

BAB II

KAJIAN PUSTAKA

A. Kajian Teoritis

1. Rancang Bangun

Hisyam et al (2025) menjelaskan bahwa perancangan perangkat lunak merupakan tahapan krusial untuk mentransformasikan kebutuhan pengguna ke dalam arsitektur sistem yang terstruktur. Rancang bangun yang sistematis dapat meminimalkan risiko kesalahan logika pada modul-modul aplikasi utama sejak fase awal pengembangan. Implementasi metode perancangan yang tepat bertindak sebagai cetak biru fungsional yang mengintegrasikan alur data bisnis secara runtut dan transparan. Oleh karena itu, kehadiran fase rancang bangun yang matang dinilai sangat penting karena mampu meningkatkan efisiensi proses manajemen internal organisasi, mengoptimalkan skalabilitas sistem saat terjadi lonjakan data, serta mempermudah para pengembang dalam melakukan pemeliharaan kode program secara berkelanjutan.

Fernando et al (2023) menyatakan bahwa rancang bangun sebagai proses menerjemahkan cetak biru desain sistem menjadi perintah logika terkomputerisasi melalui penerapan arsitektur perangkat lunak yang terstandarisasi. Rancang bangun mencakup tahapan panjang yang dimulai dari survei objek penelitian, perumusan masalah, hingga pemetaan kebutuhan pengguna secara spesifik. Struktur perancangan ini bekerja dengan mengonfigurasi komponen-komponen *database* dan antarmuka

untuk menghasilkan sistem yang sinkron dan responsif. Penggunaan pendekatan rancang bangun yang disiplin menjamin seluruh fungsionalitas sistem informasi dapat berjalan dengan tingkat kecepatan yang optimal, meminimalkan redundansi data, serta mempermudah pelacakan galat teknis secara terukur.

Saputra et al (2025) menjelaskan bahwa proses rancang bangun merupakan langkah krusial untuk menciptakan sistem informasi yang mampu mengatasi ketidakefisienan operasional dan kesalahan pencatatan secara manual. Rancang bangun yang sistematis bertindak sebagai fondasi untuk menyatukan spesifikasi kebutuhan fitur layanan seperti manajemen armada dan transaksi reservasi dengan struktur data yang terkomputerisasi. Melalui tahapan pemodelan data dan perancangan antarmuka yang jelas, alur instruksi di dalam sistem dapat diproses secara teratur dan efisien. Selain itu, tahapan rancang bangun ini juga harus diikuti dengan proses pengujian sistem yang komprehensif untuk memvalidasi performa fungsionalitas aplikasi, guna memastikan seluruh modul bebas dari kesalahan sebelum diimplementasikan pada lingkungan kerja yang sebenarnya.

Berdasarkan beberapa kajian tersebut, dapat disimpulkan bahwa rancang bangun adalah proses menerjemahkan desain dan kebutuhan sistem ke dalam bentuk instruksi terkomputerisasi serta arsitektur basis data yang terstruktur. Rancang bangun berfungsi sebagai cetak biru operasional untuk mengintegrasikan alur fungsionalitas aplikasi dan memastikan sistem

berjalan secara efisien. Dalam konteks penelitian ini, rancang bangun digunakan sebagai pedoman terstruktur untuk merancang, membangun, dan menyatukan seluruh komponen sistem informasi manajemen *rent car* Bintang Travel Kabupaten Ponorogo.

2. Sistem Informasi

Aryani & Voutama (2025) menjelaskan bahwa sistem informasi manajemen merupakan kombinasi terstruktur antara perangkat lunak aplikasi dan basis data elektronik untuk memproses data operasional organisasi. Implementasi sistem berbasis komputer ini berfungsi sebagai media penyedia informasi bagi pihak manajemen dan pengguna akhir. Struktur aplikasi tersebut mengintegrasikan modul pengelolaan inventaris, pencatatan transaksi, serta penyimpanan data identitas penyewa ke dalam satu jaringan yang sistematis. Oleh karena itu, kehadiran sistem informasi manajemen yang tertata dengan baik mampu meningkatkan efisiensi waktu pelayanan administrasi harian, meminimalkan potensi kelalaian pencatatan, serta menyediakan data pelaporan bisnis yang akurat bagi pemilik usaha.

Lahwani et al (2025) memandang sistem informasi sebagai sebuah solusi digitalisasi terintegrasi yang dirancang untuk menggantikan metode pendokumentasian manual yang rentan terhadap risiko kehilangan berkas. Sistem informasi ini menyediakan fitur-fitur fungsional yang saling berkaitan secara runtut, mulai dari pencatatan kondisi fisik aset, otomatisasi penjadwalan pemeliharaan berkala, hingga perekaman historis servis komponen. Keteraturan alur data administratif di dalam peladen server

memastikan proses pengawasan kondisi armada dapat berjalan dengan lebih transparan. Dengan adanya arsitektur sistem yang matang, platform digital mampu menekan biaya operasional yang tidak terduga akibat kerusakan mendadak serta membantu manajemen dalam mengambil keputusan berbasis keandalan data harian.

Felinta et al (2024) juga mengemukakan bahwa sistem informasi merupakan suatu kesatuan instrumen teknologi yang menyelaraskan interaksi aktivitas manusia dengan penggunaan basis data relasional guna mengoptimalkan proses bisnis. Wadah komunikasi digital ini dibangun dengan memanfaatkan bahasa pemrograman web dan kerangka kerja desain untuk memfasilitasi kebutuhan aksesibilitas informasi yang dinamis. Struktur pangkalan data pada sistem dirancang khusus untuk mengelola kapasitas data transaksi berskala besar serta menyajikan visualisasi analisis performa usaha. Pengintegrasian elemen-elemen teknologi informasi ini secara efektif mendukung kelancaran operasional harian perusahaan, mempercepat penyusunan berkas laporan tahunan, serta meningkatkan daya saing organisasi di era digital.

Berdasarkan beberapa kajian tersebut, dapat disimpulkan bahwa sistem informasi adalah kesatuan kerangka kerja berbasis teknologi internet yang mengintegrasikan aktivitas manusia, bahasa pemrograman, dan basis data terpusat untuk mengolah data mentah menjadi laporan fungsional secara otomatis. Sistem informasi memiliki alur pemrosesan data yang terstruktur guna menghasilkan output informasi yang transparan dan akurat.

Dalam konteks operasional bisnis persewaan kendaraan, sistem informasi berperan sebagai platform digital terpusat yang menyatukan manajemen inventaris armada, pemantauan jadwal servis berkala, validasi transaksi reservasi *online*, hingga otomatisasi pencatatan laporan kas keuangan.

