TotalByteWeight: total byte transfer jaringan (setelah kompresi) dari semua sumber daya yang dimuat ke halaman

#### D. Analisa Data

Setelah data dikumpulkan, data akan diproses dan diinterpretasi menggunakan analisis statistik deskriptif. Hasil analisis data dapat memberikan kesimpulan mengenai kinerja, kecepatan, efisiensi, dan kelebihan masing-masing framework. Dengan demikian, analisis data akan menjadi landasan untuk memberikan rekomendasi mengenai framework mana yang lebih cocok untuk digunakan dalam pengembangan website.

Data dianalisis dengan dua hipotesis diantaranya:

H0 : Tidak ada perbedaan signifikan antara React dan Vue.

H1 : Terdapat perbedaan signifikan antara React dan Vue.

Metode: Uji t Independen dengan asumsi data berdistribusi normal

Rumus uji t independen:

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{S_1^2/n_1 + S_2^2/n_2}} \dots\dots\dots(1)$$

X1 = rata-rata sampel pertama (React)

X2 = rata-rata sampel kedua (Vue)

S1 = simpangan baku sampel pertama (React)

S2 = simpangan baku sampel kedua (Vue)

n1 = jumlah observasi sampel pertama (React)

n2 = jumlah observasi sampel kedua (Vue)

### III. HASIL DAN PEMBAHASAN

#### A. Hasil

Pengujian atau benchmarking rendering time dilakukan sebanyak 10 kali untuk memastikan konsistensi hasil yang diperoleh. Melakukan pengujian sebanyak 10 kali dapat mengurangi kemungkinan adanya anomali atau outlier yang dapat mempengaruhi hasil secara signifikan. Dalam setiap pengulangan, framework yang diuji dijalankan dengan parameter yang sama dan dalam kondisi yang serupa untuk memastikan bahwa setiap pengujian dilakukan dalam lingkungan yang sebanding.

Dalam pengujian berulang, beberapa nilai statistik penting dihitung untuk memberikan informasi yang lebih komprehensif tentang data yang diperoleh. Nilai-nilai ini meliputi:

1. Nilai minimum (min) adalah nilai terkecil yang diamati dalam sekumpulan data.
2. Nilai maksimum (max) adalah nilai terbesar yang diamati dalam sekumpulan data.
3. Nilai rata-rata (mean) adalah hasil dari penjumlahan semua nilai dalam data dan kemudian dibagi dengan banyaknya data.
4. Nilai median adalah nilai tengah dalam sekumpulan data ketika data diurutkan secara terurut naik atau turun.
5. Simpangan baku atau standar deviasi (stddev) digunakan untuk menguji konsistensi dalam data.

Dalam konteks pengujian tipe performa, pengukuran dilakukan berdasarkan waktu rendering yang diperlukan untuk menyelesaikan tugas atau operasi tertentu. Waktu rendering ini diukur dengan menggunakan satuan millisecond (ms). Semakin kecil nilai waktu rendering, semakin baik performa CPU dalam menyelesaikan tugas dengan cepat.

Sementara itu, dalam pengujian tipe memori, pengukuran dilakukan berdasarkan jumlah memori RAM yang dibutuhkan oleh sistem atau aplikasi saat dieksekusi. Pengukuran ini menggunakan satuan megabyte (MB). Performa yang lebih baik dalam pengujian tipe memori akan ditunjukkan oleh kebutuhan memori yang lebih rendah.

