

BAB II

TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Penelitian yang mendukung Rancang Bangun Jasa WashClean Dengan Integrasi REST API disajikan pada tabel 2.1.

Tabel 2. 1 Penelitian Terdahulu

No.	Peneliti	Judul	Hasil Penelitian
1.	(Arifin et al., 2021)	Pengembangan Sistem Pengelolaan Laundry berbasis Website (Studi Kasus: Senopati Jaya Card Laundry)	Pengelolaan laundry berbasis web di Senopati Jaya Card Laundry mampu mempermudah pembukuan transaksi, mempercepat akses informasi cucian, serta mencatat riwayat perawatan mesin secara terstruktur.
2.	(Ariantara et al., 2020)	Penerapan REST API dalam Pengembangan Aplikasi Pemesanan Rental Mobil berbasis Web dan Mobile (Studi Kasus: CV. Dwi Cipta Rent Car)	Aplikasi pemesanan rental mobil berbasis REST API, memudahkan pelanggan dalam proses booking kendaraan, karena tidak perlu lagi datang langsung ke lokasi untuk melakukan pemesanan. Selain itu, pencatatan transaksi yang otomatis dapat meminimalisir kesalahan pencatatan dan mempercepat pelayanan.
3.	(Putri et al., 2021)	Aplikasi Laundry Berbasis Web Modul Admin Web Based Laundry Application Admin Module	Aplikasi laundry berbasis web yang dibangun dengan metode waterfall meningkatkan efisiensi operasional, mempermudah pencarian data, mempercepat pembuatan laporan keuangan, serta memungkinkan pemilik usaha untuk memantau transaksi secara <i>real-time</i>
4.	(Mardi et al., 2020)	Aplikasi Pengelolaan Pendapatan Dan Pengeluaran Kas Pada Rafa Laundry Berbasis Web	Aplikasi pengelolaan kas pada Rafa Laundry mempermudah pencatatan transaksi, mengelola keuangan lebih akurat, serta mempercepat laporan keuangan dan meningkatkan efisiensi operasional
5.	(Rahman, 2021)	Penerapan Metode Waterfall Pada Aplikasi Laundry Berbasis Web	Aplikasi berbasis web menyimpan dan mengelola data pelanggan serta layanan secara terstruktur, sehingga mengurangi kesalahan pencatatan. Sistem ini juga meningkatkan efisiensi dan operasional di Dinda Laundry.

2.2. REST API

Menurut penelitian dari (Maulana et al., 2024), API (*Application Programming Interface*) adalah antarmuka yang memungkinkan aplikasi untuk berkomunikasi dan berinteraksi satu sama lain. API menyediakan standar dan spesifikasi untuk pertukaran data, fungsi, dan layanan. API memfasilitasi pengembang untuk menggunakan fitur yang telah ada di aplikasi lain, sehingga menghemat waktu dan tenaga. Salah satu arsitektur *back-end* adalah *Representational State Transfer* (REST). REST adalah kumpulan aturan desain yang mengatur bagaimana data ditransfer melalui antarmuka standar seperti HTTP.

2.3. Rancang Bangun

Menurut penelitian dari (Mulyanto et al., 2020), rancang bangun adalah proses perencanaan dan pengembangan sistem yang bertujuan untuk menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada. Proses ini melibatkan analisis kebutuhan, desain sistem, implementasi, serta pengujian.

Rancang bangun sangat penting dalam pengembangan perangkat lunak karena menentukan arah dan struktur sistem yang akan dibangun. Dalam konteks penelitian ini, rancang bangun diterapkan untuk mengembangkan sistem aplikasi WashClean yang dirancang agar mampu mengelola transaksi laundry, data pelanggan, serta mitra usaha secara terpusat dan terintegrasi.

2.4. Basis Data

Menurut (Dirgantara et al., 2023), basis data adalah kumpulan data yang saling terkait dan disimpan secara terorganisir untuk mendukung berbagai aplikasi secara optimal. Penyimpanan data secara independen dari program memungkinkan

pengelolaan yang fleksibel dan aman, terutama dalam hal penambahan, pengubahan, dan pengambilan data.

Dalam penelitian ini, basis data digunakan untuk menyimpan semua data operasional laundry, termasuk data transaksi, pelanggan, dan mitra. Desain basis data yang baik sangat krusial untuk mendukung integrasi antara aplikasi web dan mobile, serta memastikan data konsisten dan dapat diakses dengan cepat oleh REST API.

