

**ANALISIS EFISIENSI PROSES *BUILD* DAN PERFORMA *SINGLE-PAGE APPLICATION* REACT, VUE, DAN SVELTE YANG DIKEMBANGKAN
MENGUNAKAN VITE SEBAGAI *BUILD TOOL***

SKRIPSI

Untuk Memenuhi Sebagian Persyaratan Gelar Sarjana S-1



Disusun oleh:

AZHAR ZAIDAN FAUZI
20106050022
STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

PROGRAM STUDI INFORMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA YOGYAKARTA

2024



KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA
FAKULTAS SAINS DAN TEKNOLOGI

Jl. Marsda Adisucipto Telp. (0274) 540971 Fax. (0274) 519739 Yogyakarta 55281

PENGESAHAN TUGAS AKHIR

Nomor : B-1111/Un.02/DST/PP.00.9/07/2024

Tugas Akhir dengan judul : Analisis Efisiensi Proses Build dan Performa Single-Page Application React, Vue, dan Svelte yang Dikembangkan Menggunakan Vite sebagai Build Tool

yang dipersiapkan dan disusun oleh:

Nama : AZHAR ZAIDAN FAUZI
Nomor Induk Mahasiswa : 20106050022
Telah diujikan pada : Rabu, 03 Juli 2024
Nilai ujian Tugas Akhir : A-

dinyatakan telah diterima oleh Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta

TIM UJIAN TUGAS AKHIR



Ketua Sidang

Dr. Agus Mulyanto, S.Si., M.Kom., ASEAN Eng.
SIGNED

Valid ID: 6690ce7c35096



Penguji I

Ir. Maria Ulfah Siregar, S.Kom., MIT., Ph.D.
SIGNED

Valid ID: 668fa98a83bd1



Penguji II

Mandahadi Kusuma, M.Eng.
SIGNED

Valid ID: 668cab15571bf



Yogyakarta, 03 Juli 2024
UIN Sunan Kalijaga

Dekan Fakultas Sains dan Teknologi

Prof. Dr. Dra. Hj. Khurul Wardati, M.Si.
SIGNED

Valid ID: 669490a20337a

LEMBAR PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama : Azhar Zaidan Fauzi

NIM : 20106050022

Dosen Pembimbing : Dr. Agus Mulyanto, S.Si., M.Kom., ASEAN Eng.

Menyatakan dengan sesungguhnya bahwa skripsi saya yang berjudul: “**Analisis Efisiensi Proses *Build* dan Performa *Single-Page Application* React, Vue, dan Svelte yang Dikembangkan Menggunakan Vite sebagai *Build Tool***” merupakan hasil penelitian saya sendiri dan tidak terdapat penelitian yang pernah diajukan untuk memperoleh gelar akademik di suatu perguruan tinggi atau lembaga pendidikan lainnya, kecuali telah disebutkan dalam sumber referensi yang digunakan. Saya juga menyatakan bahwa semua sumber yang saya gunakan dalam penulisan skripsi ini telah saya nyatakan secara benar sesuai dengan ketentuan yang berlaku. Skripsi ini diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer pada Program Studi Informatika di Universitas Islam Negeri Sunan Kalijaga Yogyakarta. Apabila di kemudian hari ditemukan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik sesuai peraturan yang berlaku.

Yogyakarta, 3 Juni 2024

Yang Menyatakan



Azhar Zaidan Fauzi

NIM: 20106050022



SURAT PERSETUJUAN SKRIPSI/TUGAS AKHIR

Hal : Persetujuan Skripsi / Tugas Akhir

Lamp :

Kepada

Yth. Dekan Fakultas Sains dan Teknologi

UIN Sunan Kalijaga Yogyakarta

di Yogyakarta

Assalamu'alaikum wr. wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan seperlunya, maka kami selaku pembimbing berpendapat bahwa skripsi Saudara:

Nama : Azhar Zaidan Fauzi

NIM : 20106050022

Judul Skripsi : Analisis Efisiensi Proses *Build* dan Performa *Single-Page Application* React, Vue, dan Svelte yang Dikembangkan Menggunakan Vite sebagai *Build Tool*

sudah dapat diajukan kembali kepada Program Studi Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu dalam Program Studi Informatika.

Dengan ini kami berharap agar skripsi/tugas akhir Saudara tersebut di atas dapat segera dimunaqasyahkan. Atas perhatiannya kami ucapkan terima kasih.

