

Development and Functional Evaluation of a Web-Based Electronic Device Repair Service Information System Using React JS

Tatas Buana^{1*}, Yeremia Alfa Susetyo²

^{1,2} Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana, Salatiga, Indonesia

Email: ^{1*}672022271@student.uksw.edu, ²yeremia.alfa@uksw.edu

(*: corresponding author)

Abstract – Electronic device repair services still commonly use manual booking processes, such as visiting the service location directly or communicating through WhatsApp. Based on the observation results, the service process is still carried out manually, and system testing was simulated using 5 service types, 10 user accounts, and 20 dummy order data. This condition causes ineffective communication, customer queues, and limited information regarding repair status and cost estimates. Based on these problems, this study aims to implement React JS in the frontend development of an electronic device repair service application. The system development used the Waterfall method, which includes requirement, system design, implementation, testing, and maintenance. The application was developed using React JS with the support of React Router DOM, React Hooks, Fetch API, Tailwind CSS, and PostgreSQL as the database. The system produces 10 user features and 8 admin features. Functional testing using Black Box Testing on 15 scenarios showed that all system functions worked according to the specified requirements with a success rate of 100%, while User Acceptance Testing (UAT) obtained a user acceptance rate of 86.40%. The main contribution of this study is the implementation of React JS to build a component-based frontend interface that supports navigation management, Single Page Application (SPA), state management, and structured user interaction in the repair service system.

Keywords: Electronic Device Repair Service, React JS, Repair Service Booking System, Single Page Application, Waterfall.

Abstrak – Layanan perbaikan barang elektronik masih banyak menggunakan proses pemesanan secara manual, seperti datang langsung ke lokasi atau melalui WhatsApp. Berdasarkan hasil observasi, proses layanan masih dilakukan manual dan pengujian disimulasikan menggunakan 5 jenis layanan, 10 akun pengguna, dan 20 data pesanan dummy. Kondisi tersebut menyebabkan komunikasi kurang efektif, antrian pelanggan, serta keterbatasan informasi mengenai status perbaikan dan estimasi biaya. Berdasarkan permasalahan tersebut, peneliti bertujuan untuk menerapkan React JS dalam pengembangan frontend aplikasi layanan perbaikan barang elektronik. Pengembangan sistem menggunakan metode Waterfall yang meliputi *requirement, system design, implementation, testing, dan maintenance*. Aplikasi dibangun menggunakan React JS dengan dukungan React Router DOM, React Hooks, Fetch API, Tailwind CSS, dan PostgreSQL sebagai basis data. Sistem menghasilkan 10 fitur pengguna dan 8 fitur admin. Pengujian fungsional menggunakan metode *Black Box Testing* terhadap 15 skenario menunjukkan seluruh fungsi sistem berjalan sesuai dengan kebutuhan yang telah ditentukan dengan tingkat keberhasilan 100%, sedangkan *User Acceptance Testing (UAT)* memperoleh tingkat penerimaan pengguna sebesar 86,40%. Kontribusi utama penelitian ini adalah penerapan React JS untuk membangun antarmuka *frontend* berbasis komponen yang mendukung pengelolaan navigasi, *Single Page Application (SPA)*, pengelolaan state, dan interaksi pengguna secara terstruktur pada sistem layanan perbaikan.

Kata Kunci: Layanan Perbaikan Perangkat Elektronik, React JS, Sistem Pemesanan Layanan Perbaikan, Single Page Application (SPA), Waterfall.

1. PENDAHULUAN

Layanan *service* elektronik merupakan bentuk transaksi antara pelanggan yang membutuhkan perbaikan barang elektronik yang mengalami kerusakan dengan tenaga profesional. Jenis-jenis layanan servis elektronik yang umum dijumpai antara lain mencakup perbaikan perangkat seperti AC, kulkas, televisi, dan kipas angin. Bahkan, peran perangkat elektronik kini mulai mengalami pergeseran fungsi, dari sekadar alat bantu menjadi kebutuhan pokok bagi sebagian orang, khususnya dalam menunjang aktivitas dan kebutuhan sehari-hari [1]. Dalam kondisi tersebut, layanan pemesanan jasa secara online menjadi solusi yang penting untuk mempermudah dan mempercepat proses pelayanan kepada pelanggan [2].

Berdasarkan hasil observasi awal pada usaha layanan perbaikan barang elektronik, tercatat rata-rata 10 pelanggan per hari. Proses pemesanan masih dilakukan melalui WhatsApp atau dengan datang langsung ke lokasi. Selain itu, belum tersedia sistem penyimpanan data pesanan serta keharusan bagi pelanggan untuk terus menanyakan perkembangan perbaikan barang melalui panggilan telepon atau pesan singkat [3]. Oleh karena itu, diperlukan sistem penyedia jasa yang dapat mengelola data pelanggan dan jadwal perbaikan secara lebih terorganisir dan terintegrasi sehingga kualitas layanan yang diberikan menjadi lebih optimal [4].

Berdasarkan permasalahan tersebut, rumusan masalah dalam penelitian ini adalah: (1) bagaimana menerapkan React JS pada pengembangan frontend sistem informasi layanan perbaikan barang elektronik; dan (2) bagaimana hasil pengujian fungsional sistem yang dikembangkan menggunakan metode *Black Box Testing* dan *User Acceptance Testing*.

React JS merupakan salah satu *library JavaScript* yang paling banyak diminati dan digunakan dalam pembuatan antarmuka pengguna (*user interface*) [5]. Salah satu keunggulan react adalah pendekatan berbasis

komponen, di mana di setiap bagian tampilan seperti formulir pemesanan, daftar status *service*, dan informasi harga dibuat dalam bagian-bagian kecil sehingga mempermudah dalam proses pengembangan. Pendekatan berbasis komponen pada React JS sesuai untuk aplikasi layanan perbaikan elektronik karena setiap bagian antarmuka, seperti formulir pemesanan, status perbaikan, dan informasi harga, dapat dikembangkan secara terstruktur dan digunakan kembali [6]. Secara teknis, React JS memungkinkan aplikasi web dibangun sebagai *Single Page Application* (SPA), yaitu aplikasi yang berjalan pada satu halaman sehingga navigasi antar tampilan dapat dilakukan tanpa memuat ulang seluruh halaman [7]. Komunikasi antara *front-end* dan *back-end* pada aplikasi ini dilakukan melalui REST API, yaitu antarmuka pemrograman yang memungkinkan pertukaran data antar sistem menggunakan protokol HTTP secara terstruktur.

Pengembangan sistem pada penelitian ini menggunakan metode Waterfall karena memiliki tahapan yang sistematis, mulai dari *requirement*, *system design*, *implementation*, *testing*, hingga *maintenance* [8]. Metode ini dipilih agar proses pengembangan aplikasi layanan perbaikan barang elektronik dapat berjalan terarah dan terdokumentasi. Pengujian dilakukan menggunakan *Black Box Testing* untuk memastikan setiap fungsi berjalan sesuai kebutuhan berdasarkan input dan output, serta *User Acceptance Testing* (UAT) untuk mengetahui tingkat penerimaan pengguna terhadap aplikasi yang dikembangkan. Beberapa penelitian sebelumnya telah membahas aplikasi layanan perbaikan dan penerapan teknologi *front-end*, diantaranya penelitian yang mengembangkan website pemesanan jasa reparasi berbasis Laravel yang bertujuan untuk membantu pemilik bengkel dalam mengelola data pesanan, jenis barang yang dapat diperbaiki, serta menyajikan laporan secara digital [9]. Selain itu, sistem tersebut juga memudahkan pelanggan dalam melakukan pemesanan layanan reparasi secara *online* tanpa harus datang langsung ke lokasi.

Penelitian lainnya menyebutkan bahwa React JS mempermudah pengembangan kualitas aplikasi iCare dengan memanfaatkan *library Axios* dalam menangani permintaan HTTP menawarkan kelebihan tersendiri jika dibandingkan dengan penggunaan Fetch API [10]. Penelitian mengenai pengembangan sistem informasi layanan servis elektronik rumah tangga berbasis web menggunakan metode *Rapid Application Development* (RAD) memberikan hasil dengan sistem yang mendukung proses pencatatan layanan servis, pelacakan status perbaikan, serta penyimpanan data secara digital yang lebih aman, terstruktur, dan mudah diakses. Selain itu, penerapan metode RAD dinilai efektif karena melibatkan pengguna secara langsung dalam proses pengembangan sehingga sistem yang dihasilkan lebih sesuai dengan kebutuhan di lapangan [11]. Penelitian *platform* layanan jasa perbaikan kendaraan pada Nung Key Immobilizer menghasilkan aplikasi berbasis web yang mampu menjembatani komunikasi antara pelanggan, pemilik perusahaan, dan teknisi melalui proses reservasi jasa perbaikan kendaraan secara lebih efisien [12]. Penelitian lainnya mengenai aplikasi Serviso menghasilkan *platform mobile* untuk pencarian dan pemesanan jasa rumah tangga dengan fitur pencarian layanan, pemesanan, manajemen voucher, dan pelacakan status pesanan dalam satu platform [13]. Perbandingan penelitian terdapat pada Tabel 1.

