

Studi Analisis Dan Optimasi Penggunaan Format Animasi Vektor (Svg) Untuk Meningkatkan Performa Pada Pengembangan Website Profil Perusahaan

Clarinta Ghita Pradyan¹, I Made Suartana²

^{1,2} Program Studi S1 Teknik Informatika, Universitas Negeri Surabaya

Corresponding Author: Clarinta Ghita Pradya, e-mail: clarintaghita.22001@mhs.unesa.ac.id

Published: Februari , 2026

ABSTRAK

Pesatnya perkembangan teknologi internet menuntut website memiliki performa optimal, di mana animasi menjadi elemen penting yang dapat memengaruhi kecepatan dan pengalaman pengguna. Format Scalable Vector Graphics (SVG) sering digunakan untuk animasi web karena skalabilitas dan kualitas visualnya, namun pada animasi kompleks, SVG dapat menurunkan performa akibat beban rendering DOM yang tinggi. Penelitian ini bertujuan menganalisis dampak penggunaan animasi SVG terhadap performa website profil perusahaan serta menguji efektivitas dua metode optimasi: code-level optimization (penyederhanaan struktur SVG) dan format-based optimization (konversi ke Lottie JSON). Metode penelitian menggunakan pendekatan eksperimen komparatif dengan pengujian pada landing page statis yang dilengkapi animasi kompleks. Parameter performa diukur menggunakan Google Lighthouse dan Chrome DevTools, mencakup First Contentful Paint (FCP), Largest Contentful Paint (LCP), Total Blocking Time (TBT), Cumulative Layout Shift (CLS), Frame Rate (FPS), penggunaan CPU, dan ukuran file. Hasil penelitian menunjukkan bahwa format-based optimization dengan Lottie JSON secara signifikan meningkatkan metrik LCP (0,40–0,58 detik), TBT (250–318 ms), CLS (0,072–0,085), dan FPS (57–58 fps) dibandingkan SVG asli, meskipun ukuran file dan penggunaan CPU lebih tinggi. Sementara itu, code-level optimization efektif menekan TBT hingga 0 ms dan mengurangi ukuran file, namun tidak berdampak besar pada FPS. Kesimpulannya, Lottie JSON lebih optimal untuk animasi kompleks dengan pergerakan dinamis, sedangkan SVG tetap efisien untuk animasi sederhana atau statis. Penelitian ini memberikan rekomendasi praktis bagi pengembang web dalam memilih format animasi berdasarkan kompleksitas desain dan tujuan optimasi performa.

Kata Kunci: Performa website, animasi SVG, Lottie JSON, optimasi animasi, Core Web Vitals, pengalaman pengguna

© The Author(s). 2021 Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0>

PENDAHULUAN

Pesatnya perkembangan teknologi informasi dan internet akhir-akhir ini telah merevolusi pola interaksi manusia baik antar individu maupun komunitas. Pertukaran informasi yang semakin aktif melalui platform digital membuat berbagai industri terdorong untuk menciptakan sebuah situs website guna menyediakan layanan server yang lebih stabil dan andal [1]. Adanya kemajuan teknologi saat ini seperti menjadi suatu keharusan bagi dunia untuk mengalihkan sistem penyampaian informasi dari bentuk klasik ke bentuk digital yang lebih fleksibel, cepat, dan efisien [2]. Terlebih, pada masa kini, situs digital seperti website sangat diandalkan oleh mayoritas pengguna internet baik untuk sekedar menyalurkan informasi maupun mendukung aktivitas mereka sehari-hari [3].

Meningkatnya pengguna internet tersebut dibuktikan oleh pernyataan APJII (Asosiasi Penyelenggara Jasa Internet Indonesia) yang mengumumkan bahwa pada tahun 2024, tercatat sebanyak 221.563.479 penduduk Indonesia telah menggunakan internet, yang mengindikasikan penetrasi internet nasional telah meraih angka sebesar 79,5% [4]. Banyaknya masyarakat yang mulai mengintegrasikan situs website dalam keseharian ini selaras dengan meningkatnya kompleksitas website secara bertahap,

sehingga performa situs website menjadi aspek krusial untuk diperhatikan [5]. Hal tersebut dikarenakan performa situs website yang rendah dapat menimbulkan dampak negatif yang luas, baik dari segi kepuasan pengguna maupun operasional perusahaan itu sendiri [6].

Situs website dengan performa rendah bahkan lambat dalam memuat konten cenderung membuat pengguna frustrasi dan memilih untuk berpaling dari situs tersebut [7]. Studi menyebutkan bahwa situs website dengan proses pemuatan konten yang melebihi 3,00 detik cenderung membuat pengunjung beralih dan meninggalkan situs tersebut [8]. Pernyataan tersebut selaras dengan temuan yang menyatakan bahwa durasi rata-rata pemuatan konten dalam website maksimal adalah 2,96 detik pada seluruh perangkat [9]. Permasalahan akan semakin meruncing apabila konten media dalam situs website seperti animasi, diunggah dengan ukuran file besar tanpa proses kompresi [10].

Peran animasi dalam performa website tidak dapat diabaikan karena memiliki pengaruh signifikan terhadap waktu muat, penggunaan sumber daya, dan persepsi pengguna. Animasi berbasis CSS mampu mengurangi beban CPU dan GPU secara signifikan dibandingkan pustaka JavaScript populer [11]. Selain itu, kehadiran animasi pada area above-the-fold secara langsung memengaruhi persepsi kecepatan muat oleh pengguna [12]. Kedua temuan ini memperkuat bahwa pemilihan dan pengelolaan animasi yang tepat merupakan langkah penting dalam pengembangan situs yang efisien dan responsif.

Scalable Vector Graphics (SVG) telah lama hadir sebagai salah satu strategi untuk mengatasi tantangan performa akibat file media berukuran besar. SVG diakui sebagai standar format vektor populer karena karakteristiknya yang tidak bergantung pada resolusi (resolution-independent) serta kemampuannya dalam skalabilitas [13]. Format SVG juga memungkinkan pencarian, indeksasi, manipulasi grafis secara fleksibel, serta kompresi tanpa mengurangi kualitas visual [14]. Melalui kemampuan penskalaan yang baik, SVG menjadi format grafis yang paling efektif untuk desain web [15].

Walaupun format SVG telah menjadi pilihan utama, SVG masih memiliki kelemahan dalam aspek performa, kompatibilitas, dan efisiensi [16]. SVG memiliki ukuran file yang cukup besar saat menangani gambar kompleks, serta mengalami penurunan performa pada dataset berjumlah besar [17]. Studi membuktikan bahwa animasi SVG mempunyai frame rate yang lebih rendah dengan rata-rata 102,3 fps dibandingkan jenis animasi lainnya [18]. Pengembang website menganggap bahwa format SVG masih belum cukup efektif untuk digunakan karena penggunaan CPU yang tinggi di semua perangkat.