Wassalamu'alaikum wr. wb.

Yogyakarta, 04 Juni 2024

Pembimbing

Dr. Agus Mulyanto, S.Si., M.Kom.,

ASEAN Eng.

NIP. 19710823 199903 1 003

LEMBAR PEDOMAN PENGGUNAAN TUGAS AKHIR

Tugas Akhir ini tidak dipublikasikan, tetapi tersedia di perpustakaan dalam lingkungan UIN Sunan Kalijaga Yogyakarta, diperkenankan dipakai sebagai referensi kepustakaan, tetapi pengutipan harus seizin penyusun, dan harus menyebutkan sumbernya sesuai dengan kebiasaan ilmiah. Dokumen Tugas Akhir ini merupakan hak milik UIN Sunan Kalijaga Yogyakarta.



ABSTRAK

Single-page application (SPA) menjadi pendekatan populer dalam pengembangan web. *Framework* dan *library* JavaScript seperti React, Vue, dan Svelte menjadi pilihan utama, di mana efisiensi proses *build* dan performa menjadi aspek penting dalam pemilihannya. Penggunaan *build tool*, seperti Vite, berperan penting dalam efisiensi proses *build* dan performa. Dengan demikian, perlu adanya penelitian yang menitikberatkan pada perbandingan yang setara sebagai rujukan pemilihan.

Penelitian ini melibatkan empat tahapan utama: perancangan arsitektur, pengembangan, proses *build* dan pengujian efisiensinya, serta pengujian performa. SPA dirancang dan dikembangkan dalam tiga tingkatan: sederhana, menengah, dan kompleks. Pengujian efisiensi proses *build* melibatkan *build time* dan *bundle size*, sedangkan performa menggunakan Lighthouse mode *navigation* dan *timespan* serta metrik-metriknya.

Hasilnya menunjukkan Svelte unggul pada efisiensi proses *build* semua tingkatan, diikuti Vue dan React. Pada performa mode *navigation*, Vue awalnya unggul, tetapi tertinggal oleh Svelte pada tingkatan menengah lalu menjadi yang terbawah, React selalu di bawah Svelte. Perbedaan performa semakin mendekat seiring meningkatnya tingkatan. Responsivitas interaksi pada mode *timespan* tidak berbeda signifikan antara mode *navigation*, dengan Vue mempertahankan keunggulan terhadap React pada tingkatan kompleks.

Kata kunci: *Single-page application*, efisiensi proses *build*, performa

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

ABSTRACT

Single-page application (SPA) is becoming a popular approach in web development. JavaScript frameworks and libraries such as React, Vue, and Svelte are the main choices, where build process efficiency and performance are important aspects in their selection. The use of build tools, such as Vite, plays an important role in build process efficiency and performance. Thus, there is a need for research that focuses on equal comparisons as a reference for selection.

This research involved four main stages: architectural design, development, build process and efficiency testing, and performance testing. SPA is designed and developed at three levels: simple, intermediate, and complex. Testing the efficiency of the build process involves build time and bundle size, while performance uses Lighthouse navigation mode and timespan and other metrics.

The results show Svelte excels in build process efficiency at all levels, followed by Vue and React. In navigation mode performance, Vue initially excelled, but fell behind Svelte at the middle level and then became the bottom, React was always below Svelte. The performance gap gets closer as the levels increase. Interaction responsiveness in timespan mode did not differ significantly between navigation modes, with Vue maintaining an advantage over React at the complex level.

Keywords: Single-page application, build process efficiency, performance



KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kepada Tuhan yang Maha Esa yang telah memberikan rahmat-Nya sehingga skripsi ini dapat diselesaikan dengan baik. Penulis menyampaikan kata pengantar ini sebagai bagian dari penyusunan skripsi dengan judul “Analisis Efisiensi Proses *Build* dan Performa *Single-Page Application* React, Vue, dan Svelte yang Dikembangkan Menggunakan Vite sebagai *Build Tool*”.

Penyusunan skripsi ini bertujuan untuk memenuhi salah satu syarat dalam menyelesaikan Program Studi Informatika pada Fakultas Sains dan Teknologi, Universitas Islam Negeri Sunan Kalijaga. Melalui penelitian ini, penulis berharap dapat memberikan kontribusi pada bidang informatika serta menghasilkan pemahaman yang lebih mendalam terkait dengan topik pemrograman web.