3. Penyewaan Mobil (Rental)

Supriyadi et al (2024) menjelaskan bahwa usaha persewaan mobil merupakan sektor jasa transportasi yang menyediakan armada kendaraan roda empat untuk memenuhi kebutuhan mobilitas harian konsumen. Pengelolaan bisnis rental konvensional sering menghadapi kendala administratif akibat pencatatan jadwal pengembalian mobil yang masih menggunakan pembukuan kertas. Implementasi sistem aplikasi persewaan dirancang khusus untuk mengotomatisasi pencatatan durasi sewa, kalkulasi biaya keterlambatan secara otomatis, serta pemantauan status ketersediaan unit kendaraan. Oleh karena itu, digitalisasi manajemen persewaan mobil sangat diperlukan karena mampu meminimalkan risiko bentrok jadwal pemesanan antar-pelanggan, mendokumentasikan bukti transaksi secara aman, serta mempercepat proses pelayanan serah terima kunci kendaraan.

Palis et al (2024) menyatakan bahwa industri penyewaan mobil yang tumbuh dengan pesat menuntut ketersediaan platform berbasis web yang tanggap terhadap perubahan. Sistem manajemen transaksi sewa yang masih menggunakan catatan buku secara manual dapat menyulitkan pengolahan informasi karena tidak terstruktur dan sangat rawan mengalami kehilangan data. Melalui pengembangan *website* manajemen rental, proses bisnis dapat

diperkaya dengan layanan yang menyajikan informasi armada, harga, serta stok kendaraan secara daring (*online*). Penerapan sistem terintegrasi ini diyakini mampu mempercepat waktu layanan kepada pelanggan sekaligus memudahkan pemilik usaha dalam mengelola transaksi dan pelaporan secara optimal.

Raihan & Eviyanti (2022) juga mengemukakan bahwa persaingan dalam usaha rental mobil yang semakin ketat mengharuskan pemilik bisnis untuk memaksimalkan pemanfaatan teknologi informasi. Praktik pengolahan data yang masih menggunakan buku untuk merekap transaksi pelanggan maupun menyusun laporan harian sering kali menimbulkan kesalahan akibat faktor kelalaian manusia (*human error*), sehingga berisiko mengakibatkan kerusakan atau hilangnya data-data penting. Oleh sebab itu, perancangan sistem informasi penyewaan berbasis web menjadi solusi strategis untuk mempermudah pemilik dalam mengolah bisnisnya. Dengan adanya sistem terkomputerisasi, seluruh proses pemesanan dan transaksi pelanggan dapat dilakukan secara daring, sehingga kegiatan operasional pengelolaan penyewaan kendaraan menjadi jauh lebih efisien.

Berdasarkan beberapa kajian tersebut, dapat disimpulkan bahwa penyewaan mobil (rental) adalah kegiatan pelayanan jasa transportasi yang sangat membutuhkan platform digital untuk menggantikan pencatatan transaksi manual yang rentan terhadap risiko kesalahan (*human error*) dan kehilangan data. Sistem informasi penyewaan berfungsi sebagai instrumen terintegrasi yang menyajikan katalog armada secara daring, mempermudah

proses pemesanan kendaraan, serta mempercepat otomatisasi pelaporan bagi pemilik usaha. Dalam konteks penelitian ini, konsep operasional penyewaan tersebut akan diimplementasikan ke dalam pengembangan Sistem Informasi *Rent Car* berbasis *website* pada Bintang Travel Kabupaten Ponorogo guna menciptakan manajemen administrasi yang efisien, aman, dan akurat.

4. Berbasis *Website*

Pariela & Afifah (2026) mengemukakan bahwa ketatnya persaingan usaha sewa kendaraan saat ini menuntut para pelaku bisnis untuk memanfaatkan teknologi informasi, sebab masih banyak pemilik rental yang menggunakan media promosi konvensional seperti spanduk atau banner di lokasi usaha. Kurangnya pemanfaatan teknologi digital ini menyebabkan perusahaan rental mobil rentan terhadap risiko kehilangan kuitansi fisik sebagai bukti pembayaran, yang pada gilirannya menghambat kelancaran proses pelaporan dan pengambilan data transaksi. Di sisi lain, keterbatasan transparansi mengenai tarif sewa dan kepastian ketersediaan armada menciptakan tantangan tersendiri bagi konsumen dalam mengambil keputusan penyewaan. Kehadiran sistem informasi berbasis *website* dirancang untuk mempermudah calon penyewa mengakses informasi terperinci mengenai tarif dan stok mobil secara transparan dari berbagai lokasi, sekaligus mendukung administrator dalam melakukan pencatatan transaksi dan pengelolaan data secara aman.

Syauqi (2026) menjelaskan bahwa transformasi digital melalui sistem informasi berbasis web mampu menjadi solusi terintegrasi untuk menggantikan pengelolaan data operasional konvensional yang masih bersifat manual (buku atau *spreadsheet* terpisah). Sistem administrasi yang berjalan tanpa komputerisasi terpusat berpotensi menimbulkan kesalahan pencatatan, keterlambatan pelayanan, serta menyulitkan manajemen dalam memantau status kendaraan maupun menyusun laporan finansial. Implementasi sistem manajemen rental mobil berbasis *website* mempermudah kontrol penuh terhadap jalannya operasional bisnis, mulai dari validasi data pelanggan, perhitungan biaya otomatis berdasarkan durasi hari sewa, pelacakan pembayaran uang muka (DP) atau pelunasan, hingga penalti denda keterlambatan pengembalian armada. Integrasi pangkalan data (*database*) berbasis web ini memastikan seluruh riwayat penyewaan terekam secara *real-time*, sehingga mempercepat penyusunan laporan manajerial dan meningkatkan daya saing usaha.

Adam et al (2026) menegaskan bahwa penyediaan layanan rental mobil berbasis *website* merupakan langkah strategis untuk meningkatkan kualitas pelayanan dan kemudahan akses informasi bagi masyarakat. Jika pengelolaan data operasional armada dikerjakan secara terpisah atau belum terpusat secara digital, calon konsumen terpaksa harus menghubungi pihak pengelola atau datang langsung ke lokasi hanya demi memastikan ketersediaan kendaraan. Melalui sistem berbasis web, platform digital ini tidak sekadar berfungsi sebagai media pemesanan (*booking*) *online* dan

unggah dokumen persyaratan transaksi (seperti bukti transfer dan SIM A), melainkan juga dikembangkan dengan menyertakan fitur analitik performa bisnis. Integrasi modul laporan visual seperti grafik jumlah penyewaan tahunan serta visualisasi data armada mobil yang paling sering diminati memberikan nilai tambah bagi pemilik usaha untuk mengevaluasi kinerja operasional bisnis secara akurat.

Berdasarkan beberapa kajian tersebut, dapat disimpulkan bahwa sistem rental mobil berbasis *website* merupakan platform teknologi informasi terintegrasi yang berfungsi mengoptimalkan aksesibilitas, transparansi, dan efisiensi operasional usaha jasa transportasi. Pengadopsian sistem berbasis web ini krusial untuk menggantikan sistem pencatatan tradisional yang rentan terhadap risiko kehilangan data bukti transaksi, bentrok jadwal pemesanan, dan lambatnya penyusunan laporan operasional. Melalui integrasi basis data terpusat, sistem ini tidak hanya memudahkan konsumen dalam memantau ketersediaan armada, memeriksa tarif sewa, dan menyelesaikan transaksi secara daring, melainkan juga mempermudah administrator dalam mengelola alur status penyewaan kendaraan secara aman serta memfasilitasi pemilik usaha dengan fitur analitik manajerial demi mendukung pengambilan keputusan bisnis yang tepat.

