

ANALISIS PERBANDINGAN FRAMEWORK FRONT-END JAVASCRIPT SOLIDJS DAN VUEJS PADA PENGEMBANGAN WEBSITE INTERAKTIF

Moch Fikri Khoirurrizal^{1*}, Cepi Rahmat Hidayat², Ruuhwan³

^{1,2,3} Universitas Perjuangan; Jl.Peta No.177, Kahuripan, Kec.Tawang, Tasikmalaya

Riwayat artikel:

Received: 3 Februari 2024

Accepted: 30 Maret 2024

Published: 2 April 2024

Keywords:

Framework;

Front-End;

JavaScript;

SolidJS;

VueJS.

Correspondent Email:

mfikri.khoirurrizal@gmail.com

Abstrak. Penelitian ini membandingkan performa dan efisiensi antara dua framework front-end JavaScript, yaitu SolidJS dan VueJS, dalam pengembangan website interaktif. Menggunakan metode kuantitatif dengan pengujian lapangan, penelitian ini menganalisis waktu proses dan penggunaan sumber daya dari kedua framework. Hasilnya menunjukkan bahwa SolidJS memiliki performa dan efisiensi yang lebih baik daripada VueJS dalam mayoritas skenario pengujian kecuali skenario. Analisis statistik juga menunjukkan perbedaan yang signifikan antara kedua framework. Sebagai rekomendasi, SolidJS layak dipertimbangkan bagi pengembang website interaktif, meskipun VueJS memiliki stabilitas yang lebih baik.

Abstract. This research compares the performance and efficiency between two front-end JavaScript frameworks, namely SolidJS and VueJS, in the development of interactive websites. Using a quantitative method with field testing, the study analyzes the processing time and resource usage of both frameworks. The results indicate that SolidJS outperforms VueJS in the majority of testing scenarios. Statistical analysis also reveals significant differences between the two frameworks. As a recommendation, SolidJS is worth considering for interactive website developers, despite VueJS having better stability.

1. PENDAHULUAN

Pengembangan website interaktif saat ini telah menjadi unsur penting bagi perusahaan, organisasi maupun individu. Website interaktif secara esensial berkaitan erat dengan pengembangan *front-end*, yang memungkinkan pengguna berinteraksi dengan konten situs secara langsung. *Front-End* mencakup elemen tampilan dan antarmuka yang dapat dilihat dan digunakan oleh pengunjung [1].

Pada awalnya, pengembangan front-end menggunakan JavaScript secara sederhana, terutama untuk fungsionalitas dasar. Namun, seiring perkembangan dunia front-end yang memasuki era tuntutan pengguna yang semakin kompleks, maka diperlukan efisiensi dalam penulisan kode untuk mencapai hasil terbaik [2]. Pemilihan framework adalah keputusan

penting untuk membantu pengembang dalam menghadapi tuntutan ini [3].

Dalam konteks pemilihan framework, SolidJS dan VueJS adalah diantaranya yang menojol. SolidJS adalah framework JavaScript baru, dikenal karena memiliki performanya yang baik [4]. VueJS adalah framework JavaScript yang sangat populer, dikenal karena performanya yang sangat baik dan pendekatan pengembangan yang ramah pengguna seperti dibekali *virtual DOM*, *Data Binding* dan lainnya [5].

Kemajuan pengembangan website interaktif menyebabkan peningkatan *traffic user* yang mempengaruhi performa website [6]. Namun di saat yang bersamaan, terdapat tantangan untuk memastikan website interaktif berfungsi dengan baik di berbagai perangkat dan browser yang

berbeda [7]. Pemilihan *framework front-end* menjadi langkah awal yang sangat penting dalam menghadapi tantangan ini [3].

Menurut penelitian terkait yang berjudul “Benchmark Comparison of Javascript Frameworks React, Vue, Angular and Svelte” oleh Wenqing Xu pada tahun 2021, dikatakan VueJS memiliki performa yang lebih baik dibandingkan dengan *framework* lainnya [2].

Penelitian serupa yang berjudul “Analisis Perbandingan Framework Front-End Javascript React dan Vue Pada Pengembangan Website” oleh Farro Axza Febsinatra Sofie dan Anita Qoiriah pada tahun 2023, dikatakan VueJS memiliki performa dan efisiensi yang lebih baik dibandingkan dengan *framework* ReactJS [1].

Berdasarkan uraian diatas, pemilihan sebuah *framework front-end* sangatlah penting karena dapat mempengaruhi performa dan efisiensi website interaktif. Oleh karena itu penelitian ini akan mencermati berbagai sisi, termasuk melibatkan SolidJS. Tujuannya adalah untuk mengetahui apakah sebuah *framework* baru yang masih kalah dalam hal popularitas, bisa bersaing dengan *framework* yang sudah mapan dan memberikan rekomendasi *framework* yang lebih informatif dalam pemilihan *framework front-end JavaScript* untuk pengembangan website interaktif.

2. TINJAUAN PUSTAKA

2.1. Website

Website adalah entitas digital yang terdiri dari berbagai halaman web yang saling terhubung dan menyertakan semua berkas terkait. Halaman-halaman ini sering disebut sebagai "homepage" dan membentuk situs web secara keseluruhan [8]. Halaman web dapat diakses melalui browser dan internet. Website yang berada dalam domain atau subdomain, dikenal sebagai WWW atau World Wide Web. Sebuah website dibangun dengan menggunakan bahasa markup HTML (*Hyper Text Markup Language*) dan dibuka menggunakan protokol internet. Di samping bahasa markup, website seringkali dikombinasikan dengan penggunaan bahasa pemrograman dinamis [9].

2.2. Front-End Development

Front-End Development adalah disiplin dalam pengembangan website yang bertugas untuk merancang, mengembangkan, dan

menjaga komponen-komponen tampilan yang berhubungan langsung dengan pengguna [2].

2.3. JavaScript

JavaScript adalah sebuah bahasa pemrograman yang diterapkan pada pengembangan web untuk menciptakan situs web yang interaktif, dinamis, dan responsif. Bahasa ini berfungsi di sisi klien, berarti dijalankan di peramban web pengguna, memungkinkan setiap pengembang untuk dapat menambahkan berbagai fitur yang dapat meningkatkan pengalaman pengguna [10].

