

Servis-Hub: Platform Digital Manajemen Dan Tracking Servis Perangkat Elektronik Berbasis Web

Muhammad Rafi Aditya¹, Jauhan Ahmad², Muhammad Setya Adjie³, Eko Prasetyo Widhi⁴

^{1,2,3,4}Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Darussalam Gontor

E-mail: 1muhammadrafiaditya12@student.cs.unida.gontor.ac.id,

2jauhanahmad80@student.cs.unida.gontor.ac.id,

3muhammadsetyaadjie14@student.cs.unida.gontor.ac.id, 4ekoprasetiowidhi@unida.gontor.ac.id.

Abstrak

Manajemen layanan perbaikan perangkat elektronik sering menghadapi kendala operasional akibat ketergantungan pada metode pencatatan manual. Permasalahan utama yang muncul meliputi risiko hilangnya data riwayat servis, ketidakakuratan rekapitulasi pendapatan, serta sulitnya pelanggan mendapatkan informasi status perbaikan secara *real-time* yang berdampak pada kepercayaan konsumen. Penelitian ini bertujuan membangun "Servis-Hub", sebuah sistem informasi berbasis web untuk mendigitalisasi operasional bengkel TI sebagai solusi atas permasalahan tersebut. Sistem dikembangkan menggunakan metode System Development Life Cycle (SDLC) model Waterfall, dengan memanfaatkan teknologi React.js untuk antarmuka pengguna yang responsif dan Supabase untuk manajemen basis data *real-time*. Hasil pengujian menunjukkan bahwa sistem berhasil memfasilitasi pendaftaran servis dengan validasi kelengkapan barang, pelacakan status perbaikan secara transparan, serta pembuatan laporan keuangan dan invoice otomatis. Implementasi sistem ini terbukti meningkatkan efisiensi administrasi, meminimalisir sengketa kelengkapan unit, serta meningkatkan transparansi layanan kepada pelanggan..

Kata kunci: Sistem Informasi Servis, React.js, Supabase, Monitoring Perbaikan, Web Application..

1. PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi di era digital saat ini telah mengubah paradigma operasional bisnis, termasuk pada sektor Usaha Mikro, Kecil, dan Menengah (UMKM) bidang jasa. Integrasi teknologi digital tidak lagi menjadi opsi, melainkan kebutuhan mendasar untuk meningkatkan daya saing dan efisiensi operasional. Dalam konteks penyedia jasa perbaikan perangkat elektronik (IT Workshop), penguasaan manajemen data berbasis komputer menjadi kompetensi krusial[1]. Peralihan dari sistem konvensional ke ekosistem digital memungkinkan pengelolaan sumber daya, stok suku cadang, dan hubungan pelanggan menjadi lebih terstruktur dan akuntabel.

Meskipun urgensi digitalisasi semakin meningkat, observasi di lapangan menunjukkan bahwa banyak bengkel servis laptop masih mengandalkan metode manajemen konvensional. Pencatatan transaksi seringkali dilakukan secara manual menggunakan buku tulis atau nota fisik yang rentan hilang dan rusak. Keterbatasan media pencatatan ini menimbulkan kendala nyata berupa fragmentasi data, kesulitan dalam merekapitulasi riwayat perbaikan, serta risiko kesalahan perhitungan estimasi biaya[2]. Dampak yang paling signifikan dari keterbatasan ini dirasakan oleh pelanggan, yaitu sulitnya mendapatkan informasi status perbaikan secara *real-time*, yang seringkali memicu ketidakpercayaan terhadap penyedia jasa.

Sistem Informasi Berbasis Web adalah Solusi Sebagai respons terhadap permasalahan tersebut, penerapan Sistem Informasi Manajemen berbasis web hadir sebagai solusi yang relevan. Berbeda dengan aplikasi *desktop* konvensional yang memerlukan instalasi khusus, aplikasi web menawarkan fleksibilitas akses lintas perangkat (*cross-platform*) yang memungkinkan pemantauan data dilakukan kapan saja dan di mana saja. Teknologi web modern, seperti *Single Page Application* (SPA)[3], menawarkan antarmuka yang interaktif, cepat, dan responsif. Keunggulan utama dari pendekatan ini adalah kemampuan integrasi data secara terpusat (*centralized*), sehingga informasi status perbaikan dapat disajikan secara transparan kepada pelanggan maupun teknisi dalam satu ekosistem yang sama.

Penelitian mengenai digitalisasi layanan servis telah banyak dilakukan sebelumnya. Studi terdahulu umumnya berfokus pada pengembangan aplikasi *Point of Sales* (POS) untuk manajemen stok dan kasir menggunakan basis data lokal[2]. Temuan umum dari penelitian-penelitian tersebut menunjukkan bahwa digitalisasi administrasi mampu meningkatkan efisiensi waktu transaksi dan akurasi laporan keuangan. Namun, mayoritas sistem yang ada masih bersifat administratif internal (fokus pada kasir) dan belum optimal dalam menjembatani komunikasi pasca-transaksi antara teknisi dan pelanggan.