Penulis menyampaikan penghargaan setinggi-tingginya kepada semua pihak yang telah memberikan dukungan, bimbingan, dan motivasi selama proses penyusunan skripsi ini. Oleh karena itu, dengan penuh hormat, ucapan terima kasih penulis sampaikan kepada:

1. Keluarga penulis, yaitu Bapak, Ibu, Kakak, dan Adik yang telah memberikan dukungan yang tak terhingga nilainya dalam seluruh aspek kehidupan penulis.
2. Bapak Dr. Agus Mulyanto, S.Si., M.Kom., ASEAN Eng., selaku dosen pembimbing yang telah mengarahkan proses penulisan skripsi ini.
3. Bapak Muhammad Galih Wonoseto, M.T., selaku dosen penasihat akademik yang telah menuntun langkah perkuliahan penulis.
4. Teman-teman mahasiswa-mahasiswi Program Studi Informatika angkatan 2020 atas memori indah yang akan terus penulis kenang.
5. Seluruh *civitas academica* UIN Sunan Kalijaga serta berbagai pihak yang tidak dapat penulis cantumkan satu per satu.

Akhir kata, penulis menyadari bahwa penyusunan skripsi ini tidak terlepas dari kekurangan dan keterbatasan. Oleh karena itu, segala kritik dan saran yang bersifat membangun sangat penulis harapkan untuk perbaikan di masa yang akan datang.

Yogyakarta, Juni 2024

Penyusun

Azhar Zaidan Fauzi



HALAMAN PERSEMBAHAN

Skripsi ini penulis persembahkan untuk keluarga penulis, terutama Bapak dan Ibu yang telah mendukung dan memberi penulis privilese untuk mengenyam pendidikan sampai ke jenjang tinggi. Penulis berikrar untuk menjunjung mulia dedikasi orang tua penulis dan menggunakan ilmu pengetahuan yang diraih untuk kebermanfaatan.



HALAMAN MOTO

-



DAFTAR ISI

HALAMAN SAMPUL	i
HALAMAN PENGESAHAN.....	ii
LEMBAR PERNYATAAN KEASLIAN SKRIPSI	iii
SURAT PERSETUJUAN SKRIPSI	iv
LEMBAR PEDOMAN PENGGUNAAN TUGAS AKHIR	v
ABSTRAK	vi
ABSTRACT.....	vii
KATA PENGANTAR	viii
HALAMAN PERSEMBAHAN	x
HALAMAN MOTO	xi
DAFTAR ISI.....	xii
DAFTAR TABEL.....	xv
DAFTAR GAMBAR	xvii
DAFTAR LAMPIRAN.....	xix
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah.....	3
1.4. Tujuan Penelitian.....	3
1.5. Manfaat Penelitian.....	4
BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI	5
2.1. Tinjauan Pustaka	5
2.2. Landasan Teori	9

2.2.1.	JavaScript	9
2.2.2.	Node.js	10
2.2.3.	Single-Page Application.....	11
2.2.4.	React.....	13
2.2.5.	Vue	14
2.2.6.	Svelte.....	16
2.2.7.	Vite.....	17
2.2.8.	Lighthouse.....	19
BAB III METODE PENELITIAN		26
3.1.	Tahapan Penelitian	26
3.1.1.	Perancangan dan Pengembangan	26
3.1.2.	Pengujian Efisiensi Proses Build dan Performa.....	29
3.2.	Lokasi dan Waktu Penelitian.....	35
3.3.	Alat dan Bahan Penelitian	36
3.3.1.	Alat pada Tahap Pengembangan.....	36
3.3.2.	Alat pada Tahap Pengujian	40
BAB IV HASIL DAN PEMBAHASAN		41
4.1.	Hasil Perancangan dan Pengembangan SPA.....	41
4.1.1.	Sederhana	41
4.1.2.	Menengah.....	45
4.1.3.	Kompleks	51
4.2.	Hasil Pengujian Efisiensi Proses <i>Build</i> dan Performa SPA.....	53
4.2.1.	Sederhana	53
4.2.2.	Menengah.....	54
4.1.3.	Kompleks	56