2.4. SolidJS

SolidJS adalah sebuah *framework front-end* yang bertujuan untuk menyediakan solusi yang sangat efisien dan performa yang baik dalam pengembangan aplikasi web [4].

2.5. VueJS

VueJS adalah *framework front-end* yang banyak dikenal dan mudah digunakan. VueJS menawarkan pendekatan pengembangan yang ramah pengguna dan mudah dipelajari [1].

2.6. DOM

DOM adalah antarmuka pemrograman aplikasi untuk mengembangkan aplikasi pada dokumen XML atau HTML. Diterbitkan oleh W3C sebagai spesifikasi antarmuka lintas platform, DOM menyediakan antarmuka terprogram standar di berbagai lingkungan dan aplikasi, dapat diimplementasikan dalam berbagai bahasa pemrograman. DOM memungkinkan penguraian dokumen XML atau HTML terstruktur dengan model objek yang merepresentasikan instruksi, elemen, entitas, dan atribut [2].

2.7. API

Application Programming Interface (API) adalah penerjemah antara *client* dan *server* memungkinkan pengembang agar dapat menggabungkan dua bagian yang berbeda. Tujuannya adalah untuk efisiensi proses pengembangan dengan memisahkan fungsi-fungsi yang ada sehingga pengembang tidak perlu membangun fitur-fitur serupa [11].

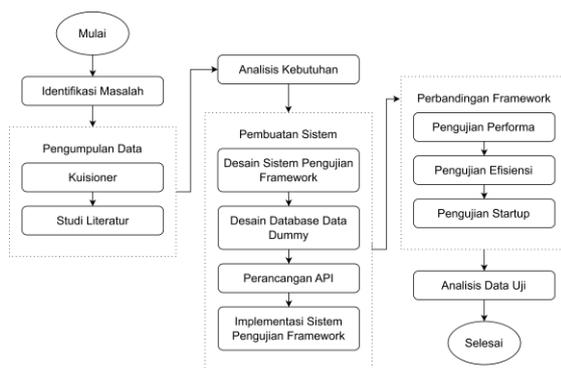
2.8. RESTful API

RESTful API adalah sebuah arsitektur yang digunakan dalam implementasi web service.

Melalui REST API, sistem-sistem dari berbagai jenis dapat berkomunikasi dan bertukar data secara sederhana. Dalam kerangka RESTful API, terdapat REST client yang memungkinkan akses terhadap data atau sumber daya yang terletak di REST server. Setiap sumber daya dibedakan dengan menggunakan ID global atau URI (*Universal Resource Identifiers*). Keistimewaan ini membuat RESTful API menjadi pilihan yang tepat untuk aplikasi yang terhubung dengan perangkat pintar [12].

3. METODE PENELITIAN

Penelitian ini dikerjakan dengan menerapkan metode kuantitatif eksperimen untuk membandingkan performa dan efisiensi SolidJS dan VueJS dalam pengembangan front-end. Langkah-langkah yang terdapat dalam penelitian ini tergambar pada gambar di bawah:



Gambar 3. 1 Metodologi Penelitian

3.1. Identifikasi Masalah

Tahap identifikasi masalah dilakukan untuk mengetahui permasalahan yang ada terkait dengan kebingungan dalam pemilihan framework front-end javascript yang cocok sesuai dengan kebutuhan proyek.

3.2. Pengumpulan Data

Pada tahap ini meliputi pengumpulan data untuk digunakan apabila diperlukan untuk memenuhi kebutuhan data dan informasi. Teknik pengumpulan data meliputi kuesioner dan studi literatur.

3.3. Analisis Kebutuhan

Pada tahap analisis kebutuhan, fokus utama adalah untuk mengidentifikasi kebutuhan pengembangan *front-end* seperti perangkat keras maupun perangkat lunak dan menetapkan spesifikasi yang sama untuk pengujian SolidJS

dan VueJS untuk mendapatkan hasil yang serupa kemudian dilakukan pengujian alternatif untuk sebagian skenario pada perangkat lain untuk mendapatkan kepastian pada pola performa dan efisiensi yang dihasilkan.

- 1) Kebutuhan perangkat keras (*Hardware*)
 - Prosesor: Intel(R) Core(TM) i7-8565U
 - RAM: 8GB
 - Penyimpanan: SSD 512 GB
 - Sistem Operasi: Windows 11 Home Single Language 64-bit
- 2) Kebutuhan perangkat keras alternatif
 - Prosesor: AMD A4-6210 APU with AMD Radeon R3 Graphics
 - RAM: 8GB
 - Penyimpanan: SSD 256 GB
 - Sistem Operasi: Windows 10 Pro 64-bit
- 3) Kebutuhan perangkat lunak (*Software*)
 - Google Chrome 119.0.6045.105 (*Official Build*) (64-bit)
 - Visual Studio Code 1.85.1 (*user setup*)
 - Solid.js 1.8.3
 - Vue.js 3.3.6
 - js-framework-benchmark
 - Chromedriver 119.0.1
 - Lighthouse
 - Node.js 20.9.0 (LTS)
 - NPM 10.1.0
 - IBM SPSS Statistics 29.0.2.0

3.4. Pembuatan Sistem

Sistem ditempatkan pada kondisi pengujian yang serupa, mengikuti skenario dan perangkat keras yang sama. Setelah sistem siap digunakan untuk pengujian, langkah berikutnya ialah melaksanakan pengujian pada setiap skenario yang telah dibangun.

3.5. Perbandingan Framework

Perbandingan framework akan dilaksanakan menggunakan sistem yang telah dibuat sebelumnya untuk mengelola komponen antarmuka pengguna dengan data dummy pada database MySQL menggunakan API. Skenario pengujian terbagi menjadi dua yaitu *DOM Test* dan *API Test*. Pada pengujian startup akan menggunakan Lighthouse sebagai library tambahan untuk melakukan pengujian. Selama proses ini, waktu yang diperlukan untuk setiap skenario, termasuk waktu rendering, akan dicatat secara rinci. Berikut adalah skenario yang akan dilakukan.