#### 1) Performa

TABEL I  
HASIL PENGUJIAN SIMULASI MEMBUAT BARIS PADA TABEL KOSONG

	Membuat 1k		Membuat 10k		Membuat 20k	
	REACT	VUE	REACT	VUE	REACT	VUE
Min	151.71	139.88	1760.1	1461.736	4275.947	3111.474
Max	158.23	151.656	5888.796	3874.217	4360.62	3184.903
Mean	155.049	146.211	2541.752	1849.737	4318.853	3139.1923
Median	154.703	145.277	1791.433	1510.441	4310.765	3126.367
Stddev	2.292	4.4005	1573.189	797.843	29.4997	24.3294

TABEL II  
HASIL PENGUJIAN SIMULASI MENGGANTI BARIS PADA TABEL KOSONG

	Mengganti 1k		Mengganti 10k		Mengganti 20k	
	REACT	VUE	REACT	VUE	REACT	VUE
Min	155.877	141.061	1729.943	1460.424	4226.047	3055.054
Max	179.917	150.617	2165.197	1489.671	4377.892	3135.239
Mean	165.199	146.9294	1820.119	1473.866	4325.168	3102.752
Median	164.442	148.2645	1772.729	1474.775	4333.184	3109.305
Stddev	8.568	3.4253	130.9729	8.7366	40.8241	21.836

TABEL III  
HASIL PENGUJIAN SIMULASI MEMPERBARUI BARIS PADA TABEL KOSONG

	Memperbarui 1k		Memperbarui 10k		Memperbarui 20k	
	REACT	VUE	REACT	VUE	REACT	VUE
Min	885.937	822.749	6113.769	5023.292	12505.85	10959.489
Max	1093.733	973.597	6846.022	5769.152	13550.098	12732.088
Mean	1019.554	923.949	6494.506	5363.883	13061.385	11983.744
Median	1027.8515	934.827	6521.176	5281.845	13131.672	12084.152
Stddev	61.3513	45.601	236.9753	287.3626	373.7517	567.5061

TABEL IV  
HASIL PENGUJIAN SIMULASI MEMILIH BARIS PADA TABEL KOSONG

	REACT	VUE
Min	885.937	822.749
Max	1093.733	973.597
Mean	1019.5544	923.9492
Median	1027.8515	934.827
Stddev	61.3513	45.601

Keterangan	Framework	Memory (MB)
Membuat 1000 baris	React	4.5144
	Vue	3.3348
Membuat 10.000 baris	React	32.1444
	Vue	23.7110
Membuat 20.000 baris	React	62.7771
	Vue	46.3715

TABEL V  
HASIL PENGUJIAN SIMULASI MENUKAR BARIS PADA TABEL KOSONG

	Menukar 1k		Menukar 10k		Menukar 20k	
	REACT	VUE	REACT	VUE	REACT	VUE
Min	619.910	92.306	8145.782	843.412	18471.924	1728.427
Max	688.882	112.348	9628.483	979.712	20342.810	1866.703
Mean	648.488	100.117	8806.793	914.8733	19470.680	1804.820
Median	647.579	98.831	8789.494	910.598	19422.664	1807.324
Stddev	20.387	5.463	536.052	50.4273	691.041	45.737

TABEL VI  
HASIL PENGUJIAN SIMULASI MENGHAPUS 1 BARIS PADA TABEL KOSONG

	REACT	VUE
Min	168.044	172.804
Max	203.452	201.779
Mean	185.6263	186.7533
Median	182.1605	185.9205
Stddev	12.0559	10.4462

TABEL VII  
HASIL PENGUJIAN SIMULASI MENAMBAH BARIS PADA TABEL KOSONG

	Menambah 1k		Menambah 10k	
	REACT	VUE	REACT	VUE
Min	395.786	346.738	4052.234	3402.297
Max	501.665	410.901	9879.442	3618.511
Mean	419.5457	369.236	5158.1976	3510.2866
Median	407.3465	361.2305	4468.5705	3505.54
Stddev	31.2115	23.407	1776.3462	75.7935

TABEL VIII  
HASIL PENGUJIAN SIMULASI MENGHAPUS BARIS PADA TABEL KOSONG

	Menghapus 1k		Menghapus 10k		Menghapus 20k	
	REACT	VUE	REACT	VUE	REACT	VUE
Min	126.735	144.246	1288.706	1295.939	2921.666	2751.416
Max	170.984	194.661	1389.754	1449.661	3432.27	3355.657
Mean	159.0529	176.0347	1347.0373	1399.8882	3125.5624	2926.0282
Median	160.8245	176.9685	1353.342	1424.1895	3062.3885	2816.641
Stddev	12.6007	16.055	38.9160	49.3931	189.4778	212.7504