5. Metode Pengembangan (*Waterfall*)

Anindita & Dewanto (2024) menerapkan metode *Waterfall* sebagai model pengembangan perangkat lunak. Model ini melibatkan lima tahapan utama. Lima tahapan tersebut meliputi tahap analisis kebutuhan, desain sistem, implementasi, pengujian program, pemeliharaan. Karakter metode ini memiliki sifat sistematis. Pengembang mengerjakan setiap tahapan secara berurutan. Pengembang tidak dapat memulai tahapan baru. Pengembang wajib menyelesaikan tahapan sebelumnya secara tuntas. Pengembang mengumpulkan data pada tahap analisis kebutuhan. Pengembang membuat rancangan aplikasi pada tahap desain sistem. Langkah ini mengarahkan proses penulisan kode program secara tepat. Pengembang melakukan pengujian program pada tahap akhir. Pengembang mengevaluasi sistem untuk menjaga fungsi aplikasi.

Siregar et al (2024) mendeskripsikan metode *Waterfall* sebagai pendekatan perbaikan pemrograman. Pendekatan ini menggambarkan pergantian fase secara lurus. Metode *Waterfall* memecah siklus hidup perangkat lunak menjadi beberapa fase spesifik. Setiap fase memiliki serangkaian tugas tersendiri. Setiap fase juga mempunyai sasaran spesifik. Seluruh fase merepresentasikan proses pengembangan perangkat lunak. Pengembang membawa hasil fase sebelumnya menuju fase selanjutnya. Pengembang memperoleh alur kerja secara terstruktur. Pengembang mudah mengikuti urutan langkah pengembangan aplikasi. Pemilihan metode ini mendukung pembuatan sistem penyewaan secara rapi.

Dinata & Nugraha (2026) menggunakan model *Waterfall* dalam pembuatan program. Model pengembangan perangkat lunak ini memiliki struktur rapi. Sistem ini menyimpan dokumentasi pengembangan secara baik. Pengembang mencatat setiap langkah secara berurutan. Metode ini menyederhanakan perancangan sistem pemesanan berbasis web. Peneliti menggunakan metode ini. Peneliti sudah mengidentifikasi kebutuhan dasar sistem sejak awal proyek. Penerapan *Waterfall* meminimalkan risiko revisi berskala besar. Pengembang mencegah perubahan mendadak pada pertengahan proyek. Pengembang menjamin kesesuaian kebutuhan pengguna terhadap sistem.

Berdasarkan beberapa kajian tersebut, metode *Waterfall* sebagai model rekayasa perangkat lunak sekuensial linier. Metode ini menuntut penyelesaian fase pengembangan secara tuntas. Keunggulan metode ini mencakup kemudahan pemahaman alur kerja. Metode ini menekan risiko perubahan proyek secara tiba-tiba. Penulis menggunakan metode *Waterfall* sebagai kerangka kerja. Penulis membagi proses ke dalam serangkaian fase berurutan. Kerangka kerja ini membentuk Sistem Informasi *Rent Car*. Sistem ini beroperasi pada Bintang Travel Kabupaten Ponorogo. Penulis menargetkan hasil perangkat lunak fungsional.

6. Pustaka React (React)

Steven & Mawardi (2024) menerapkan *React Javascript* sebagai perpustakaan pengembangan antarmuka pengguna. Perpustakaan ini berisi kumpulan kode *Javascript*. Pengembang menggunakan teknologi ini untuk

menciptakan tampilan *website* interaktif. React Javascript meningkatkan kecepatan akses halaman aplikasi. Teknologi ini mengadopsi model arsitektur *Model-View-Controller*. *React Javascript* berfokus pada lapisan tampilan *website*. Ekosistem React menyediakan berbagai paket bantuan tambahan. Paket tersebut mempermudah pengembang dalam merancang aplikasi *website* sewa kantor. Pengembang merancang modul pencarian dinamis menggunakan teknologi ini. Sistem menampilkan riwayat transaksi pelanggan secara cepat. Pengguna mengakses fitur pembayaran uang muka secara mudah.

Hakim et al (2025) menggunakan *React* untuk membangun komponen antarmuka responsif. Pustaka Javascript sumber terbuka ini berfokus pada pengembangan tampilan aplikasi. Jordan Walke mengembangkan teknologi ini pertama kali pada tahun 2013. Pengembang memanfaatkan React dalam pembuatan bagian depan aplikasi. Teknologi ini mengelola lapisan tampilan untuk perangkat komputer maupun telepon pintar. *React* menawarkan arsitektur berbasis komponen. Pengembang memecah antarmuka *website* menjadi beberapa elemen kecil. Pengembang menggunakan kembali komponen tersebut pada berbagai halaman berbeda. Pendekatan ini menghilangkan kebutuhan penulisan kode berulang. Pengembang menghemat waktu pengerjaan proyek pembuatan aplikasi. Tim pengembang menjaga konsistensi desain visual *website* secara mudah. Kualitas antarmuka pengguna mengalami peningkatan signifikan. Pengguna merasakan pengalaman navigasi secara nyaman.

Naresvari & Susetyo (2025) mengimplementasikan *React* pada pengembangan sistem *e-commerce*. *Framework* ini membagi seluruh aplikasi menjadi bagian-bagian kecil. Potongan kode tersebut mempermudah pengembang melakukan perbaikan kesalahan. *React* mendukung penerapan konsep *Single Page Application*. Sistem memperbarui komponen visual halaman web tanpa memuat ulang keseluruhan struktur halaman. Teknologi ini memanfaatkan fitur navigasi tingkat lanjut. Pengguna berpindah halaman secara halus. Kinerja aplikasi mengalami peningkatan performa. Pengembang mengurangi biaya infrastruktur server. Sistem menyajikan alur pemesanan produk secara lancar. Pelanggan mengoperasikan menu aplikasi tanpa hambatan teknis. Teknologi ini mendukung tujuan digitalisasi usaha penyewaan kendaraan.

Berdasarkan beberapa kajian tersebut, *React* sebagai pustaka *Javascript* terkemuka pada lapisan antarmuka. Pustaka ini mengadopsi arsitektur berbasis komponen independen. Teknologi ini menghasilkan tampilan *website* interaktif. *React* mempercepat waktu pemuatan halaman situs. Pengembang menggunakan kembali elemen visual pada berbagai halaman berbeda. Tim pengembang memelihara struktur kode secara mudah. Penelitian ini mengimplementasikan *React* dalam pembuatan Sistem Informasi *Rent Car* Bintang Travel Kabupaten Ponorogo. Sistem menyajikan katalog mobil secara responsif. Pelanggan menikmati proses pemesanan daring secara instan.

