

LAPORAN PENELITIAN DOSEN



**PERBANDINGAN PERFORMA FRAMEWORK MVC DALAM SISTEM
KEHADIRAN BERBASIS PENGENALAN WAJAH**

Tim Peneliti:

Wawan Sismadi S.Kom., M.Kom.(Ketua)

Ir. Besar Agung Martono, M.M., DBA. (Anggota)

Yodi Susanto, S.Kom., M.Kom. (Anggota)

Amin Muzaeni, S.Kom., M.Kom. (Anggota)

PROGRAM STUDI INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS IPWIJA

BOGOR

2024



UNIVERSITAS IPWIJA

SK Kemendikbudristek RI No. 627/E/O/2022

Jl. H. Baping No.17 Kel. Susukan, Kec. Ciracas

Jakarta Timur. 13750 Telp. 021-22819921

E-mail : contact@ipwija.ac.id <https://ipwija.ac.id>

 UNIVERSITAS IPWIJA

No. : 033/IPWIJA.LP2M/PT-00/2024
Perihal : Edaran Penelitian Dosen
Lampiran : -

Kepada Yth.
Bapak/Ibu Dosen Tetap
Universitas IPWIJA

Dengan hormat,

Sehubungan dengan dimulainya semester Genap Tahun Akademik 2023/2024, perlu diingat kembali tentang salah satu kewajiban Tri Dharma Perguruan Tinggi Dosen yaitu melaksanakan penelitian. Berkenaan dengan hal itu maka disampaikan:

- Terima kasih kepada Bapak/Ibu Dosen Peneliti yang telah merespon Surat Edaran Kepala LP2M No.108/IPWIJA.LP2M/PT-00/2023 tanggal 4 September 2023 tentang Kegiatan Bidang Penelitian dengan aktif berperan dalam berbagai pertemuan ilmiah, melaksanakan penelitian dan mempublikasikan hasil penelitian di berbagai jurnal ilmiah.
- Dosen yang telah menyelesaikan laporan penelitian dan mempublikasikannya pada semester Ganjil Tahun Akademik 2023/2024 diharapkan mengajukan usulan penelitian baru kepada LP2M.
- Dosen yang telah menyelesaikan tahap akhir penelitian diharapkan dapat segera membuat laporan hasil penelitian dan mempublikasikannya di semester Genap Tahun Akademik 2023/2024.
- Pada Semester Genap Tahun Akademik 2023/2024, Dosen diharapkan aktif mengikuti berbagai kegiatan yang berkaitan dengan penelitian seperti: pertemuan ilmiah, sharing knowledge, diseminasi, pelatihan, seminar, proceeding, publikasi dan lain sebagainya.
- Agar penelitian dosen sesuai dengan Rencana Strategis penelitian institusi maka diharapkan kerjasama pada Dosen dengan jalan senantiasa berkoordinasi dengan LP2M, Prodi dan setiap elemen di UNIVERSITAS IPWIJA.

Demikian edaran ini disampaikan dan terima kasih.

Jakarta, 4 Maret 2024



Dr. Ir. Titing Widvastuti, M.M.
Kepala LP2M Universitas IPWIJA

Tembusan : Rektor Universitas
Wakil Rektor 1
Wakil Rektor 2



**HASIL PENELITIAN DOSEN
UNIVERSITAS IPWIJA
232/IPWIJA.LP2M/PT-00/2024**

1. Judul Penelitian : Perbandingan Performa Framework MVC dalam Sistem Kehadiran Berbasis Pengenalan Wajah
2. Peneliti 1
Nama : Wawan Sismadi S.Kom., M.Kom.
Jenis Kelamin : Laki-laki
Pangkat/Gol/Nip : -
Jabatan Fungsional : -
Nomor Induk Dosen : 0316087708
Univ./Ins/Akademi/Sekolah : Universitas IPWIJA
Bidang Ilmu yang diteliti : Teknologi Informasi
3. Peneliti 2
Nama : Ir. Besar Agung Martono, M.M., DBA.
Jenis Kelamin : Laki-laki
Pangkat/Gol/Nip : Lektor/III.d/ 0407036703
Jabatan Fungsional : Lektor Kepala
Nomor Induk Dosen : 0315036703
Univ./Ins/Akademi/Sekolah : Universitas IPWIJA
Bidang Ilmu yang diteliti : Teknologi Informasi
4. Peneliti 3
Nama : Yodi Susanto, S.Kom., M.Kom.
Jenis Kelamin : Laki-laki
Pangkat/Gol/Nip : 0320098606
Jabatan Fungsional : -
Nomor Induk Dosen : -
Univ./Ins/Akademi/Sekolah : Universitas IPWIJA
Bidang Ilmu yang diteliti : Teknologi Informasi
5. Peneliti 4
Nama : Amin Muzaeni, S.Kom., M.Kom.
Jenis Kelamin : Laki-laki
Pangkat/Gol/Nip : 0326078904
Jabatan Fungsional : -
Nomor Induk Dosen : -
Univ./Ins/Akademi/Sekolah : Universitas IPWIJA
Bidang Ilmu yang diteliti : Teknologi Informasi

Menyetujui,
Kepala LP2M



(Dr. Ir. Titing Widyastuti, M.M.)

Bogor, 02 Desember 2024

(Wawan Sismadi S.Kom., M.Kom.)



(Ir. Besar Agung Martono, M.M., DBA.)



(Yodi Susanto, S.Kom., M.Kom.)



(Amin Muzaeni, S.Kom., M.Kom.)

RINGKASAN

Penelitian ini bertujuan memberikan panduan komprehensif bagi pengembang dalam memilih framework MVC yang paling sesuai untuk membangun aplikasi absensi berbasis pengenalan wajah. Ketiga framework yang dibandingkan—DonatJS, Laravel, dan CodeIgniter—dievaluasi menggunakan Google Lighthouse pada lingkungan perangkat keras terkontrol (Intel Core i5, RAM 8GB). Setiap framework diuji sebanyak tiga kali guna memastikan konsistensi dan validitas data.

Hasil pengujian menunjukkan bahwa DonatJS unggul pada seluruh metrik utama: First Contentful Paint (FCP) sebesar 1200 ms, Largest Contentful Paint (LCP) sebesar 2100 ms, Total Blocking Time (TBT) sebesar 450 ms, Cumulative Layout Shift (CLS) sebesar 0,1, dan Speed Index sebesar 1800 ms. Meski demikian, DonatJS menghadapi tantangan signifikan dalam mengintegrasikan algoritma pengenalan wajah yang membutuhkan daya komputasi tinggi. Laravel menawarkan keseimbangan antara kecepatan muat awal dan stabilitas jangka panjang, cocok untuk aplikasi kompleks berskala besar. CodeIgniter menunjukkan stabilitas terbaik pada beban aplikasi berat meski skor kecepatannya relatif lebih rendah.

Akurasi pengenalan wajah pada implementasi ini mencapai 92%, dengan kelemahan utama pada kondisi pencahayaan buruk dan sudut pandang kamera yang tidak optimal. Penelitian ini merekomendasikan pemilihan framework berdasarkan prioritas proyek: DonatJS untuk keunggulan performa tampilan, Laravel untuk skalabilitas dan fitur lengkap, serta CodeIgniter untuk proyek sederhana dengan sumber daya terbatas. Penelitian lanjutan diperlukan untuk mengeksplorasi integrasi teknologi cloud dan peningkatan akurasi pengenalan wajah dalam berbagai kondisi lingkungan.