Tabel 1. Perbandingan Penelitian

Penelitian	Metode	Teknologi	Fitur	Kontribusi
[7]	Pengembangan sistem berbasis web	Laravel	Pemesanan jasa reparasi, pengelolaan data pesanan, pengelolaan jenis barang, dan laporan digital	Membantu pemilik bengkel mengelola data layanan reparasi secara digital serta memudahkan pelanggan melakukan pemesanan tanpa datang langsung ke lokasi
[8]	Pengembangan aplikasi <i>front-end</i>	React JS dan Axios	Antarmuka aplikasi, pengelolaan komponen, dan permintaan data secara asynchronous	Menunjukkan bahwa React JS dapat digunakan untuk membangun antarmuka berbasis komponen, sedangkan Axios membantu komunikasi data antara frontend dan backend
[9]	<i>Rapid Application Development</i> (RAD)	Sistem informasi berbasis web	Pencatatan layanan servis, pelacakan status perbaikan, dan penyimpanan data digital	Menghasilkan sistem layanan servis elektronik rumah tangga yang lebih terstruktur, mudah diakses, dan sesuai kebutuhan pengguna melalui pendekatan iteratif
[10]	<i>Web Engineering</i>	Aplikasi berbasis web	Reservasi jasa perbaikan kendaraan dan komunikasi antara pelanggan, pemilik perusahaan, dan teknisi	Membantu menjembatani komunikasi layanan perbaikan kendaraan sehingga proses reservasi dan penyampaian data pelanggan kepada teknisi menjadi lebih efisien
[11]	Waterfall	Aplikasi mobile	Pencarian layanan, pemesanan jasa, manajemen voucher, dan pelacakan status pesanan	Menghasilkan platform layanan jasa rumah tangga yang meningkatkan efisiensi pencarian dan pemesanan layanan serta memperjelas informasi harga dan ketersediaan layanan

Berdasarkan Tabel 1, penelitian-penelitian sebelumnya telah membahas berbagai bentuk sistem layanan, baik berbasis web maupun mobile, dengan metode seperti RAD, *Web Engineering*, serta penerapan React JS pada domain aplikasi yang berbeda. Namun, belum banyak penelitian yang secara khusus membahas penerapan React JS sebagai frontend berbasis komponen pada sistem layanan perbaikan barang elektronik berbasis web. Penelitian sebelumnya juga belum secara menyeluruh mengintegrasikan fitur pemesanan layanan, pelacakan status perbaikan, pengelolaan pembayaran, pencetakan invoice, testimoni, dan dashboard admin dalam satu sistem. Oleh karena itu, penelitian ini dilakukan untuk mengisi kesenjangan tersebut melalui pengembangan frontend sistem informasi layanan perbaikan barang elektronik berbasis web menggunakan React JS.

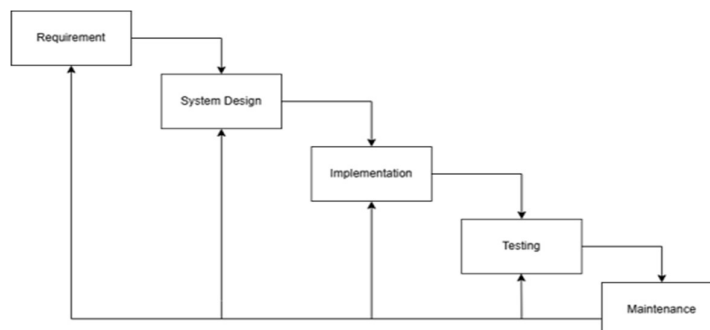
Kebaruan penelitian ini adalah penerapan React JS dengan pendekatan berbasis komponen pada aplikasi servis elektronik yang terintegrasi dengan fitur pelacakan status secara terstruktur. Penelitian ini mengimplementasikan React JS untuk membangun antarmuka pengguna yang mendukung proses pemesanan layanan, pelacakan status perbaikan, pengelolaan pembayaran, pencetakan invoice, serta dashboard administrasi dalam satu aplikasi berbasis web. Pendekatan tersebut memungkinkan setiap komponen antarmuka dikembangkan secara terstruktur dan dapat digunakan kembali, sehingga mempermudah pengelolaan serta pengembangan aplikasi.

Oleh karena itu, tujuan dari pelaksanaan riset ini adalah untuk memberikan kontribusi melalui penerapan dan pengembangan sistem informasi jasa layanan perbaikan barang elektronik menggunakan React JS guna mendukung proses layanan secara lebih terstruktur.

2. METODE PENELITIAN

2.1. Metode Pengembangan Sistem

Pada penelitian layanan perbaikan barang elektronik, pengembangan sistem dilakukan dengan menggunakan metode *Waterfall*. Metode *Waterfall* dipilih karena kebutuhan utama sistem dapat diidentifikasi dan Metode ini sesuai dengan rumusan masalah penelitian karena setiap tahap pengembangan dilakukan secara terstruktur dan berurutan. Metode *Waterfall* adalah sebuah cara pendekatan alur perangkat lunak yang terstruktur untuk menyelesaikan suatu permasalahan [8], dengan tahapan pada Gambar 1.



Gambar 1. Metode Waterfall [8]

Adapun tahapan dalam metode *Waterfall* adalah *requirement*, *system design*, *implementation*, *testing*, *maintenance*. Antara lain sebagai berikut.

a. Requirement

Pada tahap ini, dilakukan kebutuhan sistem melalui observasi dan wawancara dengan pemilik usaha jasa perbaikan barang elektronik, serta studi literatur terhadap penelitian sejenis. Observasi dilakukan untuk memahami alur layanan yang berjalan, mulai dari pemesanan, pencatatan pesanan, penyampaian status perbaikan, estimasi biaya, hingga konfirmasi pembayaran. Hasil observasi menunjukkan bahwa proses layanan masih dilakukan secara manual, baik melalui WhatsApp maupun dengan datang langsung ke toko. Kondisi tersebut menimbulkan beberapa kendala, seperti komunikasi yang kurang efektif, antrean pelanggan, pencatatan data yang belum terintegrasi, dan ketidakjelasan informasi status perbaikan.

Selain observasi, wawancara juga dilakukan dengan pemilik usaha untuk mengidentifikasi kebutuhan sistem, seperti fitur pemesanan *online*, unggah foto barang, pemantauan status perbaikan, pencatatan riwayat layanan, pengelolaan pembayaran, dan penyampaian estimasi biaya. Pemilik usaha memiliki peran sebagai *stakeholder* utama yang memberikan masukan serta memvalidasi kebutuhan fungsional sebelum sistem dirancang.

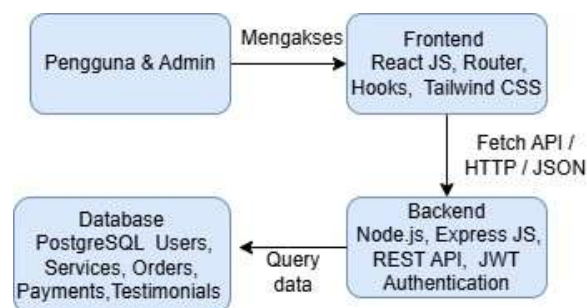
Berdasarkan permasalahan tersebut, dilakukan analisis kebutuhan untuk mengidentifikasi solusi yang diperlukan, yaitu pengembangan website atau aplikasi layanan perbaikan elektronik yang memungkinkan pelanggan melakukan pemesanan secara online dengan mengisi formulir dan mengunggah foto barang. Data yang digunakan untuk keperluan pengujian sistem merupakan data dummy, terdiri dari 5 jenis layanan perbaikan, 10

akun pengguna dummy, dan 20 data pesanan dummy yang dibuat untuk mensimulasikan kondisi penggunaan aplikasi secara nyata.

b. System Design

Pada tahap ini, dilakukan perancangan sistem digunakan *Use Case Diagram*, *Activity Diagram*, dan *Class Diagram* untuk menggambarkan interaksi pengguna serta alur kerja sistem. *Use Case Diagram* digunakan untuk menunjukkan hubungan antara aktor, yaitu pengguna dan admin, dengan fungsi-fungsi utama sistem. *Activity Diagram* digunakan untuk menggambarkan alur proses sistem dari awal hingga akhir secara terstruktur, mulai dari proses pemesanan hingga penyelesaian perbaikan. *Class Diagram* digunakan untuk menggambarkan struktur kelas dan hubungan antar objek dalam sistem secara keseluruhan.

Ketiga diagram ini digunakan sebagai acuan dalam pengembangan sistem agar sesuai dengan kebutuhan pengguna. Selain menggunakan diagram pemodelan, tahap perancangan juga mencakup penyusunan arsitektur sistem yang menunjukkan hubungan antara pengguna dan admin, *frontend*, *backend*, serta basis data. Pengguna dan admin mengakses sistem melalui antarmuka yang dikembangkan menggunakan React JS. *Frontend* kemudian berkomunikasi dengan *back-end express JS* melalui Fetch API menggunakan protokol HTTP dan format pertukaran data JSON. *Backend* menyediakan REST API untuk memproses permintaan dari *front-end* serta mengelola data yang tersimpan dalam basis data PostgreSQL.



Gambar 2. Arsitektur Sistem

Gambaran arsitektur sistem ditampilkan pada Gambar 2. Perancangan sistem ini juga mencakup komunikasi antara *frontend* dan *backend* melalui *Representational State Transfer Application Programming Interface* (REST API). Beberapa *endpoint* utama yang digunakan dalam sistem tersebut disajikan pada Tabel 1.

Tabel 2. Endpoint Utama Sistem

Endpoint	Method	Fungsi
/auth/register	POST	Mendaftarkan akun pengguna baru
/auth/login	POST	Melakukan login pengguna atau admin
/services	GET	Menampilkan daftar layanan perbaikan
/orders	POST	Membuat pesanan layanan perbaikan
/orders/user:id	GET	Menampilkan daftar pesanan pengguna
/orders/:id	GET	Menampilkan detail pesanan
/orders/:id/Status	PUT	Memperbarui status perbaikan
/payments	POST	Mengirim data pembayaran
/payments/:id/confirm	PUT	Mengonfirmasi pembayaran
/testimonials	POST	Menambahkan testimoni pengguna
/admin/orders	GET	Menampilkan seluruh data pesanan untuk admin
/admin/reports	GET	Menampilkan laporan layanan

c. Implementation

Setelah perancangan sistem, dilakukan tahapan implementasi menggunakan React JS. Pada sisi *frontend* dikembangkan menggunakan React JS dengan dukungan React Router DOM untuk mengatur navigasi, React Hooks untuk mengelola *state*, dan Tailwind CSS untuk membangun antarmuka pengguna yang responsif dan *backend* dikembangkan menggunakan Express JS yang menyediakan REST API sebagai penghubung antara *front-end* dan basis data PostgreSQL. Komunikasi antara *frontend* dan *backend* dilakukan menggunakan Fetch API melalui protokol HTTP. kode sumber aplikasi yang dikembangkan dapat diakses publik melalui URL dan dapat diakses melalui <https://github.com/Tatassz/Fixit>.

d. Testing

Setelah proses implementasi selesai, tahap berikutnya adalah *testing* atau pengujian sistem untuk memastikan bahwa seluruh fitur dapat berjalan dengan baik dan sesuai dengan kebutuhan pelanggan atau pengguna. Pengujian

dilakukan menggunakan dua metode, yaitu *Black Box Testing* dan *User Acceptance Testing* (UAT). *Black Box Testing* digunakan untuk menguji fungsionalitas sistem tanpa memperhatikan struktur kode program, sedangkan *User Acceptance Testing* (UAT) digunakan untuk mengevaluasi tingkat penerimaan pengguna terhadap aplikasi yang dikembangkan. Pengujian yang dilakukan pada seluruh fitur utama, meliputi, proses pemesanan, pelacakan status perbaikan, dan manajemen data oleh admin. Sistem dinyatakan berhasil apabila seluruh skenario pengujian menghasilkan *output* yang sesuai dengan hasil yang diharapkan.

e. Maintenance

Tahapan yang terakhir adalah *maintenance* atau pemeliharaan. Setelah dilakukan pengujian dalam aplikasi yang digunakan pelanggan, dilakukan *maintenance* agar dapat menjamin bahwa aplikasi dapat tetap berjalan secara stabil dan dapat terus berkembang sesuai kebutuhan pengguna.