Merespon hal tersebut, penerapan animasi berbasis Lottie JSON dianggap sebagai strategi efisien untuk mengoptimasi performa website tanpa mengurangi kualitas visual. Lottie merupakan format animasi berbasis JSON yang berukuran file kecil serta kompatibel pada seluruh perangkat [19]. Format ini diakui memiliki ukuran file lebih kecil hingga 70% dibandingkan format lain, sehingga mempercepat waktu pemuatan [20]. Pengujian beban menunjukkan bahwa Lottie JSON memiliki waktu muat tercepat sebesar 4,58 ms dibandingkan GIF [21]. Efisiensi penyimpanan Lottie JSON bahkan mencapai 94,69% - 95,16% dengan waktu build yang lebih cepat [22].

Berdasarkan studi tersebut, penelitian ini bertujuan melakukan analisis serta optimasi penggunaan format animasi vektor SVG untuk meningkatkan performa website profil perusahaan. Parameter pengujian diukur melalui kecepatan loading, ukuran file, penggunaan sumber daya, frame rate, dan kompatibilitas. Melalui analisis ini, diharapkan dapat memberikan rekomendasi praktis bagi pengembang web dalam menentukan format animasi yang tepat sesuai dengan kompleksitas desain dan kebutuhan kinerja situs website.

METODE PENELITIAN

Metodologi yang digunakan dalam penelitian ini memiliki beberapa tahapan sistematis sebagai panduan agar hasil yang diperoleh sesuai dengan tujuan yang telah ditetapkan sekaligus menjaga konsistensi proses penelitian agar tetap terarah sesuai perencanaan awal. Berikut adalah tahapan alur penelitian yang digambarkan dalam bentuk diagram seperti pada Gambar 1.



Gambar 1. Diagram Alur Tahapan Penelitian

A. Identifikasi Masalah

Tahap penelitian diawali dengan identifikasi permasalahan melalui kajian literatur terkait animasi SVG dan performa website. Analisis studi terdahulu mengungkap beberapa keterbatasan, seperti belum adanya spesifikasi server yang jelas dalam pengujian performa berbasis animasi SVG, parameter pengujian yang belum optimal khususnya dalam kecepatan *rendering* dan stabilitas *frame rate* serta belum adanya validasi empiris terhadap dampak implementasi animasi SVG terhadap performa website dan solusi optimasi yang sesuai. Kesenjangan ini menjadi dasar bagi penelitian ini untuk menyusun rekomendasi berbasis data yang lebih akurat dalam penerapan animasi vektor.

Penelitian ini melakukan analisis performa mencakup kecepatan muat, *frame rate*, ukuran file, penggunaan *resource*, dan kompatibilitas *browser* dengan menggunakan *Google Lighthouse*, *Chrome DevTools*, dan *Task Manager*. Selain menganalisis performa animasi SVG, penelitian juga menyajikan solusi optimasi yang bertujuan meningkatkan efisiensi dan kualitas animasi. Hasilnya diharapkan dapat menjadi referensi bagi pengembang dalam memilih dan mengoptimalkan animasi sesuai kebutuhan proyek.

B. Studi Literatur

Dalam rangka memperkuat penelitian ini, studi literatur dilakukan sebagai landasan teoritis, rujukan, dan bahan pertimbangan dalam pelaksanaan penelitian. Tahap ini mencakup pengumpulan informasi dari berbagai sumber terpercaya, seperti buku, artikel ilmiah, jurnal nasional maupun internasional, serta sumber lain yang relevan dengan topik penelitian. Adanya studi literatur ini bertujuan untuk memperoleh pemahaman yang mendalam yang berkaitan dengan penelitian, khususnya dalam membangun landasan teori yang kuat, mengidentifikasi metode pengembangan sistem yang tepat, serta menentukan tools pengujian yang sesuai. Dengan demikian, penelitian dapat dilakukan secara sistematis. Topik literatur yang digunakan sebagai acuan dalam penelitian ini mencakup *Web Animation*, *Comparative Study*, *Website Performance Optimization*, *Web Asset Optimization*, dan *Performance and Testing Methodologies*.

C. Analisis Kebutuhan

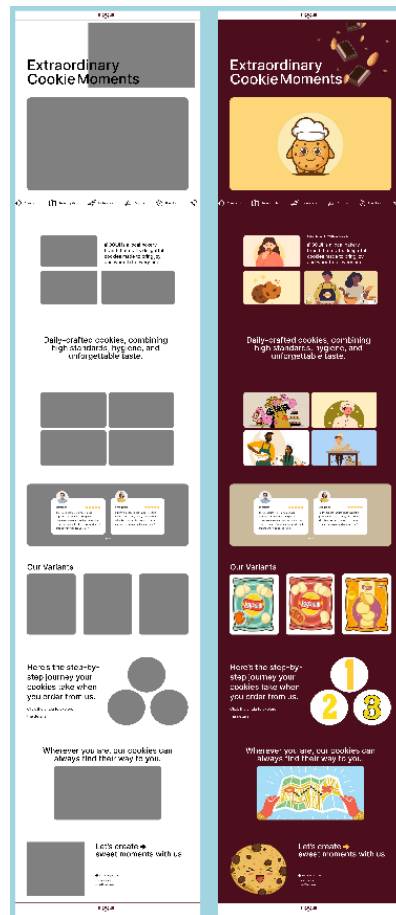
Untuk memastikan sistem website yang optimal dan menjaga performa website tetap baik, diperlukan analisis kebutuhan yang mendukung proses pengembangan dan optimasi dalam penelitian ini. Analisis kebutuhan ini mencakup tiga komponen utama, yaitu perangkat keras (*hardware*), perangkat lunak (*software*), dan teknologi (*tech stack*) yang akan digunakan dalam pembangunan sistem. Berikut adalah komponen-komponen tersebut.

1. Perangkat Keras (*Hardware*)

- a. Device Laptop: Advan Workplus Heritage
 - b. Processor: AMD Ryzen 5 7535HS
 - c. Storage: SSD 512 GB
 - d. Random Access Memory (RAM): DDR5 8 GB
 - e. Mouse
 - f. Keyboard
2. Perangkat Lunak (*Software*)
- a. System Operasi Windows 11
 - b. Visual Studio Code
 - c. Figma
 - d. Infinite Design
 - e. Adobe After Effect
 - b. Notepad
 - c. Browser Chrome
 - d. Microsoft Edge
 - e. Google Lighthouse
 - f. Chrome DevTools
 - g. Task Manager
3. Teknologi (*Tech Stack*)
- a. HTML
 - b. Tailwind CSS
 - c. React JS
 - d. Next JS
 - e. Python

D. Perancangan Sistem dan Animasi

Sebelum pengujian, dibuat sebuah prototipe *landing page* website profil perusahaan sebagai objek penelitian. Desain antarmuka diselesaikan menggunakan Figma untuk menentukan layout, skema warna, dan penempatan elemen animasi. Animasi SVG dirancang melalui dua metode berbeda untuk membandingkan karakteristiknya. Metode pertama menggunakan alur kerja desain visual: aset vektor statis dibuat di *Infinite Design*, kemudian dianimasikan dan diekspor sebagai file SVG menggunakan *Adobe After Effects*. Metode kedua menggunakan pendekatan *code-based*, di mana animasi dikodekan langsung dalam format SVG dengan memanfaatkan elemen SMIL untuk mengontrol transformasi dan *timing*. Seluruh komponen, termasuk animasi SVG yang telah dihasilkan, kemudian diintegrasikan ke dalam sebuah website yang dibangun dengan React.js dan Next.js. Gambar 2 menunjukkan tampilan akhir *mockup landing page* yang digunakan sebagai lingkungan uji.