7. PHP dan Framework Laravel

Saefudin et al (2023) menjelaskan bahwa PHP merupakan bahasa pemrograman skrip di sisi server (*server-side scripting*) yang dirancang secara dinamis untuk mengolah data dan menghasilkan aplikasi berbasis internet. Dalam praktik rekayasa teknologi modern, pengerjaan logika sistem sering kali dikombinasikan dengan kerangka kerja (*framework*) tingkat lanjut seperti Laravel untuk menata arsitektur penulisan kode program agar lebih teratur, rapi, dan mudah dipelihara. Implementasi *framework* Laravel terbukti mampu meningkatkan kinerja organisasi secara signifikan serta membantu meminimalkan kendala fragmentasi berkas data. Melalui penyediaan struktur komponen *backend* yang matang, kerangka kerja ini sangat andal digunakan untuk mengotomatisasi pengolahan informasi administratif yang kompleks serta mendukung penyajian laporan analisis internal instansi secara sistematis.

Aipina & Witriyono (2022) menyatakan bahwa pemanfaatan *framework* Laravel dalam pembangunan aplikasi berbasis web memberikan efisiensi yang tinggi bagi pengembang karena menyediakan standarisasi komponen pemrograman yang siap pakai. Laravel mempermudah pengelolaan basis data relasional melalui kueri yang aman serta memiliki ekosistem yang stabil untuk menangani alur transaksi operasional secara daring (*online*). Di samping itu, penggunaan *framework* ini memisahkan secara jelas antara struktur kendali data dengan lapisan tampilan antarmuka, sehingga proses pengembangan sistem dapat berjalan secara modular dan

bertahap. Penataan sistem yang teratur di balik layar peladen ini memastikan bahwa seluruh riwayat transaksi terekam dengan baik ke dalam *database* server, sehingga mempermudah administrator dalam melakukan pencatatan data secara transparan tanpa risiko manipulasi informasi.

Alfarisi et al (2023) mengemukakan bahwa penerapan *framework* Laravel sangat krusial untuk mengatasi keterlambatan pelayanan serta kesulitan dalam pembuatan laporan berkala akibat manajemen data yang masih bersifat manual. Laravel secara konsisten mengadopsi pola arsitektur *Model-View-Controller* (MVC) yang membagi fungsionalitas program menjadi tiga bagian inti, yaitu *Model* untuk interaksi *database*, *View* untuk merender visual antarmuka, dan *Controller* untuk memproses seluruh logika bisnis. Selain menyajikan struktur kode yang kokoh, Laravel juga dilengkapi dengan sistem keamanan data mutakhir bawaan untuk melindungi integritas data dari ancaman eksternal. Pengintegrasian kerangka kerja ini memastikan platform digital yang dibangun memiliki skalabilitas tinggi, berjalan dengan performa stabil, serta mempermudah proses validasi input data operasional sebelum disimpan secara permanen.

Berdasarkan beberapa kajian tersebut, dapat disimpulkan bahwa PHP adalah bahasa skrip peladen dinamis dan Laravel merupakan kerangka kerja ekstensinya yang mengimplementasikan arsitektur MVC (*Model-View-Controller*) untuk menghasilkan standar pengkodean yang aman, rapi, dan mudah dirawat. Kombinasi teknologi ini berfungsi sebagai penggerak utama di sisi *backend* untuk mengotomatisasi logika bisnis yang rumit,

menjaga keamanan basis data, serta mempercepat pemrosesan data operasional harian. Dalam konteks penelitian ini, PHP dan *framework* Laravel digunakan sebagai fondasi *backend* Sistem Informasi *Rent Car* Bintang Travel Kabupaten Ponorogo untuk memproses data transaksi pemesanan sewa mobil, mengamankan pertukaran data, memvalidasi status ketersediaan armada secara otomatis, serta menyusun laporan kas masuk dan keluar secara instan dan akurat.

8. Sistem Manajemen Basis Data (PostgreSQL)

Prasetyo & Susetyo (2022) menjelaskan bahwa perkembangan teknologi penyimpanan data menuntut sistem informasi untuk mengadopsi infrastruktur basis data yang tangguh, aman, dan berbasis relasi yang kuat. PostgreSQL merepresentasikan salah satu perangkat lunak *Relational Database Management System* (RDBMS) berbasis *open-source* yang mematuhi standar *Structured Query Language* (SQL) secara ketat dalam setiap pemrosesan operasinya. Salah satu fitur unggulan dari sistem *database* ini adalah fleksibilitasnya dalam mengakses komponen *information schema* secara otomatis untuk mengekstrak metadata tabel yang meliputi detail nama kolom, tipe data, hingga batasan (*constraint*). Kemudahan akses kueri terhadap skema informasi ini memberikan efisiensi yang sangat tinggi bagi para pengembang perangkat lunak dalam memetakan struktur data serta mempercepat proses pemeliharaan fungsional skema basis data dalam jangka waktu yang panjang.

Badolo et al (2025) menyatakan bahwa integrasi PostgreSQL sebagai sistem basis data di sisi belakang (*backend*) terbukti sangat efektif untuk mendukung pengelolaan data operasional organisasi secara andal, transparan, dan *real-time*. Pangkalan data relasional yang kokoh sangat diperlukan oleh aplikasi dinamis untuk menampung volume data operasional yang besar serta mengontrol hak akses pengguna melalui dasbor interaktif yang terpusat. Penerapan teknologi penyimpanan terstruktur menggunakan RDBMS ini mampu mencegah terjadinya risiko tumpang tindih data operasional pada platform digital secara signifikan. Tata kelola basis data yang dikonfigurasi dengan baik di dalam server peladen akan memberikan jaminan perlindungan data yang kuat, mengoptimalkan proses sinkronisasi informasi, serta memberikan fondasi utama bagi pihak manajemen dalam memantau ringkasan data harian secara aman.

Aziz et al (2020) juga mengemukakan bahwa jaminan keamanan dan keselamatan jalannya aktivitas operasional pada suatu instansi sangat bergantung pada keandalan sistem manajemen pangkalan data yang digunakan untuk merekam riwayat kondisi aset secara berkala. PostgreSQL memiliki keunggulan performa yang sangat tinggi dalam menangani kapasitas lalu lintas data yang padat serta mampu menjaga integritas relasi antar-tabel secara optimal ketika terjadi banyak aktivitas transaksi yang diinput secara bersamaan (*concurrency*). Sistem pengolahan *database* yang terkomputerisasi ini secara efektif meminimalkan potensi terjadinya kesalahan penginputan, keterlambatan penyajian data, serta kerusakan

berkas arsip fisik. Melalui pemanfaatan arsitektur basis data relasional yang stabil di sisi server, platform digital mampu menyajikan pelaporan status pemantauan yang valid bagi administrator guna mendukung kelancaran kegiatan operasional secara berkelanjutan.

Berdasarkan beberapa kajian tersebut, dapat disimpulkan bahwa PostgreSQL adalah sistem manajemen basis data relasional (RDBMS) berbasis *open-source* berstandar SQL yang memiliki performa tinggi dalam menangani kapasitas data skala besar, menjaga integritas relasi antar-tabel, serta memfasilitasi pengelolaan metadata melalui *information schema*. PostgreSQL berfungsi sebagai media penyimpanan data operasional terpusat untuk menjamin keamanan informasi dan mencegah anomali data saat transaksi berlangsung secara bersamaan. Dalam konteks penelitian ini, PostgreSQL diimplementasikan sebagai fondasi pangkalan data utama pada Sistem Informasi *Rent Car* Bintang Travel Kabupaten Ponorogo untuk menyusun skema tabel yang saling berelasi secara aman, mulai dari penyimpanan master data pengguna, detail armada mobil, jadwal supir, status reservasi *online*, hingga pencatatan riwayat laporan kas keuangan.