3. HASIL DAN PEMBAHASAN

3.1. Analisis Kebutuhan Sistem

Tahap awal penelitian dilakukan melalui analisis kebutuhan untuk mengidentifikasi permasalahan serta menentukan kebutuhan sistem yang diperlukan dalam pengembangan aplikasi layanan perbaikan barang elektronik. Analisis ini dilakukan dengan mengumpulkan informasi mengenai kebutuhan yang belum terpenuhi melalui observasi dan wawancara dengan pemilik toko. Hasil observasi dan wawancara disajikan pada Tabel 3, sedangkan kebutuhan fungsional pengguna dan admin ditampilkan pada Tabel 4 dan Tabel 5.

Tabel 3. Hasil Observasi

No	Aspek yang di Observasi	Hasil Observasi
1.	Proses pemesanan	Pelanggan melakukan pemesanan dengan datang langsung ke toko atau menghubungi melalui WhatsApp.
2.	Pencatatan pesanan	Data pesanan masih dicatat secara manual sehingga berpotensi terjadi kesalahan pencatatan dan kesulitan dalam pencarian data.
3.	Penyampaian status perbaikan	Informasi perkembangan perbaikan disampaikan melalui whatsapp atau telepon setelah pelanggan menghubungi pihak toko.
4.	Estimasi biaya	Estimasi biaya diberikan setelah perangkat diperiksa dan disampaikan secara manual kepada pelanggan.
5.	Pengelolaan pembayaran	Konfirmasi pembayaran masih dilakukan secara manual dan belum terintegrasi dalam suatu sistem.
6.	Penyusunan laporan	Data transaksi dan laporan belum tersusun secara otomatis sehingga membutuhkan waktu dalam proses rekapitulasi.

Tabel 4. Kebutuhan Fungsional Pengguna

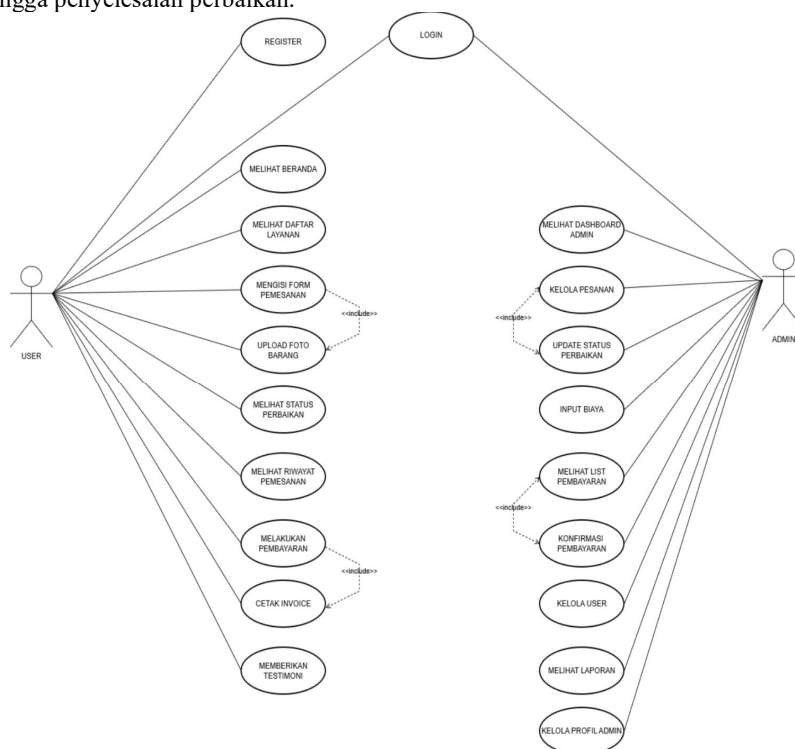
No	Fitur	Deskripsi
1.	Login	Sistem hanya dapat diakses setelah pengguna berhasil melakukan login
2.	Register	Pengguna dapat membuat akun untuk mengakses layanan
3.	Beranda	Sistem menyediakan halaman utama untuk melihat informasi layanan perbaikan
4.	Status Perbaikan	Pengguna ingin mengetahui status perbaikan barang perbaikan
5.	Layanan	Pengguna ingin melihat daftar layanan dan estimasi biaya perbaikan
6.	Formulir Pemesanan	Pengguna dapat melakukan pemesanan layanan perbaikan
7.	Upload Foto	Pengguna dapat mengunggah foto barang yang akan diperbaiki
8.	Riwayat Pemesanan	Pengguna dapat melihat riwayat layanan yang pernah dilakukan
9.	Testimoni	Pengguna dapat memberikan ulasan layanan setelah pembayaran
10.	Cetak Invoice	Pengguna dapat mencetak bukti pembayaran yang telah dilakukan

Tabel 5. Kebutuhan Fungsional Admin

No	Fitur	Deskripsi
1.	Dashboard Admin	Admin dapat ringkasan data sistem
2.	Kelola Pesanan	Admin dapat mengelola data pesanan pelanggan
3.	Kelola Pengguna	Admin dapat melihat data pengguna yang telah mendaftar dalam aplikasi
4.	List Pembayaran	Admin dapat melihat dan mengelola pembayaran
5.	Kelola Testimoni	Admin dapat mengelola Testimoni
6.	Laporan	Admin dapat melihat dan mengunduh laporan
7.	Pengaturan	Admin dapat mengelola data profil akun
8.	Konfirmasi Pembayaran	Admin dapat mengonfirmasi pembayaran pelanggan

3.2. Perancangan Sistem

Pada tahap perancangan sistem digunakan *Use Case Diagram*, *Class Diagram*, dan *Activity Diagram*. *Use Case Diagram* menggambarkan interaksi antara aktor (pengguna dan admin), *Class Diagram* menunjukkan struktur kelas beserta hubungan antarobjek, sedangkan *Activity Diagram* memodelkan alur proses sistem mulai dari pemesanan hingga penyelesaian perbaikan.



Gambar 3. Use Case Diagram

Use case diagram ini ditampilkan pada Gambar 3. Terdapat dua aktor utama dalam sistem ini, yaitu pengguna dan admin. *Pengguna* merupakan pelanggan yang mengakses aplikasi untuk melakukan pemesanan layanan perbaikan, memantau status perbaikan secara *real-time*, melakukan pembayaran, mencetak *invoice*, serta memberikan testimoni. Sedangkan admin bertugas mengelola pesanan yang masuk, memperbarui status perbaikan, menginput estimasi biaya, mengonfirmasi pembayaran, serta mengunduh laporan keuangan.

Alur proses sistem pengguna pada sistem ini digambarkan dalam bentuk *Activity Diagram* pada Gambar 4. Alur proses pemesanan pengguna aplikasi layanan perbaikan barang elektronik ini menggambarkan tahapan yang dilalui pengguna, dimulai dari membuka halaman utama sampai menyelesaikan transaksi. Proses diawali saat pengguna mengakses halaman utama. *Activity Diagram* ini dimulai dengan simbol "Start" sebagai tanda awal, dan diakhiri dengan simbol "Selesai" sebagai penutup alur proses. Pada halaman utama terdapat tampilan yang mudah digunakan untuk melihat bagaimana aplikasi tersebut memberikan layanan untuk mudah dipahami pengguna menggunakan aplikasi tersebut untuk memesan jasa layanan. Jika pengguna ingin melakukan pemesanan dan belum mempunyai akun, terdapat menu *login* dan *register*. Pada halaman *register*, pengguna dapat membuat akun terlebih dahulu, dan ketika sudah berhasil lalu langsung masuk ke halaman *login*.

Setelah berhasil masuk, terdapat halaman *dashboard*, pesanan, layanan, pembayaran, dan pengaturan. Jika Pengguna memilih keluar dari halaman, sistem akan menampilkan konfirmasi terlebih dahulu sebelum keluar dan akan mengakhiri alur prosesnya yaitu "Selesai". Bagi pengguna yang ingin melakukan pemesanan akan diarahkan ke halaman layanan untuk melihat apa saja layanan perbaikan yang tersedia. Lalu, pengguna diminta untuk melakukan pengisian data terlebih dahulu dalam formulir pemesanan yaitu seperti merek perangkat, model perangkat, deskripsi kerusakan, dan alamat pengguna.

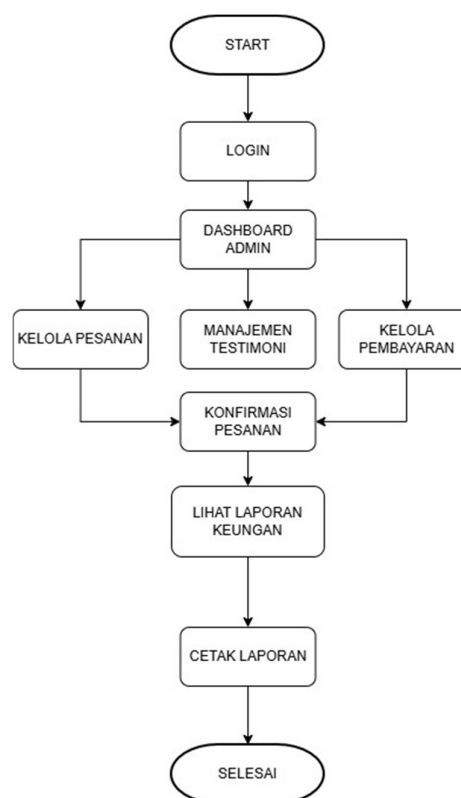
Ada proses pengisian form jika sudah selesai lalu akan diarahkan ke halaman pesanan yang di mana akan terdapat status pesanan barang pengguna. Jika status pesanan sudah selesai, pengguna akan diarahkan untuk melakukan pembayaran dan Setelah pembayaran dilakukan, pengguna tidak langsung mendapatkan konfirmasi secara otomatis, melainkan menunggu verifikasi dari pihak admin. Admin akan memeriksa bukti pembayaran yang masuk dan mengkonfirmasi status pembayaran tersebut di dalam sistem. Setelah pembayaran dikonfirmasi,

pengguna dapat mengunduh atau mencetak *invoice* sebagai bukti transaksi yang sah. Jika sudah selesai dan tidak ada aktivitas pemesanan dari pengguna, alur akan diakhiri dengan simbol "Selesai".