Berdasarkan tinjauan literatur, terdapat kesenjangan (*gap*) pada sistem yang tersedia saat ini. Penelitian sebelumnya cenderung: fokus pada implementasi teknis inventaris barang semata, kurang menekankan aspek pengalaman pelanggan (*user experience*) dalam hal pelacakan status (*tracking*)[2], dan minimnya fitur validasi kelengkapan barang (seperti tas dan charger) yang sering menjadi sumber sengketa. Oleh karena itu, dibutuhkan sebuah aplikasi manajemen servis berbasis web yang tidak hanya stabil secara teknis, tetapi juga dirancang khusus dengan fitur *tracking* transparan dan validasi kelengkapan unit untuk meminimalisir risiko operasional bengkel skala kecil-menengah.

Penelitian ini bertujuan untuk mengembangkan "Servis-Hub", sebuah platform digital berbasis web menggunakan teknologi React.js dan Supabase untuk manajemen dan pelacakan servis perangkat elektronik. Kontribusi utama yang diharapkan dari penelitian ini adalah: menyediakan alternatif media manajemen yang mengeliminasi ketergantungan pada pencatatan kertas, meningkatkan kepercayaan pelanggan melalui fitur visualisasi status perbaikan (*tracking*) yang transparan, serta menyediakan dasar pengembangan sistem administrasi bengkel yang modern dan terintegrasi.

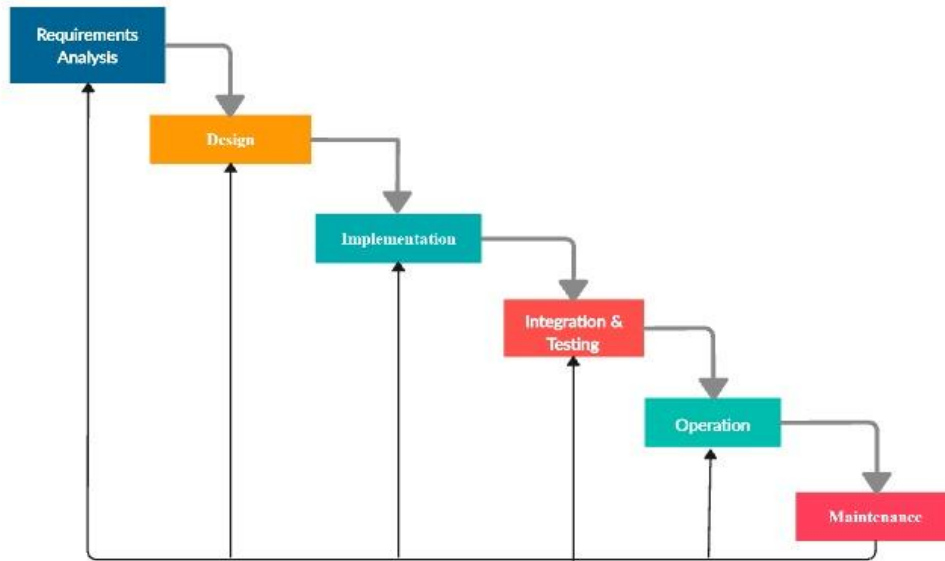
2. METODE PENELITIAN

2.1 Jenis dan Pendekatan Penelitian

Jenis penelitian yang digunakan adalah penelitian terapan (*applied research*) yang berfokus pada pengembangan solusi praktis untuk memecahkan masalah operasional di bengkel servis komputer. Pendekatan yang digunakan adalah rekayasa perangkat lunak (*software engineering*) dengan orientasi pada pengembangan aplikasi berbasis web modern. Pendekatan ini dipilih untuk menjembatani kesenjangan antara kebutuhan pencatatan manual dan efisiensi teknologi digital[4].

2.2 Metode Pengembangan Sistem

Metode pengembangan sistem yang diterapkan adalah *System Development Life Cycle (SDLC) model Waterfall*. [5] Model ini dipilih karena karakteristik kebutuhan sistem "Servis-Hub" telah terdefinisi dengan jelas sejak awal, meliputi manajemen antrean servis, pelacakan status, dan pelaporan keuangan. Pendekatan sekuensial ini memastikan setiap fase diselesaikan secara menyeluruh sebelum melangkah ke fase berikutnya, meminimalisir risiko kesalahan logika di tahap akhir



Gambar 1. Model Pengembangan Sistem Waterfall

2.3 Tahapan Penelitian

Tahapan penelitian disusun secara sistematis mengikuti alur kerja *Waterfall*, mulai dari analisis hingga pemeliharaan, sebagai berikut:

2.3.1. Analisis Kebutuhan

Tahap ini dilakukan untuk mengidentifikasi masalah pada sistem konvensional. Berdasarkan observasi, ditemukan kendala utama berupa: risiko hilangnya data riwayat servis pada buku manual, ketidakjelasan informasi status perbaikan bagi pelanggan, dan kesulitan teknisi dalam merekapitulasi total pendapatan bulanan.

2.3.2. Kebutuhan Sistem

Berdasarkan analisis masalah, dirumuskan spesifikasi kebutuhan sistem sebagai berikut:

1. Kebutuhan Fungsional:

- Sistem harus mampu melakukan manajemen pengguna (*Admin* dan *Customer*) melalui autentikasi aman.
- Sistem harus menyediakan formulir pendaftaran servis dengan fitur validasi kelengkapan aksesoris.
- Sistem wajib memiliki fitur cetak invoice otomatis berukuran A4.