4.3. Pembahasan	58
BAB V PENUTUP.....	66
5.1. Kesimpulan.....	66
5.2. Saran.....	67
DAFTAR PUSTAKA	68
LAMPIRAN.....	70



DAFTAR TABEL

Tabel 2.1. Ringkasan Tinjauan Pustaka Terkait Topik SPA	8
Tabel 3.1. Pembobotan metrik-metrik yang membentuk skor performa	32
Tabel 3.2. Warna penggambaran pada rentang skor performa	32
Tabel 3.3. Referensi nilai FCP	33
Tabel 3.4. Referensi nilai Speed Index.....	33
Tabel 3.5. Referensi nilai TBT	34
Tabel 3.6. Referensi nilai LCP	34
Tabel 3.7. Referensi nilai CLS	35
Tabel 3.8. Referensi nilai INP	35
Tabel 3.9. Spesifikasi perangkat PC.....	36
Tabel 3.10. Perangkat lunak yang digunakan dalam tahap pengembangan SPA..	37
Tabel 3.11. Dependensi NPM yang digunakan dalam tahap pengembangan SPA	37
Tabel 3.12. Dev dependensi NPM yang digunakan dalam tahap pengembangan SPA.....	38
Tabel 3.13. Perangkat lunak yang digunakan dalam tahap pengujian SPA	40
Tabel 4.1. Hasil pengujian rata-rata kriteria efisiensi proses build tingkatan sederhana.....	53
Tabel 4.2. Hasil pengujian rata-rata skor performa mode navigation tingkatan sederhana.....	54
Tabel 4.3. Hasil pengujian rata-rata metrik-metrik performa mode navigation tingkatan sederhana dalam satuan detik.....	54
Tabel 4.4. Hasil pengujian rata-rata metrik-metrik performa mode timespan tingkatan sederhana dalam satuan detik.....	54
Tabel 4.5. Hasil pengujian rata-rata kriteria efisiensi proses build tingkatan menengah	55
Tabel 4.6. Hasil pengujian rata-rata skor performa mode navigation tingkatan menengah	55
Tabel 4.7. Hasil pengujian rata-rata metrik-metrik performa mode navigation tingkatan menengah dalam satuan detik	56

Tabel 4.8. Hasil pengujian rata-rata metrik-metrik performa mode timespan tingkatan menengah dalam satuan detik	56
Tabel 4.9. Hasil pengujian rata-rata kriteria efisiensi proses build tingkatan kompleks	57
Tabel 4.10. Hasil pengujian rata-rata skor performa mode navigation tingkatan kompleks	57
Tabel 4.11. Hasil pengujian rata-rata metrik-metrik performa mode navigation tingkatan kompleks dalam satuan detik	58
Tabel 4.12. Hasil pengujian rata-rata metrik-metrik performa mode timespan tingkatan kompleks dalam satuan detik	58



DAFTAR GAMBAR

Gambar 3.1. Alur penelitian.	26
Gambar 3.2. Contoh hasil build pada Vite.	30
Gambar 3.3. Contoh hasil pengujian mode Navigation.	31
Gambar 4.1. Tampilan UI SPA tingkatan sederhana bagian pertama.	41
Gambar 4.2. Tampilan UI SPA tingkatan sederhana bagian kedua.	42
Gambar 4.3. Struktur folder SPA React tingkatan sederhana.	43
Gambar 4.4. Struktur folder SPA Vue tingkatan sederhana.	44
Gambar 4.5. Struktur folder SPA Svelte tingkatan sederhana.	45
Gambar 4.6. Tampilan UI SPA tingkatan menengah dan kompleks bagian pertama.	46
Gambar 4.7. Tampilan UI SPA tingkatan menengah dan kompleks bagian kedua.	47
Gambar 4.8. Struktur folder SPA React tingkatan menengah dan kompleks.	48
Gambar 4.9. Struktur folder SPA Vue tingkatan menengah dan kompleks.	49
Gambar 4.10. Struktur folder SPA Svelte tingkatan menengah dan kompleks.	51
Gambar 4.11. Grafik rata-rata build time React, Vue, dan Svelte pada semua tingkatan.	59
Gambar 4.12. Grafik rata-rata bundle size React, Vue, dan Svelte pada tingkatan sederhana dan menengah.	59
Gambar 4.13. Grafik rata-rata bundle size React, Vue, dan Svelte pada tingkatan kompleks.	60
Gambar 4.14. Grafik rata-rata nilai metrik mode navigation React, Vue, dan Svelte tingkatan sederhana dalam satuan detik.	61
Gambar 4.15. Grafik rata-rata nilai metrik mode timespan React, Vue, dan Svelte tingkatan sederhana.	61
Gambar 4.16. Grafik rata-rata nilai metrik mode navigation React, Vue, dan Svelte tingkatan menengah.	62
Gambar 4.17. Grafik rata-rata nilai metrik mode timespan React, Vue, dan Svelte tingkatan menengah.	63