## 2) Memori

TABEL IX  
HASIL PENGUJIAN MEMORY PAGE LOAD

Memory (MB)	
REACT	VUE
1.117508888244629	0.8392229080200195

TABEL X  
HASIL PENGUJIAN PENGGUNAAN MEMORY SETELAH MEMBUAT BARIS

TABEL XI  
HASIL PENGUJIAN PENGGUNAAN MEMORY SETELAH MEMPERBARUI BARIS

Keterangan	Framework	Memory (MB)
Memperbarui 1000 baris	React	5.0216
	Vue	3.3706
Memperbarui 10.000 baris	React	36.2729
	Vue	23.6794
Memperbarui 20.000 baris	React	71.0951
	Vue	46.2360

TABEL XII  
HASIL PENGUJIAN PENGGUNAAN MEMORY SETELAH MENGHAPUS BARIS

Keterangan	Framework	Memory (MB)
Menghapus 1000 baris	React	1.8992
	Vue	1.1867
Menghapus 10.000 baris	React	4.06999
	Vue	1.15889
Menghapus 20.000 baris	React	6.52697
	Vue	1.16522

### 3) Startup

TABEL XIII  
HASIL PENGUJIAN STARTUP TIME: WAKTU INTERAKTIF

Framework	React	Vue
Min (ms)	2631.64	2106.685
Max (ms)	2652.716	2129.745
Mean (ms)	2642.897333	2115.201667
Median (ms)	2644.336	2109.175
Stddev	10.611	12.656

TABEL XIV  
HASIL PENGUJIAN STARTUP TIME: WAKTU MULAI

Framework	React	Vue
Min (ms)	127.044	29.160
Max (ms)	147.728	39.496
Mean (ms)	134.9153	33.480
Median (ms)	132.510	33.510
Stddev	8.562535	3.61996

TABEL XV  
HASIL PENGUJIAN STARTUP TIME: MAIN THREAD COST

Framework	React	Vue
Min (ms)	504.688	447.356
Max (ms)	5152.996	467.456
Mean (ms)	2058.844	456.277
Median (ms)	518.848	454.020
Stddev	2679.624	10.238

TABEL XVI  
HASIL PENGUJIAN STARTUP TIME: MAIN THREAD COST

Keterangan	Framework	Memory (MB)
Startup Total	React	281.096

Bytes	Vue	197.369
-------	-----	---------

### B. Pembahasan

#### 1) Analisa Pengujian Performa Framework React dan Vue

Sebelum melakukan uji t independen, data pada tabel 1 sampai dengan tabel 8, yang merupakan data hasil pengujian tipe performa harus diuji dahulu untuk memastikan bahwa distribusi data normal. Penulis menggunakan uji Kolmogorov-Smirnov untuk menentukan distribusi data. Kolmogorov-Smirnov Test satu sampel adalah metode statistik yang digunakan untuk menguji apakah sebuah sampel data berasal dari distribusi tertentu, dalam hal ini, apakah sampel data ini berasal dari distribusi normal. Test ini berguna untuk mengetahui apakah distribusi data yang akan diuji serupa dengan distribusi normal, atau mungkin mengikuti distribusi yang berbeda.