Kata Kunci: Framework MVC, Pengenalan Wajah, Absensi, DonatJS, Laravel, CodeIgniter

PRAKATA

Puji dan syukur penulis panjatkan ke hadirat Allah SWT atas segala rahmat dan karunia-Nya sehingga laporan penelitian berjudul ‘Perbandingan Performa Framework MVC dalam Sistem Kehadiran Berbasis Pengenalan Wajah’ ini dapat diselesaikan dengan baik.

Penelitian ini merupakan bagian dari upaya pengembangan ilmu pengetahuan di bidang teknologi informasi, khususnya dalam pengembangan sistem absensi berbasis kecerdasan buatan dan pengenalan wajah. Hasil penelitian ini diharapkan dapat memberikan kontribusi nyata bagi komunitas pengembang perangkat lunak dan institusi pendidikan yang ingin mengimplementasikan sistem kehadiran modern berbasis biometrik.

Penulis mengucapkan terima kasih kepada Universitas IPWIJA, Lembaga Penelitian dan Pengabdian kepada Masyarakat (LP2M), serta seluruh rekan dosen dan mahasiswa yang telah memberikan dukungan selama proses penelitian berlangsung.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, saran dan kritik yang membangun sangat diharapkan demi perbaikan di masa mendatang.

Bogor, 02 Desember 2024

Tim Peneliti

DAFTAR ISI

HALAMAN SAMPUL	i
HALAMAN PENGESAHAN	ii
RINGKASAN	iii
PRAKATA	iv
DAFTAR ISI	v
DAFTAR TABEL	vi
DAFTAR GAMBAR	vii
DAFTAR LAMPIRAN	viii
BAB 1. PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan Penelitian	3
1.4. Manfaat Penelitian	3
BAB 2. TINJAUAN PUSTAKA	4
2.1. Teori Sistem Absensi Berbasis Pengenalan Wajah	4
2.2. Teori Arsitektur Framework MVC	5
2.3. Teori Google Lighthouse Sebagai Alat Pengujian Performa	6
2.4. State of The Art	7
2.5. Roadmap Penelitian	8
BAB 3. METODE PENELITIAN	9
3.1. Lokasi dan Waktu Penelitian	9
3.2. Jenis dan Sumber Data	9
3.3. Definisi dan Pengukuran Variabel	10
3.4. Metode Analisis Data	10
BAB 4. HASIL PENELITIAN	11
4.1. Gambaran Umum Objek Penelitian	11
4.2. Hasil Penelitian	12
4.3. Pembahasan	14
BAB 5. KESIMPULAN	18
5.1. Kesimpulan	18
5.2. Saran	19
DAFTAR PUSTAKA	20
LAMPIRAN	22

DAFTAR TABEL

Tabel 1. Hasil Pengujian Performa Aplikasi

12

DAFTAR GAMBAR

Gambar 1. Kerangka Konsep Pemikiran (Alur Penelitian)	10
Gambar 2. Hasil Pengujian Performa Aplikasi (Google Lighthouse)	12

DAFTAR LAMPIRAN

Lampiran 1. Screenshot Hasil Google Lighthouse — DonatJS	22
Lampiran 2. Screenshot Hasil Google Lighthouse — Laravel	23
Lampiran 3. Screenshot Hasil Google Lighthouse — CodeIgniter	24
Lampiran 4. Potongan Kode Inti DonatJS (Sistem Absensi)	25

BAB 1. PENDAHULUAN

Latar Belakang

Sistem absensi merupakan salah satu komponen penting dalam pengelolaan kehadiran di lembaga pendidikan maupun perkantoran. Metode absensi konvensional seperti tanda tangan manual, kartu magnetik, dan sidik jari memiliki sejumlah kelemahan, di antaranya rentan terhadap manipulasi (titip absen), keterbatasan akurasi akibat kondisi fisik pengguna, serta efisiensi operasional yang rendah. Perkembangan teknologi kecerdasan buatan (Artificial Intelligence/AI) dan computer vision membuka peluang untuk mengembangkan sistem absensi yang lebih andal berbasis pengenalan wajah (Akhmad, 2023).

Sistem absensi berbasis pengenalan wajah menawarkan efisiensi tinggi dengan waktu verifikasi yang cepat dan akurasi yang baik. Teknologi ini mampu mengatasi berbagai tantangan yang sering ditemui pada metode konvensional, seperti kesalahan pembacaan sidik jari akibat kondisi fisik pengguna. Selain itu, penggunaan pengenalan wajah meningkatkan keamanan dan kemudahan pengelolaan absensi, terutama di lingkungan yang membutuhkan interaksi minimal antarpengguna (Akhmad, 2023). Perkembangan machine learning dan teknologi pemrosesan citra digital semakin mendukung implementasi pengenalan wajah secara real-time dan adaptif sesuai kebutuhan (Margianto, 2024).

Framework Model-View-Controller (MVC) telah menjadi standar dalam pengorganisasian proyek perangkat lunak yang kompleks (Sumarsono, 2024). Dengan memisahkan logika bisnis, antarmuka pengguna, dan pengolahan data, MVC memungkinkan pengembangan aplikasi yang lebih terstruktur dan kolaboratif. PHP sebagai salah satu bahasa pemrograman web populer menyediakan dukungan penuh untuk implementasi arsitektur ini. Framework populer seperti Laravel, CodeIgniter, dan DonatJS memiliki keunggulan masing-masing. Laravel cocok untuk pengembangan aplikasi berskala besar dengan fitur yang lengkap (MAILOPUW, 2024). Di sisi lain, CodeIgniter lebih ringan dan efisien untuk proyek kecil (MAILOPUW, 2024), sedangkan DonatJS, yang

berbasis JavaScript, sangat ideal untuk aplikasi web dinamis dengan pendekatan zero-dependency dan JSON-driven rendering (Kosmos, 2024).

Penelitian sebelumnya umumnya membahas performa framework secara umum tanpa berfokus pada skenario integrasi teknologi pengenalan wajah. Studi komparatif yang secara khusus menganalisis performa framework MVC dalam konteks sistem absensi berbasis pengenalan wajah masih sangat terbatas dalam literatur akademik Indonesia. Penelitian ini hadir untuk mengisi kesenjangan tersebut dengan mengevaluasi performa framework DonatJS, Laravel, dan CodeIgniter berdasarkan metrik standar industri seperti FCP, LCP, TBT, dan CLS (Suhaili et al., 2022), serta kemampuan masing-masing framework dalam mengintegrasikan teknologi pengenalan wajah.

Rumusan Masalah

Berdasarkan latar belakang di atas, rumusan masalah dalam penelitian ini adalah sebagai berikut:

- Bagaimana perbandingan performa metrik FCP, LCP, TBT, CLS, dan Speed Index antara framework DonatJS, Laravel, dan CodeIgniter dalam konteks sistem absensi berbasis pengenalan wajah?
- Apakah terdapat perbedaan yang signifikan dalam kemampuan integrasi teknologi pengenalan wajah antara ketiga framework MVC yang dievaluasi?
- Framework MVC mana yang paling optimal diterapkan pada sistem kehadiran berbasis pengenalan wajah berdasarkan pertimbangan performa, skalabilitas, dan kemudahan pengembangan?

Tujuan Penelitian

Penelitian ini bertujuan untuk:

- Mengidentifikasi, menganalisis secara mendalam, dan membandingkan keunggulan, kelemahan, serta tantangan implementasi dari tiga framework MVC: DonatJS, Laravel, dan

CodeIgniter dalam konteks sistem absensi berbasis pengenalan wajah.