Gambar 4.19. Grafik rata-rata nilai metrik mode timespan React, Vue, dan Svelte tingkatan kompleks.	64
Gambar 4.18. Grafik rata-rata nilai metrik mode navigation React, Vue, dan Svelte tingkatan kompleks.	64
Gambar 4.20. Grafik rata-rata nilai metrik mode timespan React, Vue, dan Svelte tingkatan kompleks.	65



DAFTAR LAMPIRAN

- Lampiran 1. Hasil Pengujian Efisiensi Proses Build React Tingkatan Sederhana
- Lampiran 2. Hasil Pengujian Efisiensi Proses Build Vue Tingkatan Sederhana
- Lampiran 3. Hasil Pengujian Efisiensi Proses Build Svelte Tingkatan Sederhana
- Lampiran 4. Hasil Pengujian Efisiensi Proses Build React Tingkatan Menengah
- Lampiran 5. Hasil Pengujian Efisiensi Proses Build Vue Tingkatan Menengah
- Lampiran 6. Hasil Pengujian Efisiensi Proses Build Svelte Tingkatan Menengah
- Lampiran 7. Hasil Pengujian Efisiensi Proses Build React Tingkatan Kompleks
- Lampiran 8. Hasil Pengujian Efisiensi Proses Build Vue Tingkatan Kompleks
- Lampiran 9. Hasil Pengujian Efisiensi Proses Build Svelte Tingkatan Kompleks
- Lampiran 10. Hasil Pengujian Performa React Tingkatan Sederhana
- Lampiran 11. Hasil Pengujian Performa Vue Tingkatan Sederhana
- Lampiran 12. Hasil Pengujian Performa Svelte Tingkatan Sederhana
- Lampiran 13. Hasil Pengujian Performa React Tingkatan Menengah
- Lampiran 14. Hasil Pengujian Performa Vue Tingkatan Menengah
- Lampiran 15. Hasil Pengujian Performa Svelte Tingkatan Menengah
- Lampiran 16. Hasil Pengujian Performa React Tingkatan Kompleks
- Lampiran 17. Hasil Pengujian Performa Vue Tingkatan Kompleks
- Lampiran 18. Hasil Pengujian Performa Svelte Tingkatan Kompleks

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

BAB I

PENDAHULUAN

1.1. Latar Belakang

Pada era digital yang terus berkembang, aplikasi berbasis web semakin mendominasi sebagai platform utama untuk menyajikan konten dan layanan kepada pengguna. *Single-page application* (SPA) menjadi pendekatan yang populer dalam pengembangan web. SPA memungkinkan halaman web hanya dimuat sekali sehingga interaksi dengan pengguna tidak perlu memuat ulang halaman secara keseluruhan [1]. *Framework* dan *library frontend* JavaScript yang populer, seperti React dan Vue, serta yang sedang berkembang seperti Svelte, telah menjadi pilihan utama para pengembang dalam membangun halaman web berbasis SPA. Hal tersebut dibuktikan dengan masuknya React, Vue, dan Svelte dalam peringkat enam teratas, kategori *framework* dan teknologi web paling diminati pada survei yang diadakan oleh Stack Overflow di tahun 2023 [2].

Meskipun React, Vue, dan Svelte memiliki keunggulan dan keunikan masing-masing dalam membangun SPA, termasuk pada manajemen *state*, *rendering* komponen, dan pengelolaan interaksi antarmuka, efisiensinya dalam proses *build* dan performa akhir menjadi dua aspek penting yang perlu dipertimbangkan dalam memilih *framework* dan *library* untuk proyek pengembangan SPA. Efisiensi dalam proses *build* menitikberatkan pada waktu dan sumber daya yang digunakan untuk melakukan *bundle* terhadap seluruh dependensi yang digunakan [3]. Sementara itu, performa dalam definisi umum merujuk pada kapabilitas sesuatu berdasarkan keberhasilannya dalam menjalankan tugas dalam kondisi tertentu. Dalam konteks ini, performa merujuk pada sejauh mana SPA dapat memberikan respons cepat dan lancar kepada pengguna.

