

BAB 2 TINJAUAN PUSTAKA

2.1 Landasan Teori

Terdapat beberapa landasan teori yang mendasari dalam perancangan *system* penjualan biji kopi berbasis *website* ini. Landasan teori mencakup pengertian terkait domain bisnis kopi, metode pengembangan sistem, serta teknologi dan *tools* yang digunakan.

2.1.1 *Jokanti Coffee Roasters*

Jokanti Coffee Roasters merupakan usaha lokal yang berfokus pada penjualan minuman dan biji kopi hasil roasting sendiri, yang berasal dari berbagai daerah di Indonesia. Produk yang ditawarkan meliputi biji kopi Arabika, Robusta, serta campuran khusus yang dijual kepada pelanggan individu, kafe, dan *reseller*. Dalam kegiatan usahanya.



Gambar 2.1 *Jokanti Coffee Roasters*

2.1.2 *System Informasi Penjualan*

Sistem informasi pemesanan merupakan aplikasi yang dirancang untuk memfasilitasi proses pembelian produk oleh pelanggan secara digital, mulai dari pencatatan produk, keranjang belanja, proses *checkout*, hingga pengiriman. *Sistem* ini dapat membantu pelaku usaha dalam mengelola transaksi penjualan secara terstruktur, efisien, dan *real-time*. Pada bisnis penjualan biji kopi, sistem informasi

ini sangat penting untuk menangani jumlah pesanan yang tinggi, mempercepat proses layanan, serta menyediakan data transaksi untuk analisis bisnis dan strategi pemasaran.

Menurut Pratap (2021), *system* informasi adalah seperangkat komponen yang saling berinteraksi untuk mengumpulkan, mengolah, menyimpan, dan mendistribusikan informasi guna mendukung proses *operasional*, pengambilan keputusan, serta pengendalian dalam organisasi. Sistem informasi dalam konteks UMKM sangat relevan karena dapat membantu pelaku usaha dalam melakukan otomatisasi kegiatan bisnis, mempercepat pelayanan pelanggan, dan mengurangi ketergantungan terhadap proses manual yang rentan terhadap kesalahan. Dalam bisnis seperti Jokanti *Coffee Roasters*, *system* informasi dapat digunakan untuk mengelola pemesanan, stok produk, data pelanggan, hingga laporan transaksi secara terstruktur dan efisien.

2.1.3 Metode Pengembangan Rapid Application Development (RAD)

Metode Rapid Application Development (RAD) adalah model proses pengembangan perangkat lunak yang menekankan pada siklus pengembangan yang singkat, adaptif, dan bersifat iteratif. Tidak seperti metode konvensional, RAD memanfaatkan pendekatan berbasis komponen serta pembuatan prototipe untuk mempercepat proses penerjemahan kebutuhan bisnis menjadi sistem yang fungsional Sexton & Cavalli (2015). Tujuan utama dari metode ini adalah menghasilkan produk perangkat lunak secara cepat tanpa menurunkan kualitas, dengan melibatkan pengguna secara aktif pada setiap tahap pengembangan sehingga *system* yang dibangun dapat benar-benar selaras dengan kebutuhan operasional di lapangan.



Gambar 2.2 Alur RAD

Dalam metodologi RAD, terdapat langkah-langkah pengembangan aplikasi yang harus diikuti pada setiap fase pengembangan aplikasi:

1) Rencanaan Syarat-Syarat

Pada fase ini, *user* dan penganalisis bekerja sama dalam diskusi untuk mengidentifikasi tujuan aplikasi atau *sistem*, serta data yang dibutuhkan untuk mencapai tujuan tersebut. Fase ini melibatkan kedua belah pihak secara aktif dan penting, karena tidak hanya menerima ide yang diajukan. Pertemuan seperti ini seringkali disebut sebagai kolaborasi pengembangan aplikasi, sehingga memastikan bahwa informasi yang dibutuhkan oleh setiap pengguna dapat terpenuhi dengan baik.

2) *Workshop Desain* RAD

Apabila terdapat perbedaan pendapat antara *user* dan *designer* mengenai perancangan pada tahap ini, proses perancangan harus diimplementasikan dan diperbaiki. Keterlibatan *user* menjadi sangat penting dalam tahap ini untuk memastikan keberhasilan, karena mereka dapat memberikan tanggapan secara langsung yang berkaitan dengan perancangan. *User* dan *designer* sering bekerja secara kolaboratif untuk menyelesaikan masalah tersebut.