Gambar 2. Rancangan Design Website

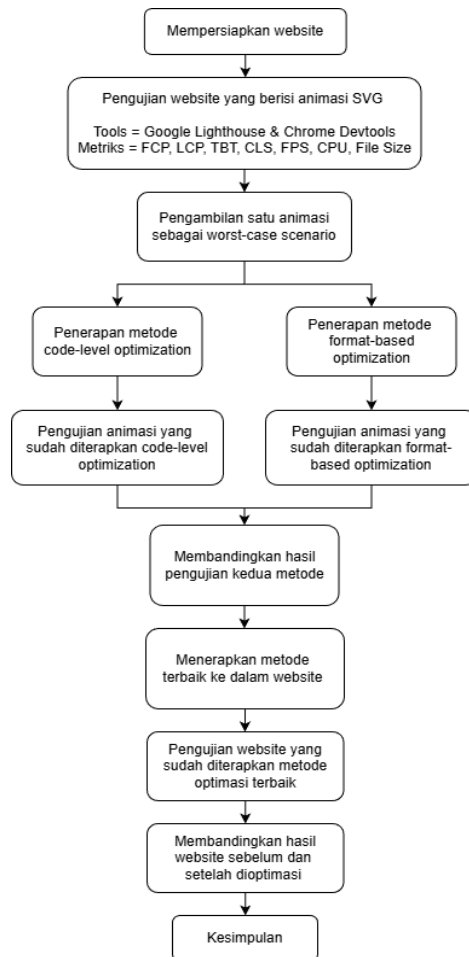
E. Implementasi Sistem dan Animasi

Implementasi animasi dilakukan dengan dua teknik yang berbeda. Pertama, dengan menggunakan tools *Adobe After Effects* (AE) di mana aset vektor statis yang telah dibuat sebelumnya dianimasikan menggunakan AE, kemudian diekspor sebagai file SVG animasi. Kedua, implementasi animasi dilakukan melalui manual *coding* menggunakan SMIL (*Synchronized Multimedia Integration Language*), di mana animasi dibuat langsung melalui penulisan kode SVG berbasis XML tanpa bantuan perangkat lunak animasi grafis, memungkinkan pengendalian gerakan secara terprogram melalui elemen `<animate>` dan `<animateTransform>`.

Sementara itu, sistem website dikembangkan dengan menggunakan *tech stack* *HTML*, *Tailwind CSS*, *React.js*, dan *Next.js*. *Next.js* dipilih sebagai kerangka kerja utama karena kemampuannya dalam optimasi performa dan *server-side rendering* [23], sedangkan *React.js* digunakan untuk membangun antarmuka yang modular [24]. Animasi SVG yang telah dibuat baik dari AE maupun kode manual kemudian diintegrasikan ke dalam komponen *React* menggunakan elemen `<svg>` atau tag `<img>`. Setelah integrasi, dilakukan verifikasi lintas perangkat dan browser untuk memastikan kompatibilitas dan fungsionalitas sebelum sistem dilakukan *deployment* ke lingkungan pengujian.

F. Pengujian Sistem

Pengujian sistem dilakukan untuk mengevaluasi dampak penggunaan animasi vektor SVG terhadap performa website serta menilai efektivitas penerapan metode optimasi yang diusulkan. Pengujian dilakukan dalam beberapa skenario yang merepresentasikan kondisi penggunaan nyata.



Gambar 3. Alur Tahap Pengujian Sistem

Skenario awal merupakan kondisi *baseline*, yaitu website yang menggunakan animasi SVG tanpa penerapan optimasi. Selanjutnya, satu animasi dengan tingkat kompleksitas tertinggi dipilih sebagai *worst-case scenario* untuk dianalisis dan diterapkan metode optimasi. Pemilihan animasi ini bertujuan untuk mengamati dampak maksimum beban *rendering* SVG terhadap performa sistem, khususnya pada metrik yang berkaitan dengan kecepatan pemuatan dan responsivitas antarmuka [25].

Dua pendekatan optimasi diterapkan dan dibandingkan dalam penelitian ini, yaitu *code-level optimization* melalui penyederhanaan struktur SVG serta *format-based optimization* melalui konversi animasi SVG ke format Lottie JSON. Masing-masing pendekatan diuji menggunakan skenario pengujian yang sama untuk memastikan konsistensi hasil. Setelah metode optimasi yang menunjukkan performa terbaik ditentukan berdasarkan hasil pengujian animasi tunggal, metode tersebut diterapkan secara menyeluruh pada seluruh animasi dalam halaman website.

Evaluasi performa dilakukan dengan mengacu pada sejumlah metrik yang merepresentasikan kecepatan pemuatan, stabilitas visual, responsivitas interaksi, kelancaran animasi, serta efisiensi penggunaan sumber daya sistem. Metrik yang digunakan meliputi *First Contentful Paint* (FCP), *Largest Contentful Paint* (LCP), *Total Blocking Time* (TBT), *Cumulative Layout Shift* (CLS), *Frame Rate* (FPS), penggunaan CPU, dan ukuran file animasi.

Tabel 1. Metrik Dan Standar Pengujian

Metrik	Aspek yang Diukur	Tools	Standar Performa
FCP	Waktu muat awal	Google Lighthouse	$\leq 1,8$ detik
LCP	Waktu muat konten utama	Google Lighthouse	$\leq 2,5$ detik
TBT	Responsivitas thread utama	Google Lighthouse	≤ 150 ms
CLS	Stabilitas tata letak	Google Lighthouse	$< 0,1$

FPS	Kelancaran animasi	Chrome DevTools	Stabil di 60 FPS
CPU	Konsumsi prosesor	Chrome DevTools	< 30% (idle), < 60% (aktif)

Penentuan metrik, tools pengujian, serta standar ideal performa pada Tabel 1 mengacu pada rekomendasi *Core Web Vitals* [26] dan praktik terbaik evaluasi performa website yang disediakan oleh *web.dev* [27], *Chrome for Developers* [28], dan *MDN Web Docs* [29]. Pendekatan ini bertujuan untuk memastikan bahwa proses evaluasi dilakukan secara objektif, terukur, dan relevan dengan kondisi penggunaan website modern.