Pemilihan *build tool* memiliki peranan dalam efisiensi proses *build* dan performa SPA, hal ini dikarenakan berbagai *build tool* dan *bundler* seperti Vite dan Webpack memiliki mekanismenya sendiri dalam proses *build* [4]. Vite merupakan

build tool yang dikembangkan untuk menyelesaikan masalah hambatan performa yang sering dialami oleh *bundler* tradisional JavaScript seperti Webpack, Rollup and Parcel. Vite menawarkan berbagai keunggulan seperti kecepatan *build* yang tinggi saat server lokal pertama kali dimuat, kemampuan ini ditunjang melalui mekanisme pemisahan modul ke dalam dua kategori, yaitu dependensi yang tidak sering berubah dan sumber kode yang sering berubah. Selain itu, Vite juga mengatasi pembaruan yang lambat dengan memanfaatkan ECMAScript Modules (ESM) sehingga memungkinkan pembaruan *Hot Module Replacement* (HMR) tanpa bergantung pada ukuran.

Berdasarkan penelitian Ferreira, dkk yang dilakukan pada tahun 2022, performa menjadi salah satu dari sembilan faktor kunci dan menempati posisi keenam dalam pemilihan *frontend framework* JavaScript [5]. Jadi, perlu adanya penelitian lebih lanjut yang berfokus untuk melakukan analisis mendalam serta ditujukan baik sebagai pendukung pemilihan atau referensi perbandingan untuk melakukan pembaruan atau membangun *framework* dan *library* baru. Selain itu, supaya pengujian dapat dilakukan secara setara, pengembangan SPA perlu menggunakan sebuah *build tool* yang mendukung *framework* dan *library* yang berkaitan. Sebagai *build tool*, Vite mendukung berbagai *framework* dan *library* termasuk React, Vue, dan Svelte, hal ini memungkinkan pengembangan SPA dengan arsitektur yang setara antara ketiga *framework* dan *library* tersebut sehingga analisis terhadap efisiensi proses *build* dan performa SPA dapat dilakukan.

Oleh karena itu, penelitian ini bertujuan untuk melakukan analisis mendalam terhadap efisiensi proses *build* dan performa SPA React, Vue, dan Svelte yang dikembangkan menggunakan Vite sebagai *build tool*. Menggunakan konsep pengembangan yang setara, penelitian ini berfokus untuk mengidentifikasi hasil dari pengujian masing-masing *framework* dan *library* sehingga dapat menjadi suatu rujukan bagi para pengembang dalam memilih teknologi yang paling sesuai dengan kebutuhan proyek mereka.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, maka rumusan masalah yang dapat diajukan adalah sebagai berikut:

1. Bagaimana merancang dan mengembangkan *single-page application* (SPA) React, Vue, dan Svelte dengan arsitektur yang setara dan menggunakan Vite sebagai *build tool*?
2. Bagaimana menganalisis efisiensi proses *build* dan performa *single-page application* (SPA) React, Vue, dan Svelte yang dikembangkan dengan arsitektur yang setara dan menggunakan Vite sebagai *build tool*?

1.3. Batasan Masalah

Untuk menghindari terjadinya penyimpangan dan pelebaran pokok masalah, maka perlu dibatasi ruang lingkup penelitian ini.

1. Bahasa pemrograman yang digunakan untuk pengembangan *single-page application* (SPA) adalah JavaScript.
2. *Build tool* yang digunakan untuk pengembangan *single-page application* (SPA) adalah Vite.
3. Pengujian efisiensi proses *build* dan performa *single-page application* (SPA) dilakukan menggunakan satu *personal computer* (PC), browser Google Chrome dengan profil khusus, dan lokal server pada PC peneliti.

1.4. Tujuan Penelitian

Berdasarkan rumusan masalah yang telah dipaparkan, maka tujuan penelitian yang dapat diajukan adalah sebagai berikut:

1. Merancang dan mengembangkan *single-page application* (SPA) React, Vue, dan Svelte dengan arsitektur yang setara dan menggunakan Vite sebagai *build tool*.
2. Mengetahui analisis proses *build* dan performa *single-page application* (SPA) React, Vue, dan Svelte yang dikembangkan dengan arsitektur yang setara dan menggunakan Vite sebagai *build tool*.