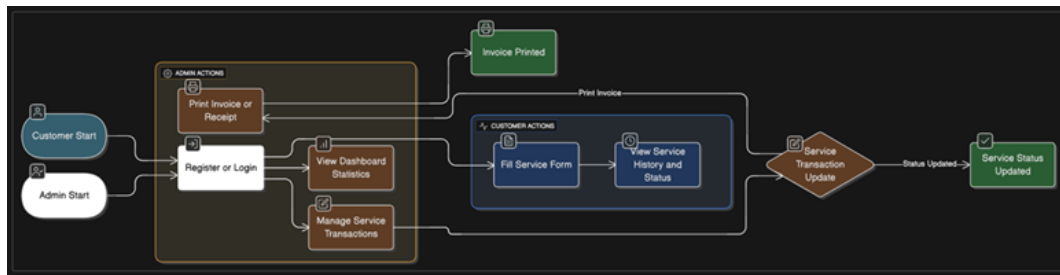
2. Kebutuhan Non-Fungsional:

- Antarmuka pengguna harus responsif (dapat diakses via *mobile* dan *desktop*).
- Pembaruan data status perbaikan harus bersifat *real-time*.

2.3.3. Perancangan Sistem

Perancangan sistem difokuskan pada dua aspek utama:

1. Perancangan Basis Data: Menggunakan skema relasional pada Supabase dengan dua entitas utama: *tabel profiles* untuk menyimpan data pengguna dan *tabel transactions* untuk menyimpan riwayat perbaikan, status, dan biaya[6].
2. Perancangan Antarmuka (UI): Menggunakan desain minimalis dengan Tailwind CSS[6] untuk memastikan kemudahan navigasi bagi pengguna awam.



Gambar 2. Use Case Diagram

2.3.4. Perancangan Algoritma Kategorisasi (Logika Bisnis)

Meskipun tidak menggunakan algoritma SPK kompleks, sistem menerapkan Algoritma Pengelompokan Otomatis (*Rule-Based*) untuk visualisasi data. Algoritma ini bekerja pada modul *Dashboard Admin* dengan langkah logis:

1. Sistem mengambil data mentah deskripsi masalah dari *database*.
2. *Algoritma Regex Matching* memindai pola teks [Kategori] pada awal deskripsi.
3. Data dikelompokkan menjadi kategori (*Hardware/Software*) untuk divisualisasikan menjadi *Pie Chart* dan perhitungan total pendapatan mingguan.

2.3.5. Implementasi Sistem

Tahap implementasi merupakan penerjemahan desain ke dalam kode program (*coding*):

- *Front-end*: Dibangun menggunakan *library React.js* dengan *build tool Vite* untuk performa tinggi.
- *Back-end*: Menggunakan Supabase sebagai penyedia API dan manajemen basis data PostgreSQL.
- *Fitur Cetak*: Mengimplementasikan *library react-to-print* untuk menghasilkan dokumen fisik nota servis secara presisi.

2.3.6. Pengujian Sistem

Pengujian dilakukan menggunakan metode *Black Box Testing*[7]. Fokus pengujian adalah memvalidasi fungsionalitas *input-output* tanpa melihat struktur kode internal. Skenario pengujian meliputi:

1. Uji validasi formulir: Memastikan data kelengkapan barang (Tas, Charger) tersimpan dengan benar.
2. Uji kalkulasi: Memastikan total biaya pada *dashboard Admin* sesuai dengan akumulasi transaksi yang berstatus "Done".
3. Uji responsivitas: Memastikan tampilan aplikasi tetap proporsional saat diakses melalui perangkat *mobile*.

2.3.7. Pemeliharaan Sistem

Tahap akhir meliputi penyebaran (*deployment*) dan pemeliharaan rutin:

- *Deployment*: Aplikasi *di-build* menjadi berkas statis untuk diunggah ke server produksi.
- *Keamanan*: Penerapan *Row Level Security* (RLS) pada Supabase dipantau secara berkala untuk memastikan data pelanggan tidak bocor ke publik[8].
- *Update*: Pembaruan library dependensi dilakukan untuk menjaga kompatibilitas dengan *browser modern*.