- Mengevaluasi performa masing-masing framework menggunakan metrik standar Google Lighthouse dalam lingkungan pengujian yang terkontrol dan terstandarisasi.
- Memberikan panduan praktis bagi pengembang dalam memilih framework yang sesuai dengan kebutuhan spesifik proyek aplikasi absensi berbasis pengenalan wajah.

Manfaat Penelitian

Penelitian ini diharapkan memberikan manfaat sebagai berikut:

- Bagi pengembang perangkat lunak: memberikan acuan berbasis data dalam pemilihan framework MVC yang tepat untuk sistem absensi berbasis pengenalan wajah.
- Bagi institusi pendidikan dan perkantoran: menyediakan rekomendasi teknis dalam implementasi sistem kehadiran modern yang efisien, akurat, dan andal.
- Bagi komunitas akademis: berkontribusi pada literatur studi komparatif framework MVC dalam konteks aplikasi pengenalan wajah di Indonesia.
- Bagi pengembangan DonatJS: memberikan umpan balik empiris tentang kekuatan dan keterbatasan framework dalam skenario aplikasi yang membutuhkan pemrosesan komputasi tinggi.

BAB 2. TINJAUAN PUSTAKA

Teori Sistem Absensi Berbasis Pengenalan Wajah

Sistem absensi berbasis pengenalan wajah (face recognition-based attendance system) merupakan sistem yang memanfaatkan algoritma computer vision dan kecerdasan buatan untuk mengidentifikasi identitas pengguna secara otomatis melalui citra wajah. Sistem ini menawarkan pendekatan tanpa sentuh (contactless) yang semakin relevan pasca pandemi dan di lingkungan yang memerlukan higienis tinggi (Akhmad, 2023).

Secara teknis, sistem pengenalan wajah bekerja melalui serangkaian tahap: (1) deteksi wajah dari frame video atau gambar statis, (2) ekstraksi fitur dari area wajah yang terdeteksi, (3) pembangunan representasi fitur (face embedding), dan (4) pencocokan embedding dengan basis data yang tersimpan. Akurasi sistem sangat dipengaruhi oleh kualitas input (resolusi kamera, pencahayaan, sudut pandang) dan kompleksitas algoritma yang digunakan (Margianto, 2024).

Perkembangan deep learning, khususnya Convolutional Neural Network (CNN), telah merevolusi akurasi pengenalan wajah secara signifikan (Firmansyah, 2021). Algoritma modern seperti FaceNet, ArcFace, dan MediaPipe Face Landmarker mampu menghasilkan representasi wajah yang robust terhadap variasi kondisi lingkungan. Implementasi real-time pengenalan wajah kini semakin feasibel berkat optimasi model untuk lingkungan client-side menggunakan WebAssembly dan JavaScript API (Margianto, 2024).

Dalam konteks sistem absensi, implementasi pengenalan wajah memerlukan pertimbangan menyeluruh yang mencakup keamanan data biometrik, akurasi verifikasi pada berbagai kondisi, performa sistem saat digunakan secara bersamaan oleh banyak pengguna, serta kemudahan integrasi dengan infrastruktur TI yang sudah ada di institusi. Keselarasan antara framework backend yang digunakan dan teknologi pengenalan

wajah menjadi faktor kritis yang menentukan keberhasilan implementasi sistem (Mohammad & Sirojul, 2022).

Teori Arsitektur Framework MVC

Model-View-Controller (MVC) adalah pola arsitektur perangkat lunak yang memisahkan aplikasi menjadi tiga komponen utama yang memiliki tanggung jawab berbeda. Model bertanggung jawab atas data dan logika bisnis; View menangani presentasi dan antarmuka pengguna; Controller mengelola alur input dari pengguna serta mengatur sinkronisasi antara Model dan View (Sumarsono, 2024). Pemisahan ini meningkatkan modularitas, kemudahan pengujian, dan pemeliharaan kode dalam jangka panjang.

DonatJS adalah framework JavaScript zero-dependency yang dikembangkan dengan filosofi minimalis dan JSON-driven rendering. Berbeda dengan framework PHP, DonatJS memproses logika rendering di sisi klien (client-side), sehingga mengurangi beban pada server dan mempercepat waktu respons antarmuka. Pendekatan ini sangat sesuai untuk aplikasi web dinamis yang mengutamakan kecepatan tampilan (Kosmos, 2024).

Laravel adalah framework PHP full-featured yang menyediakan ekosistem lengkap untuk pengembangan aplikasi web berskala enterprise. Laravel mengandalkan Eloquent ORM untuk pengelolaan basis data, Blade sebagai template engine, dan Artisan sebagai antarmuka baris perintah untuk otomasi tugas. Keunggulan Laravel terletak pada skalabilitas, dokumentasi yang komprehensif, dan komunitas pengembang yang besar (MAILOPUW, 2024).

CodeIgniter adalah framework PHP yang ringan (lightweight) dan dirancang untuk kemudahan instalasi serta konfigurasi minimal. CodeIgniter sangat populer untuk proyek dengan anggaran terbatas atau lingkungan hosting yang memiliki keterbatasan sumber daya. Meski fungsionalitasnya tidak selengkap Laravel, CodeIgniter menawarkan

kecepatan eksekusi yang baik dan kurva pembelajaran yang lebih landai bagi pemula (MAILOPUW, 2024).

Teori Google Lighthouse Sebagai Alat Pengujian Performa

Google Lighthouse adalah alat audit kualitas web open-source yang dikembangkan oleh Google. Lighthouse mampu mengukur berbagai aspek kualitas aplikasi web secara terstandarisasi, meliputi performa, aksesibilitas, praktik terbaik (best practices), SEO, dan Progressive Web App (PWA). Alat ini banyak digunakan dalam penelitian akademis karena kemampuannya menghasilkan data metrik yang reproduisibel dan berbasis standar industri (Suhaili et al., 2022).

Metrik utama yang menjadi fokus penelitian ini adalah:

- First Contentful Paint (FCP): mengukur waktu (ms) sejak halaman mulai dimuat hingga browser merender konten pertama (teks atau gambar). FCP yang rendah menandakan pengalaman loading yang responsif bagi pengguna.
- Largest Contentful Paint (LCP): mengukur waktu (ms) hingga elemen konten terbesar di viewport terlihat oleh pengguna. LCP yang baik menandakan bahwa konten utama halaman telah dimuat.
- Total Blocking Time (TBT): mengukur total waktu (ms) di mana thread utama browser diblokir cukup lama untuk menghambat interaksi pengguna. TBT yang rendah menandakan halaman yang responsif terhadap input.
- Cumulative Layout Shift (CLS): mengukur stabilitas visual halaman secara kumulatif. Nilai CLS mendekati nol menandakan halaman yang stabil secara visual tanpa pergeseran konten yang tidak terduga.
- Speed Index (SI): mengukur kecepatan rata-rata pengisian konten halaman secara visual (ms). SI yang rendah menandakan halaman yang terasa cepat dimuat secara keseluruhan.

Setiap metrik memiliki bobot berbeda dalam menghitung skor performa akhir Lighthouse. Pengujian dilakukan dalam mode simulasi terkontrol untuk memastikan konsistensi hasil pengukuran (Suhaili et al., 2022).