1.5. Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat-manfaat sebagai berikut:

1. Memberikan rancangan arsitektur *single-page application* (SPA) dengan arsitektur yang setara antara React, Vue, dan Svelte menggunakan Vite sebagai *build tool*
2. Memberikan hasil analisis dan perbandingan proses *build* dan performa *single-page Application* (SPA) React, Vue, dan Svelte yang dapat dijadikan sebagai pendukung pemilihan *framework* dan *library* SPA atau referensi studi banding untuk melakukan pembaruan atau membangun *framework* dan *library* baru.

BAB V

PENUTUP

5.1. Kesimpulan

Dalam penelitian ini, penulis melakukan analisis efisiensi proses *build* dan performa dari tiga *framework* dan *library* aplikasi web berbasis satu halaman atau *single-page application* (SPA), yaitu React, Vue, dan Svelte. Penelitian ini difokuskan pada penggunaan Vite sebagai alat pembangunan (*build tool*) untuk menunjang kesetaraan arsitektur sehingga perbandingan dapat dilakukan secara setara. Terdapat empat tahapan utama dalam penelitian ini, yaitu perancangan arsitektur SPA, pengembangan SPA, *build* SPA dan pengujian efisiensi proses *build*, serta pengujian performa SPA.

SPA dirancang dan dikembangkan secara setara pada tiga tingkatan, yaitu sederhana, menengah, dan kompleks. Perancangan SPA meliputi pemilihan dependensi, penentuan fungsionalitas SPA, dan penentuan arsitektur folder dari masing-masing tingkatan. Sedangkan pengembangan SPA meliputi pembangunan arsitektur komponen sehingga folder yang dihasilkan setara dan proses *coding* untuk menyatukan komponen-komponen dengan format file yang berbeda-beda.

Pengujian terhadap efisiensi *proses build* dilakukan dengan menjalankan perintah *build* pada Vite untuk mendapatkan nilai dari dua kriteria yang digunakan, yaitu *build time* dan *bundle size*. Sedangkan pengujian terhadap performa dilakukan dengan menjalankan Lighthouse pada SPA hasil *build* untuk mendapatkan nilai waktu dari metrik *First Contentful Paint*, *Speed Index*, *Total Blocking Time*, *Largest Contentful Paint*, *Cumulative Layout Shift*, dan *Interaction to Next Paint*.

Hasil dari pengujian menunjukkan bahwa Svelte memiliki efisiensi proses *build* yang lebih baik pada semua tingkatan yang kemudian diikuti dengan Vue lalu React. Sedangkan untuk performa mode *navigation*, seiring dengan naiknya tingkatan kompleksitas SPA yang berkorelasi dengan penambahan dependensi, Vue mengalami penurunan yang awalnya mengungguli Svelte dan React di tingkatan sederhana kemudian diungguli oleh Svelte pada tingkatan menengah lalu menjadi

yang terbawah pada tingkatan kompleks. Di lain sisi, jika dibandingkan dalam hal performa, React selalu berada di bawah Svelte. Namun, jarak perbedaan nilai dari metrik-metriknya semakin mendekat seiring dengan naiknya tingkatan kompleksitas. Hasil pengujian pada mode *navigation* tersebut tidak jauh berbeda pada hasil mode *timespan* yang berkaitan dengan responsivitas interaksi, perbedaannya hanya terletak pada Vue yang berhasil mempertahankan keunggulan metrik-metriknya dari React pada tingkatan kompleks.

5.2. Saran

Berdasarkan hasil analisis yang telah disajikan pada bab sebelumnya, terdapat beberapa aspek yang dapat diperbaiki atau diperhatikan lebih lanjut guna meningkatkan relevansi dan dampak penelitian ini. Berikut ini beberapa saran yang berguna bagi pembaca dalam memahami implikasi dari penelitian ini dan memberikan arahan untuk penelitian selanjutnya dalam bidang yang sama.