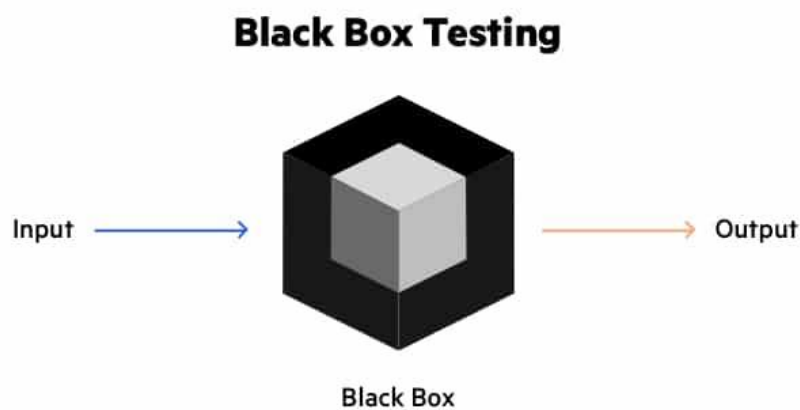
9. Pengujian Perangkat Lunak (*Black-Box Testing*)

Putra et al (2023) menjelaskan bahwa pengujian perangkat lunak merupakan tahapan krusial yang bertujuan untuk mengidentifikasi kekeliruan, meminimalkan potensi kerugian teknis, serta mengeksekusi kegagalan sistem agar dapat segera dilakukan perbaikan sebelum aplikasi dirilis secara resmi. Dalam penerapannya, pengujian ini terbagi menjadi dua

pendekatan utama, yaitu *Black-Box Testing* dan *White-Box Testing*. *Black-Box Testing* berfokus sepenuhnya pada pengujian fungsionalitas antarmuka (*interface*) berdasarkan spesifikasi klien tanpa memberikan akses ke kode program, sementara *White-Box Testing* atau pengujian terstruktur menguji jalur logika internal dengan memanfaatkan basis data kode peladen. Melalui pemodelan grafik alir (*flowgraph*) dan perhitungan kompleksitas siklomatis (*Cyclomatic Complexity*), pengembang dapat memvalidasi fungsionalitas menu pengelolaan data sewa, halaman data pengguna, hingga formulir pengelolaan merek mobil guna memastikan seluruh sistem rental berjalan semestinya.

Jeniarta et al (2024) menegaskan bahwa uji sistem menggunakan metode *Black-Box* bekerja dengan menguji respons perangkat lunak tanpa memedulikan detail implementasi internal untuk memastikan program beroperasi sesuai dengan skenario rencana. Integrasi antara pengujian *Black-Box* pada antarmuka autentikasi *login* serta halaman dasbor sangat efektif untuk memvalidasi performa algoritma pendukung keputusan sistem. Di sisi lain, pengujian *White-Box* dengan teknik *Basis Path* digunakan untuk memetakan diagram alir keputusan pengguna saat menginputkan kriteria kendaraan. Dengan menghitung kompleksitas regional melalui rumus tertentu, pengembang dapat mendefinisikan kasus uji (*test case*) pada jalur independen secara presisi, sehingga hasil rekomendasi yang dikeluarkan oleh mesin komputasi terbukti valid dan siap digunakan secara luas oleh masyarakat.

Secara konseptual, metode pengujian *Black-Box* bekerja dengan mengevaluasi kinerja sistem murni berdasarkan fungsionalitasnya, di mana pengujian dilakukan hanya dengan memberikan data masukan (*input*) dan mengamati hasil keluaran (*output*) tanpa menganalisis kode atau struktur internalnya. Ilustrasi dari konsep pengujian ini dapat dilihat pada Gambar 2.1 berikut:



Gambar 2.1 Ilustrasi Konsep Black Box Testing

Sumber : Zahra et al (2025)

Ayyasy & Voutama (2026) turut mengemukakan bahwa implementasi teknologi web dalam mentransformasi sektor penyewaan armada sangat menuntut jaminan kualitas fungsional yang optimal. Pendekatan *Black-Box Testing* menjadi metode yang sangat relevan untuk mengevaluasi kesesuaian output terhadap variasi data masukan dari sudut pandang pengguna akhir secara langsung. Pengujian terstruktur ini idealnya mencakup seluruh alur fitur krusial, mulai dari validasi format pendaftaran akun, autentikasi hak akses masuk, pembatalan pemesanan daring, hingga pengelolaan status

inventaris armada oleh administrator. Melalui dokumentasi rekapitulasi skenario uji yang ketat, ketepatan respons sistem terhadap input yang tidak valid dapat diukur dengan jelas, yang pada akhirnya berkontribusi signifikan dalam meningkatkan aspek keamanan data serta keandalan performa operasional aplikasi dalam jangka panjang.

Berdasarkan beberapa kajian tersebut, dapat disimpulkan bahwa *Black-Box Testing* adalah metode pengujian fungsional perangkat lunak yang berfokus pada evaluasi hubungan antara data masukan (*input*) dan keluaran (*output*) secara terstruktur tanpa menganalisis struktur internal kode program. Pengujian *Black-Box* berfungsi untuk mendeteksi kesalahan validasi formulir, menjaga kualitas integritas data pada basis data, serta memastikan aplikasi aman dari akses tidak sah. Dalam konteks penelitian ini, metode *Black-Box Testing* diimplementasikan untuk menguji fungsionalitas fitur autentikasi *login* pengguna, form pemesanan armada, serta modul manajemen pelaporan kas pada sistem informasi manajemen *rent car* Bintang Travel Ponorogo agar dapat beroperasi sesuai dengan spesifikasi yang diharapkan dan bebas dari galat fungsional.

10. Data Flow Diagram (DFD)

Rahmadan & Gunawan (2024) mendefinisikan *Data Flow Diagram* sebagai gambaran visual pergerakan data di dalam suatu sistem informasi. Pemodelan ini menggunakan simbol-simbol khusus untuk mempresentasikan alur informasi. Simbol tersebut memperlihatkan secara detail cara data masuk ke dalam sistem. Sistem memproses data tersebut

secara internal melalui serangkaian prosedur bisnis. Sistem menyimpan data olahan ke dalam pangkalan data secara permanen. Sistem mengeluarkan data tersebut sebagai informasi hasil bagi pihak manajemen. Alat visual ini menyajikan informasi alur data secara mudah dipahami. Pengembang merancang *Data Flow Diagram* secara tepat untuk memetakan kebutuhan sistem. Implementasi diagram ini menentukan keberhasilan dalam pembuatan aplikasi.