2.5. Laragon

Menurut (Andarsyah et al., 2022), Laragon adalah perangkat lunak sumber terbuka yang berfungsi sebagai server lokal (*localhost*) untuk mendukung pengembangan aplikasi berbasis web. Laragon mendukung berbagai sistem operasi dan menyediakan lingkungan pengembangan yang ringan, stabil, dan fleksibel, termasuk Apache, MySQL, PHP, dan Node.js.

Dalam konteks penelitian ini, Laragon digunakan sebagai lingkungan pengembangan lokal (*local development environment*) untuk membangun dan menguji aplikasi WashClean sebelum diunggah ke server produksi. Keunggulan Laragon dalam kemudahan konfigurasi dan fleksibilitas menjadikannya pilihan yang sesuai untuk mendukung proses rancang bangun sistem berbasis web dengan framework Laravel dan integrasi REST API.

2.6. Laravel

Menurut penelitian dari (Aipina & Witriyono, 2022), Laravel adalah kerangka kerja (*framework*) berbasis PHP yang bersifat *open-source* dan dirancang untuk mempermudah proses pengembangan aplikasi web. Laravel mendukung arsitektur *Model-View-Controller* (MVC) dan menyediakan fitur unggulan seperti

Blade *templating engine*, *routing* yang fleksibel, dan sistem autentikasi yang terintegrasi.

Dalam penelitian ini, Laravel digunakan sebagai *back-end* utama dalam rancang bangun sistem WashClean. Alasan pemilihan Laravel adalah karena kemampuannya dalam memisahkan logika bisnis dan tampilan, serta dukungan ekosistem yang kuat untuk integrasi REST API. Hal ini memungkinkan sistem dapat dikembangkan secara modular dan mudah dikembangkan kembali di masa depan.

2.7. Metode Waterfall

Menurut (Rahman, 2021), Requirement Analysis (Analisis Kebutuhan): Mengidentifikasi kebutuhan perangkat lunak dengan berkomunikasi langsung dengan pengguna melalui wawancara, survei, atau diskusi.

1. *Design System* (Desain Sistem): Spesifikasi kebutuhan sistem untuk menentukan perangkat keras dan perangkat lunak secara menyeluruh.
2. *Coding* (Pengkodean): Pengembangan unit kecil perangkat lunak (program) yang nantinya akan diintegrasikan.
3. *Testing* (Pengujian): Seluruh unit diintegrasikan dan diuji untuk memastikan tidak ada kegagalan atau kesalahan pada sistem.
4. *Maintenance* (Pemeliharaan) Memperbaiki kesalahan yang ditemukan setelah sistem berjalan dan menambahkan fitur baru sesuai kebutuhan pengguna.

2.8. Figma

Menurut (Muhyidin et al., 2020), Figma adalah alat desain antarmuka berbasis cloud yang memungkinkan kolaborasi secara *real-time* antar desainer.

Figma digunakan untuk membuat prototipe aplikasi, mockup tampilan web atau mobile, serta mendesain *user interface* (UI) yang interaktif.

Dalam pengembangan WashClean, Figma digunakan pada tahap desain antarmuka pengguna, khususnya untuk merancang UI aplikasi mobile dan dashboard web admin. Figma dipilih karena kemampuannya mendukung desain kolaboratif dan *prototyping* yang cepat, sehingga memudahkan dalam proses validasi desain terhadap kebutuhan pengguna.

2.9. Blackbox Testing

Menurut (Muhammad Jibril et al., 2024), *black box testing* adalah metode pengujian perangkat lunak yang berfokus pada pengujian fungsionalitas aplikasi tanpa mengetahui struktur internal kode. Tujuannya untuk memastikan bahwa fitur sistem berfungsi sesuai dengan spesifikasi.

Dalam penelitian ini, metode *black box testing* digunakan untuk menguji fungsionalitas utama aplikasi WashClean, seperti fitur pemesanan layanan laundry, manajemen pelanggan, dan sinkronisasi data melalui REST API. Pengujian dilakukan dari sisi pengguna dan mitra untuk memastikan aplikasi bekerja sesuai kebutuhan pengguna akhir.