

PERANCANGAN WEBSITE SHOWROOM MOBIL BERBASIS LARAVEL UNTUK EFISIENSI AKSES DAN UKURAN

Radja Ravine Salfriandry¹, Riza Ibnu Adam², Carudin³

^{1,2,3}Universitas Singaperbangsa Karawang; Jl. HS.Ronggo Waluyo, Puseurjaya, Telukjambe Timur, Karawang, Jawa Barat 41361; (0267) 641177

Keywords:

Laravel, Car Showroom, Agile Kanban, Access Speed, Optimization, Image Compression

Correspondent Email:

2110631170091@student.unsika.ac.id

Abstrak. Dalam era digital yang semakin kompetitif, performa website menjadi faktor krusial dalam menentukan kualitas layanan dan kepuasan pengguna. Kecepatan akses serta efisiensi ukuran berkas merupakan dua aspek utama yang berpengaruh terhadap pengalaman pengguna dalam mengakses sebuah platform daring. Berdasarkan kebutuhan tersebut, dilakukan perancangan dan pengembangan website showroom mobil berbasis Laravel dengan fokus pada pengurangan ukuran berkas dan peningkatan kecepatan akses, menggunakan studi kasus pada showroom ARR Motor. Metode yang digunakan adalah pendekatan Agile dengan model Kanban, yang meliputi tahapan analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan evaluasi. Strategi optimasi yang diterapkan mencakup kompresi gambar, lazy loading, caching, serta optimalisasi query database melalui teknik eager loading. Hasil pengujian dengan Laravel Debugbar dan User Acceptance Test (UAT) menunjukkan adanya peningkatan signifikan terhadap efisiensi waktu muat dan penggunaan sumber daya sistem. Temuan ini menunjukkan bahwa penerapan teknik optimasi dalam framework Laravel mampu menghasilkan website showroom yang efisien, ringan, dan responsive.



Copyright © [JITET](#) (Jurnal Informatika dan Teknik Elektro Terapan). This article is an open access article distributed under terms and conditions of the Creative Commons Attribution (CC BY NC)

Abstract. In an increasingly competitive digital era, website performance is a crucial factor in determining service quality and user satisfaction. Access speed and file size efficiency are two main aspects that affect user experience in accessing an online platform. Based on these needs, a Laravel-based car showroom website design and development is carried out with a focus on reducing file size and increasing access speed, using a case study at ARR Motor showroom. The method used is the Agile approach with the Kanban model, which includes the stages of needs analysis, system design, implementation, testing, and evaluation. The optimization strategy applied includes image compression, lazy loading, caching, and database query optimization through eager loading techniques. Test results with Laravel Debugbar and User Acceptance Test (UAT) showed a significant improvement in the efficiency of load time and system resource usage. These findings show that the application of optimization techniques in the Laravel framework can produce an efficient, lightweight, and responsive showroom website.

1. PENDAHULUAN

Dalam era digital saat ini, performa website memegang peranan penting dalam menentukan kualitas layanan dan kepuasan pengguna. Waktu muat halaman yang lambat dapat menyebabkan pengguna meninggalkan situs sebelum mengakses informasi secara menyeluruh. Analisis performa *website* menggunakan alat seperti Pingdom Tools dan Page Speed Insights menunjukkan bahwa waktu muat halaman yang cepat dapat meningkatkan kenyamanan pengguna dan menurunkan potensi *bounce rate* [1]. Studi lain juga mengidentifikasi permasalahan serupa serta menawarkan solusi untuk meningkatkan performa, di antaranya melalui penerapan teknik optimasi seperti *cache browser*, aktivasi *gzip compression*, dan penghapusan *rendering-blocking resources*. Hasilnya, waktu muat halaman dan respons sistem membaik meskipun situs menerima banyak permintaan, yang menunjukkan bahwa strategi optimasi teknis memiliki peran penting dalam meningkatkan efisiensi dan kenyamanan akses website [2]. Namun, penerapan strategi tersebut masih belum banyak dieksplorasi secara mendalam dalam pengembangan website berbasis Laravel, terutama dalam konteks bisnis digital seperti showroom mobil.

Selain optimasi melalui framework dan pengurangan ukuran file, peningkatan kecepatan akses juga dapat dicapai melalui efisiensi pada sisi teknis lainnya, seperti query database, kompresi gambar, dan teknik rendering. Penggunaan stored procedure pada sistem berbasis MySQL terbukti mampu mempercepat proses manipulasi data, meskipun memerlukan kode yang lebih kompleks dan waktu implementasi yang lebih panjang [3]. Di sisi lain, kompresi gambar dapat mengurangi ukuran file hingga 60% dan mempercepat waktu muat, namun beberapa metode kurang efektif jika digunakan pada gambar berukuran besar [4]. Teknik lazy loading juga dapat mempercepat waktu muat dengan mengurangi beban server, namun jika tidak disertai penyesuaian UI yang tepat seperti pengaturan tata letak dan transisi, pengguna dapat mengalami keterlambatan tampilan konten yang berdampak pada kepuasan pengguna [5].

2. TINJAUAN PUSTAKA

2.1. Website

Website adalah kumpulan halaman yang saling terhubung dalam sebuah domain atau subdomain pada jaringan internet, yang dirancang untuk menyajikan informasi dalam berbagai format, seperti teks, gambar, video, dan animasi. Sebagai salah satu fasilitas utama di era digital, website memungkinkan pengguna untuk mengakses informasi dari mana saja dan kapan saja melalui perangkat yang terhubung ke internet. *Website* terdiri dari kumpulan halaman yang saling terkait, dengan *homepage* sebagai pusatnya. Website tidak hanya menjadi sarana komunikasi dan penyebaran informasi, tetapi juga alat penting dalam berbagai sektor, termasuk pendidikan, bisnis, dan pemerintahan [6].

2.2. HTML (HyperText Markup Language)

HTML (*HyperText Markup Language*) adalah bahasa yang digunakan untuk membuat dan menyusun halaman *web*, memungkinkan teks, gambar, video, dan elemen lain untuk ditampilkan di *browser*. Secara umum, HTML berfungsi sebagai kerangka kerja utama dari sebuah situs *web*, memungkinkan integrasi dengan CSS untuk mempercantik tampilan dan JavaScript untuk menambahkan interaktivitas. Implementasi HTML yang efisien juga dapat mendukung pengelolaan sistem informasi berbasis *web* untuk meningkatkan aksesibilitas dan efektivitas, seperti yang dijelaskan dalam penelitian terkait perancangan sistem informasi berbasis *web* [7].