Performa:

1. Membuat 10.000, 20.000, 30.000 baris setelah halaman dimuat (*API Test*).
2. Mengganti 10.000, 20.000, 30.000 baris dengan baris yang baru (*API Test*).
3. Memperbarui teks setiap baris pada kelipatan ke-10 pada elemen dengan 10.000, 20.000, 30.000 baris (*DOM Test*).
4. Memilih baris sebagai respons terhadap klik pada baris tersebut (*DOM Test*).
5. Mengganti posisi dua baris pada elemen dengan 10.000, 20.000, 30.000 baris (*DOM Test*).
6. Menghapus satu baris pada komponen tabel (*DOM Test*).
7. Menambahkan 10.000, 20.000, 30.000 baris pada elemen yang sudah memiliki 1.000 baris (*API Test*).
8. Membersihkan semua baris (1.000, 10.000, 20.000) (*DOM Test*).

Efisiensi:

1. Penggunaan memori setelah halaman dimuat.
2. Penggunaan memori setelah membuat 10.000, 20.000, 30.000 baris.
3. Penggunaan memori setelah memperbarui teks setiap baris kelipatan ke-10 pada elemen dengan 10.000, 20.000, 30.000.
4. Penggunaan memori setelah membuat dan membersihkan semua baris (1.000, 10.000, 20.000).

Startup:

1. *TimeToConsistentlyInteractive*: waktu yang diperlukan untuk mencapai tingkat interaktivitas yang konsisten pada halaman setelah proses startup.
2. *ScriptBootUpTime*: total waktu yang diperlukan untuk menguraikan, mengkompilasi, dan mengeksekusi semua skrip pada halaman selama proses startup.
3. *MainThreadWorkCost*: total waktu yang diperlukan untuk menjalankan tugas pada thread utama, meliputi perhitungan tata letak dan tugas lainnya yang terjadi selama proses startup.
4. *TotalByteWeight*: total jumlah byte yang ditransfer melalui jaringan (pasca kompresi) dari berbagai sumber daya yang dimuat ke dalam halaman.

3.6. Analisis Data Uji

Dalam tahap analisis data uji ini, fokus utama diberikan pada pemahaman statistik

deskriptif menggunakan alat bantu IBM SPSS *Statistics*, pembentukan hipotesis, dan rencana awal untuk pengujian lebih lanjut. Statistik deskriptif akan digunakan untuk memberikan gambaran karakteristik umum dari data, termasuk mean, median, dan standar deviasi waktu eksekusi SolidJS dan VueJS berdasarkan hasil pengujian. Data akan dianalisis menggunakan dua hipotesis, yaitu:

- a. H_0 : Tidak ada perbedaan signifikan antara SolidJS dan VueJS
- b. H_0 : Terdapat perbedaan signifikan antara SolidJS dan VueJS

Selanjutnya, rencana awal melibatkan analisis lebih lanjut menggunakan uji t independen untuk mengevaluasi perbedaan signifikan antara waktu eksekusi SolidJS dan VueJS. Namun, pendekatan ini akan dipertimbangkan secara kontekstual dan disesuaikan selama tahap analisis, terutama dengan mempertimbangkan distribusi data. Jika distribusi waktu eksekusi tidak normal, uji t independen tidak akan dijalankan dan, sebagai gantinya, akan beralih ke uji Mann-Whitney U.

4. HASIL DAN PEMBAHASAN

Pengujian framework dikerjakan sebanyak 15 iterasi untuk setiap skenario. Hal ini dilakukan untuk menghindari anomali maupun outlier, sekaligus untuk mengetahui rata-rata bahkan simpangan baku pada hasil pengujian.

4.1. Perancangan Antarmuka

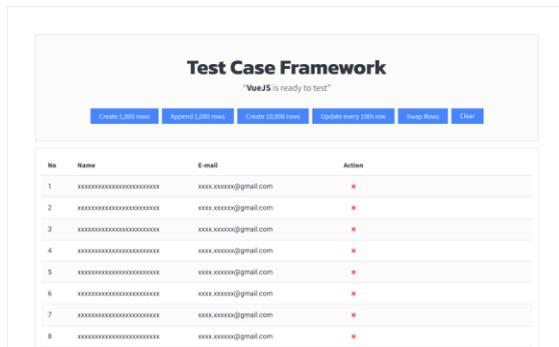
Perancangan antarmuka berfokus untuk merinci struktur halaman tanpa memasukkan data aktual, memberikan gambaran awal terkait tampilan antarmuka yang akan digunakan.

- a. Wireframe



Gambar 4. 1 Wireframe Setelah Pengambilan Data

- b. User Interface



Gambar 4. 2 User Interface Setelah Pengambilan Data

4.2. Desain Database Data Dummy

Database ini akan diisi dengan 35.000 baris data dummy sesuai dengan volume pengujian maksimal, yaitu 30.000, ditambah 5.000 untuk memastikan variasi data yang memadai.

| Users |
|---------------------|
| +id: int(11) |
| +name: varchar(50) |
| +email: varchar(50) |

Gambar 4. 3 Desain database data dummy (users)

4.3. Perancangan API

Perancangan API pada pengujian ini dibuat agar pengujian bisa terlaksana secara optimal karena mencerminkan kebutuhan *real case* di lapangan dalam menggunakan *framework front-end* JavaScript.

Tabel 4. 1 Endpoint API

| | |
|-----------------|---|
| Endpoint | /api.php?limit={limit} |
| Metode | GET |
| Fungsi | Mengambil data dari database dengan jumlah terbatas sesuai parameter <i>limit</i> . |

4.4. Perbandingan Framework

Pengujian framework dilakukan sebanyak 15 kali dalam setiap skenario untuk memastikan hasil pengujian yang konsisten. Hal ini juga membantu untuk mengidentifikasi anomali atau outlier yang mungkin memengaruhi hasil pengujian secara signifikan.