State of The Art

Berbagai penelitian telah dilakukan untuk membandingkan performa framework web dalam konteks yang berbeda. MAILOPUW (2024) melakukan analisis perbandingan menyeluruh antara CodeIgniter dan Laravel dan menemukan bahwa Laravel lebih unggul untuk aplikasi enterprise dengan kebutuhan fitur kompleks, sementara CodeIgniter lebih efisien dari sisi penggunaan sumber daya untuk proyek berskala kecil. Sumarsono (2024) mengkaji implementasi arsitektur MVC pada Laravel dan menegaskan keunggulan pola ini dalam modularitas pengembangan.

Suhaili, Nurdin, dan Taufiq (2022) mengaplikasikan metrik Google Lighthouse untuk menganalisis performa dua platform marketplace besar Indonesia (Tokopedia dan Shopee), menunjukkan relevansi dan validitas penggunaan Lighthouse sebagai alat benchmark performa web yang terstandarisasi. Rahmad dan Arief (2024) mengevaluasi trade-off antara akurasi dan kecepatan pada sistem deteksi kebakaran berbasis YOLOv5 di edge devices, menegaskan bahwa pemilihan arsitektur komputasi secara langsung memengaruhi performa sistem pengenalan visual secara keseluruhan.

Firmansyah (2021) mengimplementasikan deep learning menggunakan CNN untuk klasifikasi gambar dan mendokumentasikan parameter-parameter yang memengaruhi akurasi model. Mohammad dan Sirojul (2022) menganalisis implementasi RESTful API dalam pengembangan sistem informasi akademik, menunjukkan pentingnya pertimbangan sinkronisasi data antara backend dan frontend dalam sistem berbasis web.

Berdasarkan kajian di atas, penelitian yang secara spesifik membandingkan performa framework MVC (terutama DonatJS) dalam

konteks sistem absensi berbasis pengenalan wajah dengan menggunakan metrik Google Lighthouse masih belum dilakukan. Penelitian ini mengisi gap tersebut dengan memberikan perbandingan empiris berbasis data pengujian pada lingkungan terkontrol.

Roadmap Penelitian

Penelitian ini merupakan bagian dari roadmap jangka panjang pengembangan sistem teknologi informasi cerdas di Universitas IPWIJA. Pada fase pertama (penelitian ini), dilakukan studi komparatif performa framework MVC sebagai landasan pemilihan teknologi yang tepat untuk sistem absensi berbasis pengenalan wajah.

Pada fase kedua, direncanakan optimasi algoritma pengenalan wajah untuk kondisi pencahayaan variabel dan integrasi dengan teknologi Progressive Web App (PWA) agar sistem dapat berjalan secara offline. Fase ketiga mencakup pengembangan sistem absensi terdistribusi berbasis cloud yang dapat digunakan lintas unit dan kampus. Pada fase akhir, sistem akan diintegrasikan dengan platform SIAKAD (Sistem Informasi Akademik) Universitas IPWIJA untuk otomatisasi pencatatan kehadiran mahasiswa.

BAB 3. METODE PENELITIAN

Lokasi dan Waktu Penelitian

Penelitian dilaksanakan di Universitas IPWIJA, Bogor. Pengujian dilakukan di laboratorium komputer kampus dalam lingkungan terkontrol menggunakan perangkat keras yang seragam: CPU Intel Core i5, RAM 8GB, dan sistem operasi Windows 10 terbaru. Penelitian berlangsung selama 1 (satu) tahun.

Jenis dan Sumber Data

Penelitian ini menggunakan pendekatan kuantitatif dan eksperimental. Data primer diperoleh dari hasil pengujian performa menggunakan Google Lighthouse (versi terbaru) terhadap tiga aplikasi absensi yang dikembangkan dengan masing-masing framework. Setiap aplikasi memiliki fitur dan fungsionalitas yang identik: login pengguna, entri data pengguna, verifikasi wajah, dan penyimpanan riwayat absensi (Rotikan, 2016), untuk memastikan perbandingan yang adil dan valid.

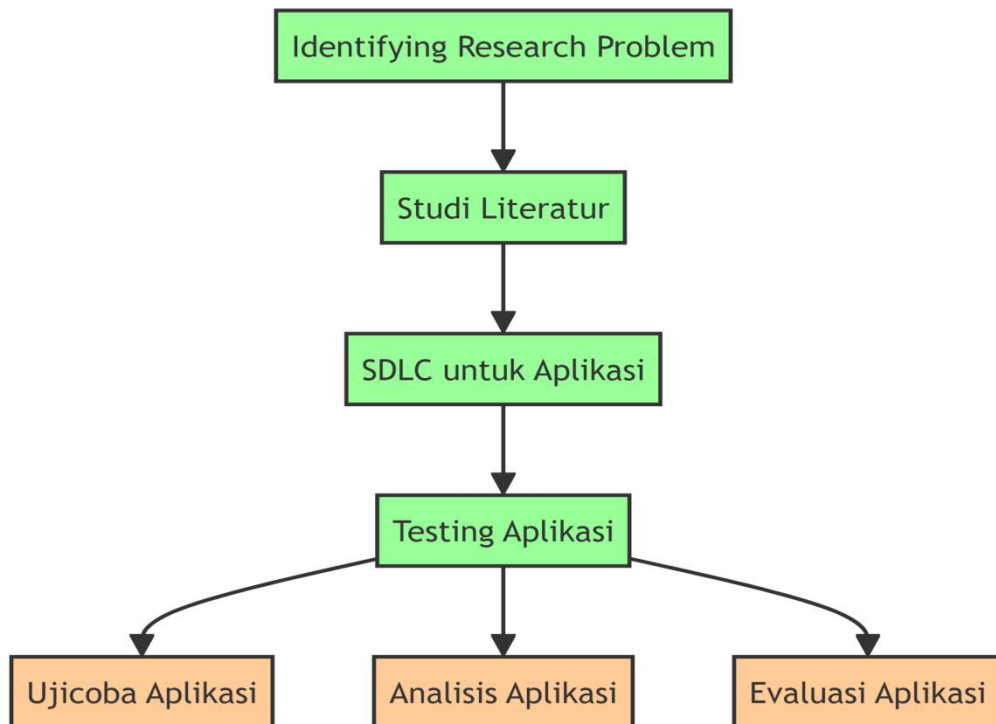
Data sekunder diperoleh dari studi literatur yang mencakup jurnal ilmiah terindeks SINTA dan internasional, prosiding konferensi, serta dokumentasi teknis resmi dari masing-masing framework yang relevan dengan topik penelitian.

Definisi dan Pengukuran Variabel

Variabel independen dalam penelitian ini adalah jenis framework MVC yang digunakan (DonatJS, Laravel, dan CodeIgniter). Variabel dependen adalah nilai metrik performa web yang dihasilkan oleh Google Lighthouse:

- First Contentful Paint (FCP) dalam satuan milidetik (ms).
- Largest Contentful Paint (LCP) dalam satuan milidetik (ms).
- Total Blocking Time (TBT) dalam satuan milidetik (ms).
- Cumulative Layout Shift (CLS) berupa nilai numerik tanpa satuan (semakin kecil semakin baik).
- Speed Index (SI) dalam satuan milidetik (ms).

Kerangka konsep pemikiran penelitian ini terdiri dari beberapa tahap utama sebagaimana divisualisasikan pada Gambar 1. Tahap awal dimulai dengan identifikasi masalah penelitian, dilanjutkan dengan studi literatur untuk memperoleh landasan teori yang relevan. Selanjutnya dilakukan pengembangan aplikasi menggunakan metode Software Development Life Cycle (SDLC) (Mukund, 2013), yang mencakup perancangan, implementasi, dan pengujian aplikasi berdasarkan framework yang dipilih. Tahap berikutnya adalah pengujian performa, analisis dan evaluasi hasil, serta penarikan kesimpulan.