#### One-Sample Kolmogorov-Smirnov Test

N		490
Normal Parameters <sup>a,b</sup>	Mean	2569.33767
	Std. Deviation	3782.282428
Most Extreme Differences	Absolute	.268
	Positive	.268
	Negative	-.256
Test Statistic		.268
Asymp. Sig. (2-tailed) <sup>c</sup>		<.001
Monte Carlo Sig. (2-tailed) <sup>d</sup>	Sig.	.000
	99% Confidence Interval	Lower Bound .000
		Upper Bound .000

Gbr. 2 Hasil uji Kolmogorov-Smirnov pada data performa

Berdasarkan gbr. 2, nilai Asymp. Sig. (2-tailed) dan Monte Carlo Sig. (2-tailed) adalah 0.000. Angka ini menunjukkan bahwa p-value yang dihasilkan sangat kecil, jauh lebih kecil dari tingkat signifikansi umum (misalnya 0.05). Karena p-value sangat rendah, maka dapat disimpulkan bahwa data yang dianalisis tidak berasal dari distribusi normal. Oleh karena itu, uji t independen tidak dapat dilakukan. Sehingga penulis menggunakan uji Mann-Whitney U (juga dikenal sebagai uji Wilcoxon Rank-Sum)

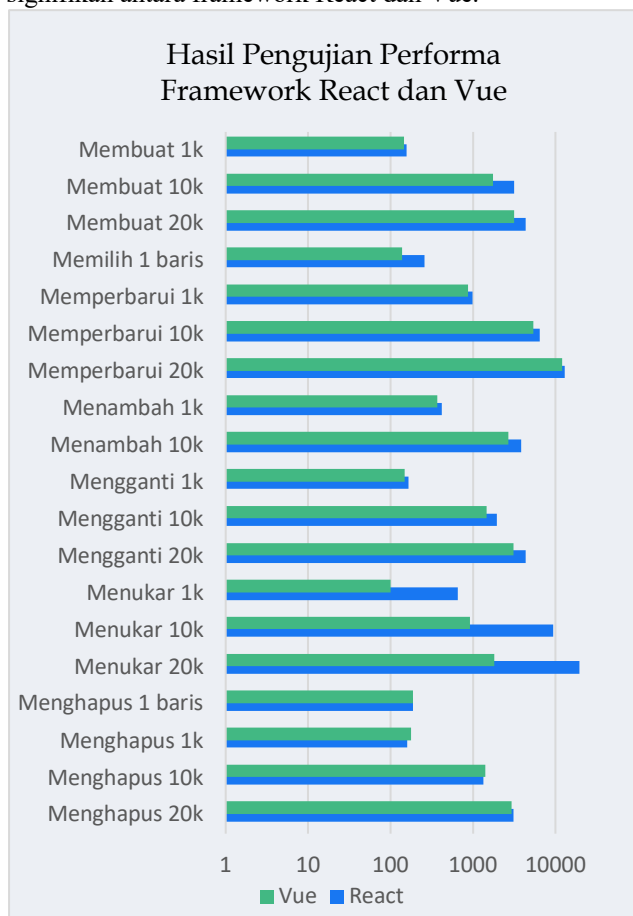
Uji Mann-Whitney U adalah metode statistik non-parametrik yang digunakan untuk membandingkan median dua kelompok data yang independen. Uji ini juga dikenal sebagai uji Wilcoxon Rank-Sum karena menggunakan perangkikan data sebagai dasar perhitungannya. Uji Mann-Whitney U sangat berguna ketika data tidak memenuhi asumsi distribusi normal yang diperlukan untuk uji t-test atau ketika ingin membandingkan dua kelompok data yang independen tanpa mengasumsikan bentuk distribusi tertentu.

### Test Statistics<sup>a</sup>

Mann-Whitney U	24463.000
Wilcoxon W	51028.000
Z	-3.476
Asymp. Sig. (2-tailed)	<.001

Gbr. 3 Hasil uji Mann-Whitney pada data performa

Dengan nilai p-value sebesar 0.001, pada gbr. 3, yang lebih kecil dari tingkat signifikansi umum (misalnya 0.05), maka dapat disimpulkan bahwa median dari kedua kelompok data berbeda secara signifikan. Hal ini menunjukkan bahwa ada perbedaan performa yang signifikan antara framework React dan Vue.