2.3. CSS (Cascading Style Sheet)

CSS (*Cascading Style Sheets*) adalah bahasa yang digunakan untuk mengatur tampilan dan format elemen pada halaman *web*. CSS memungkinkan pengembang untuk memisahkan konten dari presentasi, sehingga desain dapat dikelola secara lebih efisien. Dengan CSS, elemen seperti warna, *font*, *layout*, dan responsivitas halaman dapat diatur secara terpusat, mendukung kemudahan pemeliharaan serta fleksibilitas desain *web*. Penggunaan CSS juga dapat meningkatkan pengalaman pengguna dengan tampilan yang lebih konsisten dan menarik pada berbagai perangkat [8].

2.4. Javascript

JavaScript adalah bahasa pemrograman yang banyak digunakan untuk menambahkan interaktivitas dan fungsi dinamis pada halaman web. Awalnya dikembangkan untuk pengembangan *front-end*, JavaScript kini juga mendukung pengembangan *back-end* melalui teknologi seperti Node.js, yang memungkinkan eksekusi kode di sisi *server*. Dengan *sintaks* yang fleksibel dan dukungan terhadap paradigma pemrograman imperatif, berorientasi objek, dan fungsional, JavaScript menawarkan berbagai kemampuan untuk menciptakan aplikasi web yang efisien, responsif, dan kompleks [9].

2.5. Bootstrap

Bootstrap adalah *framework* CSS yang dirancang untuk mempermudah pengembangan tampilan *website* yang responsif dan modern. Dengan menyediakan komponen-komponen siap pakai seperti *grid system*, tombol, formulir, dan navigasi, Bootstrap memungkinkan pengembang untuk membangun desain yang konsisten dengan efisiensi tinggi. *Framework* ini juga mendukung kompatibilitas lintas *browser* dan perangkat, sehingga menjadi pilihan populer dalam pengembangan *website e-commerce* maupun aplikasi berbasis web lainnya [10].

2.6. PHP (Hypertext Preprocessor)

PHP (*Hypertext Preprocessor*) merupakan bahasa pemrograman yang berjalan di sisi server dan dirancang khusus untuk pengembangan web. PHP memungkinkan programmer menyisipkan perintah-perintah yang dapat diproses oleh perangkat lunak web server seperti Apache maupun IIS, sebelum hasilnya dikirimkan ke browser pengguna [11]. Artinya, seluruh logika yang dituliskan menggunakan PHP akan dieksekusi terlebih dahulu di server, lalu hasil akhir berupa HTML akan ditampilkan kepada pengguna. Salah satu contoh implementasinya adalah kemampuan untuk menampilkan tanggal dan waktu secara otomatis setiap kali halaman diakses. Selain itu, PHP juga mendukung integrasi dengan berbagai sistem basis data seperti MySQL, PostgreSQL, dan SQLite, serta memiliki banyak pustaka pendukung yang mempermudah proses pengembangan aplikasi web yang dinamis dan interaktif.

2.7. Laravel

Laravel adalah *framework* PHP *open-source* yang dirancang untuk mempermudah pengembangan aplikasi berbasis web dengan menerapkan pola *Model-View-Controller* (MVC). *Framework* ini menyediakan berbagai fitur seperti *sintaksis* yang jelas, *library* bawaan, serta sistem *templating* yang memungkinkan pengembang untuk membangun aplikasi dengan lebih cepat dan efisien. Laravel pertama kali diperkenalkan oleh Taylor Otwell pada tahun 2012 dan terus berkembang hingga saat ini sebagai salah satu *framework* PHP yang populer karena kemampuannya dalam meningkatkan kualitas *software* dengan biaya pengembangan yang lebih rendah serta pengalaman pengguna yang optimal [12].

2.8. Showroom Mobil

Showroom mobil merupakan tempat usaha yang menyediakan layanan penjualan dan pertukaran kendaraan, terutama mobil, untuk memenuhi kebutuhan transportasi masyarakat yang beragam. Showroom memiliki peran penting dalam mengidentifikasi preferensi konsumen terhadap jenis dan tipe kendaraan untuk menentukan strategi pemasaran yang efektif. Dengan memahami pola perpindahan minat konsumen, showroom dapat mengantisipasi tren pasar dan menyesuaikan stok kendaraan sesuai kebutuhan pelanggan [13].

2.9. Unified Model Language (UML)

Unified Modeling Language (UML) adalah bahasa pemodelan grafis standar yang digunakan untuk memvisualisasikan, merancang, dan mendokumentasikan sistem perangkat lunak berorientasi objek. UML menyediakan berbagai jenis diagram yang membantu pengembang memahami struktur dan perilaku sistem secara menyeluruh, sehingga mendukung proses pengembangan dan pemeliharaan sistem yang lebih terstruktur [14].

2.10. Black Box Testing

Black box testing adalah metode pengujian perangkat lunak yang menguji fungsionalitas sistem tanpa melihat struktur internal atau kode program. Pengujian dilakukan dengan memberikan input dan

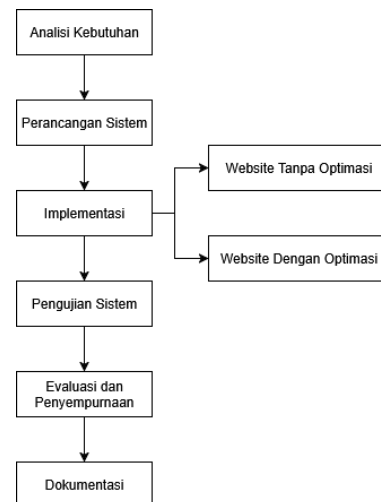
mengamati output untuk memastikan sistem bekerja sesuai spesifikasi. Metode ini efektif dalam memverifikasi apakah fitur utama telah berjalan sesuai kebutuhan pengguna [15].

3. METODE PENELITIAN

Metode penelitian yang digunakan dalam pengembangan sistem informasi *showroom* mobil berbasis Laravel adalah metode *agile* dengan pendekatan *kanban*. Pendekatan ini dipilih karena memberikan fleksibilitas tinggi dalam proses pengembangan perangkat lunak, memungkinkan perubahan kebutuhan secara dinamis, serta memfasilitasi kolaborasi tim yang berkelanjutan. Melalui papan *kanban*, seluruh proses kerja dibagi menjadi beberapa tahapan yang divisualisasikan secara jelas, sehingga memudahkan pemantauan progres dan identifikasi hambatan. Pendekatan *kanban* juga memungkinkan pengelolaan waktu dan sumber daya secara efisien karena setiap tugas dikerjakan secara berurutan berdasarkan prioritas. Metode ini dianggap sesuai untuk proyek berskala kecil hingga menengah yang membutuhkan iterasi cepat, evaluasi berkelanjutan, dan peningkatan kualitas sistem berdasarkan masukan pengguna.