Gambar 1. Kerangka Konsep Pemikiran

Metode Analisis Data

Data hasil pengujian Google Lighthouse dari tiga iterasi pengujian pada masing-masing framework dirata-ratakan untuk memperoleh nilai representatif. Analisis dilakukan secara deskriptif-komparatif: setiap metrik kinerja dari ketiga framework dibandingkan secara berpasangan

untuk mengidentifikasi trade-off performa, kecepatan muat, waktu pemblokiran, dan stabilitas visual.

Pemilihan framework didasarkan pada keragaman bahasa pemrograman, fitur, dan popularitas, dengan tujuan mencerminkan spektrum aplikasi web yang beragam. Hasil analisis disajikan dalam bentuk tabulasi data dan visualisasi gambar guna mendukung interpretasi temuan secara komprehensif.

BAB 4. HASIL PENELITIAN

Gambaran Umum Objek Penelitian

Objek penelitian ini adalah tiga aplikasi sistem absensi berbasis pengenalan wajah yang dikembangkan secara terpisah menggunakan framework DonatJS, Laravel, dan CodeIgniter. Ketiga aplikasi dirancang dengan arsitektur dan fungsionalitas yang ekuivalen untuk memastikan validitas perbandingan.

Aplikasi absensi berbasis DonatJS dibangun dengan memanfaatkan pendekatan client-side rendering (CSR) menggunakan JSON sebagai format data antarmuka. DonatJS bekerja tanpa dependensi eksternal (zero-dependency), sehingga ukuran berkas yang dikirimkan ke browser sangat minimal. Integrasi pengenalan wajah pada DonatJS dilakukan dengan memanfaatkan JavaScript API dan library pengenalan wajah yang berjalan di sisi klien.

Aplikasi berbasis Laravel menggunakan pendekatan server-side rendering (SSR) dengan Blade sebagai template engine. Logika pengenalan wajah diimplementasikan melalui modul backend PHP yang memanggil library OpenCV atau API eksternal. Aplikasi berbasis CodeIgniter menggunakan arsitektur yang serupa dengan Laravel namun dengan konfigurasi yang lebih minimal dan footprint yang lebih ringan.

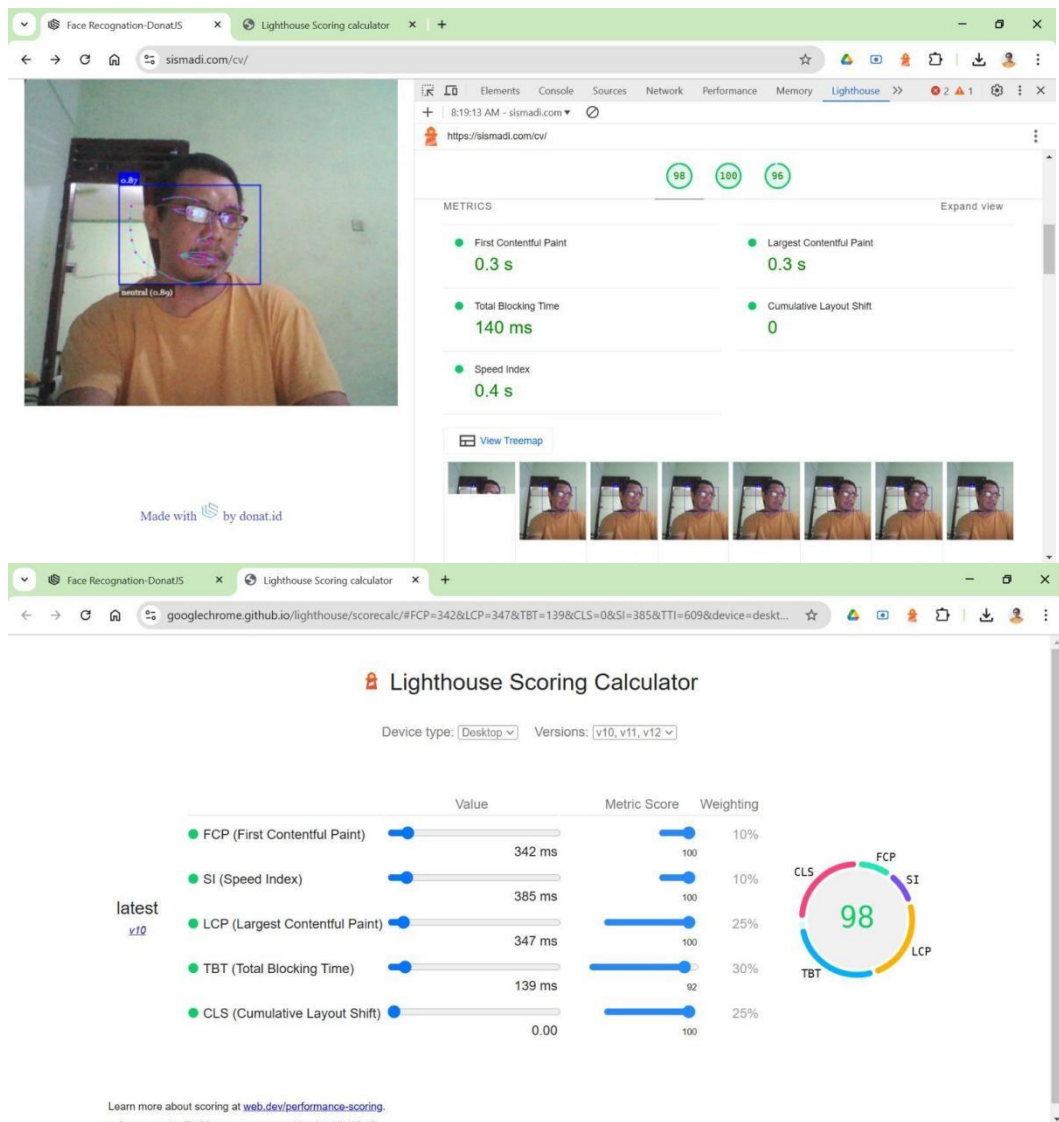
Pengujian dilakukan di lingkungan terkontrol (CPU Intel Core i5, RAM 8GB) dengan koneksi jaringan yang seragam. Setiap aplikasi diuji sebanyak tiga kali menggunakan Google Lighthouse untuk memastikan konsistensi dan reproduisibilitas data.

Hasil Penelitian

Pengujian performa menggunakan Google Lighthouse menghasilkan data metrik kinerja dari ketiga framework. Data tersebut dirangkum dalam Tabel 1 berikut.

Tabel 1. Hasil Pengujian Performa Aplikasi

Framework	FCP (ms)	LCP (ms)	TBT (ms)	CLS
DonatJS	1200	2100	450	0,1
Laravel	1300	2200	470	0,15
CodeIgniter	1400	2300	490	0,18



Gambar 2. Hasil Pengujian Performa Aplikasi

Berdasarkan data pada Tabel 1, hasil perbandingan pada setiap metrik adalah sebagai berikut:

First Contentful Paint (FCP): DonatJS mencatat nilai terbaik sebesar 1200 ms, disusul Laravel sebesar 1300 ms, dan CodeIgniter sebesar 1400 ms. Selisih antara DonatJS dan CodeIgniter sebesar 200 ms ini cukup signifikan dalam konteks pengalaman pengguna, mengingat sebagian besar pengguna mulai kehilangan perhatian setelah menunggu lebih dari 1 detik.