Gbr. 4 Grafik hasil pengujian performa Framework React dan Vue

Berdasarkan grafik hasil pengujian tipe performa yang dapat dilihat pada gbr. 4, dapat disimpulkan bahwa Vue menunjukkan kinerja yang lebih baik dibandingkan dengan React dalam hal waktu rendering. Pada sebagian besar pengujian, Vue memiliki waktu rendering yang lebih singkat dibandingkan dengan React, menunjukkan kemampuan Vue dalam merespons perubahan dengan lebih

cepat. Hal ini menunjukkan bahwa Vue dapat memberikan pengalaman pengguna yang lebih responsif dan lancar.

### 2) Analisa Pengujian Efisiensi Memori Framework React dan Vue

Seperti sebelumnya, data hasil pengujian tipe memori pada tabel 9 sampai dengan tabel 12 harus diuji dahulu untuk memastikan bahwa distribusi data normal sebelum melakukan uji t independent.

#### One-Sample Kolmogorov-Smirnov Test

		Memory (MB)
N		32
Normal Parameters <sup>a,b</sup>	Mean	20.82870395
	Std. Deviation	21.59448630
Most Extreme Differences	Absolute	.246
	Positive	.246
	Negative	-.177
Test Statistic		.246
Asymp. Sig. (2-tailed) <sup>c</sup>		<.001
Monte Carlo Sig. (2-tailed) <sup>d</sup>	Sig.	<.001
	99% Confidence Interval	Lower Bound .000
		Upper Bound .000

Gbr. 5 Hasil uji Kolmogorov-Smirnov pada data penggunaan memori

Karena nilai p-value (Asymp. Sig. dan Monte Carlo Sig.) pada gbr. 5 kurang dari tingkat signifikansi umum (misalnya 0.05), maka dapat disimpulkan bahwa distribusi data tidak berasal dari distribusi normal. Oleh karena itu, perlu dilakukan Uji Mann-Whitney U untuk menentukan perbedaan kedua sampel.

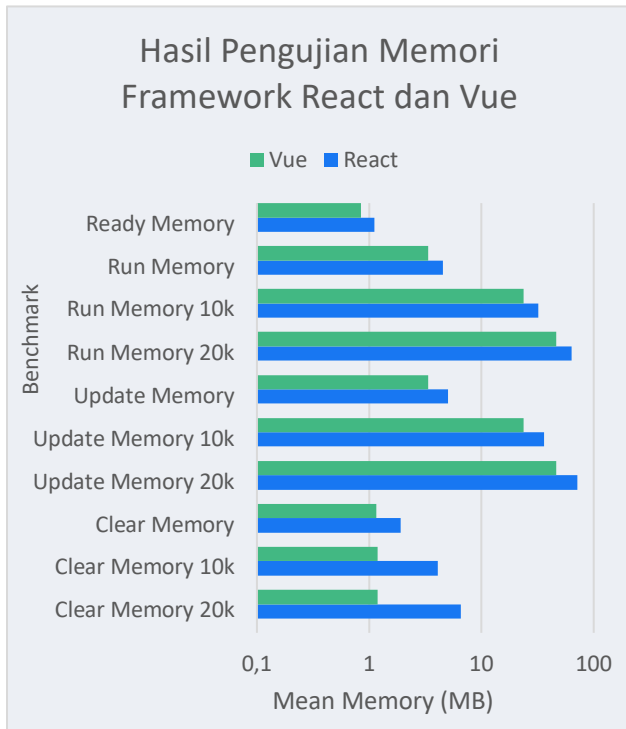
### Test Statistics<sup>a</sup>

		Memory (MB)
Mann-Whitney U		93.000
Wilcoxon W		229.000
Z		-1.319
Asymp. Sig. (2-tailed)		.187
Exact Sig. [2*(1-tailed Sig.)]		.196 <sup>b</sup>

Gbr. 6 Hasil uji Mann-Whitney pada data penggunaan memori

Dengan nilai p-value (Asymp. Sig. dan Exact Sig.) pada gbr. 6 lebih besar dari tingkat signifikansi yang umum digunakan (misalnya 0.05), maka berdasarkan hasil uji Mann-Whitney, dapat disimpulkan bahwa tidak ada perbedaan yang signifikan dalam penggunaan memori antara framework React dan Vue.