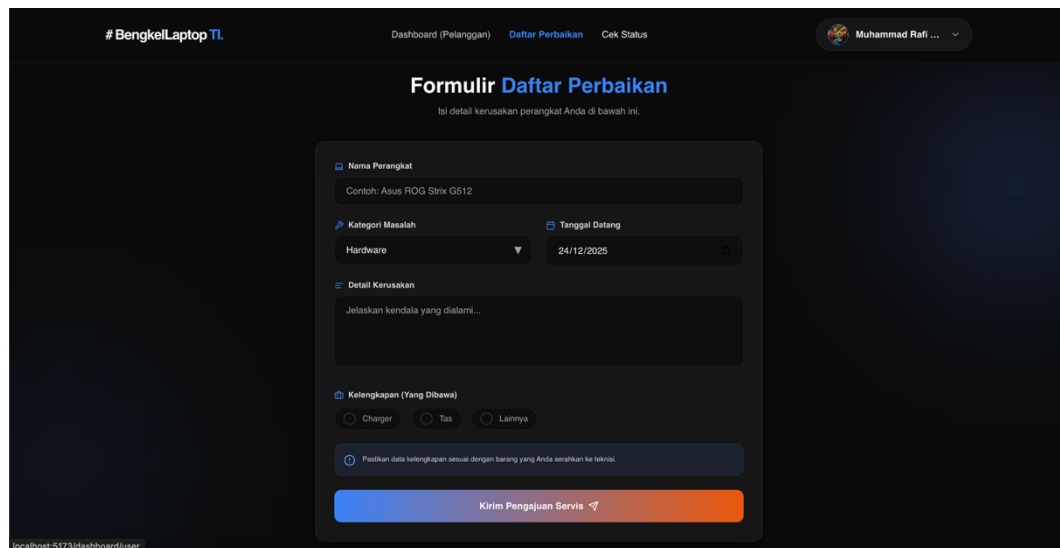
3. HASIL DAN PEMBAHASAN

3.1 Analisis Kebutuhan Fungsional dan Perancangan (Fase 1 & 2)

3.1.1. Logika Validasi Kelengkapan dan Formulir Servis

Formulir Pendaftaran Servis dirancang untuk mengatasi masalah sengketa kelengkapan barang yang sering terjadi di bengkel konvensional. Berbeda dengan pencatatan manual, sistem ini menerapkan validasi input terstruktur. Pada implementasinya, formulir tidak hanya mencatat kerusakan, tetapi juga mewajibkan pengguna memvalidasi aksesoris bawaan. Data kelengkapan (seperti *Charger*, Tas, atau Lainnya) diproses oleh

logika *frontend* menjadi *string* terformat yang digabungkan dengan deskripsi masalah utama sebelum dikirim ke basis data. Hal ini memastikan teknisi menerima informasi yang presisi mengenai kondisi awal perangkat saat diserahkan.

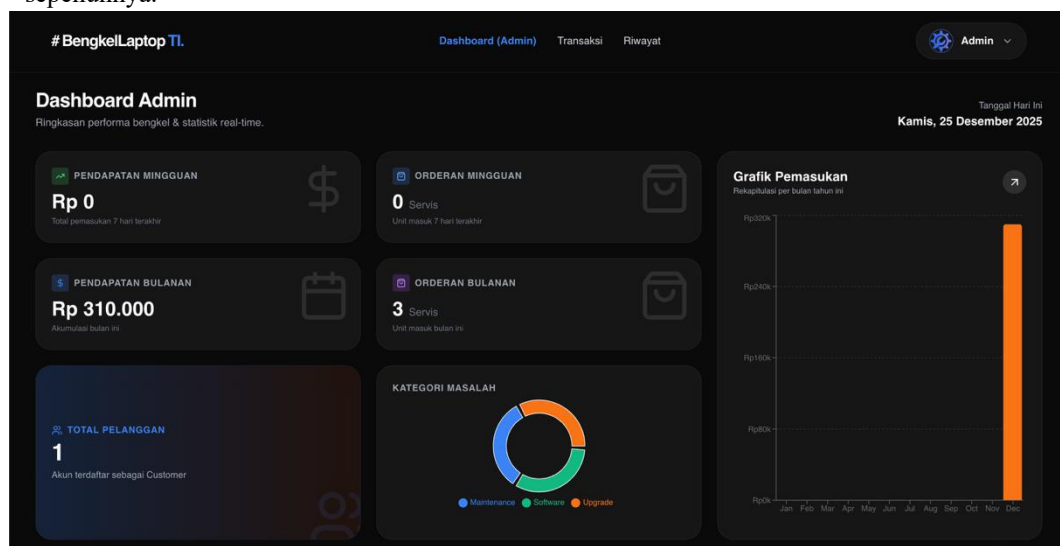


Gambar 3. Formulir Pendaftaran Servis dengan Validasi Kelengkapan

3.1.2. Implementasi Manajemen Akses (*RBAC 2 Level*)

Penerapan *Role-Based Access Control* (RBAC) terbukti efektif dalam memisahkan lingkungan kerja antara pelanggan dan teknisi. Penerapan model RBAC ini sangat krusial dalam meningkatkan sistem manajemen izin untuk memastikan data sensitif tetap terjaga[9]. Sistem membagi hak akses menjadi dua entitas utama:

- Customer*: Dibatasi hanya untuk operasi *Create* (mengajukan servis) dan *Read* (melihat riwayat sendiri).
- Admin*: Memiliki akses penuh (*Full Access*) untuk melihat seluruh transaksi, mengubah status pengerjaan, dan mengakses laporan keuangan. Pemisahan ini diatur langsung pada level basis data menggunakan *Row Level Security* (RLS) di Supabase, sehingga manipulasi data oleh pihak tidak berwenang dapat dicegah sepenuhnya.



Gambar 4. Antarmuka Dashboard Admin

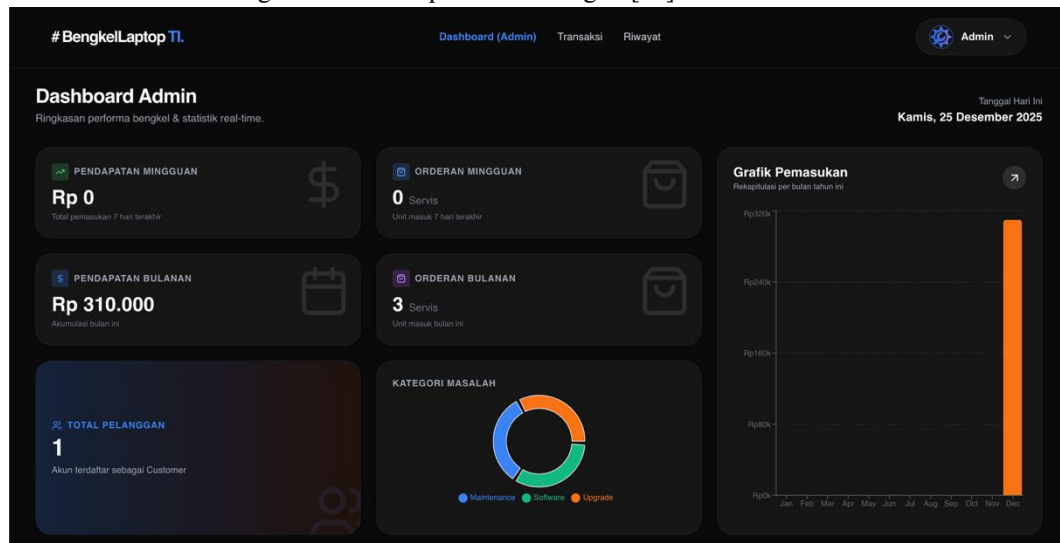
3.2 Implementasi Modul Utama dan Integrasi Alur Kerja (Fase 3)