1) Performa

Tabel 4. 2 Hasil Pengujian Membuat Baris Data

| (ms) | Membuat | | | | | |
|------|---------|------|-------|------|-------|------|
| | 10k | | 20k | | 30k | |
| | Solid | Vue | Solid | Vue | Solid | Vue |
| Min | 1488 | 1587 | 2969 | 3360 | 4492 | 5054 |
| Max | 1818 | 1756 | 3743 | 3481 | 4992 | 5454 |
| Mean | 1551 | 1676 | 3288 | 3411 | 4782 | 5148 |

| (ms) | Membuat | | | | | |
|--------|---------|------|-------|------|-------|------|
| | 10k | | 20k | | 30k | |
| | Solid | Vue | Solid | Vue | Solid | Vue |
| Median | 1531 | 1689 | 3281 | 3403 | 4825 | 5126 |
| Stddev | 88 | 52 | 155 | 32 | 156 | 117 |

Tabel 4. 3 Hasil Pengujian Mengganti Baris Baru

| (ms) | Mengganti | | | | | |
|--------|-----------|------|-------|------|-------|------|
| | 10k | | 20k | | 30k | |
| | Solid | Vue | Solid | Vue | Solid | Vue |
| Min | 1563 | 1682 | 3271 | 3458 | 4589 | 5147 |
| Max | 1895 | 1985 | 3837 | 4012 | 5327 | 5280 |
| Mean | 1609 | 1803 | 3370 | 3523 | 4921 | 5217 |
| Median | 1586 | 1803 | 3317 | 3484 | 4972 | 5209 |
| Stddev | 83 | 79 | 151 | 137 | 185 | 35 |

Tabel 4. 4 Hasil Pengujian Memperbarui Baris Data

| (ms) | Memperbarui | | | | | |
|--------|-------------|------|-------|------|-------|------|
| | 10k | | 20k | | 30k | |
| | Solid | Vue | Solid | Vue | Solid | Vue |
| Min | 740 | 909 | 1649 | 1878 | 2442 | 2761 |
| Max | 1497 | 1252 | 2897 | 3059 | 3423 | 4402 |
| Mean | 896 | 964 | 1978 | 2107 | 2712 | 3117 |
| Median | 881 | 950 | 1841 | 1985 | 2611 | 2834 |
| Stddev | 178 | 82 | 357 | 304 | 275 | 617 |

Tabel 4. 5 Hasil Pengujian Memilih Baris Data

| (ms) | Memilih | | | | | |
|--------|---------|-------|-------|-------|-------|-------|
| | 10k | | 20k | | 30k | |
| | Solid | Vue | Solid | Vue | Solid | Vue |
| Min | 99.4 | 121.4 | 202.0 | 234.3 | 282.5 | 340.0 |
| Max | 119.4 | 150.2 | 237.1 | 264.7 | 324.9 | 368.3 |
| Mean | 109.2 | 132.5 | 214.0 | 249.6 | 303.3 | 354.3 |
| Median | 109.4 | 130.1 | 211.4 | 248.7 | 302.4 | 354.2 |
| Stddev | 5.6 | 8.1 | 9.9 | 9.3 | 9.6 | 8.5 |

Tabel 4. 6 Hasil Pengujian Menukar Baris Data

| (ms) | Menukar | | | | | |
|--------|---------|------|-------|------|-------|------|
| | 10k | | 20k | | 30k | |
| | Solid | Vue | Solid | Vue | Solid | Vue |
| Min | 950 | 1080 | 1981 | 2216 | 2938 | 3218 |
| Max | 1179 | 1172 | 2405 | 2396 | 3302 | 4165 |
| Mean | 1066 | 1122 | 2266 | 2328 | 3221 | 3377 |
| Median | 1076 | 1121 | 2288 | 2342 | 3227 | 3334 |
| Stddev | 53 | 22 | 93 | 52 | 90 | 223 |

Tabel 4. 7 Hasil Pengujian Menghapus Satu Baris Data

| (ms) | Menghapus Satu | | | | | |
|--------|----------------|-------|-------|------|-------|------|
| | 10k | | 20k | | 30k | |
| | Solid | Vue | Solid | Vue | Solid | Vue |
| Min | 665.5 | 907.7 | 1439 | 1918 | 2047 | 2528 |
| Max | 788.9 | 997.9 | 1627 | 2187 | 2313 | 2869 |
| Mean | 704.2 | 952.1 | 1581 | 1986 | 2238 | 2748 |
| Median | 692.8 | 950.8 | 1607 | 1984 | 2271 | 2810 |
| Stddev | 38.8 | 23.9 | 58 | 62 | 73 | 115 |

Tabel 4. 8 Hasil Pengujian Menambah Baris Data

| (ms) | Menambah | | | | | |
|--------|----------|------|-------|------|-------|------|
| | 10k | | 20k | | 30k | |
| | Solid | Vue | Solid | Vue | Solid | Vue |
| Min | 1582 | 1632 | 3047 | 3185 | 4467 | 4784 |
| Max | 1723 | 1698 | 3231 | 4996 | 5007 | 5573 |
| Mean | 1631 | 1662 | 3096 | 3544 | 4664 | 5038 |
| Median | 1627 | 1658 | 3073 | 3488 | 4567 | 4902 |
| Stddev | 37 | 20 | 56 | 440 | 212 | 254 |

Tabel 4. 9 Hasil Pengujian Menghapus Baris Data

| (ms) | Menghapus | | | | | |
|--------|-----------|------|-------|-------|-------|-------|
| | 10k | | 20k | | 30k | |
| | Solid | Vue | Solid | Vue | Solid | Vue |
| Min | 66.7 | 65.5 | 128.9 | 124.4 | 193.5 | 187.1 |
| Max | 78.3 | 77.3 | 166.7 | 151.4 | 234.6 | 234.6 |
| Mean | 72.0 | 71.6 | 143.3 | 131.3 | 211.7 | 207.2 |
| Median | 72.0 | 71.2 | 142.3 | 129.9 | 213.0 | 203.5 |
| Stddev | 3.6 | 2.5 | 8.8 | 7.3 | 10.5 | 14.5 |

2) Efisiensi

Tabel 4. 10 Hasil Pengujian Page Load

| (MB) | Memory | |
|--------|-------------|-------------|
| | Solid | Vue |
| Min | 0.528544426 | 0.785040855 |
| Max | 0.574753761 | 0.808870316 |
| Mean | 0.539505768 | 0.788134638 |
| Median | 0.532431602 | 0.785803795 |
| Stddev | 0.015071743 | 0.006123378 |