Gbr. 7 Grafik hasil pengujian memori pada framework React dan Vue

Berdasarkan grafik hasil pengujian tipe memori pada gbr. 7, dapat disimpulkan bahwa Vue memiliki keunggulan dalam penggunaan sumber daya memori (RAM) yang lebih efisien dibandingkan dengan React. Pengujian menunjukkan bahwa Vue membutuhkan lebih sedikit memori dalam operasionalnya, baik dalam situasi dengan jumlah data yang besar maupun kecil walaupun tidak signifikan. Hal ini menandakan bahwa Vue dapat berjalan dengan lebih lancar dan efisien, terutama pada perangkat dengan sumber daya terbatas atau spesifikasi rendah.

### 3) Analisa Pengujian Startup Framework React dan Vue

Seperti kedua analisis sebelumnya, data hasil pengujian tipe startup pada tabel 13 sampai dengan tabel 16 harus diuji untuk memastikan bahwa distribusi data normal sebelum melakukan uji t independent.

### One-Sample Kolmogorov-Smirnov Test

		Memory (MB)
N		48
Normal Parameters <sup>a,b</sup>	Mean	922.0145579
	Std. Deviation	1192.321621
Most Extreme Differences	Absolute	.347
	Positive	.347
	Negative	-.227
Test Statistic		.347
Asymp. Sig. (2-tailed) <sup>c</sup>		<.001
Monte Carlo Sig. (2-tailed) <sup>d</sup>	Sig.	.000
	99% Confidence Interval	Lower Bound .000
		Upper Bound .000

a. Test distribution is Normal.

b. Calculated from data.

c. Lilliefors Significance Correction.

d. Lilliefors' method based on 10000 Monte Carlo samples with starting seed 1502173562.

Gbr. 8 Hasil uji Kolmogorov-Smirnov pada data startup

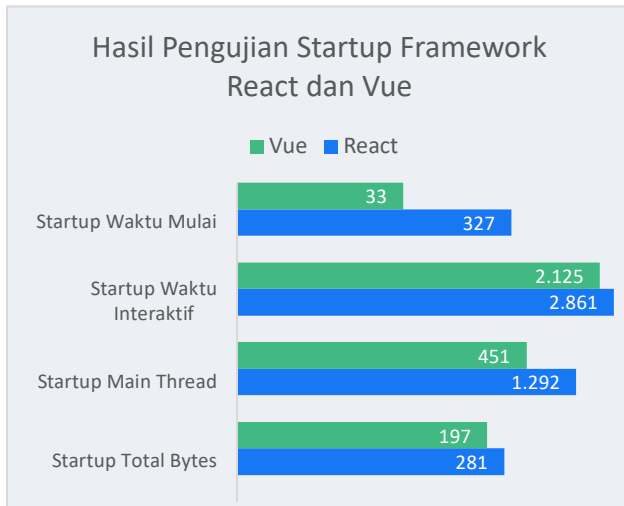
Berdasarkan hasil uji diatas, nilai p-value (Asymp. Sig. dan Monte Carlo Sig.) pada gbr. 8 adalah 0.000. Angka ini menunjukkan bahwa p-value sangat kecil, jauh lebih kecil dari tingkat signifikansi umum (misalnya 0.05). Karena p-value sangat rendah, maka tidak ada cukup bukti statistik untuk menyatakan bahwa data berasal dari distribusi normal. Oleh karena itu, perlu dilakukan Uji Mann-Whitney U untuk menentukan perbedaan kedua sampel.