3.2.1. Implementasi Sistem

Fitur unggulan pada sisi *Admin* adalah dasbor analitik yang menyajikan rekapitulasi performa bisnis secara *real-time*. Menggunakan pustaka visualisasi data, sistem secara otomatis menghitung total pendapatan dari transaksi yang berstatus "*Done*" dan menampilkannya dalam dua format:

- Grafik Batang (*Bar Chart*): Untuk membandingkan tren pendapatan mingguan dan bulanan.
- Diagram Lingkaran (*Pie Chart*): Untuk memetakan distribusi kategori kerusakan (*Hardware vs Software*), membantu manajemen dalam pengambilan keputusan stok suku cadang.

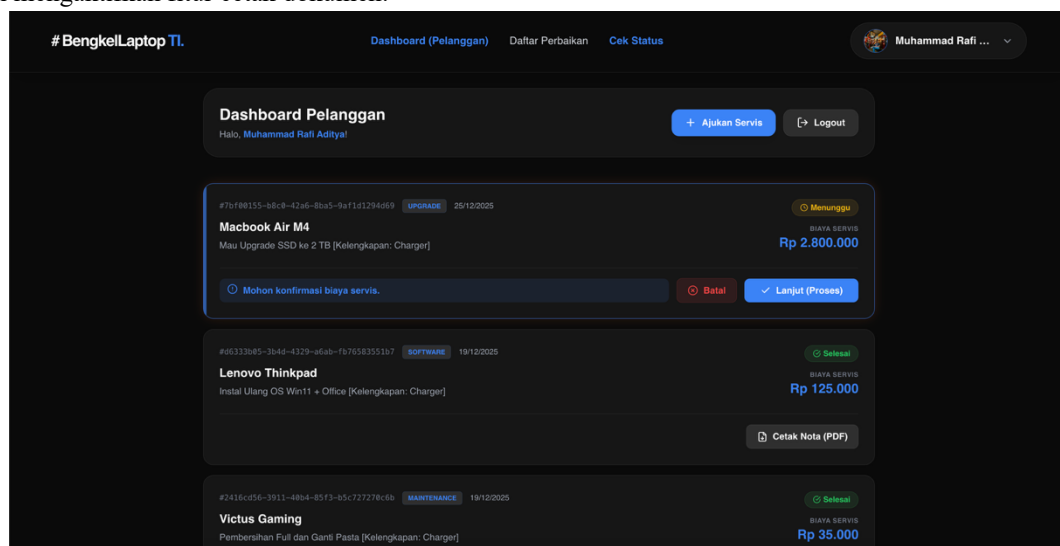
Penggunaan kombinasi grafik batang dan diagram lingkaran ini bertujuan untuk memberikan penilaian visual yang akurat dalam membandingkan tren data operasional bengkel[10].



Gambar 5. Visualisasi Statistik Pendapatan dan Kategori Kerusakan

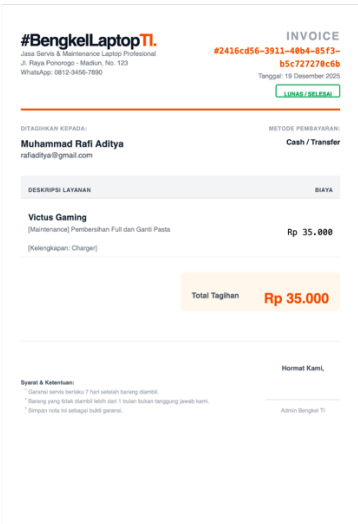
3.2.2. Order Status Tracking (OST) dan Invoice Digital

Modul *Order Status Tracking* (OST) memungkinkan transparansi penuh dalam siklus perbaikan. Status pengerjaan diperbarui oleh Admin dan langsung ter-sinkronisasi di dasbor pelanggan. Setelah servis selesai, sistem mengaktifkan fitur cetak dokumen.



Gambar 6. Antarmuka Cek Status Pelanggan

Komponen *Invoice* dirancang dengan format kertas A4 yang siap cetak (*print-ready*). Dokumen ini memuat detail krusial seperti Nomor Referensi, Identitas Pelanggan, Rincian Biaya, dan Syarat Garansi yang digenerate secara dinamis dari data transaksi. Keberadaan fitur ini menghilangkan kebutuhan akan penulisan nota manual yang rentan kesalahan hitung.