Tahap pengujian selanjutnya dilakukan pada website yang telah sepenuhnya menerapkan metode optimasi terbaik. Hasil pengujian pada kondisi ini kemudian dibandingkan dengan kondisi awal sebelum optimasi untuk mengevaluasi perubahan performa website secara keseluruhan. Seluruh hasil pengujian dan analisis perbandingan disajikan dan dibahas secara rinci pada Bab III (Hasil Penelitian dan Pembahasan) untuk menilai efektivitas penerapan optimasi animasi SVG terhadap peningkatan performa website.

G. Menarik Kesimpulan

Setelah seluruh tahapan penelitian diselesaikan, penarikan kesimpulan dilakukan berdasarkan pada hasil pengujian animasi SVG pada setiap kondisi pengujian, mulai dari animasi SVG awal pada halaman website, animasi SVG yang dipilih sebagai *worst-case scenario*, hingga animasi yang telah diterapkan dua pendekatan optimasi yang berbeda. Selanjutnya, metode optimasi yang menunjukkan performa paling sesuai berdasarkan hasil perbandingan tersebut diterapkan pada seluruh animasi dalam halaman website dan diuji kembali. Melalui proses perbandingan antara kondisi website sebelum dan sesudah penerapan metode optimasi, penarikan kesimpulan dilakukan untuk mengevaluasi dampak dan kesesuaian metode optimasi animasi SVG dalam konteks implementasi website secara menyeluruh.

HASIL DAN PEMBAHASAN

A. Pembuatan dan Implementasi Animasi SVG

Tahap awal penelitian berfokus pada pembuatan animasi vektor dalam format *Scalable Vector Graphics* (SVG) sebagai elemen interaktif utama pada website profil perusahaan. Pengembangan animasi dilakukan secara *hybrid* melalui dua pendekatan utama, yaitu penggunaan perangkat lunak *Adobe After Effects* dan penulisan kode manual menggunakan *Synchronized Multimedia Integration Language* (SMIL).

Proses pembuatan menggunakan *Adobe After Effects* diawali dengan perancangan aset grafis pada aplikasi *Infinite Design* yang kemudian diolah lebih lanjut dalam *Adobe Illustrator* untuk penataan *layer*. Aset tersebut kemudian diimpor ke dalam *Adobe After Effects* untuk diberikan pergerakan menggunakan *keyframe* pada properti seperti posisi, skala, dan rotasi. Tahap akhir dari proses ini adalah ekspor animasi ke format SVG menggunakan *plugin Bodymovin*. Di sisi lain, pendekatan SMIL dilakukan dengan menulis kode SVG murni secara manual pada *text editor*. Struktur animasi didefinisikan melalui elemen `<animate>` dan `<animateTransform>` untuk mengatur durasi, pola gerak, serta pengulangan tanpa ketergantungan pada perangkat lunak grafis eksternal. Karakteristik teknis kedua metode ini memiliki perbedaan signifikan; animasi dari *After Effects* unggul dalam presisi visual namun memiliki struktur file yang kompleks, sementara SMIL lebih ringan dan mudah dimodifikasi kodenya namun terbatas untuk animasi yang sangat kompleks.

Implementasi animasi ke dalam website dilakukan menggunakan teknologi *React.js*, *Next.js*, dan *Tailwind CSS* untuk membangun komponen yang terstruktur dan responsif. Website dibagi menjadi beberapa bagian utama seperti *header*, *hero section*, dan *footer*. Animasi hasil *After Effects* diintegrasikan menggunakan metode *linked SVG* melalui tag `<img>`, sedangkan animasi berbasis SMIL disematkan secara langsung (*inline*) ke dalam struktur HTML. Seluruh elemen animasi diatur melalui CSS untuk memastikan proporsi tampilan tetap stabil dan responsif di berbagai resolusi layar. Hasil implementasi akhir menunjukkan bahwa seluruh animasi dapat dijalankan secara stabil pada kerangka website tanpa menimbulkan konflik antar komponen.



Gambar 4. Tampilan Keseluruhan Website Profil Perusahaan Sebagai Studi Kasus Penelitian

B. Pengujian Performa SVG Awal

Pengujian awal dilakukan pada skenario utama, yaitu halaman website profil perusahaan utuh yang mengimplementasikan animasi SVG secara menyeluruh. Proses pengujian dijalankan sebanyak lima kali iterasi pada perangkat dengan prosesor AMD Ryzen 5 untuk memastikan konsistensi data. Lingkungan uji mencakup tiga kondisi jaringan berbeda, yaitu LAN, *Wi-Fi* Rumah, dan *Wi-Fi* Kampus guna meminimalkan bias faktor eksternal.

Hasil pengujian menunjukkan bahwa metrik kecepatan pemuatan awal, yaitu *First Contentful Paint* (FCP) dan *Largest Contentful Paint* (LCP), berada pada kategori baik (Good) di seluruh kondisi jaringan dengan nilai rata-rata di bawah ambang batas standar masing-masing sebesar 1,8 detik dan 2,5 detik. Namun, ditemukan kendala signifikan pada metrik stabilitas dan responsivitas *thread* utama. Total *Blocking Time* (TBT) tercatat sangat tinggi, mencapai rentang 762–784 ms, yang melebihi standar kategori *fast* (0–150 ms), menandakan beban pemrosesan yang berat pada *thread* utama browser akibat kompleksitas elemen vektor. Selain itu, nilai *Cumulative Layout Shift* (CLS) berada pada angka 0,801–0,898 yang termasuk kategori buruk (*Poor*), dipicu oleh pergeseran tata letak elemen selama proses *render* animasi berlangsung. Penggunaan CPU tercatat cukup efisien di bawah 60%, sementara frame rate (FPS) berada di kisaran 55 fps, sedikit di bawah standar ideal 60 fps. Ringkasan hasil pengujian awal ini disajikan pada Tabel 2.

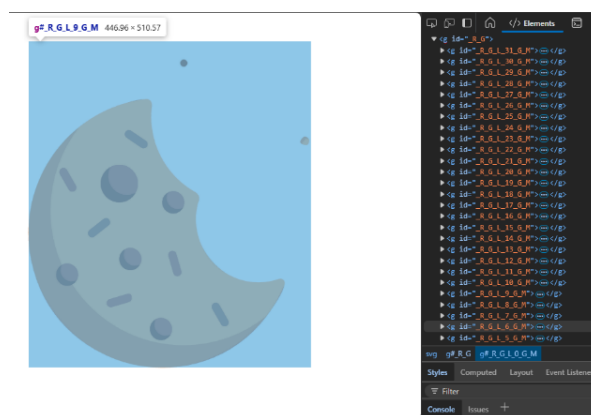
Tabel 2. Hasil Pengujian Performa Website Dengan Animasi Svg

Metrik	LAN	Rumah	Kampus	Standar Ideal
FCP (detik)	0,300	0,360	0,540	$\leq 1,8$
LCP (detik)	0,480	1,040	1,200	$\leq 2,5$
TBT (ms)	762	768	784	≤ 150
CLS (skor)	0,801	0,898	0,895	$< 0,1$
CPU (%)	34,04	32,80	34,00	< 60
FPS (fps)	55,98	55,38	55,32	~ 60

C. Pengambilan Satu Animasi Sebagai Sampel

Setelah dilakukan analisis performa animasi SVG pada halaman website penuh, penelitian ini kemudian mengambil satu animasi sebagai sampel untuk menerapkan dan menguji metode optimasi secara langsung pada kondisi yang paling berat. Selain itu, pengujian difokuskan pada satu animasi terisolasi pada halaman web sementara agar proses penerapan metode optimasi dan evaluasi hasilnya dapat dilakukan secara terkontrol dan objektif, tanpa pengaruh dari animasi atau elemen lain [30].