Alur proses admin digambarkan dalam *Activity Diagram* pada Gambar 5. Admin memulai alur dengan membuka halaman utama dan melakukan *login* menggunakan akun super admin. Setelah berhasil masuk, admin dapat mengakses *dashboard* yang menampilkan ringkasan data pendapatan, total pesanan, dan jumlah pengguna terdaftar. Selanjutnya, admin mengelola pesanan masuk melalui halaman kelola pesanan. Admin memperbarui status perbaikan sesuai kondisi barang, dari "Menunggu" menjadi "Proses" saat perbaikan berlangsung, dan "Selesai" ketika perbaikan telah usai. Setelah status selesai, sistem menginformasikan estimasi biaya kepada pengguna untuk kemudian dilanjutkan ke proses pembayaran. Konfirmasi pembayaran dilakukan admin melalui halaman *list* pembayaran. Selain mengelola pesanan, admin juga dapat mengelola testimoni yang diberikan pengguna. Setelah seluruh pesanan terselesaikan, admin dapat mengakses halaman laporan untuk melihat dan mencetak laporan keuangan. Alur diakhiri dengan simbol "Selesai".



Gambar 4. Activity Diagram Pengguna

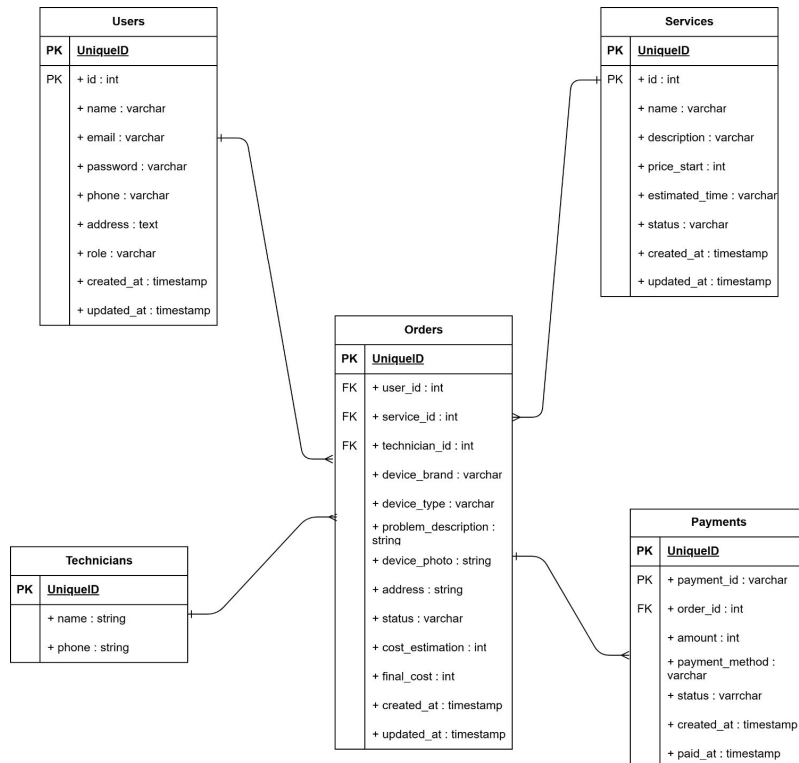


Gambar 5. Activity Diagram Admin

Dalam proses pengembangan aplikasi layanan jasa perbaikan barang elektronik, *database* berperan sebagai media penyimpanan utama yang mendukung seluruh aktivitas sistem. Data dikelola menggunakan *database* PostgreSQL bernama *layanan_elektronik*, yang terdiri dari lima entitas inti, yaitu *users*, *services*, *orders*, *payments*, dan *technicians*. Masing-masing entitas memiliki relasi yang saling terhubung untuk memastikan pengelolaan serta integrasi data berjalan dengan baik. Sebagai contoh, entitas *users* berelasi dengan *orders* melalui *user_id*, sementara *orders* berhubungan dengan *services* melalui *service_id*. Relasi tersebut menjadi dasar penting dalam proses pengolahan dan sinkronisasi data di dalam sistem. Gambaran hubungan antar entitas ini terdapat pada *Class Diagram* yang ditampilkan pada Gambar 6.

Pada hubungan antara *users* dan *orders*, setiap pengguna yang telah mendaftar melalui *endpoint* POST */auth/register* dapat membuat lebih dari satu pesanan perbaikan melalui *endpoint* POST */orders*, namun setiap pesanan hanya terkait dengan satu pengguna saja. Untuk relasi antara *services* dan *orders*, setiap jenis layanan perbaikan seperti Perbaikan TV, AC, Kulkas, Mesin Cuci, dan Kipas Angin dapat digunakan dalam banyak pesanan, tetapi setiap pesanan hanya mengacu pada satu jenis layanan tertentu. Sementara itu, pada relasi *orders* dengan *payments*, setiap pesanan yang telah memiliki status berhasil dapat memiliki data pembayaran yang dibuat

melalui *endpoint* POST /payments, di mana setiap data pembayaran tersebut selalu terhubung ke satu pesanan tertentu. Selain itu, terdapat pula relasi antara *technicians* dan *orders*, di mana satu teknisi dapat menangani lebih dari satu pesanan, tetapi setiap pesanan hanya ditangani oleh satu teknisi. Relasi ini dihubungkan melalui kolom *technician_id* pada entitas *orders*, yang merujuk ke entitas *technicians*. Dengan pemisahan ini, data teknisi seperti nama dan nomor telepon cukup disimpan satu kali pada entitas *technicians*, sehingga mengurangi duplikasi data dan memudahkan pembaruan apabila terjadi perubahan data teknisi.



Gambar 6. Class Diagram Sistem

3.3. Implementasi

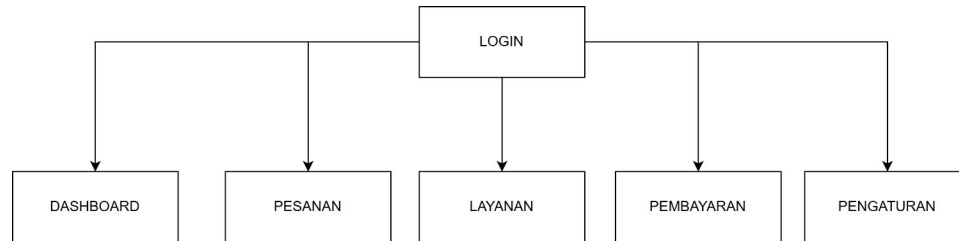
Pada aplikasi jasa layanan perbaikan barang elektronik dikembangkan menggunakan React JS dan PostgreSQL. React JS digunakan pada sisi *frontend* dan PostgreSQL sebagai *database* utama. Pada sisi *frontend*, beberapa teknologi pendukung digunakan sebagai berikut:

- React Router Dom digunakan untuk proses navigasi berlangsung meskipun terdapat banyak halaman atau komponen, proses perpindahan halaman tidak memerlukan *reload browser* dan dapat meningkatkan pengalaman pengguna karena konten dapat ditampilkan secara dinamis melalui pengelolaan URL yang terstruktur setara dengan aplikasi *Single Page Application* (SPA).
- Fetch API memungkinkan secara asynchronous mampu memisahkan proses pengambilan data di latar belakang dari proses rendering antarmuka. Dengan demikian, pengalaman pengguna (UX) tetap berjalan lancar dan responsif.
- Tailwind CSS memungkinkan pengembang menuliskan gaya secara lebih efisien tanpa perlu membuat aturan CSS dari awal yang mendukung pengembangan antarmuka pengguna yang modern dan responsif melalui penyediaan utilitas yang fleksibel dalam proses perancangan [14], khususnya pada pengembangan aplikasi jasa layanan perbaikan barang elektronik.
- React Hooks digunakan dalam manajemen *state* seperti “useState” dan “useEffect” [15].

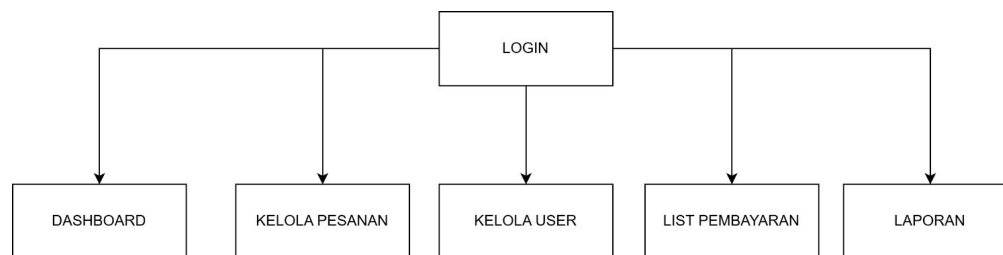
Pada sisi *backend*, Express JS menangani pemrosesan permintaan dari *frontend* melalui REST API, kemudian berkomunikasi dengan PostgreSQL untuk mengelola data, seperti *create*, *update*, dan *delete*. Sistem ini dirancang untuk melayani dua jenis pengguna, yaitu pengguna dan admin di mana masing-masing memiliki peran yang berbeda. Rancangan *database* sistem ini digambarkan dalam *Entity Relationship Diagram* pada Gambar 5. *Entity Relationship Diagram* menjelaskan hubungan antar entitas dalam sistem, mencakup tabel *users*, *services*, *orders*, *payments*, dan *technicians* yang saling berhubungan satu sama lain.