### Test Statistics<sup>a</sup>

Mann-Whitney U	210.000
Wilcoxon W	510.000
Z	-1.610
Asymp. Sig. (2-tailed)	.107

Gbr. 9 Hasil uji Mann-Whitney pada data startup

Karena nilai p-value (Asymp. Sig.) pada gbr. 9 adalah 0.107 dan lebih besar dari tingkat signifikansi umum (misalnya 0.05), maka berdasarkan hasil uji Mann-Whitney dapat disimpulkan bahwa tidak ada perbedaan yang signifikan dalam startup antara framework React dan Vue.



Gbr. 10 Hasil Pengujian Startup Framework React dan Vue

Berdasarkan grafik hasil pengujian tipe startup pada gbr. 10, dapat disimpulkan bahwa Vue memiliki keunggulan dalam hal waktu startup aplikasi yang lebih cepat dibandingkan dengan React. Pengujian menunjukkan bahwa Vue membutuhkan waktu yang lebih sedikit untuk memulai aplikasi, baik itu dalam hal waktu mulai, waktu interaktif, total byte, maupun beban kerja pada thread utama.

#### IV. KESIMPULAN

Dalam penelitian ini, dilakukan analisis perbandingan antara Framework Front-end JavaScript React dan Vue pada pengembangan website. Tujuan utama penelitian ini adalah untuk mengidentifikasi perbedaan performa dan efisiensi antara kedua framework tersebut. Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan beberapa hal sebagai berikut:

##### a. Performa

Dalam hal performa, terdapat perbedaan yang signifikan antara React dan Vue. Vue memiliki kecepatan eksekusi yang lebih cepat dalam melakukan proses tertentu dibandingkan dengan React. Hal ini terlihat dari tabel pengujian diatas. Waktu yang dibutuhkan dalam rendering komponen dan melakukan pembaruan pada website.

##### b. Penggunaan Memori

Dalam hal efisiensi, Vue juga memiliki keunggulan karena menggunakan pendekatan yang lebih ringan dalam mengelola state komponen. Vue memiliki mekanisme yang lebih efisien dalam menangani perubahan data dan mengoptimalkan pembaruan pada tampilan website. Hal ini menyebabkan penggunaan sumber daya yang lebih hemat dan performa yang lebih baik dalam skenario-skenario tertentu.

##### c. Startup

Berdasarkan hasil penelitian ini, dapat disimpulkan bahwa Vue memiliki keunggulan yang signifikan dalam startup performance dibandingkan dengan React. Keunggulan dalam startup membuat Vue menjadi pilihan yang lebih unggul dalam

pengembangan website yang responsif, efisien, dan berorientasi kepada pengguna.

#### REFERENSI

- [1] M. Mehra, M. Kumar, A. Maurya, and C. Sharma, "Mern stack web development," *Annals of the Romanian Society for Cell Biology*, vol. 25, no. 6, pp. 11756-11761, 2021.
- [2] E. Saks, "JavaScript Frameworks: Angular vs React vs Vue," Bachelor, Metropolia University of Applied Sciences, HAAGA-HELIA University of Applied Sciences, 2019.
- [3] T. Wulandari, E. P. Rudiyanto, and M. M. Pasaribu, "Perbandingan Framework Untuk Pengembangan Aplikasi Web Object Oriented Programming Menggunakan Moose," *Jurnal Sains Sosio Humaniora*, vol. 5, no. 2, pp. 929-935, 2021.
- [4] M. Pikkanen, "React and Vue performance comparison," Bachelor, Information Technology, Metropolia University of Applied Sciences, 2021.
- [5] W. Xu, "Benchmark Comparison of JavaScript Frameworks React, Vue, Angular and Svelte," *Hämtad den*, vol. 5, 2021.
- [6] K. Boczkowski and B. Pańczyk, "Comparison of the performance of tools for creating a SPA application interface-React and Vue.js," *Journal of Computer Sciences Institute*, vol. 14, pp. 73-77, 2020.
- [7] S. Aggarwal and J. Verma, "Comparative analysis of MEAN stack and MERN stack," *International Journal of Recent Research Aspects*, vol. 5, no. 1, pp. 127-132, 2018.