Sampel animasi SVG yang diambil disini adalah “Coba_CokotPart7.svg” dengan ukuran mencapai 7 MB dan struktur animasi yang memiliki jumlah elemen besar, serta struktur path yang berlapis-lapis. Pemilihan animasi ini didasarkan pada pertimbangan bahwa animasi dengan beban render paling berat akan memberikan gambaran paling jelas mengenai efektivitas metode optimasi. Jika sebuah pendekatan mampu meningkatkan performa secara signifikan pada kondisi terburuk, maka besar kemungkinan metode tersebut juga efektif untuk animasi lain yang tingkat kompleksitasnya lebih rendah.



Gambar 5. Sampel Animasi Coba_CokotPart7.svg

D. Penerapan Strategi Optimasi

Penentuan metode optimasi yang paling efektif dilakukan melalui pendekatan stress testing dengan memilih satu sampel animasi yang mewakili kondisi terburuk (*worst-case scenario*). Sampel yang dipilih adalah file "Coba_CokotPart7.svg" yang memiliki ukuran mencapai 7,8 MB dengan struktur *path* yang sangat padat dan berlapis-lapis. Pengujian pada sampel terisolasi ini bertujuan untuk mengevaluasi efektivitas dua metode optimasi, yaitu *Code-Level Optimization* dan *Format-Based Optimization*.

1) *Code-Level Optimization*: Strategi ini berfokus pada penyederhanaan struktur internal kode SVG tanpa mengubah format dasar maupun tampilan visualnya [31]. Berdasarkan identifikasi struktur, ditemukan adanya redundansi hingga 31 *hidden layer* yang tidak terhapus otomatis oleh perangkat lunak ekspor. Optimasi dilakukan dengan menghapus *layer* berulang hingga menyisakan 8 *layer* fungsional. Hasilnya, ukuran file berkurang drastis dari 7,8 MB menjadi 241 KB.

2) *Format-Based Optimization*: Strategi kedua dilakukan dengan mengonversi format animasi vektor ke format yang lebih modern, yaitu Lottie JSON. Metode ini dipilih karena Lottie memiliki karakteristik *rendering* yang lebih efisien dibandingkan SVG berbasis DOM. Konversi dilakukan menggunakan *plugin* Bodymovin pada *Adobe After Effects* untuk menghasilkan file berbasis data *keyframe*.

Untuk menentukan metode terbaik, dilakukan perbandingan performa antara SVG awal, hasil optimasi kode, dan hasil konversi format pada berbagai metrik. Ringkasan perbandingan tersebut disajikan pada Tabel 3.

Tabel 3. Perbandingan Hasil Pengujian Metode Optimasi Pada Sampel Worst-Case

Metrik	SVG Awal	Code Level	Format Based
FCP (detik)	~0,26	~0,33	_0,2
LCP (detik)	~0,78	~0,40	~0,20

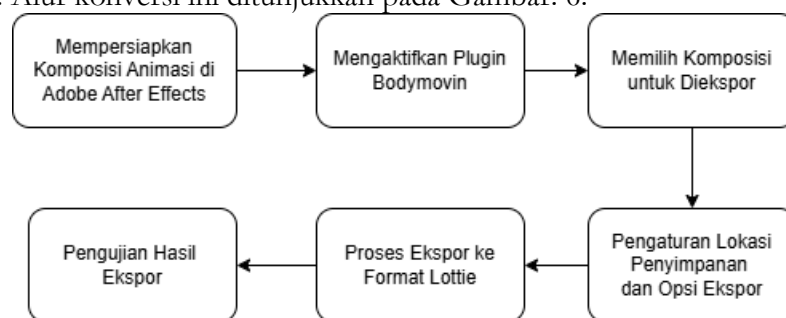
TBT (ms)	~266	0	~29
CLS (skor)	0	0	0
CPU (%)	~ 69,64	~16,88	~14,55
FPS (fps)	~52,14	~51,62	~57,5
Ukuran File	7.882kb	241kb	835kb

Berdasarkan data pada Tabel 3, meskipun *code-level optimization unggul* dalam menekan ukuran file dan TBT, metode *Format-Based Optimization* (Lottie) menunjukkan performa yang jauh lebih konsisten dalam meningkatkan metrik visual *runtime* seperti LCP, FPS, dan efisiensi CPU. Lottie mampu mempertahankan kelancaran animasi mendekati 60 fps dan menekan beban CPU hingga di bawah ambang batas 30%. Oleh karena itu, metode *format-based* ditetapkan sebagai solusi optimasi terbaik untuk diterapkan pada seluruh elemen animasi di website profil perusahaan.

E. Menerapkan Metode Konversi Format Based

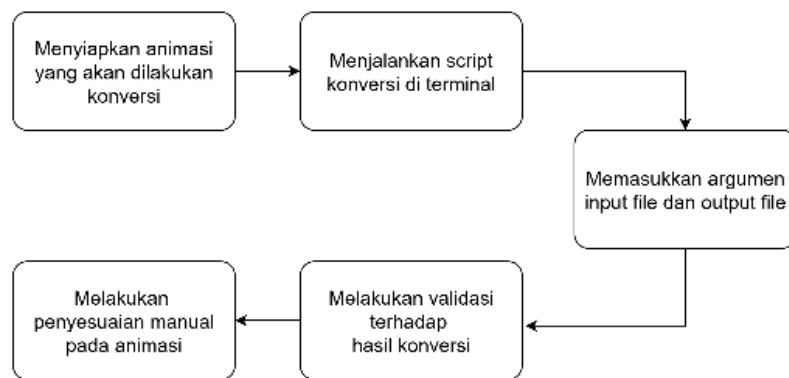
Berdasarkan hasil uji sementara sebelumnya, metode *format-based optimization* kemudian ditetapkan sebagai metode terbaik dan diterapkan ke seluruh animasi SVG yang terdapat pada website awal. Proses konversi dilakukan menggunakan *Adobe After Effects* untuk animasi yang berasal dari *Adobe After Effects*, serta *converter* berbasis Python untuk animasi SVG yang dibuat secara manual menggunakan SMIL. Selanjutnya, dilakukan perbandingan performa website sebelum dan sesudah penerapan metode optimasi guna mengevaluasi dampak keseluruhan terhadap kinerja website.