3.1. Rancangan Penelitian

Dalam membangun sistem website *showroom* mobil berbasis Laravel untuk *showroom* ARR MOTOR, tahapan penelitian ini dirancang agar fleksibel dan adaptif terhadap perubahan kebutuhan. Pendekatan *Agile* dengan metode *Kanban* memungkinkan pengembangan sistem dilakukan secara iteratif dan berkelanjutan. Alur rancangan penelitian yang diterapkan dapat dilihat pada Gambar 3.1.



Gambar 3.1 Tahapan Rancangan Penelitian

3.1.1. Analisis Kebutuhan

Perancangan sistem ini mencakup berbagai aspek, termasuk desain antarmuka pengguna (UI/UX), arsitektur sistem, serta strategi optimasi yang digunakan dalam pengembangan website *showroom* mobil berbasis Laravel. Desain UI/UX dibuat agar responsif dan mudah digunakan oleh calon pelanggan *showroom*. Arsitektur sistem ditentukan dengan struktur database yang sesuai dan alur proses yang efisien dalam framework Laravel. Untuk meningkatkan kecepatan akses, diterapkan strategi optimasi seperti *caching*, *lazy loading*, kompresi gambar dan optimasi query yang akan diuji nantinya menggunakan *library* PHP yaitu Laravel Debugbar dan DevTools.

Selain itu, diagram UML digunakan untuk menggambarkan model sistem yang dikembangkan, termasuk diagram *use case*, *sequence diagram*, dan *class diagram* yang merepresentasikan hubungan antar komponen dalam sistem. Desain sistem juga memanfaatkan papan *Kanban* untuk mengelola tugas secara efektif dengan kategori seperti *To Do*, *In Progress*, dan *Done* guna memastikan kelancaran proses pengembangan.

3.1.2. Perancangan Sistem

Perancangan sistem ini mencakup berbagai aspek, termasuk desain antarmuka pengguna (UI/UX), arsitektur sistem, serta strategi optimasi yang digunakan dalam pengembangan website *showroom* mobil berbasis Laravel. Desain UI/UX dibuat agar responsif dan mudah digunakan oleh calon pelanggan *showroom*. Arsitektur sistem ditentukan dengan struktur database yang sesuai dan alur proses yang

efisien dalam *framework* Laravel. Untuk meningkatkan kecepatan akses, diterapkan strategi optimasi seperti *caching*, *lazy loading*, kompresi gambar, dan optimasi *query* yang akan diuji nantinya menggunakan *library* PHP, yaitu Laravel Debugbar dan DevTools.

Selain itu, diagram *UML* digunakan untuk menggambarkan model sistem yang dikembangkan, termasuk diagram *use case*, *sequence diagram*, dan *class diagram* yang merepresentasikan hubungan antarkomponen dalam sistem. Desain sistem juga memanfaatkan papan *kanban* untuk mengelola tugas secara efektif dengan kategori seperti *To Do*, *In Progress*, dan *Done* guna memastikan kelancaran proses pengembangan.

3.1.3. Implementasi

Pengembangan sistem dilakukan secara iteratif menggunakan pendekatan *kanban* untuk memastikan fleksibilitas dan efisiensi kerja. Setiap tugas dipilih berdasarkan prioritas dari kolom *To Do*, kemudian dipindahkan ke *In Progress* dengan menerapkan batas *Work In Progress* agar fokus tetap terjaga dan tidak terjadi beban berlebih dalam pengerjaan. Setelah fitur dikembangkan, dilakukan pengujian sementara guna memastikan kesesuaian dengan spesifikasi yang telah ditentukan. Jika fitur dinyatakan lolos uji, tugas dipindahkan ke kolom *Done*, namun apabila ditemukan kekurangan, tugas dikembalikan ke *In Progress* untuk diperbaiki. Proses ini dilakukan secara berulang sehingga pengembangan sistem menjadi lebih terstruktur, risiko kesalahan berkurang, dan setiap fitur yang diterapkan telah melalui tahap validasi yang memadai.

3.1.4. Pengujian Sistem

Pengujian sistem dilakukan dalam beberapa tahap untuk memastikan bahwa *website* showroom mobil berbasis Laravel yang dikembangkan dapat berjalan secara optimal sesuai dengan kebutuhan pengguna. Metode pengujian yang digunakan antara lain:

1. *Black box testing*, yang bertujuan untuk memverifikasi bahwa seluruh fitur pada sistem bekerja sesuai dengan spesifikasi fungsional tanpa melihat struktur internal kode. Pengujian ini dilakukan dengan memberikan sejumlah input pada sistem dan mengevaluasi output yang dihasilkan.

2. *Kuisi*oner, digunakan untuk mengukur tingkat kepuasan pengguna terhadap tampilan dan kemudahan penggunaan sistem.

Dengan menggabungkan metode pengujian tersebut, sistem diuji baik dari sisi fungsionalitas, performa server, maupun responsivitas sisi klien, guna memastikan kualitas sistem secara menyeluruh sebelum diterapkan.

3.1.5. Evaluasi dan Penyempurnaan

Hasil pengujian digunakan untuk mengevaluasi performa sistem. Penyempurnaan dilakukan dengan fokus pada optimalisasi kecepatan akses menggunakan teknik optimasi seperti *caching*, *lazy loading*, kompresi gambar, dan optimasi *query* untuk mendapatkan hasil kualitas UI/UX berdasarkan umpan balik pengguna. Pengujian awal dilakukan menggunakan *alpha testing* berupa *black box testing*, yaitu metode pengujian yang berfokus pada fungsionalitas sistem tanpa melihat struktur kode internal.

Melalui pengujian ini, pengembang dapat mengidentifikasi adanya kekeliruan fungsional maupun ketidaksesuaian sistem dengan kebutuhan pengguna. Setelah proses evaluasi internal selesai, dilakukan pula *beta testing* dengan melibatkan pengguna nyata untuk menguji aspek *usability*, *performance*, *functionality*, *stability*, dan *user satisfaction*. Umpan balik dari kedua tahap pengujian ini menjadi dasar dalam melakukan penyempurnaan sistem secara iteratif, baik dari sisi teknis maupun pengalaman pengguna.

3.1.6. Dokumentasi

Seluruh proses pengembangan, mulai dari analisis kebutuhan hingga evaluasi, didokumentasikan untuk memudahkan pemeliharaan dan pengembangan lebih lanjut.

4. HASIL DAN PEMBAHASAN

4.1. Hasil Penelitian

Penelitian ini menghasilkan sebuah *website* showroom mobil berbasis Laravel dengan fokus pada pengurangan ukuran serta peningkatan kecepatan akses. *Website* ini dikembangkan berdasarkan kebutuhan showroom ARR MOTOR sebagai sarana katalog mobil yang informatif serta efisien secara kebutuhan antar calon pembeli maupun pemilik. Pengembangan dilakukan dengan pendekatan *Agile*

menggunakan metode Kanban, melalui tahapan iteratif mulai dari analisis kebutuhan, perancangan, implementasi, hingga evaluasi.