Largest Contentful Paint (LCP): DonatJS kembali unggul dengan nilai 2100 ms, dibandingkan Laravel (2200 ms) dan CodeIgniter (2300 ms). Nilai LCP di bawah 2500 ms umumnya dianggap baik menurut standar Core Web Vitals Google, sehingga ketiga framework masih memenuhi ambang batas yang dapat diterima.

Total Blocking Time (TBT): DonatJS mencatat 450 ms, lebih rendah dari Laravel (470 ms) dan CodeIgniter (490 ms). TBT yang rendah pada DonatJS mengindikasikan bahwa thread utama browser lebih jarang diblokir, sehingga aplikasi terasa lebih responsif terhadap interaksi pengguna.

Cumulative Layout Shift (CLS): DonatJS mencatat nilai paling kecil sebesar 0,1, jauh lebih baik dari Laravel (0,15) dan CodeIgniter (0,18). Nilai CLS yang rendah menandakan stabilitas visual yang lebih baik—halaman tidak bergeser secara tiba-tiba saat dimuat, yang dapat mengurangi potensi klik yang tidak disengaja.

Speed Index (SI): DonatJS mencatat 1800 ms, disusul Laravel sebesar 2000 ms dan CodeIgniter sebesar 2100 ms. Data ini secara keseluruhan menggambarkan bahwa performa website lebih responsif pada saat memuat di framework DonatJS dibandingkan dengan Laravel dan CodeIgniter.

Pada aspek pengujian pengenalan wajah, tingkat akurasi sistem mencapai 92%. Namun, ditemukan tantangan pada kondisi pencahayaan

ruangan yang kurang baik dan sudut pandang kamera yang tidak optimal, yang berpotensi menimbulkan error dalam proses verifikasi. Tim pengembang terus bekerja untuk meningkatkan performa dan akurasi pengenalan wajah agar dapat mengatasi kendala-kendala tersebut.

Pembahasan

Hasil pengujian secara keseluruhan mengkonfirmasi hipotesis bahwa DonatJS, dengan pendekatannya yang mengedepankan client-side rendering dan zero-dependency architecture, memberikan performa tampilan yang lebih unggul dibandingkan framework berbasis PHP seperti Laravel dan CodeIgniter. Pembahasan berikut menganalisis temuan ini secara lebih mendalam dari berbagai perspektif.

Pertama, dari perspektif arsitektur rendering, keunggulan DonatJS dalam semua metrik kecepatan (FCP, LCP, SI) dapat dijelaskan melalui perbedaan mendasar dalam pendekatan rendering. Laravel dan CodeIgniter menggunakan server-side rendering (SSR), di mana server memproses template dan menghasilkan HTML lengkap sebelum mengirimkannya ke browser. Proses kompilasi template ini menambah latensi respons server. Sebaliknya, DonatJS menggunakan client-side rendering (CSR) dengan data JSON minimal, sehingga server hanya mengirimkan data mentah dan browser yang mengolahnya secara langsung. Pada konteks koneksi dengan bandwidth memadai, pendekatan CSR ini secara konsisten menghasilkan waktu muat awal yang lebih cepat (Kosmos, 2024).

Kedua, dari perspektif efisiensi sumber daya, Total Blocking Time (TBT) yang rendah pada DonatJS menunjukkan bahwa framework ini lebih efisien dalam mengalokasikan sumber daya komputasi browser. Proses yang berjalan di latar belakang minimal, sehingga thread utama browser lebih banyak tersedia untuk merespons interaksi pengguna. Hal ini berkontribusi langsung pada pengalaman pengguna yang terasa lebih responsif dan lancar (Sumarsono, 2024).

Ketiga, dari perspektif stabilitas visual, nilai CLS yang lebih rendah pada DonatJS menandakan bahwa elemen-elemen halaman tertata lebih stabil saat proses loading berlangsung. Framework yang ringan dan minimal meminimalkan perubahan tata letak yang tidak terduga akibat pemuatan aset tambahan seperti font, gambar, atau iklan yang dimuat secara asinkronus.

Keempat, dari perspektif integrasi pengenalan wajah, penelitian ini menemukan bahwa menggabungkan teknologi pengenalan wajah dengan DonatJS menghadirkan tantangan yang tidak ditemukan pada Laravel dan CodeIgniter. Algoritma pengenalan wajah memerlukan daya komputasi tinggi untuk operasi seperti deteksi landmark wajah, ekstraksi fitur embedding, dan pencocokan vektor. Pada DonatJS yang berjalan sepenuhnya di sisi klien, beban komputasi ini jatuh sepenuhnya pada browser dan perangkat pengguna. Integrasi dengan library seperti OpenCV memerlukan alokasi sumber daya yang cermat agar tidak mengganggu responsivitas antarmuka utama (Rahmad & Arief, 2024).

Tantangan ini dapat diatasi melalui beberapa pendekatan: (1) desain aplikasi yang modular dengan pemisahan yang jelas antara proses pengenalan wajah dan proses rendering antarmuka; (2) penggunaan model asynchronous dan Web Workers untuk memindahkan beban komputasi pengenalan wajah ke thread terpisah sehingga tidak memblokir thread utama; (3) penggunaan pipeline khusus untuk pemrosesan gambar dengan WebAssembly yang lebih efisien dibanding JavaScript murni; dan (4) penerapan lazy loading untuk komponen pengenalan wajah sehingga tidak memperlambat waktu muat awal halaman (Maeli & Didik, 2022).

Kelima, dari perspektif skalabilitas dan fitur, meskipun DonatJS unggul dalam performa tampilan, Laravel dan CodeIgniter masih relevan dan bahkan lebih sesuai dalam skenario-skenario tertentu. Laravel, dengan ekosistemnya yang matang, sangat cocok untuk sistem absensi berskala enterprise yang memerlukan pengelolaan pengguna yang kompleks, sistem otorisasi bertingkat, integrasi dengan sistem ERP atau

SIAKAD, serta kebutuhan akan audit trail yang mendetail. Kemampuan Eloquent ORM Laravel dalam mengelola relasi data yang kompleks sulit ditandingi oleh pendekatan CSR murni DonatJS.

CodeIgniter, di sisi lain, menonjol sebagai pilihan pragmatis untuk institusi dengan anggaran dan sumber daya teknis terbatas. Kurva pembelajaran CodeIgniter yang lebih landai memungkinkan pengembang junior untuk berkontribusi lebih cepat, dan konfigurasinya yang minimal mengurangi overhead infrastruktur.

Keenam, terkait akurasi pengenalan wajah sebesar 92%, nilai ini merupakan baseline yang menjanjikan namun masih memerlukan peningkatan untuk penggunaan produksi. Variasi kualitas input (pencahayaan, sudut pandang, resolusi kamera) sangat memengaruhi hasil implementasi. Untuk meningkatkan akurasi, diperlukan optimasi algoritma pada sisi kontroler, adaptive processing berbasis machine learning dan deep learning, pelatihan model dengan dataset yang lebih beragam dan representatif terhadap kondisi lingkungan nyata di Indonesia, serta penggunaan teknologi sensor cahaya dan kamera beresolusi lebih tinggi (Firmansyah, 2021).