1. Konversi Menggunakan *Adobe After Effects*: Animasi yang berasal dari aset desain visual dikonversi menggunakan *plugin* Bodymovin. Proses ini dimulai dengan mengimpor aset vektor ke dalam *Adobe Illustrator* untuk merapikan *layer* agar terbaca sebagai *shape layer* yang terpisah di *After Effects*. Setelah animasi siap, Bodymovin mengeksport seluruh data *keyframe* dan transformasi menjadi *file* berekstensi .json. Alur konversi ini ditunjukkan pada Gambar. 6.



Gambar 6. Alur Konversi Menggunakan Adobe After Effect

2. Konversi Menggunakan Skrip Python: Untuk animasi SVG yang dibuat secara manual menggunakan kode SMIL, konversi dilakukan menggunakan skrip Python berbasis pustaka *python-lottie*. Penggunaan skrip ini memungkinkan konversi langsung tanpa melalui perangkat lunak grafis dengan cara memanggil nama *file input* SVG dan menentukan nama *file output* JSON [32]. Namun, karena hasil konversi otomatis sering kali berupa gambar statis, dilakukan penyuntingan manual pada struktur data JSON untuk menambahkan properti *keyframe* pergerakan agar animasi berfungsi serupa dengan versi SMIL aslinya. Alur konversi berbasis skrip ini dapat dilihat pada Gambar. 7.



Gambar 7. Alur Konversi Menggunakan Script Python

Setelah seluruh file JSON berhasil divalidasi melalui LottieFiles Previewer, animasi diimplementasikan ke dalam website menggunakan *library lottie-react*. File tersebut dimuat secara dinamis ke dalam komponen khusus agar dapat dirender dengan parameter *looping* dan skala yang sesuai dengan kebutuhan tata letak halaman.

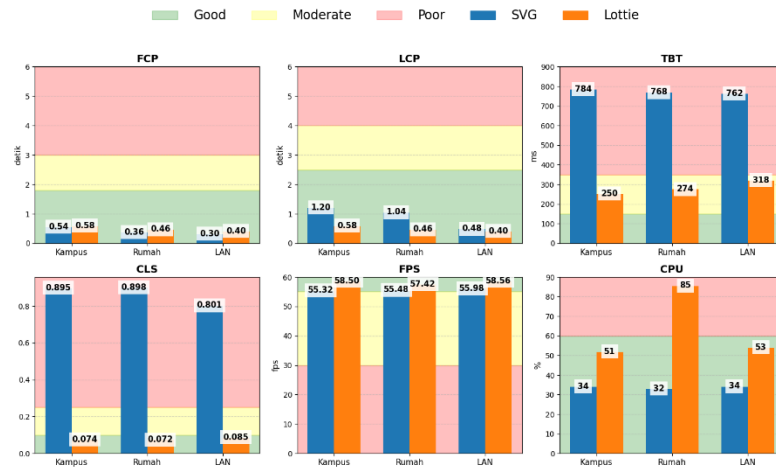
F. Analisis Perbandingan Sebelum dan Sesudah Optimasi

Analisis perbandingan difokuskan pada kinerja website profil perusahaan dalam skenario utama (halaman utuh) untuk mengevaluasi efektivitas konversi dari format SVG ke Lottie JSON. Berdasarkan hasil pengujian metrik pemuatan awal, animasi SVG menunjukkan keunggulan pada *First Contentful Paint* (FCP) dengan rata-rata 0,30–0,54 detik, lebih cepat sekitar 0,04–0,1 detik dibandingkan Lottie. Hal ini dikarenakan SVG diproses langsung oleh *browser* sebagai elemen DOM secara progresif. Sebaliknya, pada metrik *Largest Contentful Paint* (LCP), Lottie terbukti lebih cepat dengan rentang 0,40–0,58 detik, unggul hingga 0,62 detik dibandingkan SVG karena *engine* Lottie memuat seluruh struktur animasi secara menyeluruh sejak awal. Rangkuman perbandingan metrik utama ini disajikan pada Tabel 4.

Tabel 4. Perbandingan Rata-Rata Metrik Performa Svg Sebelum Dan Sesudah Diterapkan Format-Based Optimization (Website Utuh)

Metrik	SVG	Lottie (Format- Based Optimization)	Selisih/ Kenaikan
FCP (detik)	0,40	0,48	SVG lebih cepat 0,08 detik
LCP (detik)	0,90	0,48	Lottie lebih cepat 0,42 detik
TBT (ms)	771	280	Lottie lebih baik 491ms
CLS (skor)	0,86	0,07	Lottie lebih stabil 0,79 skor
CPU (%)	33,61	63,68	SVG menggunakan CPU ~30% lebih sedikit
FPS (fps)	55,58	58,16	Lottie lebih lancar 2,6 fps
Ukuran File(KB)	2.700KB	3.776KB	SVG Lebih rendah 1.076KB

Peningkatan performa yang paling signifikan terlihat pada stabilitas tata letak dan responsivitas halaman. Implementasi Lottie berhasil menekan angka *Cumulative Layout Shift* (CLS) secara drastis hingga kategori *Good* (0,072–0,085), sedangkan SVG berada pada kategori buruk atau *Poor* (0,801–0,898). Optimasi ini mampu mengurangi pergeseran tata letak hingga $\pm 90\%$. Selain itu, Lottie mampu menurunkan *Total Blocking Time* (TBT) sebesar 444–534 ms dibandingkan SVG, yang secara signifikan mengurangi beban pemrosesan pada *main thread*. Visualisasi perbandingan metrik stabilitas dan responsivitas ini dapat dilihat pada Gambar. 8.



Gambar 8. Visualisasi Hasil Perbandingan Pengujian SVG Sebelum dan Sesudah Diterapkan Format-Based Optimization pada Halaman Penuh

Meskipun Lottie unggul dalam stabilitas visual dan kelancaran animasi dengan rata-rata 57,42–58,56 FPS, format ini memiliki konsekuensi pada penggunaan sumber daya perangkat. Tercatat penggunaan CPU pada Lottie (51,74–85,40%) jauh lebih tinggi dibandingkan SVG (32,8–34,04%) karena adanya beban eksekusi *engine* JavaScript. Selain itu, total ukuran file Lottie (3,7 MB) tercatat 28,5% lebih besar daripada SVG (2,7 MB). Secara keseluruhan, hasil penelitian menunjukkan bahwa Lottie JSON merupakan alternatif yang lebih optimal untuk menangani animasi kompleks guna menjaga pengalaman pengguna tetap stabil dan responsif.

G. Kesimpulan Perbandingan Kedua Format Animasi

Berdasarkan hasil pengujian keseluruhan pada website profil perusahaan, ditemukan perbedaan performa yang signifikan antara format SVG awal dan format Lottie (JSON) setelah penerapan optimasi. Pada metrik pemuatan awal, SVG lebih unggul dalam *First Contentful Paint* (FCP) dengan waktu 0,30–0,54 detik, atau sekitar 0,04–0,12 detik lebih cepat dibanding Lottie karena SVG dapat dirender langsung oleh *browser* secara progresif tanpa inisialisasi *engine* tambahan. Namun, Lottie menunjukkan keunggulan pada *Largest Contentful Paint* (LCP) dengan kecepatan 0,40–0,58 detik, unggul $\pm 0,6$ –0,7 detik dibandingkan SVG karena seluruh struktur animasinya dimuat secara menyeluruh oleh *engine* Lottie sejak awal pemuatan.