4.1.1. Analisis Kebutuhan

Analisis kebutuhan dilakukan melalui observasi dan wawancara dengan pihak showroom ARR Motor. Hasilnya menunjukkan 6 kebutuhan utama:

1. Katalog Mobil Interaktif
Website harus menampilkan daftar mobil secara lengkap, termasuk detail seperti merek, tahun, harga, status, dan gambar dari berbagai sudut.
2. Pengelolaan Data Admin
Admin membutuhkan panel backend untuk menambah, mengubah, menghapus data mobil, mengunggah banyak gambar, serta mengatur status mobil secara manual.
3. Performa Cepat dan Ringan
Dibutuhkan optimasi seperti lazy loading, kompresi gambar, caching, dan eager loading query agar website tetap cepat meski diakses dari perangkat dengan koneksi lambat.
4. Desain Responsif
Website harus tampil baik di semua perangkat (*smartphone, tablet, desktop*) dengan UI/UX yang sederhana namun profesional.
5. Efisiensi Bandwidth dan Server
Ukuran file dan jumlah *request* ke server perlu dikurangi agar *website* tetap ringan dan efisien saat dihosting.
6. Penggunaan Laravel 11
Laravel 11 dipilih karena stabil, ringan, dan memiliki fitur seperti caching dan eager loading yang mendukung performa tinggi serta efisiensi pengelolaan data.

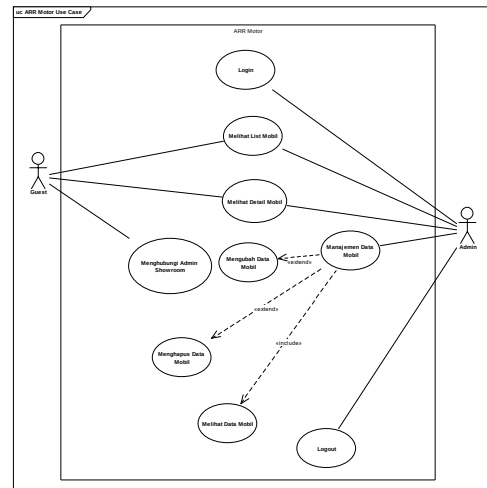
4.1.2. Perancangan Sistem

Perancangan sistem merupakan tahap krusial dalam pengembangan perangkat lunak, di mana penulis mulai merancang model sistem baru yang akan dibangun. Pada tahap ini, dilakukan pendefinisian alur proses dan kebutuhan fungsional sistem secara terstruktur. Keberhasilan dalam perancangan akan sangat menentukan kualitas dan efektivitas sistem yang dikembangkan.

a) Use Case Diagram

Use case diagram digunakan untuk menggambarkan interaksi antara aktor (pengguna sistem) dengan

fungsi-fungsi utama yang disediakan oleh sistem. Diagram ini menunjukkan bagaimana admin dan pengguna umum (*guest*) dapat mengakses fitur-fitur yang tersedia di dalam sistem showroom mobil.

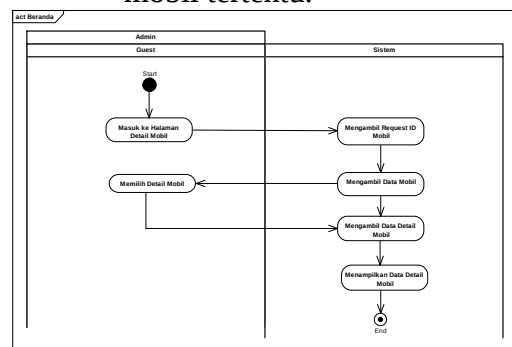


Gambar 4.1 Use Case Diagram Showroom Mobil

b) Activity Diagram

Activity diagram menggambarkan alur aktivitas suatu proses dalam sistem, baik dari sisi pengguna maupun sistem, sehingga alur bisnis dan interaksi dapat dipahami secara jelas dan sistematis.

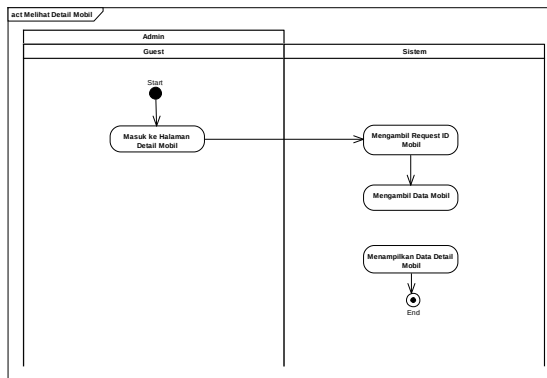
1) *Activity diagram* halaman beranda menunjukkan alur saat admin dan *guest* mengakses situs. Sistem menampilkan halaman dan data mobil, lalu menampilkan detail saat pengguna memilih mobil tertentu.



Gambar 4.2 Activity Diagram Halaman Beranda

2) Activity Diagram Detail Mobil

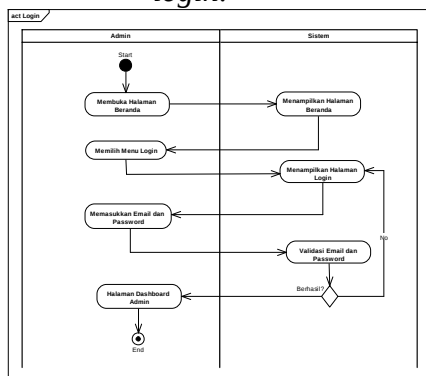
Activity diagram melihat detail mobil menggambarkan alur saat pengguna memilih mobil dari beranda. Sistem mengambil data berdasarkan ID mobil dan menampilkan detail seperti gambar, spesifikasi, tahun, harga, dan status.



Gambar 4.3 Activity Diagram Detail Mobil

3) Activity Diagram Login Admin

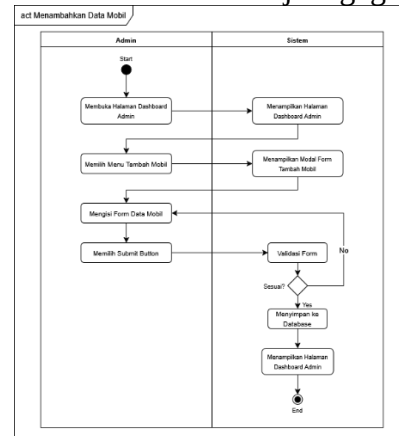
Activity diagram login admin menggambarkan alur saat admin mengakses sistem. Setelah login, sistem memvalidasi data dan memeriksa peran. Jika admin, diarahkan ke dashboard; jika bukan, kembali ke halaman login.