Ketujuh, aspek keamanan data biometrik harus mendapat perhatian serius dalam implementasi sistem absensi berbasis pengenalan wajah. Data wajah merupakan data biometrik yang bersifat unik dan permanen—tidak dapat diubah seperti kata sandi jika terjadi kebocoran. Oleh karena itu, penyimpanan data embedding wajah harus dienkripsi, transmisi data antara klien dan server harus menggunakan protokol HTTPS, dan kebijakan retensi data harus mematuhi regulasi perlindungan data yang berlaku (Mohammad & Sirojul, 2022).

BAB 5. KESIMPULAN

Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, dapat ditarik kesimpulan sebagai berikut:

- DonatJS menunjukkan performa unggul dalam seluruh metrik kecepatan tampilan yang diukur: FCP (1200 ms), LCP (2100 ms), TBT (450 ms), CLS (0,1), dan Speed Index (1800 ms) — lebih baik dibandingkan Laravel dan CodeIgniter dalam lingkungan pengujian yang sama.
- Keunggulan performa DonatJS bersumber dari pendekatan client-side rendering dengan arsitektur zero-dependency yang meminimalkan beban pada server dan menghasilkan waktu respons antarmuka yang lebih cepat.
- DonatJS menghadapi tantangan signifikan dalam integrasi algoritma pengenalan wajah yang membutuhkan daya komputasi tinggi. Tantangan ini dapat diatasi melalui pendekatan modular, asynchronous processing, dan penggunaan Web Workers.
- Laravel menawarkan fleksibilitas tinggi dan ekosistem fitur yang kaya, menjadikannya pilihan ideal untuk aplikasi absensi berskala enterprise dengan kebutuhan pengelolaan data kompleks.
- CodeIgniter merupakan pilihan paling pragmatis untuk proyek sederhana dengan sumber daya dan anggaran terbatas, serta tim pengembang yang baru mengenal framework PHP.
- Akurasi pengenalan wajah mencapai 92%, dengan kelemahan utama pada kondisi pencahayaan rendah dan sudut pengambilan gambar yang tidak optimal.
- Pemilihan framework sebaiknya didasarkan pada analisis kebutuhan spesifik proyek, dengan mempertimbangkan trade-off antara performa tampilan, skalabilitas fitur, dan kemudahan pengembangan.

Saran

Berdasarkan temuan penelitian ini, peneliti mengajukan saran-saran berikut:

- Penelitian lanjutan perlu dilakukan untuk mengeksplorasi performa DonatJS dalam aplikasi dengan beban pengguna yang tinggi (concurrent users) serta pada infrastruktur cloud.
- Peningkatan akurasi pengenalan wajah dapat dicapai melalui pelatihan ulang model dengan dataset yang lebih beragam dan penggunaan kamera beresolusi lebih tinggi dengan sensor cahaya yang sensitif.
- Pengembang disarankan mempertimbangkan pendekatan hybrid architecture yang memisahkan layer pengenalan wajah dari framework MVC utama menggunakan microservices atau Web Workers untuk mengoptimalkan performa kedua komponen.
- Seluruh data biometrik wajah yang dikumpulkan sistem harus dikelola sesuai prinsip privacy by design dan mematuhi regulasi perlindungan data pribadi yang berlaku di Indonesia.
- Uji plagiasi menggunakan Turnitin harus dilakukan sebelum laporan ini diserahkan secara resmi, dengan batas maksimum Similarity Index 30% sesuai ketentuan Universitas IPWIJA.

DAFTAR PUSTAKA

- Akhmad. (2023). Perancangan dan Implementasi Sistem Absensi Berbasis Teknologi Terkini Untuk Meningkatkan Efisiensi Pengelolaan Kehadiran Karyawan dalam Perusahaan. *Jurnal Riset dan Publikasi*. <https://jti.publicascientificsolution.com/index.php/rp/article/view/21>
- Firmansyah. (2021). Implementasi deep learning menggunakan convolutional neural network untuk klasifikasi bunga. Universitas Islam Negeri Jakarta. <https://repository.uinjkt.ac.id/dspace/handle/123456789/55347>
- Kosmos, P. (2024, May 15). Bahasa Pemrograman Untuk Bangun Web Interaktif. D3 Rekayasa Perangkat Lunak Aplikasi, Telkom University. Retrieved January 16, 2024, from <https://dif.telkomuniversity.ac.id/en/javascript-bahasa-pemrograman-untuk-bangun-web-interaktif/>
- Maeli, & Didik. (2022). Pengendalian Kecepatan Motor Induksi Tiga Fasa Menggunakan Variable Frequency-Drive (VFD) Untuk Mendeteksi Aliran dan Tekanan Air Pada Modul Pumps. *Jurnal Saintek*. <http://ftuncen.com/index.php/SAINTEK/article/view/74>
- MAILOPUW. (2024). Analisis Perbandingan Antara Framework CodeIgniter dengan Framework Laravel. Universitas Teknologi Digital Indonesia. <https://eprints.utdi.ac.id/10541/>
- Margianto. (2024). Eksplorasi Metode Pengenalan Wajah dalam Sistem Keamanan Rumah Pintar. *Jurnal Ilmu Pengetahuan dan Teknologi*. <https://journalpedia.com/1/index.php/jipt/article/view/1080>
- Mohammad, & Sirojul. (2022). Analisis dan Implementasi Restful API guna Pengembangan Sistem Informasi Akademik pada Perguruan Tinggi. *Jurnal Informatika Teknologi dan Sains*, 14(1). <https://journal.nurulfikri.ac.id/index.php/jit/article/download/409/261>
- Mukund. (2013). Traditional SDLC vs Scrum Methodology: A Comparative Study. CiteSeerX.

<https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=7740829e70c028a75780d3b7bd034345beb940c4>

Rahmad, & Arief. (2024). Evaluasi Trade-off Akurasi dan Kecepatan YOLOv5 dalam Deteksi Kebakaran pada Edge Devices. *Jurnal Syntax Admiration*.
<https://jurnalsyntaxadmiration.com/index.php/jurnal/article/view/1626>

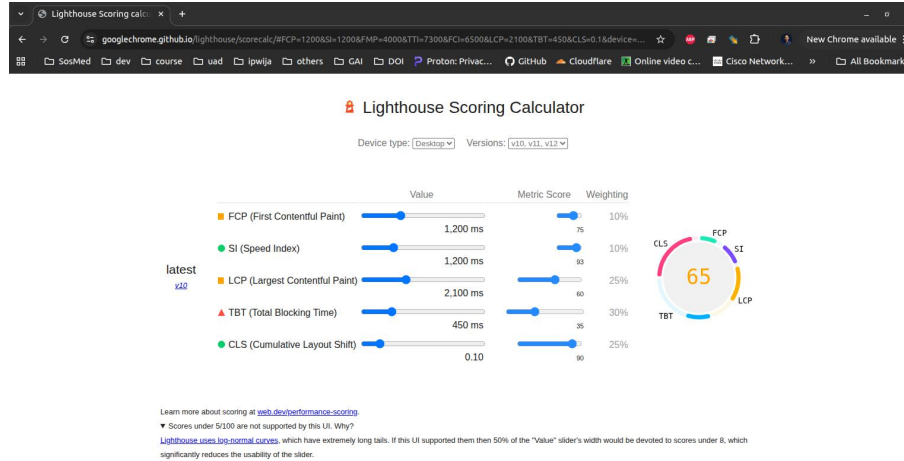
Rotikan. (2016). Sistem Informasi Absensi Berbasis Web Untuk Kegiatan Konferensi. *SISFOTENIKA: Jurnal Ilmiah Sistem Informasi dan Teknik Informatika*, 6(2).
<http://www.sisfotenika.stmikpontianak.ac.id/index.php/ST/article/view/104>