Dari aspek stabilitas dan kelancaran visual, Lottie terbukti jauh lebih efektif dalam menekan angka *Cumulative Layout Shift* (CLS) ke rentang 0,072–0,085, sementara SVG mencapai angka kritis 0,801–0,898 yang berpotensi mengganggu stabilitas tata letak. Peningkatan responsivitas juga terlihat pada metrik *Total Blocking Time* (TBT) yang berkurang menjadi 250–318 ms dibandingkan SVG yang mencapai 762–784 ms, serta capaian *frame rate* (FPS) yang lebih stabil di kisaran 57–58 fps. Hal ini disebabkan oleh mekanisme *rendering* Lottie berbasis JSON yang mampu mengurangi beban pemrosesan DOM dan *recalculation style* yang sering memicu pergeseran *layout* pada format SVG.

Meskipun Lottie unggul dalam metrik stabilitas visual dan responsivitas, format ini memiliki konsekuensi pada penggunaan CPU yang lebih tinggi (51,74–85,40%) dibandingkan SVG yang lebih efisien (32,80%–34,04%). Secara keseluruhan, Lottie menjadi format yang lebih optimal untuk menangani animasi kompleks dan penuh karena mampu menjaga konsistensi LCP, TBT, CLS, dan FPS meskipun melibatkan banyak elemen pergerakan. Sebaliknya, SVG tetap menjadi pilihan tepat untuk animasi sederhana guna menjaga efisiensi sumber daya perangkat, konsumsi CPU yang rendah, serta ukuran file yang lebih kecil.

KESIMPULAN

Penelitian ini berhasil menganalisis pengaruh penggunaan animasi vektor SVG terhadap performa website profil perusahaan melalui evaluasi dua pendekatan optimasi, yaitu code-level optimization dan format-based optimization. Berdasarkan hasil pengujian kuantitatif menggunakan parameter teknis seperti FCP, LCP, TBT, CLS, FPS, konsumsi CPU, dan ukuran file, dapat disimpulkan bahwa penggunaan animasi SVG dengan tingkat kompleksitas tinggi berpengaruh signifikan terhadap penurunan performa website. Meskipun SVG mampu menghasilkan waktu tampil awal (FCP) yang relatif cepat (0,30–0,54 detik), namun pada skenario halaman penuh, format ini menyebabkan hambatan pada stabilitas tampilan dan responsivitas, yang ditunjukkan oleh nilai TBT yang tinggi (762–784 ms) serta angka CLS yang buruk (0,801–0,898).

Strategi optimasi yang terbukti paling efektif untuk mengatasi keterbatasan tersebut adalah pendekatan format-based optimization melalui konversi animasi ke format Lottie JSON. Penerapan Lottie secara terukur mampu meningkatkan stabilitas visual dengan menekan angka CLS hingga berada pada rentang 0,072–0,085 (kategori Good) serta menurunkan nilai LCP menjadi 0,40–0,58 detik. Selain itu, Lottie mampu menjaga kelancaran animasi pada kisaran 57–58 FPS yang lebih stabil dibandingkan SVG. Meskipun demikian, penggunaan Lottie memiliki kelemahan berupa konsumsi CPU yang lebih tinggi (mencapai 85,40%) dan ukuran file yang rata-rata 40% lebih besar dibandingkan format SVG awal.

Hasil penelitian ini menegaskan bahwa tidak ada satu format animasi yang ideal untuk seluruh kondisi implementasi. Pemilihan format harus disesuaikan dengan karakteristik animasi dan tujuan penggunaan; format SVG tetap menjadi pilihan paling efisien untuk animasi ringan atau elemen antarmuka sederhana guna menjaga efisiensi sumber daya perangkat, sedangkan format Lottie JSON merupakan solusi paling optimal untuk menangani animasi kompleks yang menjadi elemen visual utama guna menjaga responsivitas dan stabilitas halaman tanpa mengorbankan kualitas visual.

Berdasarkan hasil penelitian ini, dapat disimpulkan bahwa penerapan optimasi animasi SVG memberikan peningkatan performa website pada aspek yang diuji. Meskipun demikian, pengembangan penelitian selanjutnya masih terbuka, khususnya dengan memperluas objek pengujian pada berbagai jenis website, seperti website dinamis, e-commerce, maupun aplikasi mobile, guna memperoleh gambaran performa dalam konteks teknis yang lebih beragam. Selain aspek teknis, penelitian lanjutan juga dapat mempertimbangkan evaluasi dari perspektif pengalaman pengguna (user experience), termasuk kelancaran animasi, tingkat kepuasan, dan kenyamanan interaksi. Di sisi lain, eksplorasi teknik optimasi tambahan, seperti pemanfaatan Offscreen Canvas, Web Workers, Lazy Loading khusus animasi, atau pendekatan hybrid yang menggabungkan SVG dan Lottie, serta perbandingan dengan format animasi vektor lain seperti Canvas, WebM, atau WebGL, berpotensi menghasilkan rekomendasi pemilihan format animasi yang lebih detail sesuai dengan kompleksitas, kebutuhan interaktivitas, dan target platform.

REFERENSI

- [1] Ibrahim, I. M., Ameen, S. Y., Yasin, H. M., Omar, N., Kak, S. F., Rashid, Z. N., Salih, A. A., Salim, N. O. M., & Ahmed, D. M. (2021). Web Server Performance Improvement Using Dynamic Load Balancing Techniques: A Review. *Asian Journal of Research in Computer Science*, 10(1), 47–62. <https://doi.org/10.9734/ajrcos/2021/v10i130234>
- [2] Suprpto, A., & Sasongko, D. (2021). Evaluasi Performa Website Berdasarkan Pengujian Beban Dan Stress Menggunakan Loadimpact (Studi Kasus Website Iain Salatiga). *Network Engineering Research Operation*, 6(1). <https://doi.org/10.21107/nero.v6i1.198>
- [3] Aziz, N. S., Sulaiman, N. S., Hassan, W. N. I. T. M., Zakaria, N. L., & Yaacob, A. (2021). A Review of Website Measurement for Website Usability Evaluation. *Journal of Physics: Conference Series*, 1874(1). <https://doi.org/10.1088/1742-6596/1874/1/012045>
- [4] APJII. (2024). *APJII Jumlah Pengguna Internet Indonesia Tembus 221 Juta Orang*. APJII (Asosiasi Penyelenggara Jasa Internet Indonesia). <https://apjii.or.id/berita/d/apjii-jumlah-pengguna-internet-indonesia-tembus-221-juta-orang>
- [5] Sohail, M. (2023). Importance of Website Optimization in Full-Stack Development. In *University of Virginia: Vol. VIII* (Issue I). University of Virginia.