Gambar 4.4 Activity Diagram Login Admin

4) Activity Diagram Menambahkan Data Mobil

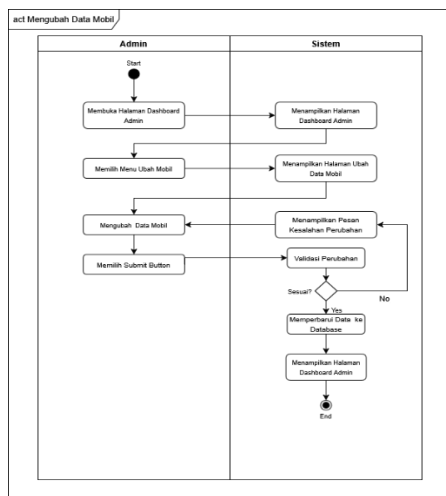
Activity diagram menambahkan data mobil menjelaskan alur saat admin mengisi dan mengirim formulir mobil baru. Sistem memvalidasi input, lalu menyimpan data ke database jika valid, atau menampilkan pesan kesalahan jika gagal.



Gambar 4.5 Activity Diagram Menambahkan Data Mobil

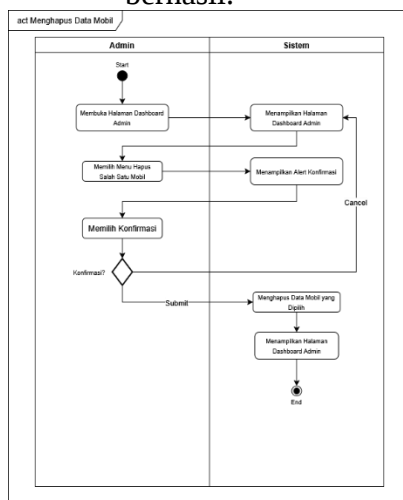
5) Activity Diagram Mengubah Data Mobil

Activity diagram mengubah data mobil menjelaskan proses saat admin memperbarui informasi mobil. Setelah memilih tombol ubah, sistem menampilkan formulir yang dapat diedit. Data divalidasi, lalu diperbarui jika valid, atau ditampilkan pesan kesalahan jika tidak.



Gambar 4.6 Activity Diagram Mengubah Data Mobil

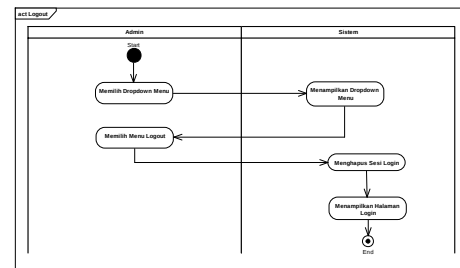
- 6) *Activity Diagram Menghapus Data Mobil*
Activity diagram menghapus data mobil menggambarkan alur saat admin menghapus mobil dari sistem. Setelah memilih menu hapus, sistem menampilkan konfirmasi. Jika disetujui, data dihapus dari *database* dan ditampilkan notifikasi berhasil.



Gambar 4.7 Activity Diagram Menghapus Data Mobil

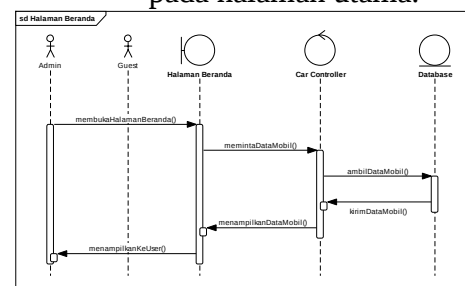
- 7) *Activity Diagram Logout*
Activity diagram logout menjelaskan alur saat admin keluar dari sistem.

Setelah memilih menu akun dan opsi *logout*, sistem menghapus sesi login dan mengarahkan admin ke halaman *login* disertai notifikasi berhasil.



Gambar 4.8 Activity Diagram Logout

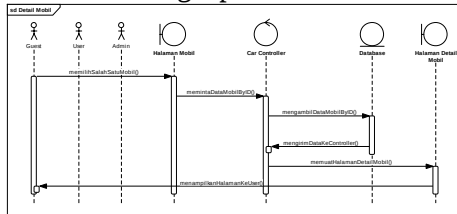
- c) *Sequence Diagram*
Sequence diagram digunakan untuk menggambarkan urutan interaksi antar objek dalam sistem berdasarkan waktu. Diagram ini memvisualisasikan proses dari saat pengguna mengakses fitur hingga sistem merespons secara berurutan.
- 1) *Sequence Diagram Halaman Beranda*
Sequence diagram pada Gambar 4.9 menunjukkan alur saat pengguna mengakses beranda, sistem mengambil data mobil dari *database*, lalu menampilkan katalog pada halaman utama.



Gambar 4.9 Sequence Diagram Halaman Beranda

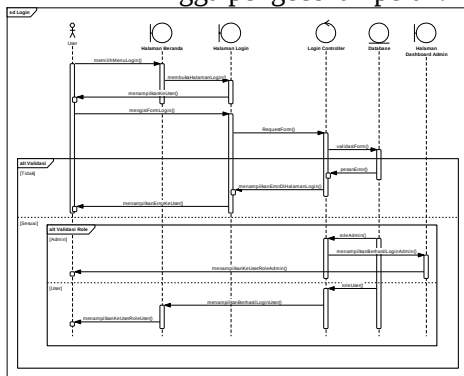
- 2) *Sequence Diagram Halaman Detail Mobil*
Sequence diagram pada Gambar 4.10 menunjukkan alur saat pengguna memilih mobil dari katalog, lalu

sistem mengambil data berdasarkan ID dan menampilkan lengkap mobil.



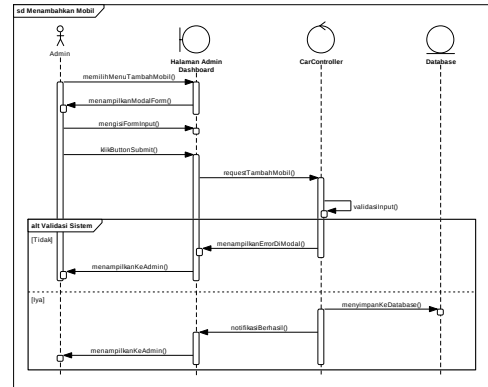
Gambar 4.10 Sequence Diagram Halaman Detail Mobil

- 3) *Sequence Diagram Login*
Sequence diagram pada Gambar 4.11 menunjukkan proses *login* pengguna, mulai dari input email dan kata sandi, validasi data, hingga pengecekan peran.