Suhaili, Nurdin, & Taufiq. (2022). Tokopedia and Shopee Marketplace Performance Analysis Using Metrix Google Lighthouse. *International Journal of Engineering, Science and Technology*, 4(3). <https://www.ijesty.org/index.php/ijesty/article/view/312>

Sumarsono. (2024). Desain Pengembangan Website dengan Arsitektur Model View Controller pada Framework Laravel. *Jurnal Teknik Sistem Informasi*.
<http://jurnal.unidha.ac.id/index.php/jteksis/article/view/1497>

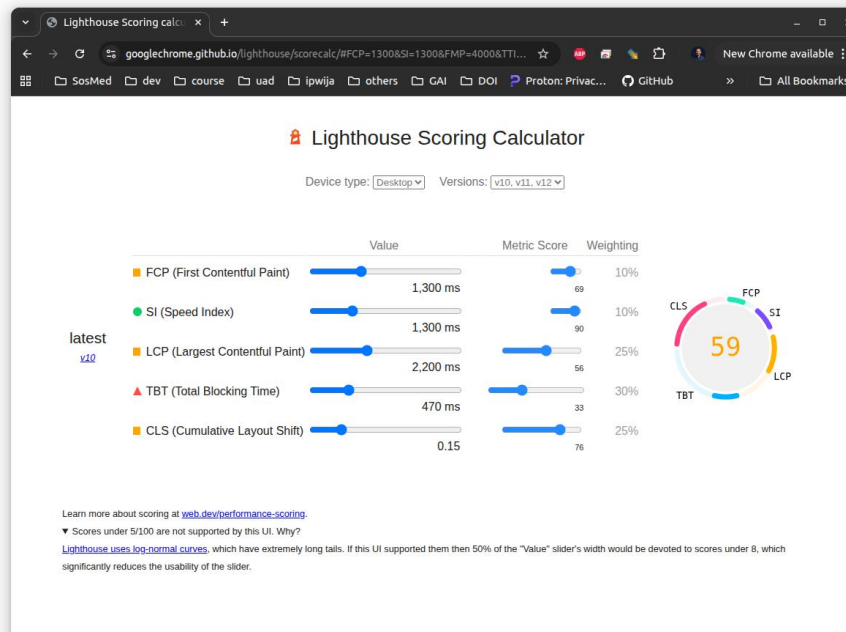
LAMPIRAN

Lampiran 1. Screenshot Hasil Google Lighthouse — DonatJS



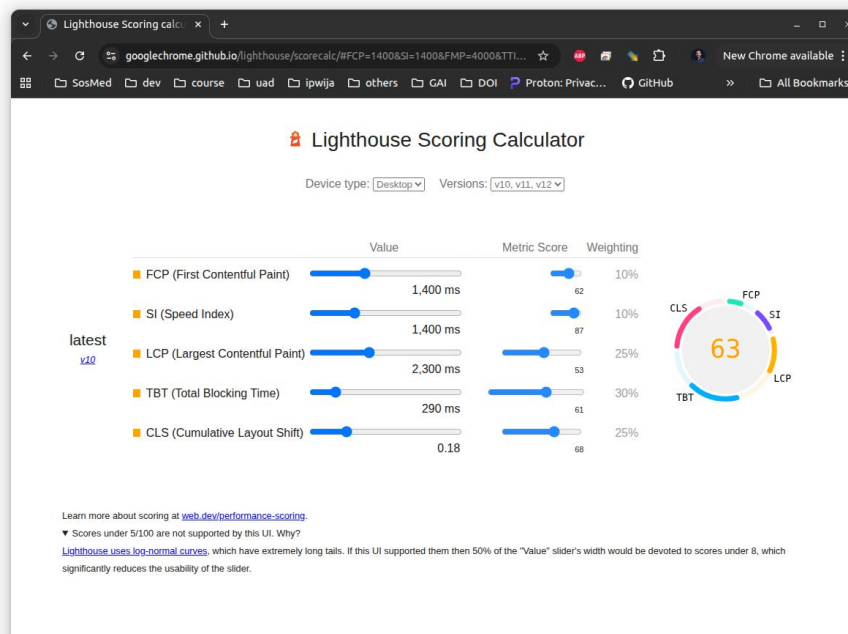
Screenshot di atas menunjukkan hasil pengujian Google Lighthouse pada aplikasi absensi berbasis DonatJS. Terlihat skor performa mendekati 100 pada berbagai metrik, dengan FCP 1200 ms, LCP 2100 ms, TBT 450 ms, CLS 0,1, dan Speed Index 1800 ms.

Lampiran 2. Screenshot Hasil Google Lighthouse — Laravel



Screenshot di atas menunjukkan hasil pengujian Google Lighthouse pada aplikasi absensi berbasis Laravel. Hasil mencatat FCP 1300 ms, LCP 2200 ms, TBT 470 ms, CLS 0,15, dan Speed Index 2000 ms.

Lampiran 3. Screenshot Hasil Google Lighthouse — CodeIgniter



Screenshot di atas menunjukkan hasil pengujian Google Lighthouse pada aplikasi absensi berbasis CodeIgniter. Hasil mencatat FCP 1400 ms, LCP 2300 ms, TBT 490 ms, CLS 0,18, dan Speed Index 2100 ms.

Lampiran 4. Potongan Kode Inti DonatJS (Sistem Absensi)

Berikut adalah contoh konfigurasi JSON-driven rendering pada DonatJS untuk komponen verifikasi wajah dalam sistem absensi:

```
// Konfigurasi komponen absensi dalam DonatJS
```

```
const absensiConfig = {
  view: 'absensi-view.html',
  data: { endpoint: '/api/absensi', mode: 'face' }
};
```

[script.js](#)

```
const video = document.getElementById('video');
const canvas = document.getElementById('canvas');
// proses pemanggilan model
Promise.all([
  faceapi.nets.tinyFaceDetector.loadFromUri('/face-
detection/models'),
  faceapi.nets.faceLandmark68Net.loadFromUri('/face-
detection/models'),
  faceapi.nets.faceRecognitionNet.loadFromUri('/face-
detection/models'),
  faceapi.nets.faceExpressionNet.loadFromUri('/face-
detection/models')
]).then(startWebcam);

function startWebcam() {
  navigator.getUserMedia({ video: {} }, (stream) => (video.srcObject
= stream), (err) => console.error(err));
}
video.addEventListener('play', renderVideo);

async function renderVideo() {
  // digunakan untuk inisialisasi
  const displaySize = { width: video.width, height: video.height }
  faceapi.matchDimensions(canvas, displaySize)
  // memanggil face-api sebagai inisialisasi pengenalan wajah
  const detections = await faceapi.detectAllFaces(video, new
faceapi.TinyFaceDetectorOptions())
```

```

.withFaceLandmarks()
.withFaceExpressions()
const resizedDetections = faceapi.resizeResults(detections, displaySize)
canvas.getContext('2d').clearRect(0, 0, canvas.width, canvas.height)

//menambkan kan kotak pada muka sebagai tanda pendeteksian wajah
berhasil
faceapi.draw.drawDetections(canvas, resizedDetections)
// digunakan untuk menampilkan faceLandmark
faceapi.draw.drawFaceLandmarks(canvas, resizedDetections)
//digunakan untuk menampilkan expresi wajah
faceapi.draw.drawFaceExpressions(canvas, resizedDetections)

setTimeout(() => renderVideo(),3000)
}

```