Gambar 7. Tampilan Cetak Invoice Digital

3.3 Implementasi dan Hasil Pengujian Sistem (Fase 4)

3.3.1. Hasil Pengujian Black Box

Pengujian fungsional dilakukan menggunakan metode *Black Box Testing* untuk memverifikasi kesesuaian output sistem dengan logika bisnis yang dirancang. Metode ini, khususnya dengan teknik *Equivalence Partitions*, terbukti efektif dalam memvalidasi fungsionalitas kritis tanpa harus mengetahui struktur internal kode program[11]. Fokus pengujian mencakup validasi input, akurasi perhitungan biaya, dan keamanan akses.

Tabel 1. Ringkasan Hasil *Black Box Testing* Fungsionalitas Kritis

| No. | Fungsionalitas Kritis yang Diuji | Hasil yang Diharapkan | Hasil Verifikasi | Status |
|-----|---|--|------------------|--------|
| 1 | Validasi Kelengkapan (Form Servis) | Data aksesoris (Tas/ <i>Charger</i>) tersimpan akurat dalam deskripsi masalah. | Berhasil | Lulus |
| 2 | Kalkulasi Biaya Servis | Total biaya pada <i>Invoice</i> sesuai dengan input <i>Admin</i> dan format Rupiah valid. | Berhasil | Lulus |
| 3 | Pemisahan Hak Akses (RBAC) | Akun <i>Customer</i> tidak dapat mengakses menu/URL khusus <i>Admin</i> . | Berhasil | Lulus |
| 4 | Sinkronisasi Status (<i>Tracking</i>) | Perubahan status di <i>Admin</i> (<i>Pending ke Done</i>) langsung terlihat di <i>Customer</i> . | Berhasil | Lulus |
| 5 | Visualisasi Grafik | Grafik batang menampilkan nominal pendapatan yang sesuai dengan data tabel transaksi. | Berhasil | Lulus |

Pengujian penerimaan pengguna (*User Acceptance Test*) dilakukan dengan melibatkan responden yang terdiri dari teknisi dan pengguna umum. Metode ini digunakan sebagai standar untuk mengukur tingkat kepuasan serta memastikan bahwa sistem yang dibangun telah memenuhi kebutuhan fungsional dari sisi pengguna akhir[12]. Hasil pengujian menunjukkan bahwa antarmuka sistem dinilai intuitif (*user-friendly*), terutama pada fitur *tracking* status yang dianggap sangat membantu mengurangi ketidakpastian informasi layanan. Fitur cetak *invoice* otomatis juga dinilai mempercepat proses administrasi di meja kasir secara signifikan.

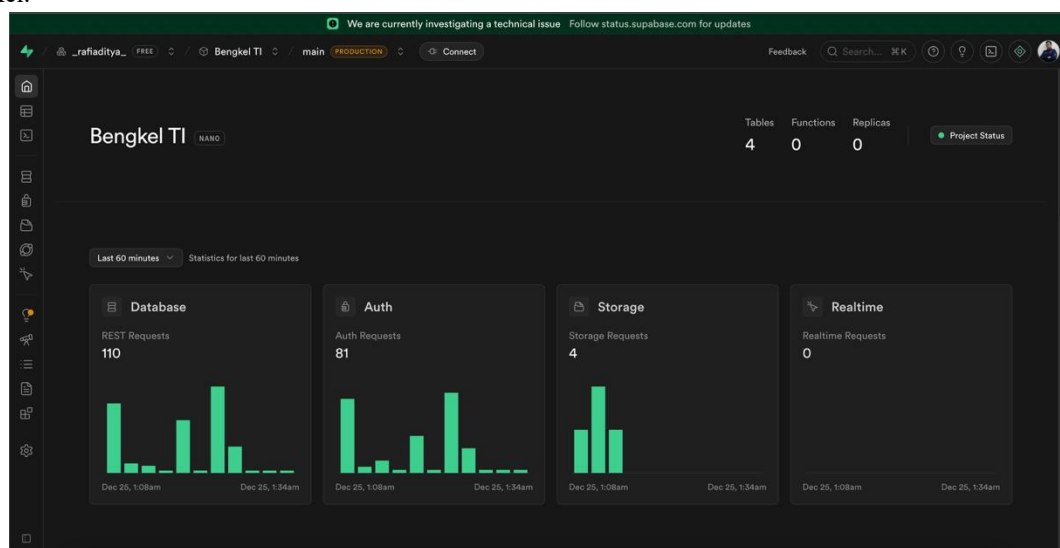
3.4 Operasi dan Pemeliharaan (Fase 5)

Tahap operasi dan pemeliharaan merupakan fase krusial untuk memastikan keberlangsungan sistem "Servis-Hub" di lingkungan produksi. Berbeda dengan pengembangan lokal, fase ini berfokus pada stabilitas akses, keamanan data, dan pemeliharaan performa aplikasi berbasis *cloud*[13].

3.4.1. Deployment dan Stabilitas Sistem

Aplikasi sisi klien (*Client-side*) yang dibangun dengan React.js telah melalui proses optimasi (*production build*) menggunakan *tool* Vite. Langkah optimasi ini sejalan dengan berbagai teknik peningkatan performa pada ReactJS untuk meminimalkan waktu muat dan meningkatkan responsivitas aplikasi secara keseluruhan[14]. Proses ini mengonversi kode sumber menjadi berkas statis yang ringan untuk meminimalkan waktu muat (*load time*) di sisi pengguna.