Muliadi et al (2020) menggunakan *Data Flow Diagram* untuk merancang aliran proses sistem informasi. Bagan ini menggambarkan arus data dalam suatu operasional perusahaan. Pengembang menggunakan sejumlah simbol standar untuk merepresentasikan perpindahan informasi. Simbol ini menunjukkan perpindahan data pada setiap tahapan proses bisnis. Perancangan aliran data bermula dari pembuatan Diagram Konteks. Diagram Konteks memperlihatkan interaksi sistem informasi terhadap lingkungan luar organisasi. Diagram ini selalu mengandung satu proses utama dengan nomor identitas nol. Proses utama tersebut menggambarkan hubungan masukan dan keluaran sistem secara menyeluruh. Pengembang memecah diagram konteks menuju tingkatan proses selanjutnya secara bertahap. Tingkatan lanjutan ini memperlihatkan alur informasi pemesanan secara lebih spesifik.

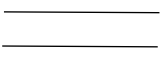
Zawawi et al (2026) menjadikan *Data Flow Diagram* sebagai instrumen gambaran sistem secara komprehensif. Pemodelan ini menjelaskan alur pengolahan data secara logis dan terstruktur. Sistem

menampilkan pergerakan data dari entitas sumber yang relevan. Data tersebut melewati tahapan proses penyimpanan dalam basis data server. Sistem menghasilkan keluaran informasi pada akhir rangkaian proses. Pengembang menempatkan entitas utama sebagai penyedia data transaksi harian. Admin bertindak sebagai penerima laporan informasi sistem. Diagram tingkat dasar memuat alur pencatatan barang masuk secara kronologis. Pemodelan ini memfasilitasi kebutuhan perancangan arsitektur aplikasi berbasis *website*. Pengembang mencegah risiko kehilangan data operasional melalui pemetaan ini.

Berdasarkan beberapa kajian tersebut, *Data Flow Diagram* terbukti sebagai alat pemodelan visual terstruktur yang krusial. Diagram ini memetakan pergerakan informasi secara logis dan sistematis. Pengembang mengawali proses perancangan melalui pembuatan Diagram Konteks. Sistem memecah proses utama menjadi beberapa tingkatan spesifik untuk detail teknis. Penggunaan simbol standar mempermudah komunikasi alur kerja perangkat lunak kepada pemangku kepentingan. Peneliti menerapkan *Data Flow Diagram* dalam pengembangan Sistem Informasi *Rent Car* Bintang Travel Kabupaten Ponorogo. Pemodelan ini menguraikan interaksi pelanggan terhadap fitur pemesanan kendaraan daring. Administrator memahami siklus pengolahan data transaksi secara transparan melalui diagram ini.

Simbol-simbol standar yang umum digunakan dalam DFD untuk merancang alur sistem perangkat lunak dapat dilihat pada Tabel 2.1 berikut ini:

Tabel 2.1 Simbol-Simbol DFD

No.	Bentuk Simbol	Nama Simbol	Fungsi Simbol
1.		<i>External Entity</i> (Persegi Panjang)	Menggambarkan entitas luar yang berinteraksi dengan sistem
2.		<i>Process</i> (Lingkaran)	Menggambarkan proses pengolahan data di dalam sistem
3.		<i>Data Flow</i> (Anak Panah)	Menunjukkan arah aliran data antar komponen
4.		<i>Data Store</i> (Dua Garis Sejajar)	Menggambarkan tempat penyimpanan data

11. *Entity Relationship Diagram (ERD)*

Ardila & Pascya (2026) menjelaskan bahwa *Entity Relationship Diagram (ERD)* merupakan sebuah model data konseptual yang digunakan untuk merepresentasikan struktur logis dari basis data secara visual. Perancangan ERD memegang peranan yang sangat vital dalam pengembangan sistem informasi karena berfungsi untuk memetakan objek data, menentukan atribut penjabar, serta mengatur hubungan (*relationship*) antar-entitas secara terstruktur. Melalui tahapan identifikasi yang matang, model konseptual ini mampu mengurangi redundansi data, menekan risiko terjadinya duplikasi informasi, serta meningkatkan keakuratan berkas operasional perusahaan secara signifikan. Oleh karena itu, kehadiran ERD

dinilai sangat krusial bagi tim pengembang karena bertindak sebagai dasar atau cetak biru (*blueprint*) utama dalam merancang skema tabel basis data relasional yang terintegrasi sebelum proses pembuatan kode program *backend* dimulai.

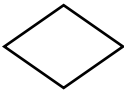
Badin et al (2025) menyatakan bahwa struktur pangkalan data yang dirancang menggunakan ERD harus mampu memfasilitasi kebutuhan transaksi operasional secara daring (*online*) untuk memberikan kenyamanan bagi pengguna. Model data konseptual ini tersusun atas beberapa komponen inti yang saling berkaitan, seperti entitas yang merepresentasikan objek nyata di dunia bisnis, atribut yang mencatat kolom informasi spesifik, serta kata penghubung yang menerangkan interaksi logis antar-tabel. Pengolahan pangkalan data relasional yang terkomputerisasi secara rapi melalui ERD bertujuan untuk memastikan bahwa seluruh data master dapat diakses kembali secara efisien oleh berbagai modul fungsional aplikasi. Dengan adanya keteraturan visual ini, pengelola sistem dapat melacak riwayat pemesanan dan memvalidasi laporan administratif secara langsung tanpa menghadapi kendala teknis akibat hilangnya berkas pembukuan fisik.

Bachtiar et al (2022) juga mengemukakan bahwa perbaikan tata kelola operasional pada usaha menengah sangat bergantung pada kekuatan pondasi ERD yang digunakan sebagai dasar perancangan antarmuka (*UI/UX design*). Model grafis ERD dilengkapi dengan aturan kardinalitas yang menjelaskan batasan jumlah hubungan antar-entitas secara integral melalui dua konsep hubungan utama, yaitu satu-ke-banyak (*one-to-many*) dan banyak-ke-satu

(*many-to-one*). Penandatanganan relasi antar-basis data tersebut dihubungkan dengan memanfaatkan arsitektur kunci tamu (*foreign key*) guna menjamin aspek integritas referensial data secara menyeluruh di dalam server peladen. Penerapan rancangan *database* relasional yang kokoh ini secara efektif meminimalkan potensi terjadinya kesalahan penginputan transaksi, mempercepat proses pencarian data operasional, serta menghasilkan nilai efisiensi yang tinggi bagi kelangsungan bisnis organisasi di era digital.

Berdasarkan beberapa kajian tersebut, dapat disimpulkan bahwa *Entity Relationship Diagram* (ERD) adalah alat pemodelan data relasional berupa diagram grafis konseptual yang memetakan objek entitas, komponen atribut, dan batasan tipe hubungan kardinalitas (*one-to-many*) menggunakan sistem *foreign key*. ERD berfungsi sebagai fondasi utama perancangan arsitektur *database* terpusat untuk mencegah anomali data serta menjadi acuan dalam penyusunan desain antarmuka aplikasi. Dalam konteks penelitian ini, ERD digunakan untuk merancang skema basis data terintegrasi pada Sistem Informasi *Rent Car* Bintang Travel Kabupaten Ponorogo agar penyimpanan data master pengguna (*users*), spesifikasi mobil (*Cars*), penugasan supir (*drivers*), status reservasi pelanggan, hingga detail mutasi kas masuk dan keluar dapat terekam secara aman, konsisten, dan saling terhubung.