3.4. Perancangan Antarmuka

Sisi *frontend* aplikasi jasa layanan perbaikan barang elektronik dapat diakses oleh dua pengguna, yaitu *user* dan *admin*. Keduanya memiliki rancangan menu yang berbeda. Rancangan menu pengguna dapat terdapat pada Gambar 7, sedangkan rancangan menu admin terdapat pada Gambar 8.



Gambar 7. Rancangan Menu Pengguna Aplikasi Layanan Perbaikan

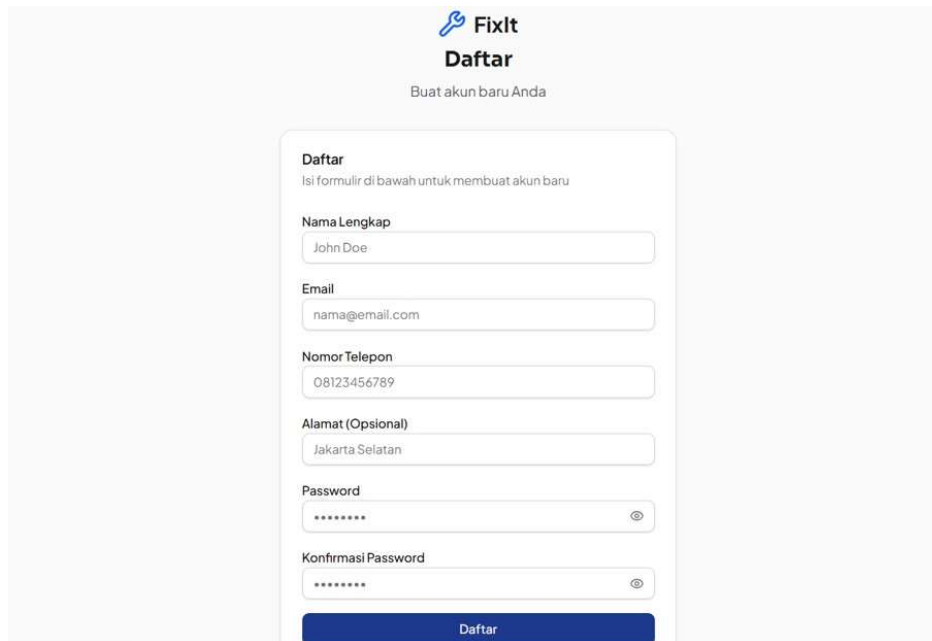


Gambar 8. Rancangan Menu Admin Aplikasi Layanan Perbaikan

3.5. Tampilan Layar

a. Tampilan Layar Pengguna

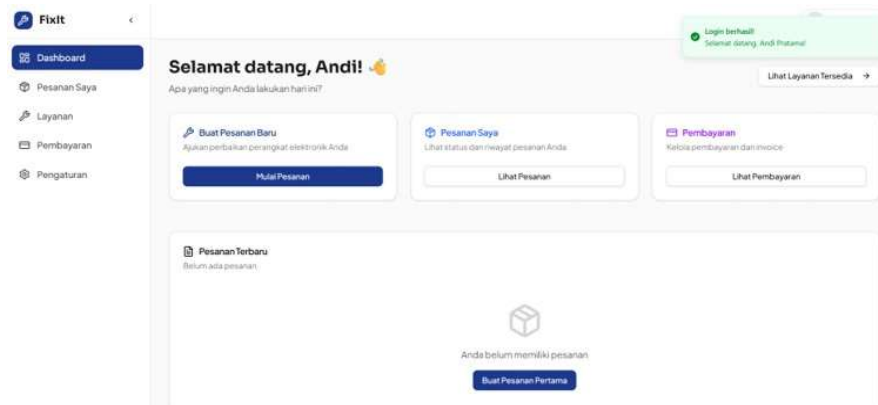
Pengguna yang belum memiliki akun diwajibkan melakukan registrasi sebelum dapat menggunakan layanan aplikasi. Pada proses registrasi pengguna mengisi nama lengkap, email, nomor telepon, alamat, *password*, dan konfirmasi *password*. Tampilan layar registrasi terdapat pada gambar 9.



The image shows a registration form titled "Daftar" (Register) for "FixIt". The form includes fields for "Nama Lengkap" (Full Name), "Email", "Nomor Telepon" (Phone Number), "Alamat (Opsional)" (Optional Address), "Password", and "Konfirmasi Password" (Confirm Password). A "Daftar" button is at the bottom. The form is set against a light gray background with a subtle grid pattern.

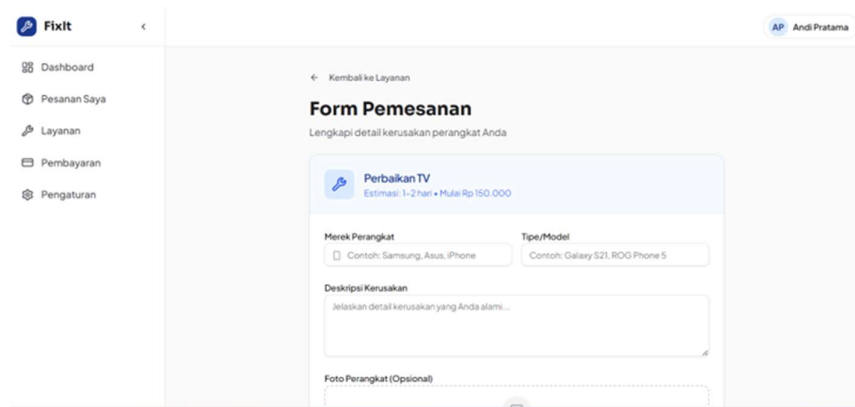
Gambar 9. Tampilan Layar Registrasi

Setelah registrasi, pengguna dapat melakukan *login* dan masuk ke halaman *dashboard* yang menampilkan akses cepat untuk membuat pesanan baru, melihat pesanan, dan mengelola pembayaran. Tampilan *dashboard* pengguna terdapat pada Gambar 10.



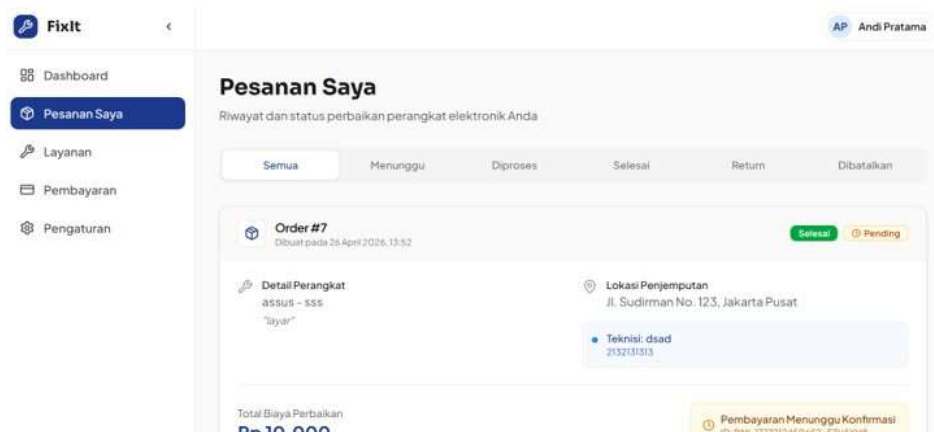
Gambar 10. Tampilan Layar Dashboard Pengguna

Pengguna dapat mengisi formulir pemesanan dengan memilih jenis layanan, mengisi deskripsi kerusakan, dan mengunggah foto kerusakan barang. Tampilan layar formulir pemesanan terdapat pada Gambar 11.



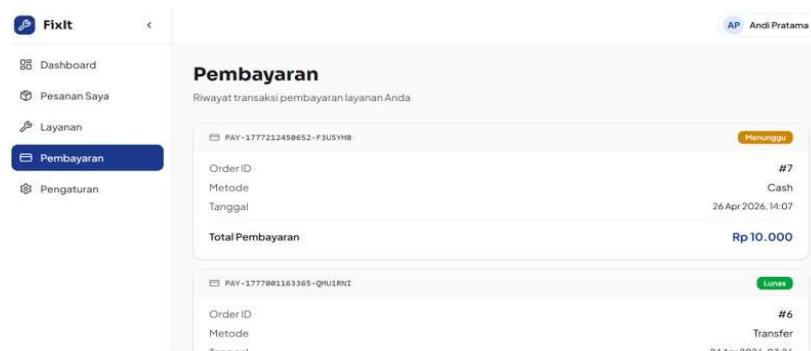
Gambar 11. Tampilan Layar Formulir Pemesanan

Setelah pengisian formulir pemesanan selesai dan pesanan dibuat, pengguna dapat memantau status perbaikan secara *real-time* melalui halaman pesanan. Tampilan layar halaman pesanan terdapat pada Gambar 12.



Gambar 12. Tampilan Layar Status Perbaikan

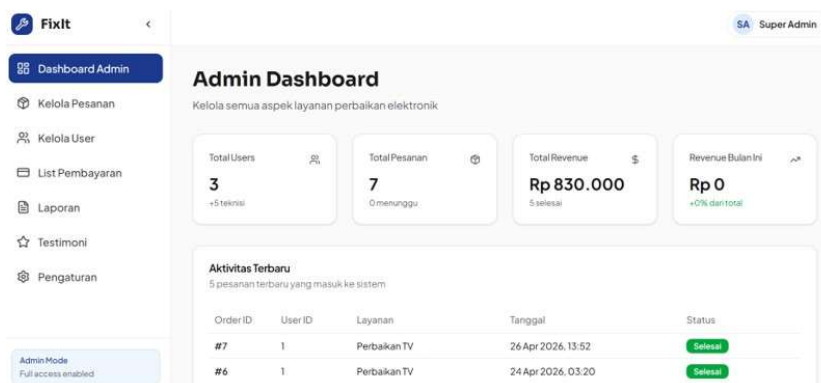
Setelah perbaikan selesai, maka status perbaikan akan berubah menjadi “selesai” dan pengguna dapat melakukan pembayaran dan mencetak *invoice* melalui halaman pembayaran. Tampilan layar halaman pembayaran terdapat pada Gambar 13.