Pemantauan stabilitas dilakukan dengan memeriksa latensi koneksi antara antarmuka web dan layanan Supabase. Hasil pemantauan menunjukkan bahwa fitur *real-time subscription* yang digunakan untuk memperbarui status servis di dasbor pelanggan berjalan stabil tanpa gangguan signifikan pada jam operasional bengkel.



Gambar 8. Dashboard Project Supabase

3.4.2. Pemeliharaan Data dan Keamanan (Security Maintenance)

Pemeliharaan keamanan difokuskan pada penerapan kebijakan akses data. Administrator secara berkala meninjau log akses pada panel Supabase untuk memastikan tidak ada anomali aktivitas. Tinjauan komprehensif terhadap langkah-langkah keamanan ini termasuk autentikasi dan kontrol akses merupakan standar krusial dalam melindungi sistem basis data dari ancaman eksternal[15].

Selain itu, fitur Manajemen Hapus Data (*Reset*) atau pembersihan transaksi lama dilakukan secara hati-hati oleh *Super Admin*. Mengingat sistem ini menggunakan relasi data yang ketat (integritas referensial antara *tabel profiles* dan *transactions*), prosedur pemeliharaan basis data dilakukan untuk mencegah penumpukan data sampah (*junk data*) yang dapat memperlambat kinerja pencarian riwayat servis.

3.4.3. Dokumentasi dan Transfer Pengetahuan

Untuk mendukung operasional mandiri oleh staf bengkel, penelitian ini juga menghasilkan dokumen Standar Operasional Prosedur (SOP) penggunaan aplikasi. Proses transfer pengetahuan pasca-implementasi ini

sangat penting bagi staf operasional TI untuk menjamin keberlanjutan penggunaan sistem di masa mendatang[16]. SOP ini mencakup panduan langkah demi langkah untuk:

- Registrasi pelanggan baru dan verifikasi akun.
- Prosedur input kelengkapan barang (*Tas/Charger*) yang benar untuk menghindari sengketa.
- Langkah-langkah pencetakan *invoice digital* dan penanganan kendala koneksi printer.

4. SIMPULAN

Penelitian ini berhasil merancang dan membangun "Servis-Hub", sebuah *platform* sistem informasi berbasis web untuk mendigitalisasi manajemen operasional bengkel komputer yang sebelumnya berjalan secara manual. Melalui penerapan *metode System Development Life Cycle (SDLC) model Waterfall*, sistem dikembangkan secara terstruktur menggunakan teknologi *React.js* untuk antarmuka pengguna yang responsif dan *Supabase* sebagai basis data *real-time* berbasis *cloud*.

Hasil pengujian fungsional menunjukkan bahwa sistem mampu mengatasi permasalahan administrasi dan transparansi layanan secara efektif. Fitur validasi kelengkapan pada formulir pendaftaran terbukti meminimalisir risiko sengketa barang antara teknisi dan pelanggan. Selain itu, integrasi fitur pelacakan status (*tracking*) dan pencetakan *invoice digital* secara otomatis telah meningkatkan efisiensi waktu pelayanan di meja kasir. Bagi manajemen bengkel, ketersediaan dasbor visualisasi data pendapatan memberikan kemudahan dalam memantau kesehatan bisnis secara akurat dan *real-time*.

Secara keseluruhan, implementasi Servis-Hub memberikan dampak positif signifikan berupa peningkatan akuntabilitas data transaksi, efisiensi alur kerja teknisi, serta peningkatan kepercayaan pelanggan melalui transparansi proses perbaikan. Sistem ini siap diterapkan sebagai solusi modern untuk standar baru pelayanan jasa servis elektronik.

5. SARAN

Berdasarkan hasil implementasi dan evaluasi sistem "Servis-Hub", terdapat beberapa aspek yang dapat dikembangkan pada penelitian selanjutnya untuk menyempurnakan fungsionalitas sistem:

1. Integrasi Notifikasi Otomatis: Penelitian selanjutnya disarankan untuk mengintegrasikan fitur *WhatsApp Gateway* atau notifikasi *Email*. Hal ini bertujuan agar pelanggan mendapatkan pesan otomatis di ponsel mereka setiap kali status servis diperbarui (misalnya: "Servis Selesai"), tanpa harus mengecek dasbor web secara berkala.
2. Penambahan Modul Pembayaran Digital (*Payment Gateway*): Sistem saat ini masih berfokus pada pencatatan transaksi tunai dan transfer manual. Pengembangan di masa depan dapat menambahkan integrasi dengan layanan *Payment Gateway* (seperti Midtrans atau Xendit) untuk mendukung pembayaran via QRIS dan E-Wallet secara langsung di aplikasi.
3. Manajemen Stok Suku Cadang (*Inventory System*): Disarankan untuk menambahkan fitur manajemen inventaris yang terhubung langsung dengan transaksi servis. Dengan fitur ini, stok suku cadang (seperti RAM, SSD, atau Baterai) akan berkurang secara otomatis saat digunakan oleh teknisi dalam perbaikan, sehingga kontrol stok menjadi lebih akurat.
4. Pengembangan Versi Aplikasi *Mobile (Native App)*: Mengingat basis teknologi yang digunakan adalah *React.js*, pengembangan selanjutnya dapat memperluas jangkauan platform menjadi aplikasi *mobile* (Android/iOS) menggunakan kerangka kerja *React Native*. Aplikasi *native* memungkinkan pemanfaatan fitur kamera ponsel untuk memindai *barcode* nota atau memotret kondisi fisik laptop secara langsung.
5. Implementasi Fitur *Chatbot*: Untuk meningkatkan responsivitas layanan pelanggan, sistem dapat dilengkapi dengan fitur *Chatbot* sederhana untuk menjawab pertanyaan umum (*FAQ*) seputar jam operasional, estimasi biaya servis, dan lokasi bengkel secara otomatis 24 jam.