1. Penelitian selanjutnya diharapkan dapat memperluas cakupan penelitian dengan menambahkan *framework* atau *library* lainnya yang mungkin juga relevan dan populer dalam pengembangan SPA.
2. Meskipun Vite digunakan dalam penelitian ini untuk menjamin kesetaraan arsitektur, penelitian selanjutnya dapat mempertimbangkan penggunaan *build tool* alternatif untuk menguji apakah hasilnya konsisten dengan temuan dalam penelitian ini.
3. Penelitian selanjutnya diharapkan dapat melakukan pengujian yang lebih mendalam pada tingkat kompleksitas yang lebih tinggi atau dalam skenario dunia nyata untuk memahami bagaimana ketiga *framework* dan *library* tersebut berperilaku saat digunakan dalam proyek yang lebih besar dan kompleks.
4. Mengingat penelitian ini juga menyoroti performa SPA, penelitian selanjutnya dapat lebih fokus pada strategi atau teknik yang dapat digunakan untuk meningkatkan performa SPA yang dikembangkan dengan menggunakan masing-masing *framework* dan *library* yang telah diteliti.

DAFTAR PUSTAKA

- [1] M. Mikowski dan J. C. Powell, *Single Page Web Applications: JavaScript End-to-end*. Manning, 2013. [Daring]. Tersedia pada: <https://books.google.co.id/books?id=hSjctPBsPfIC>
- [2] Stack Overflow, "Stack Overflow 2023 Developer Survey." Diakses: 6 Juni 2024. [Daring]. Tersedia pada: <https://survey.stackoverflow.co/2023/>
- [3] ISO/IEC 25010:2011, "Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models." Diakses: 31 Januari 2024. [Daring]. Tersedia pada: <https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en>
- [4] E. You, "Why Vite." Diakses: 25 Januari 2024. [Daring]. Tersedia pada: <https://vitejs.dev/guide/why.html>
- [5] F. Ferreira, H. S. Borges, dan M. T. Valente, "On the (un-)adoption of JavaScript front-end frameworks," *Softw Pract Exp*, vol. 52, no. 4, hlm. 947–966, Apr 2022, doi: 10.1002/spe.3044.
- [6] D. Schuster, "React vs. Vue: Comparing JavaScript Frameworks from a Beginner's Perspective," 2023.
- [7] S. Krause, "Official results for js web frameworks benchmark." Diakses: 6 Februari 2024. [Daring]. Tersedia pada: <https://krausest.github.io/js-framework-benchmark/>
- [8] K. A. Kern, "Comparing Modern Front-End Frameworks," 2022.
- [9] P. Markus, "React and Vue performance comparison," 2021.
- [10] W. Xu, "Benchmark Comparison of JavaScript Frameworks React, Vue, Angular and Svelte," 2021.
- [11] E. Saks, "JavaScript frameworks: Angular vs React vs Vue," 2019.
- [12] V. Siahaan dan R. H. Sianipar, *JavaScript: Dari A Sampai Z*. dalam 1. Sparta Publisher, 2018. [Daring]. Tersedia pada: <https://books.google.co.id/books?id=IYxeDwAAQBAJ>
- [13] T. Oxley, N. Rajlich, T. J. Holowaychuk, dan A. Young, *Node.js in Action*. Manning, 2017. [Daring]. Tersedia pada: <https://books.google.co.id/books?id=yzgzEAAAQBAJ>

- [14] Z. Hemel, “Facebook’s React JavaScript User Interfaces Library Receives Mixed Reviews.” Diakses: 25 Januari 2024. [Daring]. Tersedia pada: <https://www.infoq.com/news/2013/06/facebook-react/>
- [15] C. Macrae, *Vue.js: Up and Running: Building Accessible and Performant Web Apps*. O’Reilly Media, 2018. [Daring]. Tersedia pada: <https://books.google.co.id/books?id=appNDwAAQBAJ>
- [16] R. Harris, “Svelte 3: Rethinking reactivity.” Diakses: 25 Januari 2024. [Daring]. Tersedia pada: <https://svelte.dev/blog/svelte-3-rethinking-reactivity>
- [17] A. Roggio, “Google Lighthouse: Monitor Site Performance, SEO, Accessibility.” Diakses: 26 Januari 2024. [Daring]. Tersedia pada: <https://www.practicalecommerce.com/google-lighthouse-monitor-site-performance-seo-accessibility>
- [18] M. Edgar, *Speed Metrics Guide: Choosing the Right Metrics to Use When Evaluating Websites*. Apress, 2024. [Daring]. Tersedia pada: https://books.google.co.id/books?id=_7Jb0AEACAAJ