Tabel 4. 11 Hasil Pengujian Membuat Baris Data

| (MB) | Membuat | | | | | |
|--------|---------|------|-------|------|-------|-------|
| | 10k | | 20k | | 30k | |
| | Solid | Vue | Solid | Vue | Solid | Vue |
| Min | 29.0 | 34.9 | 57.1 | 68.9 | 85.3 | 102.7 |
| Max | 29.1 | 35.0 | 57.5 | 69.1 | 85.9 | 103.0 |
| Mean | 29.0 | 34.9 | 57.2 | 69.1 | 85.5 | 102.9 |
| Median | 29.0 | 34.9 | 57.2 | 69.1 | 85.4 | 102.8 |
| Stddev | 0.01 | 0.01 | 0.10 | 0.05 | 0.16 | 0.10 |

Tabel 4. 12 Hasil Pengujian Memperbarui Baris Data

| (MB) | Memperbarui | | | | | |
|--------|-------------|------|-------|------|-------|-------|
| | 10k | | 20k | | 30k | |
| | Solid | Vue | Solid | Vue | Solid | Vue |
| Min | 29.1 | 34.9 | 57.2 | 68.8 | 85.8 | 102.4 |
| Max | 29.1 | 34.9 | 57.3 | 68.9 | 86.5 | 102.7 |
| Mean | 29.1 | 34.9 | 57.3 | 68.9 | 86.0 | 102.6 |
| Median | 29.1 | 34.9 | 57.3 | 68.9 | 85.9 | 102.6 |
| Stddev | 0.01 | 0.00 | 0.04 | 0.02 | 0.21 | 0.10 |

Tabel 4. 13 Hasil Pengujian Menghapus Baris Data

| (MB) | Menghapus | | | | | |
|--------|-----------|------|-------|------|-------|-------|
| | 10k | | 20k | | 30k | |
| | Solid | Vue | Solid | Vue | Solid | Vue |
| Min | 0.8 | 34.7 | 0.8 | 68.3 | 0.8 | 101.9 |
| Max | 0.8 | 34.9 | 0.8 | 68.4 | 0.8 | 102.1 |
| Mean | 0.8 | 34.8 | 0.8 | 68.4 | 0.8 | 102.0 |
| Median | 0.8 | 34.9 | 0.8 | 68.3 | 0.8 | 102.0 |
| Stddev | 0.02 | 0.05 | 0.02 | 0.05 | 0.01 | 0.06 |

3) Startup

Tabel 4. 14 Hasil Pengujian Time Waktu Interaktif

| (ms) | Waktu Interaktif | |
|--------|------------------|---------|
| | Solid | Vue |
| Min | 1453.92 | 2130.50 |
| Max | 2663.20 | 2818.07 |
| Mean | 2137.73 | 2366.04 |
| Median | 2057.62 | 2277.71 |
| Stddev | 500.54 | 238.40 |

Tabel 4. 15 Hasil Pengujian Time Waktu Mulai

| (ms) | Waktu Mulai | |
|--------|-------------|----------|
| | Solid | Vue |
| Min | 16.208 | 75.516 |
| Max | 66.804 | 87.684 |
| Mean | 21.01707 | 81.17867 |
| Median | 17.312 | 80.856 |
| Stddev | 16.208 | 75.516 |

Tabel 4. 16 Hasil Pengujian Time Main Thread Cost

| (ms) | Main Thread Cost | |
|--------|------------------|-----------|
| | Solid | Vue |
| Min | 320.632 | 345.296 |
| Max | 344.028 | 370.828 |
| Mean | 334.27947 | 357.30613 |
| Median | 334.748 | 355.876 |
| Stddev | 5.46600 | 7.37962 |

Tabel 4. 17 Hasil Pengujian Time Total Byte Weight

| Keterangan | Framework | Memory (MB) |
|---------------|-----------|-------------|
| Startup Total | Solid | 266.3642578 |
| Bytes | Vue | 311.8525391 |

4) Performa Alternatif

Tabel 4. 18 Hasil Pengujian Alternatif Membuat Baris Data

| (ms) | Membuat 20k | |
|--------|-------------|-------------|
| | Solid | Vue |
| Min | 17811.148 | 19186.055 |
| Max | 20248.132 | 21437.105 |
| Mean | 19035.54207 | 19956.01587 |
| Median | 19101.787 | 19984.095 |
| Stddev | 877.567746 | 517.4434065 |

Tabel 4. 19 Hasil Pengujian Alternatif Memperbarui Baris Data

| (ms) | Memperbarui 20k | |
|--------|-----------------|-----------|
| | Solid | Vue |
| Min | 5681.502 | 6454.055 |
| Max | 9154.025 | 12663.165 |
| Mean | 6677.5554 | 7934.8756 |
| Median | 6970.737 | 7700.797 |
| Stddev | 951.2147 | 1350.1388 |

Tabel 4. 20 Hasil Pengujian Alternatif Menghapus Baris Data

| (ms) | Menghapus 20k | |
|------|---------------|---------|
| | Solid | Vue |
| Min | 697.288 | 665.419 |

| (ms) | Menghapus 20k | |
|--------|---------------|-------------|
| | Solid | Vue |
| Max | 811.432 | 766.734 |
| Mean | 733.5774667 | 717.9493333 |
| Median | 711.719 | 714.351 |
| Stddev | 43.8585 | 43.9146 |

5) Efisiensi Alternatif

Tabel 4. 21 Hasil Pengujian Alternatif Menghapus Baris Data

| (ms) | Menghapus 20k | |
|--------|---------------|-------------|
| | Solid | Vue |
| Min | 0.671934128 | 68.21344185 |
| Max | 0.783432961 | 68.50563526 |
| Mean | 0.699191157 | 68.3410183 |
| Median | 0.681660652 | 68.35668564 |
| Stddev | 0.038913585 | 0.067111058 |

6) Startup Alternatif

Tabel 4. 22 Hasil Pengujian Alternatif Time Waktu Interaktif

| (ms) | Waktu Interaktif | |
|--------|------------------|-------------|
| | Solid | Vue |
| Min | 2101.7495 | 2755.7465 |
| Max | 2230.264 | 2783.732 |
| Mean | 2146.2392 | 2768.315067 |
| Median | 2147.183 | 2770.0845 |
| Stddev | 33.11927058 | 7.073788031 |