DAFTAR PUSTAKA

- [1] Atharva Gurao, "Electronic Appliances Repair Tracking Portal," *J. Inf. Syst. Eng. Manag.*, vol. 10, no. 34s, hal. 539–549, Apr 2025, doi: 10.52783/jisem.v10i34s.5828.
- [2] Faris, Hartono, dan R. E. Pambudi, "Optimalisasi Proses Pemesanan Jasa Servis Komputer dan Laptop melalui Aplikasi Web Menggunakan Framework Codeigniter," *Sienna*, vol. 5, no. 2, hal. 122–141, Des

- 2024, doi: 10.47637/sienna.v5i2.1638.
- [3] D. V Kornienko, S. V Mishina, dan M. O. Melnikov, “The Single Page Application architecture when developing secure Web services,” *J. Phys. Conf. Ser.*, vol. 2091, no. 1, hal. 012065, Nov 2021, doi: 10.1088/1742-6596/2091/1/012065.
- [4] R. Choirudin dan A. Adil, “Implementasi Rest Api Web Service dalam Membangun Aplikasi Multiplatform untuk Usaha Jasa,” *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 18, no. 2, hal. 284–293, Mei 2019, doi: 10.30812/matrik.v18i2.407.
- [5] F. Adetya, “The ABC Laundry Service Information System Based on Web using SDLC Method,” *J. Media Inf. Teknol.*, vol. 1, no. 2, hal. 53–62, Okt 2024, doi: 10.69616/mit.v1i2.187.
- [6] V. Aienobe dan M. Z. Iqbal, “RESPONSIVE WEB DESIGN FOR ENHANCED USER EXPERIENCE (UX) AND USER INTERFACE (UI),” *Int. J. Comput. Appl.*, vol. 187, no. 34, hal. 72–93, Agu 2025, doi: 10.5120/ijca2025925533.
- [7] A. Maulana, A. Kurniawan, W. Keumala, V. R. Sukma, dan A. Saifudin, “Pengujian Black Box pada Aplikasi Penjualan Berbasis Web Menggunakan Metode Equivalents Partitions (Studi Kasus: PT Arap Store),” *J. Teknol. Sist. Inf. dan Apl.*, vol. 3, no. 1, hal. 50, Feb 2020, doi: 10.32493/jtsi.v3i1.4307.
- [8] N. S. Wisidagama dan F. Marikkar, “WATERFALL MODEL OVER PCD.UCT MODEL REVIEW,” *Autom. Technol. Bus. Process.*, vol. 16, no. 3, hal. 126–131, Okt 2024, doi: 10.15673/atbp.v16i3.2927.
- [9] K. Li dan B.-W. Min, “Improvement of Permission Management System Based on RBAC(Role-Based Access Control) Model,” *Int. J. CONTENTS*, vol. 19, no. 4, hal. 86–97, Des 2023, doi: 10.5392/IJoC.2023.19.4.086.
- [10] A. Hill, “Are pie charts evil? An assessment of the value of pie and donut charts compared to bar charts,” *Inf. Vis.*, vol. 24, no. 1, hal. 3–23, Jan 2025, doi: 10.1177/14738716241259432.
- [11] B. A. Priyaangga, D. B. Aji, M. Syahroni, N. T. S. Aji, dan A. Saifudin, “Pengujian Black Box pada Aplikasi Perpustakaan Menggunakan Teknik Equivalence Partitions,” *J. Teknol. Sist. Inf. dan Apl.*, vol. 3, no. 3, hal. 150, Agu 2020, doi: 10.32493/jtsi.v3i3.5343.
- [12] I Dewa Gde Satria Pramana Erlangga, Sugiarto Sugiarto, dan Afina Lina Nurlaili, “PENGUJIAN USER ACCEPTANCE TEST PADA APLIKASI BANGBELI,” *J. Inform. Dan Teknologi Komput.*, vol. 3, no. 3, hal. 213–219, Nov 2023, doi: 10.55606/jitek.v3i3.2003.
- [13] M. C. Ungan, “Standardization through process documentation,” *Bus. Process Manag. J.*, vol. 12, no. 2, hal. 135–148, Mar 2006, doi: 10.1108/14637150610657495.
- [14] A. Javeed, “Performance Optimization Techniques for ReactJS,” in *2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, IEEE, Feb 2019, hal. 1–5. doi: 10.1109/ICECCT.2019.8869134.
- [15] Habeeb Omotunde dan Maryam Ahmed, “A Comprehensive Review of Security Measures in Database Systems: Assessing Authentication, Access Control, and Beyond,” *Mesopotamian J. CyberSecurity*, vol. 2023, hal. 115–133, Agu 2023, doi: 10.58496/MJCSC/2023/016.
- [16] R. Santhanam, L. Seligman, dan D. Kang, “Postimplementation Knowledge Transfers to Users and Information Technology Professionals,” *J. Manag. Inf. Syst.*, vol. 24, no. 1, hal. 171–199, Jul 2007, doi: 10.2753/MIS0742-1222240105.