Gambar 13. Tampilan Layar Pembayaran

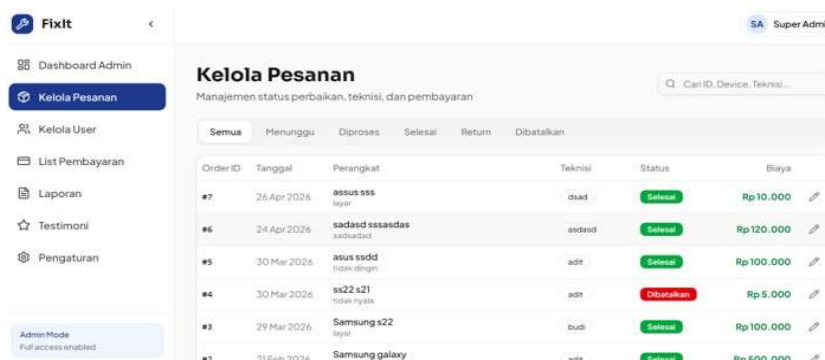
b. Tampilan Layar Admin

Admin dapat mengakses halaman *dashboard* yang berfungsi sebagai pusat informasi sistem. Pada halaman ini ditampilkan ringkasan data penting, seperti jumlah total pengguna, jumlah total pesanan, total pendapatan yang diperoleh, serta daftar aktivitas pesanan terbaru. Informasi tersebut membantu admin dalam memantau kondisi sistem dan aktivitas layanan secara lebih efektif. Tampilan halaman *dashboard* admin ditunjukkan pada Gambar 14.



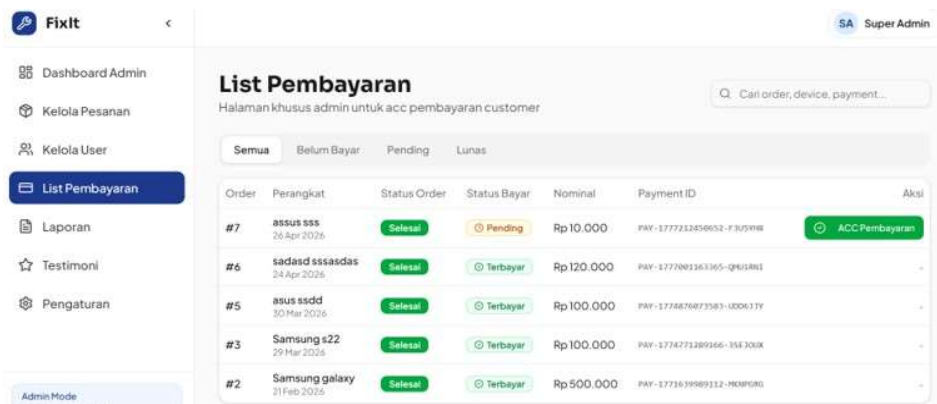
Gambar 14. Tampilan Layar Dashboard Admin

Tampilan halaman kelola pesanan admin berfungsi untuk mengelola seluruh data pesanan perbaikan yang masuk ke dalam sistem. Admin dapat mengonfirmasi, memperbarui status proses perbaikan, melihat detail pesanan, serta mengelola data pesanan sesuai kebutuhan. Fitur ini membantu admin dalam memantau perkembangan dalam setiap pesanan secara terstruktur sehingga proses pengelolaan menjadi lebih efektif. Tampilan halaman kelola pesanan admin ditunjukkan pada Gambar 15.



Gambar 15. Tampilan Layar Kelola Pesanan Admin

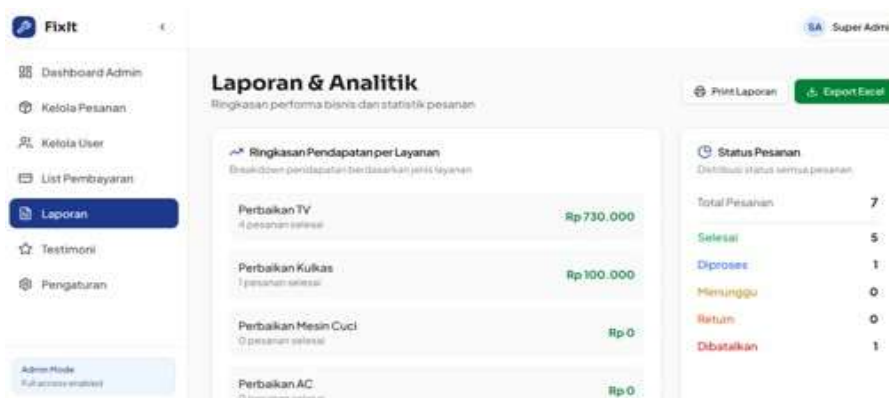
Halaman list pembayaran yang digunakan oleh admin untuk melakukan proses verifikasi dan konfirmasi pembayaran pelanggan. Halaman ini menampilkan informasi secara lengkap. Admin dapat memeriksa kesesuaian pembayaran yang telah dilakukan pelanggan, kemudian memberikan konfirmasi dengan validasi pembayaran. Melalui fitur ini, proses validasi pembayaran menjadi lebih terstruktur.



Order	Perangkat	Status Order	Status Bayar	Nominal	Payment ID	Aksi
#7	asus s55 26 Apr 2026	Selesai	Pending	Rp 10.000	PAY-177723250652-F3U5W8	ACC Pembayaran
#6	sadasd sssasdas 24 Apr 2026	Selesai	Terbayar	Rp 120.000	PAY-1777001363365-QNUSM1	-
#5	asus ssdd 30 Mar 2026	Selesai	Terbayar	Rp 100.000	PAY-1778876873583-UDK3TY	-
#3	Samsung s22 29 Mar 2026	Selesai	Terbayar	Rp 100.000	PAY-1774271289366-35430K	-
#2	Samsung galaxy 21 Feb 2026	Selesai	Terbayar	Rp 500.000	PAY-1773639989312-NCMPUR0	-

Gambar 16. Tampilan Layar List Pembayaran Admin

Admin dapat mengakses halaman laporan yang berfungsi untuk melihat ringkasan pendapatan dari seluruh layanan perbaikan yang telah diselesaikan. Pada halaman ini ditampilkan informasi mengenai total pendapatan, jumlah transaksi, serta data transaksi. Selain itu, halaman ini juga menyediakan fitur untuk mengekspor laporan ke dalam format digital dan mencetak laporan secara langsung. Fitur tersebut membuat penyusunan laporan keuangan menjadi lebih efektif, akurat, dan efisien.



Perbaikan	Jumlah Pesanan Selesai	Total Pendapatan
Perbaikan TV	4 pesanan selesai	Rp 730.000
Perbaikan Kulkas	1 pesanan selesai	Rp 100.000
Perbaikan Mesin Cuci	0 pesanan selesai	Rp 0
Perbaikan AC	0 pesanan selesai	Rp 0

Gambar 17. Tampilan Layar Laporan Admin

3.6. Pengujian Sistem

Setelah implementasi berhasil, dilakukan pengujian yang digunakan pada aplikasi jasa layanan perbaikan barang elektronik dengan menggunakan metode *Blackbox* dan *User Acceptance Testing*. Metode *Blackbox* berfokus pada pengujian fungsionalitas sistem yang telah dirancang, dengan tujuan memastikan bahwa sistem dapat berjalan sesuai dengan kebutuhan dan harapan pengguna yang di mana ini sejalan dengan tujuan dirancangnya aplikasi jasa layanan perbaikan barang elektronik [16]. *Black box testing* merupakan metode pengujian perangkat lunak yang berfokus pada pengujian fungsionalitas aplikasi tanpa memperhatikan struktur internal atau cara kerja sistem di dalamnya [17]. Pengujian dilakukan menggunakan 15 skenario, dan seluruh skenario berhasil menghasilkan output sesuai harapan. Berdasarkan hasil tersebut, tingkat keberhasilan pengujian mencapai 100%. Pengujian *User Acceptance Testing* (UAT) dilakukan dengan melibatkan pengguna akhir untuk mengevaluasi kesesuaian sistem terhadap kebutuhan pengguna serta memastikan bahwa seluruh fungsi yang dikembangkan telah memenuhi persyaratan yang telah ditentukan [18]. Tabel 6, Tabel 7, dan Tabel 8 menunjukkan hasil survei kepada pengguna dan tes pengujian pada halaman pelanggan dan admin.

Tabel 6. Hasil Pengujian Halaman Pelanggan

No	Nama Pengujian	Prosedur Pengujian	Hasil yang Diharapkan	Status
1.	Register akun pengguna pesanan	Isi nama, email, nomor telepon, alamat, password, dan konfirmasi password	Sistem menyimpan data pengguna dan masuk ke halaman login pengguna	Berhasil
2.	Login pengguna	Memasukkan email dan password yang telah terdaftar, lalu menekan tombol login	Sistem memvalidasi dan Masuk ke halaman utama dengan tampilan awal	Berhasil
3.	Melihat daftar layanan Perbaikan	Memilih daftar layanan perbaikan yang akan dipesan	Sistem memperlihatkan daftar layanan perbaikan yang tersedia	Berhasil
4.	Memantau status perbaikan	Melihat status perbaikan barang yang dipesan	Sistem memperlihatkan status perbaikan barang elektronik	Berhasil
5.	Memesan layanan perbaikan	Memilih layanan, mengisi formulir pemesanan, lalu menekan tombol "Buat Pesanan"	Sistem menampilkan formulir pemesanan dan menyimpan data pemesanan	Berhasil
6.	Melakukan pembayaran	Menekan tombol "Bayar Sekarang", memilih metode pembayaran dan mengkonfirmasi pembayaran	Sistem berhasil memproses pembayaran, menyimpan data, dan menampilkan status menunggu konfirmasi admin	Berhasil
7.	Edit profil pengguna	Isi nama, email, nomor telepon, dan alamat untuk merubah data pengguna	Sistem berhasil memperbarui data profil pengguna dan informasi terbaru	Berhasil

Hasil pengujian menunjukkan bahwa seluruh fungsi pada halaman pelanggan berjalan sesuai dengan harapan. Seluruh proses, mulai dari registrasi, pemesanan layanan perbaikan, pembayaran, melihat daftar layanan, dan edit profil pengguna, berhasil dilakukan tanpa ada kendala selama proses pengujian.