4.5. Analisis Data Uji

1) Performa

Sebelum memulai analisis data performa dari tabel 4.2 hingga tabel 4.9 dengan menerapkan metode uji t independen, terlebih dahulu dilakukan uji normalitas dengan menerapkan metode One Sample Kolmogorov-Smirnov pada dataset tersebut.

| One-Sample Kolmogorov-Smirnov Test | | Millisecond (ms) | |
|--|-------------------------|------------------|------|
| N | | 240 | |
| Normal Parameters ^{a,b} | Mean | 2075.9566 | |
| | Std. Deviation | 1261.77395 | |
| Most Extreme Differences | Absolute | .174 | |
| | Positive | .174 | |
| | Negative | -.154 | |
| Test Statistic | | .174 | |
| Asymp. Sig. (2-tailed) ^c | | <.001 | |
| Monte Carlo Sig. (2-tailed) ^d | Sig. | <.001 | |
| | 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 299883525.

Gambar 4. 4 Hasil uji Kolmogorov-Smirnov pada data performa

Sesuai dengan hasil dari gambar 4.4, nilai Asymp. Sig. (2-tailed) dan Monte Carlo Sig. (2-tailed) adalah <0.001, menyatakan bahwa p-

value yang didapat lebih rendah dari tingkat signifikansi umum (contohnya 0.05). Dikarenakan p-value yang rendah tersebut, artinya data yang dianalisis tidak terdistribusi normal. Dengan begitu, uji t independen batal untuk dapat dilaksanakan, dan akan digunakan uji Mann-Whitney U sebagai penggantinya.

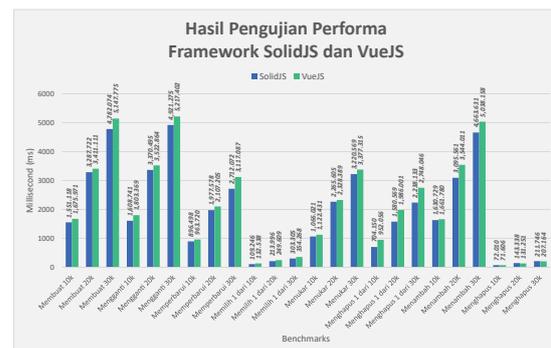
Berikut adalah tahap uji perbandingan menggunakan metode MannWhitney U pada data hasil pengujian untuk membandingkan dua kelompok data yang bersifat independen.

| Test Statistics ^a | |
|------------------------------|------------------|
| | Millisecond (ms) |
| Mann-Whitney U | 5987.000 |
| Wilcoxon W | 13247.000 |
| Z | -2.256 |
| Asymp. Sig. (2-tailed) | .024 |

a. Grouping Variable: Frameworks

Gambar 4. 5 Hasil uji Mann-Whitney U pada data performa

Berdasarkan Gambar 4.5, nilai Asymp. Sig. (2-tailed) adalah 0.024, menandakan bahwa p-value yang didapatkan lebih rendah dari tingkat signifikansi umum (contohnya 0.05). Dengan p-value yang rendah, hipotesis nol (H0) dapat ditolak, sehingga hipotesis alternatif (H1) dapat diterima. Kesimpulannya adalah bahwa kedua kelompok data menunjukkan perbedaan yang signifikan dalam performa antara framework SolidJS dan VueJS.



Gambar 4. 6 Grafik hasil pengujian performa framework SolidJS dan VueJS

Hasil grafik pengujian performa, seperti yang terlihat pada Gambar 4.6, memperlihatkan bahwa SolidJS memiliki kinerja yang optimal dibanding VueJS dalam hal waktu rendering. SolidJS secara konsisten memberikan waktu rendering yang lebih singkat dibandingkan dengan VueJS dalam sebagian besar pengujian, kecuali saat menghapus seluruh data, SolidJS menunjukkan waktu rendering yang lebih cepat

dibandingkan dengan VueJS, menandakan kemampuan SolidJS dalam merespon perubahan dengan lebih cepat.

2) Efisiensi

Sebelum memulai analisis data efisiensi dari tabel 4.10 hingga tabel 4.13 dengan menerapkan metode uji t independen, terlebih dahulu dilakukan uji normalitas dengan menerapkan metode One Sample Kolmogorov-Smirnov pada dataset tersebut.

One-Sample Kolmogorov-Smirnov Test

| | | Memory (MB) | |
|--|-------------------------|-------------|------|
| N | | 120 | |
| Normal Parameters ^{a,b} | Mean | 40.3700 | |
| | Std. Deviation | 31.17750 | |
| Most Extreme Differences | Absolute | .330 | |
| | Positive | .273 | |
| | Negative | -.330 | |
| Test Statistic | | .330 | |
| Asymp. Sig. (2-tailed) ^c | | <.001 | |
| Monte Carlo Sig. (2-tailed) ^d | Sig. | <.001 | |
| | 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 299883525.

Gambar 4. 7 Hasil uji Kolmogorov-Smirnov pada data Efisiensi

Sesuai dengan Gambar 4.7, nilai Asymp. Sig. (2-tailed) dan Monte Carlo Sig. (2-tailed) menunjukkan <0.001, menyatakan bahwa p-value yang didapat lebih rendah dibandingkan dengan tingkat signifikansi umum (contohnya 0.05). Dengan p-value yang lebih rendah, artinya data yang dianalisis tidak terdistribusi normal. Dengan begitu, uji t independen batal dilaksanakan, dan sebagai alternatif, uji Mann-Whitney U akan digunakan.