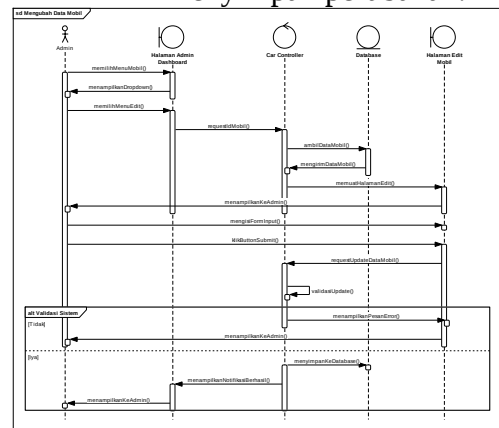
Gambar 4.11 Sequence Diagram Login

- 4) *Sequence Diagram Menambahkan Data Mobil*
Sequence diagram Gambar 4.12 menggambarkan proses admin menambah data mobil, mulai dari pengisian formulir hingga validasi. Jika valid, data disimpan dan muncul notifikasi; jika tidak, ditampilkan pesan kesalahan.



Gambar 4.12 Sequence Diagram Menambahkan Data Mobil

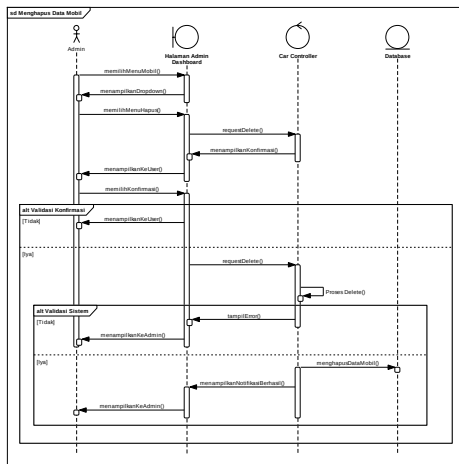
- 5) *Sequence Diagram Mengubah Data Mobil*
Sequence diagram Gambar 4.13 menunjukkan proses admin mengedit data mobil, mulai dari mengambil data, memvalidasi, hingga menyimpan perubahan.



Gambar 4.13 Sequence Diagram Mengubah Data Mobil

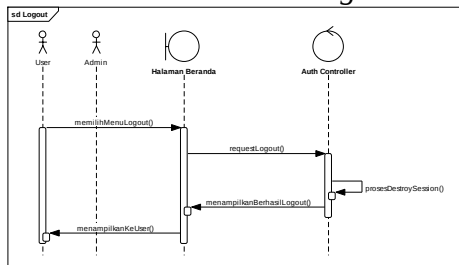
- 6) *Sequence Diagram Menghapus Data Mobil*

Sequence diagram Gambar 4.14 menggambarkan proses admin menghapus data mobil, dimulai dari konfirmasi, validasi, hingga penghapusan data dari *database*.



Gambar 4.14 Sequence Diagram Menghapus Data Mobil

7) *Sequence Diagram Logout*
Gambar 4.15 menunjukkan proses *logout* pengguna, mulai dari pemilihan menu *logout*, penghentian sesi oleh sistem, hingga pengembalian ke halaman beranda dalam kondisi tidak *login*.

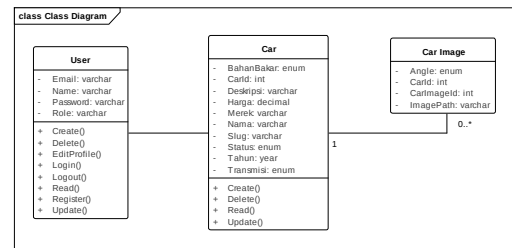


Gambar 4.15 Sequence Diagram Logout

d) Class Diagram

Class diagram pada Gambar 4.16 menunjukkan tiga kelas utama dalam sistem showroom, yaitu User, Car, dan CarImage. Kelas User menangani data pengguna dan autentikasi. Kelas Car menyimpan informasi mobil, sedangkan CarImage menyimpan data gambar mobil dengan relasi *one-to-many* ke Car. Diagram ini menggambarkan

struktur objek dan relasi antar entitas dalam sistem.

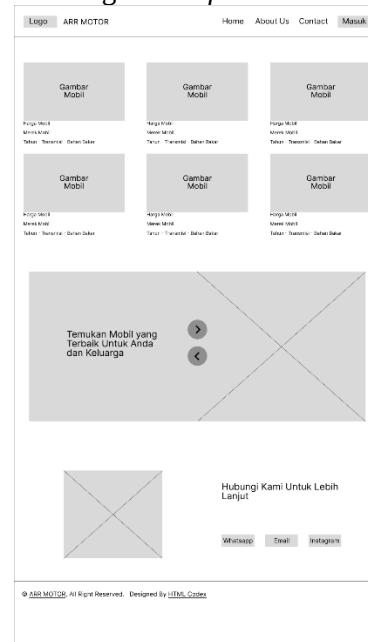


Gambar 4.16 Class Diagram

4.1.3. Perancangan Antarmuka

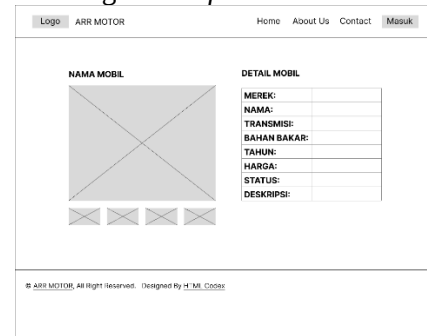
Desain antarmuka mencakup *wireframe* sebagai panduan visual halaman, penempatan elemen seperti navigasi dan katalog mobil secara terstruktur, serta pemilihan warna, tipografi, dan ikon yang selaras dengan identitas showroom. Tujuannya adalah menciptakan tampilan profesional yang memudahkan pengguna dan meningkatkan minat dalam menjelajahi katalog.

a) Rancangan Wireframe Beranda



Gambar 4.17 Wireframe Beranda

b) Rancangan Wireframe Detail Mobil



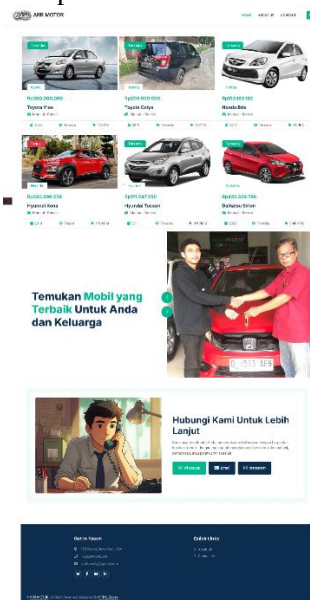
Gambar 4.18 Wireframe Detail Mobil

4.1.4. Implementasi

Implementasi sistem ini menghasilkan website *showroom* mobil berbasis web dengan dua antarmuka utama, yaitu pengguna dan administrator. Pengguna dapat mengakses halaman seperti beranda, katalog produk, tentang kami, dan hubungi kami. Sementara itu, administrator memiliki akses ke dashboard, *login*, dan daftar mobil. Seluruh fitur telah diuji untuk memastikan sistem berjalan dengan baik dan sesuai kebutuhan.