Tabel 7. Hasil Pengujian Halaman Admin

No	Nama Pengujian	Prosedur Pengujian	Hasil yang Diharapkan	Status
1.	Login admin	Isi email dan password	Sistem masuk ke halaman dashboard admin	Berhasil
2.	Melihat dashboard	Membuka halaman dashboard	Sistem menampilkan ringkasan jumlah pengguna, jumlah pesanan, total pendapatan, dan aktivitas pesanan terbaru	Berhasil
3.	Kelola pesanan	Membuka halaman kelola pesanan dan mengubah status pesanan	Sistem berhasil memperbarui status pesanan sesuai perubahan yang dilakukan	Berhasil
4.	Kelola pengguna	Membuka halaman kelola pengguna	Sistem menampilkan daftar seluruh data pengguna yang terdaftar	Berhasil
5.	Konfirmasi pembayaran	Membuka halaman List Pembayaran, memeriksa bukti pembayaran, lalu menekan tombol konfirmasi	Sistem berhasil memperbarui status pembayaran menjadi berhasil dikonfirmasi	Berhasil
6.	Melihat laporan	Membuka halaman laporan	Sistem menampilkan ringkasan pendapatan dan data transaksi layanan	Berhasil
7.	Mengekspor laporan	Menekan tombol "Export", atau "Print" pada halaman laporan	Sistem berhasil mengekspor atau mencetak laporan sesuai data yang ditampilkan	Berhasil
8.	Edit Profil Admin	Mengubah nama, email, nomor telepon, dan alamat, lalu menyimpan perubahan	Sistem berhasil memperbarui data profil admin dan menampilkan informasi terbaru	Berhasil

Berdasarkan hasil pengujian pada Tabel 7, menunjukkan bahwa seluruh fungsi pada halaman admin berjalan dengan baik dan sesuai dengan rancangan sistem. Pengujian mencakup proses login admin, pengelolaan data pengguna, konfirmasi pembayaran pelanggan, penyajian laporan pendapatan, serta fitur ekspor dan cetak laporan berjalan dengan hasil yang diharapkan tanpa ditemukan kendala selama proses pengujian.

Tabel 8. Hasil User Acceptance Testing

No	Pernyataan	Presentase
1.	Tampilan aplikasi mudah dipahami.	87,69%
2.	Navigasi antar halaman sudah digunakan.	84,62%
3.	Menu-menu pada aplikasi mudah ditemukan	80,00%
4.	Proses registrasi dan login berjalan dengan mudah.	80,00%
5.	Fitur pemesanan layanan berjalan dengan baik.	84,62%
6.	Monitoring status perbaikan mudah dipahami.	87,69%
7.	Informasi estimasi biaya mudah dimengerti.	89,23%
8.	Proses konfirmasi pembayaran berjalan dengan baik.	86,15%

No	Pernyataan	Presentase
9.	Invoice berhasil ditampilkan dengan jelas.	92,31%
10.	Riwayat pesanan mudah diakses.	87,69%
11.	Tampilan aplikasi menarik.	84,62%
12.	Warna dan desain aplikasi nyaman dilihat.	84,62%
13.	Tata letak informasi sudah rapi.	86,15%
14.	Aplikasi berjalan dengan cepat.	84,62%
15.	Perpindahan halaman terasa lancar tanpa kendala.	89,23%
16.	Aplikasi bekerja sesuai harapan tanpa mengalami error.	89,23%
17.	Aplikasi membantu proses pemesanan layanan perbaikan.	90,77%
18.	Aplikasi memudahkan pemantauan status perbaikan.	87,69%
19.	Secara keseluruhan saya puas menggunakan aplikasi ini.	84,62%

Hasil pengujian *User Acceptance Testing* (UAT) dilakukan terhadap 13 responden dengan menggunakan 19 pernyataan dan skala Likert 1–5 dan menunjukkan bahwa skor aktual yang diperoleh adalah 1.067 dari total skor ideal sebesar 1.235. Persentase tingkat penerimaan pengguna dihitung menggunakan Persamaan (1).

$$\%(UAT) = \frac{\text{Skor Aktual}}{\text{Skor Ideal}} \times 100\% \quad (1)$$

Selanjutnya, nilai pada Persamaan (1) disubstitusikan dengan hasil pengujian sehingga diperoleh:

$$\%(UAT) = \frac{1067}{1235} \times 100\% \quad (2)$$

Hasil perhitungan tersebut menghasilkan:

$$\%(UAT) = 86,40\% \quad (3)$$

Berdasarkan hasil perhitungan pada Persamaan (3), diperoleh tingkat penerimaan pengguna sebesar 86,40%. Nilai tersebut termasuk dalam kategori sangat baik. Nilai tertinggi terdapat pada fitur cetak invoice sebesar 92,31%, yang menunjukkan bahwa fitur tersebut dinilai dapat digunakan dengan baik oleh responden. Sementara itu, nilai terendah sebesar 80,00% terdapat pada aspek kemudahan menemukan menu serta proses registrasi dan login. Hasil tersebut menunjukkan bahwa meskipun aplikasi telah diterima dengan baik, susunan navigasi dan tampilan formulir autentikasi masih dapat disederhanakan agar lebih mudah digunakan.

3.7. Diskusi

Hasil *Black Box Testing* menunjukkan bahwa seluruh 15 skenario pengujian berhasil menghasilkan keluaran yang sesuai dengan hasil yang diharapkan, sehingga diperoleh tingkat keberhasilan fungsional sebesar 100%. Hasil tersebut menunjukkan bahwa fungsi utama pada sisi pengguna dan admin telah berjalan sesuai dengan kebutuhan yang ditentukan. Fungsi tersebut meliputi registrasi, login, pemesanan layanan, pemantauan status perbaikan, pembayaran, pengelolaan pesanan, konfirmasi pembayaran, serta penyajian laporan. Meskipun pengujian *black box* berhasil, pengujian performa dan keamanan belum dilakukan.

React JS sesuai digunakan dalam pengembangan aplikasi ini karena mendukung pendekatan berbasis komponen. Komponen seperti navigasi, formulir pemesanan, daftar layanan, status perbaikan, dan tabel pesanan dapat dikembangkan secara terpisah serta digunakan kembali pada beberapa halaman. Pendekatan tersebut membuat struktur *front-end* lebih terorganisasi dan memudahkan proses pengembangan. React Hooks digunakan untuk mengelola *state* dan proses pengambilan data, sedangkan React Router DOM mendukung perpindahan tampilan tanpa memuat ulang seluruh halaman. Komunikasi antara frontend dan backend dilakukan melalui REST API sehingga data pesanan, pembayaran, dan status perbaikan dapat ditampilkan secara dinamis.

Meskipun seluruh skenario berhasil pada pengujian akhir, beberapa penyesuaian tetap dilakukan selama proses pengembangan. Pada sisi *front-end* memiliki tantangan utama terdapat pada sinkronisasi data antara React JS dan REST API, terutama agar perubahan status pesanan oleh admin dapat langsung ditampilkan dengan benar pada halaman pengguna. Selain itu, proses pengambilan data juga perlu dilengkapi dengan penanganan *loading* dan *error*. Pada sisi *back-end*, penyesuaian dilakukan pada validasi data, pembagian hak akses pengguna dan admin, serta pengelolaan data pesanan dan pembayaran. Setelah dilakukan perbaikan pada *endpoint* dan proses validasi, seluruh fungsi dapat berjalan sesuai dengan hasil yang diharapkan.

Tabel 9. Perbandingan Sistem Manual dan Sistem Berbasis Web

No.	Aspek	Sistem Manual	Sistem Berbasis Web
1.	Pemesanan layanan	Pelanggan datang langsung atau menghubungi melalui WhatsApp	Pelanggan mengisi formulir pemesanan melalui aplikasi
2.	Pencatatan	Dicatat secara manual dan berisiko sulit	Tersimpan dalam basis data dan dapat

No.	Aspek	Sistem Manual	Sistem Berbasis Web
3.	pesanan	ditemukan	ditampilkan kembali
3.	Informasi Status	Pelanggan menghubungi toko melalui WhatsApp atau telepon	Pelanggan melihat status melalui halaman pesanan
4.	Estimasi biaya	Disampaikan secara manual kepada pelanggan	Dicatat oleh admin dan ditampilkan melalui sistem
5.	Pembayaran	Konfirmasi dilakukan melalui komunikasi langsung	Bukti pembayaran dikirim dan diverifikasi melalui sistem
6.	Penyusunan laporan	Direkap secara manual	Data transaksi ditampilkan, diekspor, dan dicetak melalui halaman laporan
7.	Kemudahan pengguna	Proses bergantung pada komunikasi dengan pihak toko	Hasil UAT memperoleh tingkat penerimaan 86,40%
8.	Kepuasan pengguna	Belum diukur pada sistem manual	Diukur melalui UAT dengan hasil 86,40%

Berdasarkan hasil observasi, proses layanan sebelum sistem dikembangkan masih dilakukan secara manual. Pelanggan harus pesan dengan datang langsung ke toko atau menghubungi pihak toko melalui WhatsApp. Data pesanan dicatat secara manual, sedangkan informasi perkembangan perbaikan dan estimasi biaya disampaikan melalui WhatsApp atau telepon. Proses tersebut menyebabkan pelanggan bergantung pada respons pihak toko untuk memperoleh informasi mengenai pesannya. Setelah sistem aplikasi berbasis web dikembangkan proses pemesanan, pemantauan status perbaikan, pembayaran, dan pengelolaan data dapat dilakukan melalui satu sistem yang terintegrasi dan terstruktur serta data pelanggan dan pesanan juga tersimpan dan lebih mudah dikelola oleh admin.

Hasil UAT sebesar 86,40% menunjukkan bahwa aplikasi diterima dengan sangat baik oleh responden. Sistem ini dinilai mampu mempermudah proses layanan dan memberikan akses informasi yang lebih praktis dibandingkan proses manual sebelumnya. Hasil penelitian ini sejalan dengan penelitian [8] yang menunjukkan bahwa React JS dapat digunakan untuk membangun antarmuka web yang dinamis dan berbasis komponen. Sementara itu, penelitian [7] menunjukkan bahwa sistem pemesanan jasa reparasi berbasis web dapat membantu proses pengelolaan pesanan dan penyusunan laporan secara digital.