Test Statistics^a

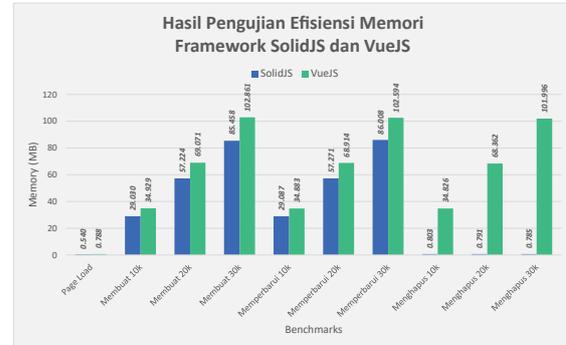
| | Memory (MB) |
|------------------------|-------------|
| Mann-Whitney U | 579.000 |
| Wilcoxon W | 2409.000 |
| Z | -6.409 |
| Asymp. Sig. (2-tailed) | <.001 |

a. Grouping Variable: Frameworks

Gambar 4. 8 Hasil uji Mann-Whitney U pada data efisiensi

Berdasarkan hasil dari Gambar 4.8, nilai Asymp. Sig. (2-tailed) menunjukkan <0.001, menandakan bahwa p-value yang dihasilkan sangat kecil dibandingkan dengan tingkat signifikansi umum (contohnya 0.05). Dengan p-value yang rendah, hipotesis nol (H0) ditolak dan hipotesis alternatif (H1) diterima. Dengan

demikian, terdapat perbedaan signifikan antara kedua kelompok data, menunjukkan adanya perbedaan efisiensi yang signifikan antara framework SolidJS dan VueJS.



Gambar 4. 9 Grafik hasil pengujian efisiensi framework SolidJS dan VueJS

Grafik hasil pengujian efisiensi pada Gambar 4.9 menunjukkan bahwa SolidJS memiliki tingkat efisiensi penggunaan sumber daya memori (RAM) yang lebih baik daripada VueJS. SolidJS membutuhkan jumlah memori yang lebih rendah dalam operasinya, baik pada saat skenario melibatkan volume data yang besar maupun yang kecil. Terutama pada skenario penghapusan data, SolidJS menunjukkan perbedaan yang signifikan.

3) Startup

Sebelum memulai analisis data startup dari tabel 4.14 hingga tabel 4.17 dengan menerapkan metode uji t independen, terlebih dahulu dilakukan uji normalitas dengan menerapkan metode One Sample Kolmogorov-Smirnov pada dataset tersebut.

One-Sample Kolmogorov-Smirnov Test

| | | Millisecond (ms) | |
|--|-------------------------|------------------|------|
| N | | 90 | |
| Normal Parameters ^{a,b} | Mean | 882.9265 | |
| | Std. Deviation | 1007.63478 | |
| Most Extreme Differences | Absolute | .361 | |
| | Positive | .361 | |
| | Negative | -.195 | |
| Test Statistic | | .361 | |
| Asymp. Sig. (2-tailed) ^c | | <.001 | |
| Monte Carlo Sig. (2-tailed) ^d | Sig. | <.001 | |
| | 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 2000000.

Gambar 4. 10 Hasil uji Kolmogorov-Smirnov pada data Startup

Sesuai dengan Gambar 4.14, nilai Asymp. Sig. (2-tailed) dan Monte Carlo Sig. (2-tailed)

menunjukkan <0.001, menyatakan bahwa p-value yang didapat lebih rendah dibandingkan dengan tingkat signifikansi umum (contohnya 0.05). Dengan p-value yang lebih rendah, artinya data yang dianalisis tidak terdistribusi normal. Dengan begitu, uji t independen batal untuk dapat dilaksanakan, sebagai gantinya, akan digunakan uji Mann-Whitney U.

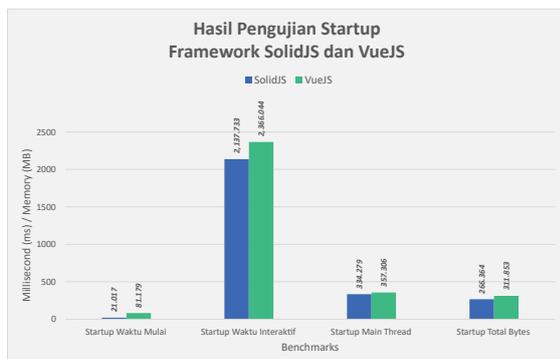
Test Statistics^a

| | Millisecond (ms) |
|------------------------|------------------|
| Mann-Whitney U | 747.000 |
| Wilcoxon W | 1782.000 |
| Z | -2.143 |
| Asymp. Sig. (2-tailed) | .032 |

a. Grouping Variable: Frameworks

Gambar 4. 11 Hasil uji Mann-Whitney U pada data startup

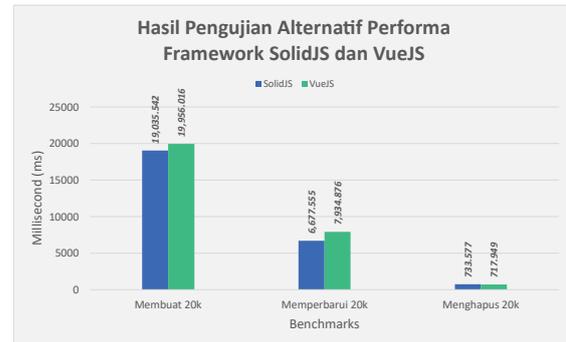
Berdasarkan hasil dari Gambar 4.11, nilai Asymp. Sig. (2-tailed) menunjukkan <0.032, menunjukkan bahwa p-value yang dihasilkan kecil dibandingkan dengan tingkat signifikansi umum (contohnya 0.05). Dengan p-value yang sangat rendah, hipotesis nol (H0) ditolak, dan sebagai hasilnya, hipotesis alternatif (H1) diterima. Kesimpulan dari hasil ini adalah bahwa terdapat perbedaan signifikan antara kedua kelompok data. Hal ini menandakan bahwa terdapat perbedaan signifikan pada saat startup antara framework SolidJS dan VueJS.



Gambar 4. 12 Grafik hasil pengujian startup framework SolidJS dan VueJS

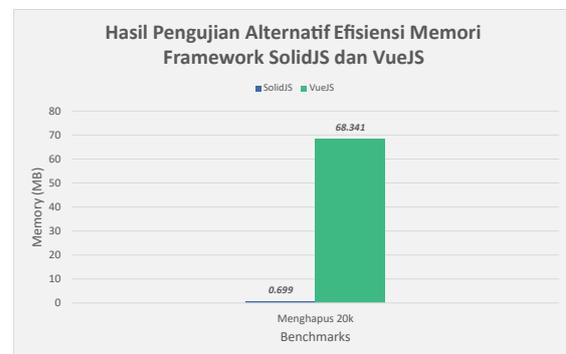
Sesuai dengan grafik pengujian startup pada Gambar 4.12, dapat diambil kesimpulan bahwa SolidJS lebih unggul pada saat melakukan startup website apabila dibandingkan dengan VueJS. SolidJS memerlukan waktu yang lebih singkat untuk memulai aplikasi, termasuk waktu memulai, waktu interaktif, total byte, dan beban kerja di thread utama.