Tabel 2.2 Definisi ERD

No.	Bentuk Simbol	Nama Simbol	Fungsi Simbol
1.		<i>Persegi panjang</i>	Objek atau data yang akan disimpan dalam <i>database</i>
2.		<i>atribut</i>	Karakteristik atau informasi yang dimiliki oleh suatu entitas
3.		<i>relasi</i>	Hubungan antara dua atau lebih entitas dalam <i>database</i>
4.		<i>garis</i>	Menunjukkan hubungan atau keterkaitan antar entitas dalam ERD

12. Flowchart

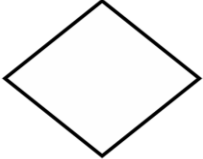
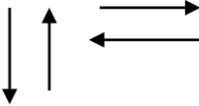
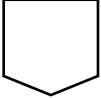
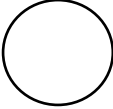
Fauzi (2020) menjelaskan bahwa diagram alir (*flowchart*) merupakan representasi gambar atau bagan yang memperlihatkan urutan langkah-langkah logis dari suatu program komputer serta hubungan antar-proses beserta pernyataannya. Tujuan utama dari perancangan *flowchart* adalah untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, terurai, rapi, dan jelas dengan menggunakan simbol-simbol yang standar. Di dalam rekayasa perangkat lunak, diagram alir ini bertindak sebagai gambar dari penjelasan atau deskripsi dari sebuah algoritma berpikir. Oleh karena itu, pemahaman tentang *flowchart* dinilai sangat penting bagi para pengembang karena mempermudah logika manusia untuk menganalisis berbagai pilihan alternatif cara kerja serta mempermudah implementasi alur data tersebut ke dalam bahasa pemrograman.

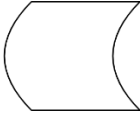
Khesya (2021) menyatakan bahwa diagram alir (*flowchart*) memegang peranan penting untuk menunjukkan alur program secara realistis, logis, dan berurutan untuk memecahkan suatu permasalahan. Alur logika tersebut dijelaskan menggunakan gambar bersimbol khusus untuk mempermudah pembacaan sistem jika dibandingkan dengan penulisan teks deskriptif konvensional. Di dalam penerapannya pada teknologi informasi, *flowchart* digunakan untuk mendefinisikan batas operasional aplikasi, melacak masukan (*input*) formulir, hingga mengontrol keluaran (*output*) informasi agar tidak menyimpang dari skenario bisnis yang diharapkan. Penataan alur algoritma secara visual ini memastikan seluruh tim pengembang memiliki acuan logika berpikir yang sama sehingga meminimalkan ambiguitas atau adanya jalur proses komputasi yang terlewat saat aplikasi dibangun.

Noija et al (2023) juga mengemukakan bahwa perbaikan efisiensi kinerja operasional dan transparansi laporan pada suatu instansi bisnis sangat bergantung pada kejelasan bagan *flowchart* yang dirancang oleh analis sistem. Struktur pembuatan *flowchart* yang profesional wajib mengikuti panduan standar internasional agar alur logika dapat dibaca dengan mudah, yaitu dibuat mengalir secara konsisten mengikuti urutan arah dari bagian atas ke bawah dan dari kiri ke kanan. Jalur aktivitas juga harus dimulai (*start*) dan diakhiri (*stop*) secara tegas menggunakan simbol terminal yang baku. Setiap keputusan logis di dalam sistem juga perlu dikontrol secara presisi menggunakan simbol percabangan kondisi untuk

mendukung standardisasi perancangan alur penelitian dan algoritma program

Tabel 2.3 Simbol *Flowchart*

No.	Bentuk Simbol	Nama Simbol	Fungsi Simbol
1.		<i>Terminator Symbol</i>	Menunjukkan titik awal (<i>start</i>) atau titik akhir (<i>stop</i>) dari suatu proses.
2.		<i>Input-Output Symbol</i>	Menunjukkan proses memasukkan data (<i>input</i>) atau menghasilkan informasi (<i>output</i>).
3.		<i>Processing Symbol</i>	Menunjukkan proses pengolahan data atau aktivitas operasional.
4.		<i>Decision Symbol</i>	Menunjukkan titik pengambilan keputusan yang menghasilkan percabangan (contoh: Ya/Tidak).
5.		<i>Flow</i>	Menyatakan jalannya suatu proses atau alur pada sistem
6		<i>cetak</i>	Menunjukkan output berupa dokumen yang dicetak atau dihasilkan sistem
7		<i>Office connector</i>	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda
8		<i>connector</i>	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama
9		<i>Predefined process</i>	Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal

10		<i>Storage data</i>	Menyatakan input berasal dari disk atau output disimpan ke disk
----	---	---------------------	---

B. Kajian Empiris

Penelitian pertama dilakukan oleh Pangestu et al (2024) yang menggunakan model pengembangan *Waterfall* dalam merancang sistem informasi rental mobil. Hasil penelitian tersebut menunjukkan bahwa penggunaan digitalisasi melalui *website* sangat efektif dalam memudahkan proses pemesanan kendaraan. Sistem ini terbukti dapat mengelola data reservasi dan transaksi secara *online*, sehingga mengurangi kesalahan dalam pencatatan manual dan meningkatkan efisiensi operasional perusahaan secara keseluruhan. Perbedaan penelitian tersebut dengan penelitian ini terletak pada fungsi yang cakupannya lebih luas, penelitian terdahulu hanya berfokus pada digitalisasi transaksi sewa, sedangkan penelitian ini menambahkan modul untuk mengelola perawatan armada serta otomatisasi laporan keuangan yang terhubung langsung dengan proses pemesanan.

Penelitian selanjutnya dilakukan oleh Ramadhan et al (2025), yang berfokus pada pengembangan *website* reservasi *online* yang terintegrasi dengan *API Bot* Telegram sebagai notifikasi *real-time* menggunakan metode *Waterfall*. Hasil penelitian menunjukkan bahwa implementasi *bot* Telegram sangat efektif dan andal untuk memberikan umpan balik informasi secara instan kepada pihak manajemen, sehingga meminimalisir keterlambatan respons operasional. Perbedaan penelitian tersebut dengan penelitian ini terletak pada objek implementasi dan tujuan notifikasinya, penelitian terdahulu

berfokus pada pengelolaan reservasi restoran, sedangkan penelitian ini mengadaptasi keandalan integrasi *API Bot* Telegram tersebut secara khusus untuk mengotomatisasi sistem pengingat (*reminder*) jadwal perawatan armada kendaraan kepada administrator Bintang Travel.