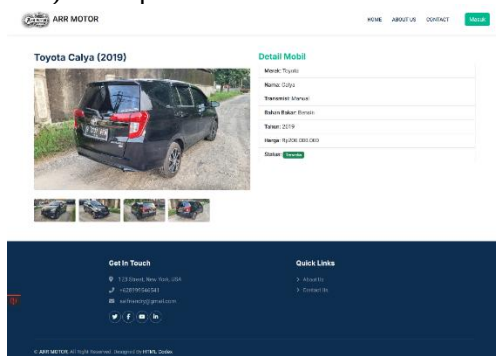
4.1.4.1. Implementasi Tampilan Pengguna

a) Tampilan Beranda

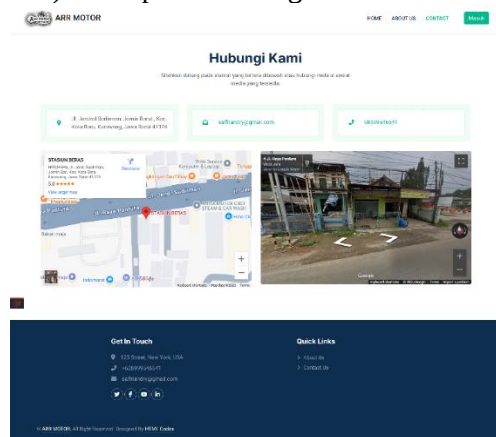


Gambar 4.18 Tampilan Beranda

b) Tampilan Detail Mobil



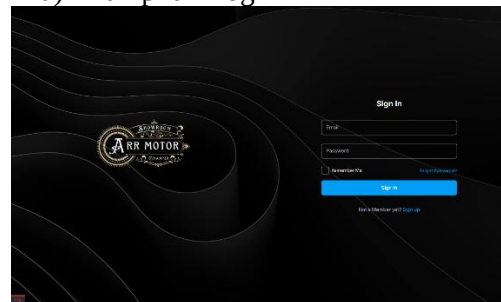
Gambar 4.19 Tampilan Detail Mobil
c) Tampilan Hubungi Kami



Gambar 4.20 Tampilan Tentang Kami

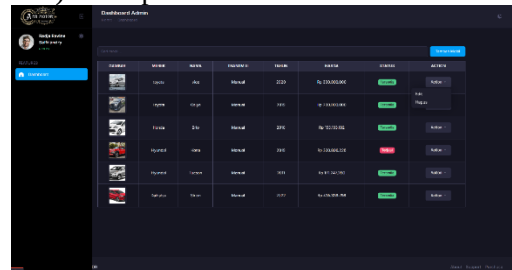
4.1.4.2. Implementasi Tampilan Admin

a) Tampilan Login



Gambar 4.21 Tampilan Login

b) Tampilan Dashboard Admin



Gambar 4.22 Tampilan Dashboard Admin

4.1.5. Pengujian Sistem

Pengujian sistem dilakukan dengan metode *Black Box* untuk menguji fungsi sistem dan skala Likert untuk menilai kepuasan pengguna terhadap tampilan dan kinerja sistem.

4.1.5.1. Black Box Testing

Pengujian dilakukan menggunakan metode *black box* untuk memastikan fungsi sistem *showroom* mobil berjalan dengan benar dan

minim kesalahan pada seluruh halaman aplikasi.

Skenario	Planning	Respon Sistem	Hasil
Memasukkan Halaman Beranda	Menampilkan Halaman Beranda	Menampilkan Halaman Beranda	Berhasil
Memasukkan Halaman Tentang Kami	Menampilkan profil showroom dan visi misi	Menampilkan profil showroom dan visi misi	Berhasil
Memilih Halaman Tentang Kami	Menampilkan profil showroom dan visi misi	Menampilkan profil showroom dan visi misi	Berhasil
Memilih Menu Detail Mobil	Menampilkan Halaman Detail Mobil	Menampilkan Halaman Detail Mobil	Berhasil
Memilih Menu Login	Menampilkan Halaman Login	Menampilkan Halaman Login	Berhasil
Memilih Menu Tambah Mobil	Menampilkan Modal Form Tambah Mobil	Menampilkan Modal Form Tambah Mobil	Berhasil
Memilih Menu Edit Mobil	Menampilkan Halaman Edit Mobil	Menampilkan Halaman Edit Mobil	Berhasil
Memilih Menu Hapus Mobil	Menampilkan modal konfirmasi	Menampilkan modal konfirmasi	Berhasil

Gambar 4.23 Pengujian Black Box

Berdasarkan Pengujian *Black Box* yang telah dilakukan, dapat disimpulkan bahwa perancangan *website showroom* mobil ini berjalan sesuai kebutuhan dan sesuai dengan yang diharapkan.

4.1.5.2. Pengujian Pengguna

Pengujian dilakukan melalui kuisiomer kepada calon pelanggan dan pelaku *showroom* mobil guna membandingkan kelayakan sistem sebelum dan sesudah optimasi.

Responden	1	2	3	4	5	6	7	8	9	10	Total	Maks
R1	5	5	5	5	5	5	5	5	5	5	50	50
R2	3	3	3	2	3	3	3	3	2	3	27	50
R3	2	2	3	1	4	3	2	3	3	2	25	50
R4	3	3	4	3	4	3	3	4	4	4	35	50
R5	4	4	3	4	3	3	3	3	4	4	35	50
R6	3	4	3	2	3	3	3	3	3	3	30	50
R7	2	2	3	2	4	4	3	4	3	3	30	50
R8	2	2	3	3	2	3	2	3	3	2	25	50
R9	2	2	3	3	3	3	3	3	3	3	28	50
R10	5	4	5	3	3	4	5	4	3	4	40	50
R11	3	3	4	4	3	2	2	2	2	3	28	50
R12	3	3	4	3	2	3	3	3	4	3	31	50
R13	4	3	2	3	3	2	3	2	3	3	28	50
R14	3	4	3	3	4	3	3	3	4	3	32	50
R15	4	4	3	3	3	4	3	4	3	4	35	50
Jumlah											480	750