Perbedaan penelitian ini terletak pada integrasi fitur yang lebih lengkap, meliputi pemesanan layanan, pemantauan status perbaikan, pembayaran, pencetakan *invoice*, testimoni, dan pengelolaan laporan admin dalam satu sistem. Kontribusi penelitian ini adalah menghasilkan rancangan *front-end* berbasis komponen yang dapat digunakan kembali serta mengintegrasikan React JS dengan REST API Express JS dan basis data PostgreSQL pada sistem informasi layanan perbaikan barang elektronik yang dapat digunakan kembali pada beberapa halaman sehingga proses pengembangan menjadi lebih terstruktur. React Router DOM juga mendukung perpindahan halaman tanpa memuat ulang seluruh halaman, sesuai dengan konsep *Single Page Application* yang dijelaskan pada penelitian [13]. React Hooks digunakan untuk mengelola state dan proses pemanggilan data, sedangkan Tailwind CSS membantu penyusunan antarmuka yang responsif melalui kelas utilitas sebagaimana dijelaskan pada [14] dan [15].

Penelitian ini masih memiliki beberapa keterbatasan. Pengujian fungsional hanya dilakukan melalui 15 skenario *Black Box Testing*, sedangkan UAT melibatkan 13 responden. Oleh karena itu, hasil penerimaan pengguna belum dapat menggambarkan seluruh calon pengguna aplikasi layanan perbaikan barang elektronik. Penelitian ini juga belum mencakup *usability testing* dengan instrumen khusus, pengujian keamanan, pengujian performa, *load testing*, dan evaluasi skalabilitas sistem. Dengan demikian, hasil penelitian baru menunjukkan bahwa fungsi yang diuji telah berjalan sesuai kebutuhan dan aplikasi diterima dengan baik pada skala pengujian yang dilakukan. Hasil tersebut belum dapat digunakan untuk menilai keamanan, performa, dan keandalan sistem apabila diterapkan pada penggunaan yang lebih luas.

4. KESIMPULAN

Penelitian ini menunjukkan bahwa penggunaan React JS pada pengembangan aplikasi jasa layanan perbaikan barang elektronik dapat menghasilkan sistem berbasis web yang lebih terstruktur dan terintegrasi. Sistem yang dibangun dapat membantu pengguna dan admin dalam mengelola proses layanan perbaikan, sehingga sistem yang sebelumnya dilakukan secara manual dapat dilaksanakan dengan lebih mudah dan efisien. Kontribusi utama penelitian ini adalah penerapan arsitektur *frontend* berbasis komponen yang dapat digunakan kembali serta terintegrasi dengan REST API Express JS dan basis data PostgreSQL. Penerapan tersebut dapat menjadi salah satu contoh penggunaan React JS dalam membangun aplikasi layanan berbasis web yang lebih mudah dikembangkan, dikelola, dan dipelihara.

Meskipun demikian, penelitian ini masih memiliki beberapa keterbatasan. jumlah responden yang terlibat masih relatif sedikit dan pengujian yang dilakukan belum mencakup evaluasi keamanan, performa, *usability* secara

khusus, *load testing*, serta skalabilitas sistem. Oleh karena itu, hasil penelitian ini belum dapat menggambarkan tingkat keandalan sistem secara menyeluruh apabila diterapkan dalam skala penggunaan yang lebih besar. Penelitian selanjutnya disarankan untuk melibatkan lebih banyak responden, melakukan pengujian keamanan dan performa secara mendalam, serta mengevaluasi kemampuan sistem ketika diakses oleh banyak pengguna secara bersamaan. Pengembangan berikutnya juga dapat mencakup notifikasi *real-time*, integrasi *payment gateway*, dan pengembangan aplikasi *mobile* agar sistem semakin lengkap dan siap diterapkan dalam lingkungan operasional yang lebih luas.

DAFTAR PUSTAKA

- [1] H. H. Anggara, M. Y. H. Utomo, M. B. B. Sutigar, and A. S. Fitri, "Perancangan aplikasi layanan jasa reparasi alat elektronik dengan metode Object Oriented Analysis Design (OOAD) berbasis web," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, no. 3, pp. 3118–3123, 2024, doi: 10.36040/jati.v8i3.9590.
- [2] A. F. Hudzaifah, M. F. Latief, "Perancangan dan Implementasi Sistem Reservasi Klinik Berbasis Web dengan Framework React JS: Studi Kasus Klinik Pratama Agha Depok," *DBESTI: Journal of Digital Business and Technology Innovation*, vol. 3, no. 1, pp. 134–141, 2026, doi: 10.54914/dbesti.v3i1.2246.
- [3] M. Abdilah, V. Yasin, A. B. Yulianto, "Rancang Bangun Aplikasi Manajemen Sistem Pelayanan Penyediaan Jasa Berbasis Online," *Jurnal Manajemen Informatika Jayakarta*, vol. 2, no. 1, pp. 103–114, 2022 doi: 10.52362/jmijayakarta.v2i1.722
- [4] R. Ferdinan and M. Fachrie, "Pengembangan Aplikasi Jasa Service Peralatan Elektronik Berbasis Android Menggunakan Payment Gateways," *Jurnal Indonesia : Manajemen Informatika dan Komunikasi*, vol. 5, no. 1, pp. 954–963, 2024, doi: 10.35870/JIMIK.V5I1.563.
- [5] M. D. Dermawan, H. Suroyo, A. Wijaya, and Jemakmun, "Pengembangan Web Admin Aplikasi Pulsa Dengan React JS Menggunakan Metode Rapid Application Development (RAD)," *Jurnal Teknologi dan Sistem Informasi Bisnis*, vol. 6, no. 2, pp. 300–309, 2024, doi: 10.47233/jteksis.v6i2.1337.
- [6] M. Effendy and A. Habib, "Rancang Bangun Sistem Informasi Pemesanan Jasa Perbaikan Elektronik Berbasis Website Dengan Penerapan Metode Eksponent Smoothing Pada CV OTW Computer Gusaha Surabaya," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 9, no. 5, pp. 8263–8270, Jul. 2025, doi: 10.36040/JATI.V9I5.14937.
- [7] L. Apriyanti, M. Ismu Rahayu, A. Winarni, and F. Riki Setiadi, "PMB Pembaruan Aplikasi Pendaftaran Mahasiswa Baru Stmik Bandung Menggunakan Siklus SPA (Single Page Application)," *Jurnal Bangkit Indonesia*, vol. 14, no. 2, pp. 73–78, 2025, doi: 10.52771/BANGKITINDONESIA.V14I2.460.
- [8] N. Raptama and Riki, "Analysis and Design of Information Systems for Make-Up Using the Waterfall Methodology," *bit-Tech*, vol. 5, no. 2, pp. 54–66, 2022, doi: 10.32877/BT.V5I2.553.
- [9] A. Maulana, I. Purnamasari, I. Maulana, U. S. Karawang, J. H. Ronggo Waluyo, and T. Timur, "Rancang Bangun Website Layanan Jasa Reparasi Alat Elektronik Rumah Tangga Menggunakan Framework Laravel (Studi Kasus: Cv. Xyz)," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 3, pp. 2830–7062, 2024, doi: 10.23960/jitet.v12i3.4859.
- [10] S. Rahmadhani, M. Dyvani, W. Wildana, H. W. Arumdanie, and L. Hakim, "Penerapan React JS dan Axios untuk Pengembangan Front-end Aplikasi iCare," *SESSION (Software Development, Digital Business Intelligence, and Computer Engineering)*, vol. 2, no. 2, pp. 40–46, 2024, doi: 10.57203/session.v2i02.2024.40-46.
- [11] R. Febriansyah, "Perancangan Sistem Informasi Layanan Servis Elektronik Rumah Tangga Berbasis Web," *HUMANITIS: Jurnal Humaniora, Sosial dan Bisnis*, vol. 2, no. 11, pp. 21–32, 2025. [Online]. Available: <https://ecosemica.net/index.php/HUMANITIS/article/view/735>. Accessed: Jul. 10, 2026.
- [12] Y. Nurfauzi, P. Sokibi, and M. Hatta, "Rancang Bangun Aplikasi Layanan Jasa Teknisi Kendaraan Menggunakan Metode Web Engineer," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, no. 6, pp. 12655–12663, 2024, doi: 10.36040/JATI.V8I6.11954.
- [13] R. D. Ritonga, I. D. Lesmana, and M. A. Kesuma, "Perancangan Sistem Informasi Layanan Jasa Rumah Tangga Berbasis Mobile Menggunakan Metode Waterfall," *REMIK: Riset dan E-Jurnal Manajemen Informatika Komputer*, vol. 10, no. 1, pp. 249–260, 2026, doi: 10.33395/REMIK.V10I1.15754.
- [14] G. D. Pamungkas, Y. Parwati, and B. D. Putranto, "Pengembangan Aplikasi Pendaftaran Siswa Baru Berbasis Web Dengan React.Js dan Tailwind CSS," *Jurnal Algoritma*, vol. 22, no. 1, pp. 37–48, 2025, doi: 10.33364/ALGORITMA/V.22-1.2135.
- [15] A. Musyaffa, M. Zulfa, and M. S. Alim, "Rancang bangun PureCompute platform e-commerce untuk belanja laptop berbasis website," *Jurnal SINTA: Sistem Informasi dan Teknologi Komputasi*, vol. 1, no. 1, pp. 21–29, 2024, doi: 10.61124/sinta.v1i1.9.

- [16] M. Huzaifa, R. Elisa Nalawati, N. Marcheta, and M. Fadhilah, “Penerapan React. Js dalam Perancangan Sistem Coffee Shop Berbasis Web,” *Journal of Social Science Research*, vol. 4, no. 3, pp. 17918–17928, 2024, doi: 10.31004/innovative.v4i3.12673
- [17] S. D. Pratama, L. Lasimin, and M. N. Dadaprawira, “Pengujian Black Box Testing Pada Aplikasi Edu Digital Berbasis Website Menggunakan Metode Equivalence Dan Boundary Value,” *Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD*, vol. 6, no. 2, pp. 560–569, 2023, doi: 10.53513/JSK.V6I2.8166.
- [18] A. Aliyah, N. Hartono, A. Azhari Muin, “Penggunaan User Acceptance Testing (UAT) Pada Pengujian Sistem Informasi Pengelolaan Keuangan Dan Inventaris Barang,” *Switch: Jurnal Sains dan Teknologi Informasi*, vol. 3, no. 2, pp. 42–58, 2025, doi: 10.62951/SWITCH.V3I1.330.