Penelitian berikutnya oleh Pradana & Supriyono (2026) mengembangkan sebuah aplikasi manajemen sewa mobil yang lengkap menggunakan metode *Software Development Life Cycle (SDLC)* berbasis *Waterfall*. Hasil penelitian menunjukkan bahwa sistem yang dibangun mampu mengatur jadwal transaksi sewa dan mengotomatisasi pencatatan keuangan secara terpusat serta efisien. Perbedaan penelitian tersebut dengan penelitian ini terletak pada teknologi arsitektur pengembangan yang digunakan; penelitian ini menerapkan arsitektur *decoupled* yang memisahkan antarmuka (*Frontend* React) dan logika server (*Backend* Laravel), serta memakai basis data Postgresql untuk menjaga skalabilitas data dibandingkan arsitektur monolitik konvensional.

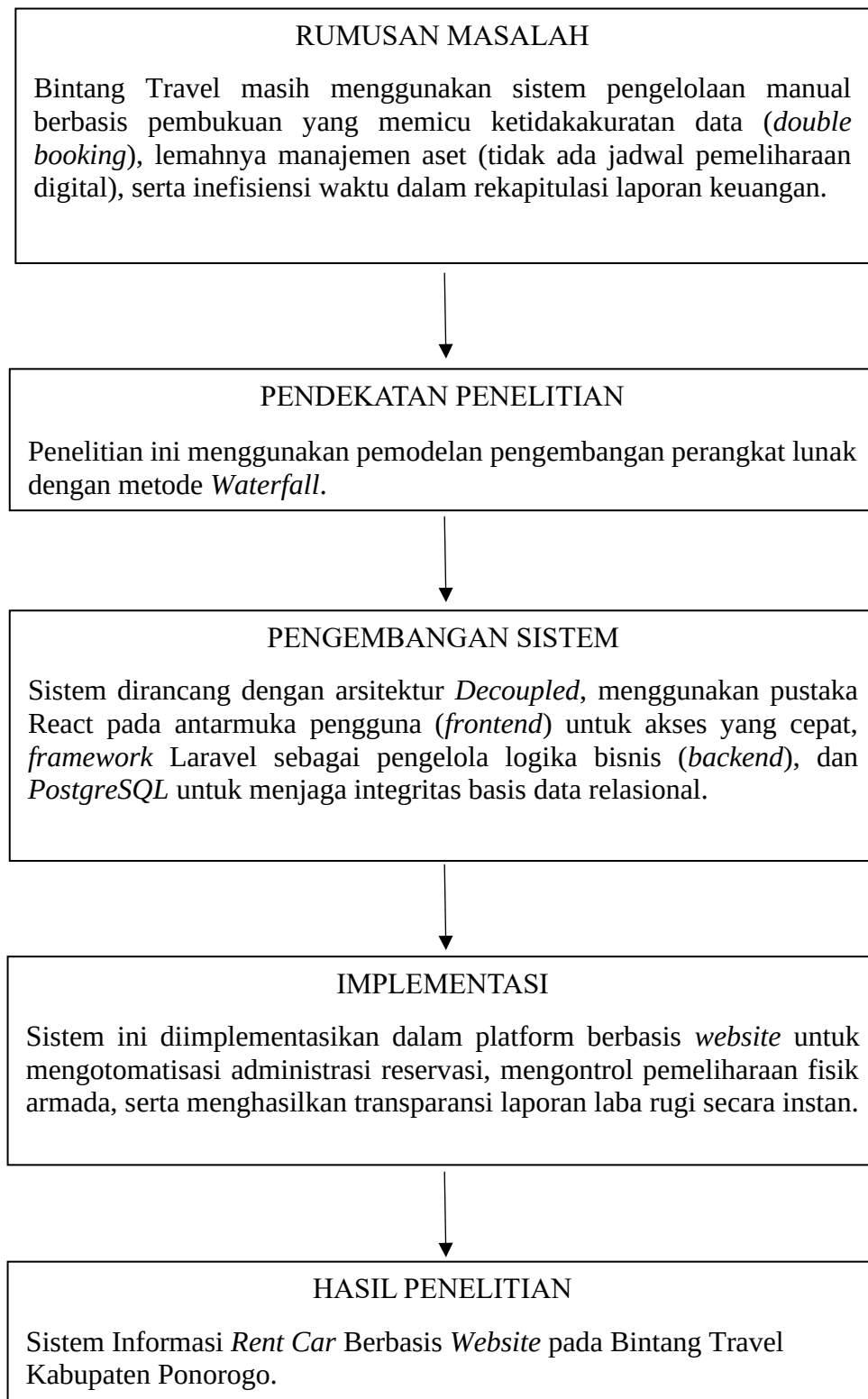
Penelitian selanjutnya dilakukan oleh Bilal & Hanif (2025) yang merancang dan membangun aplikasi penyewaan kendaraan berbasis *web* di PT. Duta Rental Mobil menggunakan metode *Waterfall*. Hasil penelitian tersebut menegaskan bahwa sistem yang terkomputerisasi mampu meningkatkan efisiensi operasional dan meminimalisir kesalahan pencatatan data yang sebelumnya dilakukan secara manual. Perbedaan penelitian tersebut dengan penelitian ini terletak pada pemilihan teknologi dan fitur pendukung; penelitian terdahulu menggunakan *stack* PHP dan MySQL konvensional, sedangkan penelitian ini menerapkan arsitektur *decoupled* yang lebih modern dengan

React sebagai *frontend*, *Laravel* sebagai *backend*, serta integrasi notifikasi *Telegram* untuk pengingat servis otomatis.

Penelitian terbaru oleh Steven & Mawardi (2024) menggunakan pustaka React dalam pengembangan antarmuka pengguna (*Frontend Development*). Hasil penelitian menunjukkan bahwa pemanfaatan React dapat mempercepat waktu pemuatan (*rendering*) komponen halaman secara signifikan. Teknologi ini menghasilkan kualitas antarmuka (*user interface*) yang interaktif dan memberikan pengalaman navigasi yang nyaman bagi pengguna. Perbedaan penelitian tersebut dengan penelitian ini terletak pada karakteristik objek yang disewakan penelitian terdahulu berfokus pada penyewaan properti statis, sementara penelitian ini mengadaptasi ketangguhan React untuk menangani sistem penyewaan aset bergerak, seperti mobil, yang memerlukan pembaruan status ketersediaan secara *real-time*.

Berdasarkan tinjauan terhadap beberapa penelitian terdahulu, penerapan teknologi informasi di bidang penyewaan terbukti berhasil mengatasi berbagai kendala administratif. Meskipun penelitian sebelumnya telah membahas fitur seperti modul rental, penjadwalan, dan pengingat servis secara terpisah, masih terdapat celah penelitian untuk mengintegrasikan seluruh fitur tersebut ke dalam satu sistem yang utuh. Penelitian ini hadir untuk memadukan keunggulan dari berbagai referensi ke dalam desain arsitektur *decoupled* yang modern, guna menciptakan Sistem Informasi *Rent Car* yang dirancang spesifik untuk menyelesaikan kompleksitas operasional di Bintang Travel Kabupaten Ponorogo.

C. Kerangka Berpikir



Gambar 2.2 Bagan Kerangka Berpikir