Gambar 4.24 Kuisiomer Pengguna Sebelum Optimasi

Responden	1	2	3	4	5	6	7	8	9	10	Total	Maks
R1	5	5	5	5	5	5	5	5	5	5	50	50
R2	4	5	4	5	4	4	5	5	5	4	45	50
R3	4	5	4	4	5	5	4	5	5	4	45	50
R4	5	5	4	5	5	4	4	4	5	4	45	50
R5	5	5	4	5	4	5	4	4	5	4	45	50
R6	4	4	5	5	5	4	5	4	4	5	45	50
R7	3	3	3	3	4	4	3	2	2	2	29	50
R8	4	3	4	5	5	5	5	5	4	5	45	50
R9	5	4	5	5	4	4	3	4	4	4	42	50
R10	4	5	5	5	5	4	4	5	5	4	46	50
R11	3	4	4	5	5	5	5	5	5	4	45	50
R12	4	4	4	5	5	5	5	5	5	5	47	50
R13	4	5	5	5	5	4	4	5	4	5	46	50
R14	5	5	5	4	3	4	3	4	4	5	42	50
R15	4	5	4	5	4	5	5	5	5	5	47	50
Jumlah											664	750

Gambar 4.25 Kuisiomer Pengguna Sesudah Optimasi

5. KESIMPULAN

Berdasarkan hasil penelitian dan pembahasan yang telah diuraikan sebelumnya, dapat disimpulkan bahwa:

- Website Showroom Mobil Berbasis Laravel menggunakan metode Agile dengan pendekatan Kanban berhasil dibuat.
- Hasil pengujian menunjukkan bahwa sistem berfungsi dengan baik. Melalui pengujian black box, seluruh fitur berjalan sesuai spesifikasi, dan dari hasil kuisiomer, mayoritas responden memberikan penilaian positif terhadap antarmuka, kecepatan akses, dan kemudahan navigasi, sehingga sistem dinilai layak untuk digunakan secara operasional.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada pihak-pihak terkait yang telah memberi dukungan terhadap penelitian ini.

DAFTAR PUSTAKA

- [1] S. Yason, Sudirman, dan A. Yunus, "Analisis performa website SCLEAN menggunakan Pingdom Tools dan Page Speed Insights," Kharisma Tech, vol. 17, no. 1, hlm. 113–124, Mei 2022, doi: <https://doi.org/10.55645/kharismatech.v17i1.213>.
- [2] Arlinta Christy Barus, Eka Stephani Sinambela, Ivani Purba, Jhonathan Simatupang, M. Marpaung, and N. Pandjaitan, "Performance Testing and Optimization of DiTenun Website," Journal of Applied Science, Engineering, Technology, and Education, vol. 4, no. 1, pp. 45–54, Jun. 2022, doi: <https://doi.org/10.35877/454ri.asci841>.
- [3] T. Hastono, "Optimasi Query Sistem Informasi Menggunakan Stored Procedure MySQL," Jurnal Dinamika Informatika, vol. 8, no. 2, pp. 79–89, Sep. 2019.

- [4] E. M. Laias, A. M. Hanashi, dan M. Alnass, "Compression study for image optimization techniques available for Joomla-based websites," *Paripex - Indian Journal of Research*, vol. 12, no. 4, pp. 1–2, 2023.
- [5] H. Park and Kim Cha Jong, "Design of Adaptive Web and Lazy Loading Components for Web Application Development," *The Journal of the Korean Institute of Information and Communication Engineering*, vol. 23, no. 5, pp. 516–522, Jan. 2019, doi: <https://doi.org/10.6109/jkiice.2019.23.5.516>.
- [6] I. Arthalita dan R. Prasetyo, "Penggunaan website sebagai sarana evaluasi kegiatan akademik siswa di SMA Negeri 1 Punggur Lampung Tengah," *JIKI (Jurnal Ilmu Komputer & Informatika)*, vol. 1, no. 2, hlm. 93–108, Jul. 2020, doi: <https://doi.org/10.24127/jiki.v1i2.678>.
- [7] B. Fachri, H. Hendry, and M. Zen, "Perancangan Sistem Informasi Posyandu Ibu Dan Anak Berbasis Web," *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 5, no. 1, pp. 49–54, Feb. 2023, doi: <https://doi.org/10.47233/jteksis.v5i1.737>.
- [8] I. P. Sari, A. Al-Khowarizmi, dan I. H. Batubara, "Implementasi aplikasi mobile learning sistem manajemen soal dan penilaian berbasis Android," *IHSAN: Jurnal Pengabdian Masyarakat*, vol. 3, no. 2, hlm. 178–183, 2021, doi: <https://doi.org/10.30596/ihsan.v3i2.7556>.
- [9] Muhammad Fardan, Dary Mochamad Rifqie, Rosidah, Akhmad Affandi, Sudarmanto Jayanegara, and M. Miftach Fakhri, "Peningkatan Kompetensi Back End Web Programming: Pelatihan Bahasa Pemrograman JavaScript bagi Mahasiswa," *Jurnal Sipakatau: Inovasi Pengabdian Masyarakat*, pp. 40–48, Apr. 2024, doi: <https://doi.org/10.61220/jsipakatau.v1i3.246>.
- [10] A. H. Nugraha, "Aplikasi penjualan hoodie & crewneck pada toko thrift Overmoon berbasis website menggunakan Bootstrap, PHP, dan MySQL," *Jurnal Ilmiah Multidisiplin*, vol. 1, no. 3, hlm. 113–117, Mei 2022, doi: <https://doi.org/10.56127/jukim.v1i03.237>.
- [11] I. Dusan, B. Sulaeman, dan R. Suppa, "Rancang Bangun Aplikasi Penjualan Berbasis Web (Toko Akbar)," *Jurnal Informatika dan Teknik Elektro Terapan (JITET)*, vol. 13, no. 1, pp. 1626–1636, Jan. 2025, doi: <https://doi.org/10.23960/jitet.v13i1.6022>.
- [12] G. R. U. Sinaga and S. Samsudin, "Implementasi Framework Laravel dalam Sistem Reservasi pada Restoran Cindelaras Kota Medan," *Jurnal Janitra Informatika dan Sistem Informasi*, vol. 1, no. 2, pp. 73–84, Oct. 2021, doi: <https://doi.org/10.25008/janitra.v1i2.131>.
- [13] Nazaruddin and S. Sarbaini, "Evaluasi Perubahan Minat Pemilihan Mobil dan Market Share Konsumen di Showroom Pabrikan Honda," *Jurnal Teknologi dan Manajemen Industri Terapan*, vol. 1, no. II, pp. 97–103, Jun. 2022, doi: <https://doi.org/10.55826/tmit.v1iII.27>.
- [14] K. Nistrina dan L. Sahidah, "Unified Modelling Language (UML) untuk perancangan sistem informasi penerimaan siswa baru di SMK Marga Insan Kamil," *J-SIKA*, vol. 4, no. 1, hlm. 17–23, Jun. 2022.
- [15] Ni, K. Oka, and I. Nyoman Piarsa, "Implementasi Black Box Testing pada Sistem Informasi Manajemen Dosen," *JITTER Jurnal Ilmiah Teknologi dan Komputer*, vol. 2, no. 3, pp. 535–535, Nov. 2021, doi: <https://doi.org/10.24843/jtrti.2021.v02.i03.p12>.