- [6] Sova, E., Makmun, & Tasliza, S. S. (2024). Performance Analysis of Business Startup Website Using Gt-Metrix With Waterfall Software Development Life Cycle Method. *International Journal Science and Technology*, 3(1), 56–63. <https://doi.org/10.56127/ijst.v3i1.1338>
- [7] Octaviani, A., & Andraini, R. (2022). Analisis Website Fakultas Teknik Universitas Negeri Jakarta Menggunakan Pingdom Tools dan GTmetrix. *SINTESIA: Jurnal Sistem Dan Teknologi Informasi Indonesia*, 1(2), 76–80. <https://tools.pingdom.com/>
- [8] Susanti, R., & Cahyono, N. (2024). ANALISIS DAN PERBANDINGAN PERFORMA WEBSITE PENERIMAAN. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(6), 12044–12050.
- [9] Gallino, S., Karacaoglu, N., & Moreno, A. (2023). Need for Speed: The Impact of In-Process Delays on Customer Behavior in Online Retail. *Operations Research*. <https://doi.org/10.1287/opre.2022.2262>
- [10] Xilogianni, C., Doukas, F.-R., Drivas, I. C., & Kouis, D. (2022). Speed Matters: What to Prioritize in Optimization for Faster Websites. *Analytics*, 1(2), 175–192. <https://doi.org/10.3390/analytics1020012>
- [11] BENŮ, M., & ŮLVECKÝ, M. (2024). Measuring the performance of techniques for dynamic 2D animation in web browsers. *Journal of Applied Mathematics, Statistic, and Informatics*, 20(2), 77–110.
- [12] Jahromi, H. Z., Delaney, D. T., & Hines, A. (2020). How crisp is the crease? A subjective study on web browsing perception of above-the-fold. *Proceedings of the 2020 IEEE Conference on Network Softwarization: Bridging the Gap Between AI and Network Softwarization, NetSoft 2020*, 43–50. <https://doi.org/10.1109/NetSoft48620.2020.9165497>
- [13] Imbleau, S. C. (2022). *Understanding Hardware-Accelerated 2D Vector Graphics*. Appalachian State University.
- [14] Maolana, A. (2024). Analisis Perbandingan Hasil Kompresi Citra Format PNG Dengan SVG Untuk Penyimpanan File Gambar. *Dinamik*, 29(1), 13–18. <https://doi.org/10.35315/dinamik.v29i1.9362>
- [15] Kurniawan, I., Maulana, M. R., & Rusli, C. Y. (2022). Perbandingan Kinerja Browser Pada Pemanfaatan Animasi Svg (Scalable Vector Graphic). *IC-Tech*, 17(2), 32–39. <https://doi.org/10.47775/icttech.v17i2.243>
- [16] Taryadi, T., Binabar, S. W., & Yuniarto, E. (2021). Akurasi Presentasi Web dengan Animasi Scalable Vector Graphics: Studi Kasus Animasi Pembelajaran Komunikasi bagi Anak Autis. *Proceedings Series on Physical & Formal Sciences*, 1(1), 134–140. <https://doi.org/10.30595/pspfs.v1i.145>
- [17] Rajatheva, R. (2023). Performance Challenges with Data Visualizations in Browser Environment. Aalto University.
- [18] Loknar, N. S., Ivančević, T. K., Bekavac, M., & Valenko, S. I. (2023). Technologies for Web Animations and their Positive and Negative Sides Regarding Web Page Metrics. *Tehnicki Glasnik*, 17(4), 566–571. <https://doi.org/10.31803/TG-20230621113640>
- [19] LottieFiles. (2025a). *Lottie: The industry standard for motion design*. <https://lottiefiles.com/>
- [20] LottieLab. (2024). How We Make Lotties 10x Smaller. <https://www.lottielab.com/blog/how-we-make-lotties-10x-smaller>
- [21] Wardhani, D. (2024). KOMPARASI ANIMASI FORMAT LOTTIE , JSON , GIF DAN MP4. *Jurnal Ilmu Komputer, Sistem Informasi Dan Teknologi Informasi (Innotech)*, 1(1), 64–72.
- [22] Hidayat, T., & Sungkowo, B. D. (2020). Comparison of Memory Consumptive against the Use of Various Image Formats for App Onboarding Animation Assets on Android with Lottie JSON. *2020 3rd International Conference on Computer and Informatics Engineering, IC2IE 2020*, 376–381. <https://doi.org/10.1109/IC2IE50715.2020.9274612>
- [23] Nugroho, R. P. A., & Sugandi, A. (2024). Meningkatkan Performa Frontend dengan Menggunakan Framework Next . Js dalam Pengembangan Website. *Journal of Cyber Health and Computer (JOCHAC)*, 2(2), 14–19. <https://doi.org/10.18196/jochac.v3i4.14>
- [24] Nurrokhim, R. F., Siregar, F. F., & Prabowo, A. (2025). Perancangan Sistem Collaborative Shopping dan Pemanfaatan Toko Offline untuk Optimalisasi E-Commerce. *Journal of Multidisciplinary Research and Development*, 7(4), 2502–2511.
- [25] Schoeberl, M., Jellum, E., Lin, S., Jerad, C., Maroun, E. J., Lohstroh, M., & Lee, E. A. (2025). Worst-case performance for real-time systems. Cambridge University Press, 1–8.

- [26] DebugBear. (2026, Januari 26). Core Web Vitals metrics and thresholds. DebugBear. Diakses 10 Desember 2025, dari <https://www.debugbear.com/docs/core-web-vitals-metrics>
- [27] Web performance. (n.d.). web.dev. <https://web.dev/performance?hl=id>
- [28] Google. (2024, November 7). Overview of the Performance panel (Indonesian version). Chrome for Developers. Diakses 10 Desember, dari <https://developer.chrome.com/docs/devtools/performance/overview?hl=id>
- [29] MDN contributors. (2025, April 11). Measuring performance. MDN Web Docs. Diakses 15 Desember, 2025, dari https://developer.mozilla.org/en-US/docs/Learn_web_development/Extensions/Performance/Measuring_performance
- [30] Al-emran, A., Kapur, P., Pfahl, D., & Ruhe, G. (2008). Simulating Worst Case Scenarios and Analyzing Their Combined Effect in Operational Release Planning. Springer, 269–281.
- [31] Chen, S., Dong, X., Xu, H., Wu, X., Tang, F., Zhang, H., Yan, Y., Wu, L., Zhang, W., Hou, G., Shen, Y., Lu, W., & Zhuang, Y. (2025). *SVGenius: Benchmarking LLMs in SVG Understanding, Editing and Generation*. arXiv. <https://doi.org/10.1145/3746027.3758287>
- [32] Basaglia, M. (2025). *Python-Lottie Script Documentation*. <https://gitlab.com/mattbas/python-lottie>