4) Performa Alternatif



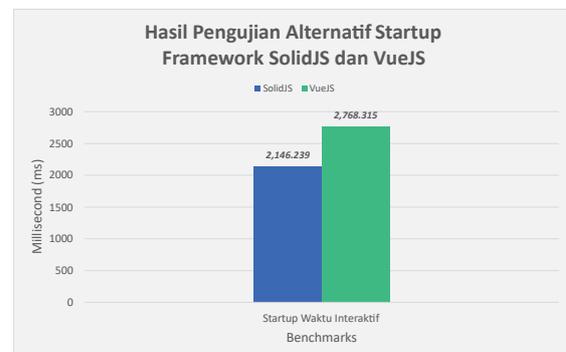
Gambar 4. 13 Grafik hasil pengujian performa alternatif

5) Efisiensi Alternatif



Gambar 4. 14 Grafik hasil pengujian efisiensi alternatif

6) Startup Alternatif



Gambar 4. 15 Grafik hasil pengujian startup alternatif

5. KESIMPULAN

Dengan adanya Analisis Perbandingan Framework Front-End JavaScript SolidJS dan VueJS Pada Pengembangan Website Interaktif, dapat diambil kesimpulan sebagai berikut:

- Hasilnya pengujian menunjukkan bahwa SolidJS unggul dalam performa, efisiensi, dan waktu startup. Dalam 15 iterasi pengujian untuk setiap skenario, SolidJS secara konsisten menunjukkan keunggulan, kecuali pada satu skenario performa

- penghapusan data. Meskipun terdapat sedikit keterlambatan dalam skenario tersebut, SolidJS tetap menonjol dengan penggunaan memori yang efisien setelah penghapusan data. Konsistensi pola hasil pengujian antara pengujian utama dan pengujian sampel alternatif pada perangkat lain semakin memperkuat kesimpulan bahwa SolidJS lebih baik daripada VueJS.
- b. Hasilnya pengujian menunjukkan bahwa VueJS memiliki tingkat stabilitas yang lebih baik dibandingkan dengan SolidJS, seperti yang terlihat dari simpangan baku (stddev) pada hasil pengujian. Di sisi lain, SolidJS menonjol dengan performa dan efisiensi yang lebih baik dibandingkan dengan VueJS, meskipun kurang stabil.
 - c. Telah berhasil melibatkan SolidJS dalam analisis framework front-end sehingga memberikan perspektif yang lebih luas dalam pemahaman tentang pemilihan framework. SolidJS meskipun mungkin kurang dikenal dibandingkan dengan VueJS, membuktikan dirinya sebagai pesaing yang layak.

UCAPAN TERIMA KASIH

Terima kasih yang sebesar-besarnya kepada keluarga dan semua pihak-pihak yang sudah mendukung penulis dalam penelitian ini.

DAFTAR PUSTAKA

- [1] F. A. F. Sofi'ie And A. Qoiriah, "Analisis Perbandingan Framework Front-End Javascript React Dan Vue Pada Pengembangan Website," *Journal Of Informatics And Computer Science (Jinacs)*, Pp. 157–164, 2023.
- [2] W. Xu, "Benchmark Comparison Of Javascript Frameworks React, Vue, Angular And Svelte," *Hämtad Den*, Vol. 5, 2021.
- [3] H. Jihadi And A. F. Syarabil, "Perbandingan React Js Dan Vue Js Dalam Pengembangan Aplikasi Web Interaktif: Sebuah Studi Komparatif," *Jurnal Sistem Informasi Bisnis (Junsibi)*, Vol. 4, No. 2, Pp. 70–79, 2023.
- [4] M. Siahaan And V. O. Vianto, "Comparative Analysis Study Of Front-End Javascript Frameworks Performance Using Lighthouse Tool," *Jurnal Mantik*, Vol. 6, No. 3, Pp. 2462–2468, 2022.
- [5] C. Chastro And E. Darmawan, "Perbandingan Pengembangan Front End Menggunakan Blade Template Dan Vue Js," *Jurnal Strategi-Jurnal Maranatha*, Vol. 2, No. 2, Pp. 302–313, 2020.
- [6] A. Geovanny, "Analisis Rendering Performa Antara Server Side Dan Client Side Pada Web Application," *Jurnal Ilmiah Betrik*, Vol. 13, No. 03 Desember, Pp. 311–319, 2022.
- [7] M. Persson, "Javascript Dom Manipulation Performance: Comparing Vanilla Javascript And Leading Javascript Front-End Frameworks." 2020.
- [8] A. Amarulloh, K. Kurniasih, And M. Muchlis, "Analisis Perbandingan Performa Web Service Rest Menggunakan Framework Laravel, Django, Dan Node Js Untuk Akses Data Dengan Aplikasi Website," *Jurnal Teknik Informatika*, Vol. 9, No. 1, Pp. 14–19, 2023.
- [9] R. Y. Endra, Y. Aprilinda, Y. Y. Dharmawan, And W. Ramadhan, "Analisis Perbandingan Bahasa Pemrograman Php Laravel Dengan Php Native Pada Pengembangan Website," *Expert: Jurnal Manajemen Sistem Informasi Dan Teknologi*, Vol. 11, No. 1, Pp. 48–55, 2021.
- [10] M. Levlín, "Dom Benchmark Comparison Of The Front-End Javascript Frameworks React, Angular, Vue, And Svelte," 2020.
- [11] E. Junirianto And S. Suswanto, "Web Design And Application Programming Interface (Api) Smart Farming Application," *Tepian*, Vol. 2, No. 1, Pp. 33–37, 2021.
- [12] W. Hadinata And L. Stianingsih, "Analisis Perbandingan Performa Restfull Api Antara Express. Js Dengan Laravel Framework," *Jurnal Informatika Dan Teknik Elektro Terapan*, Vol. 12, No. 1, 2024.