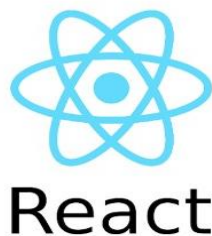
3) Implementasi

Pada fase ini, *user* dan *designer* menyetujui perancangan *system*, para *programmer* akan mengembangkan perancangan tersebut menjadi program. Setelah program tersebut siap, baik beberapa bagian atau keseluruhan, program tersebut akan diuji untuk memeriksa keberadaan *bug* sebelum

diimplementasikan di dalam organisasi. Pada tahap ini, *user* memiliki kesempatan untuk memberikan komentar mengenai sistem yang telah dibuat dan memberikan persetujuan terhadap sistem tersebut. Kontribusi *user* sangat penting agar sistem yang dikembangkan dapat memenuhi kepuasan *user*.

2.1.4 *ReactJS*

Seiring meningkatnya kebutuhan untuk membangun antarmuka aplikasi yang efisien dan mudah dikelola, *tools* menjadi hal yang penting dalam mendukung pengembangan front-end secara optimal. Salah satu library yang banyak dibahas adalah *React JS*. Menurut pendapat Shrivastava (2020), menyatakan bahwa *React JS* adalah *Library Javascript* yang bersifat *OpenSource* yang dimana mayoritas digunakan untuk membangun *User Interface* (UI) secara spesifik untuk satu *page* dalam aplikasi. Awalnya, *ReactJS* Adalah *JavaScript Library* yang dikembangkan oleh Jordan Walke sekitar tahun 2013. *ReactJS* ini *ReactJS* ini dapat mendesain tampilan sederhana untuk setiap tingkatan dalam aplikasi, sehingga dapat digunakan untuk membuat dan mengembangkan pembuatan aplikasi berbasis *website*. Popularitasnya dapat diukur oleh aplikasi –aplikasi yang menggunakannya react ini seperti *Facebook*, *WhatsApp*, *Netflix*, *Instagram*, *Airbnb*, *American Express*, *Dropbox*, *Ebay*, dan ratusan penyedia jasa pembuatan aplikasi berbasis web memanfaatkan kemampuan *React JS* digunakan sebagai salah satu framework untuk membuat bagian *front-end* dari sebuah aplikasi. *ReactJS* berfungsi untuk mengatur *Layer* pada tampilan untuk Desktop maupun versi mobile Aplikasi. Dengan *React*, user bisa membuat *Reusable Component* atau jenis komponen yang bisa digunakan kembali tanpa harus membangunnya dari awal lagi.



Gambar 2.3 *React*

2.1.5 *Node.js*

Menurut pendapat Sarreshtedari & Akhaee,(2014), *Node.js* adalah pilihan ideal untuk *system* penjualan karena arsitektur non-blocking I/O miliknya

memungkinkan server menangani ribuan transaksi dan koneksi secara bersamaan dengan sangat cepat, yang sangat krusial saat menghadapi lonjakan trafik seperti momen *flash sale*. Selain itu, ekosistem *library* (NPM) yang sangat luas memudahkan pengembang untuk mengintegrasikan sistem pembayaran (*payment gateway*), layanan pengiriman, hingga sinkronisasi stok secara *real-time*, sementara penggunaan bahasa *JavaScript* di seluruh tumpukan teknologi (*full-stack*) membuat proses pengembangan aplikasi menjadi lebih efisien dan mudah dikelola.



Gambar 2.4 *Node.js*

2.1.6 *Express.js*

Menurut pendapat Sexton & Cavalli (2015) *Express.js* adalah sebuah kerangka kerja (*framework*) untuk *Node.js* yang digunakan untuk membuat aplikasi web, terutama bagian backend atau server. *Express* membantu developer dalam menangani permintaan dari pengguna, mengatur jalur atau rute (*routing*), serta mengelola data yang masuk dan keluar dari *server* dengan cara yang lebih mudah dan cepat. *Express* banyak digunakan karena ringan, fleksibel, dan memiliki banyak fitur yang mendukung pembuatan aplikasi *modern*. Dalam proyek ini, *Express.js* digunakan untuk membangun sistem backend yang menangani proses pemesanan, pembayaran, dan pengiriman biji kopi. *Express* juga memudahkan dalam menghubungkan server dengan *database PostgreSQL* menggunakan *Sequelize*. Dengan *Express*, pengembangan sistem bisa dilakukan lebih cepat karena sintaksnya sederhana dan mudah dipahami, sangat cocok untuk metode pengembangan RAD yang butuh banyak revisi dan percobaan.



Gambar 2.5 *Express.js*

2.1.7 *Sequelize ORM*

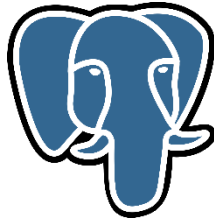
Menurut pendapat Philip (2020) Alat bantu (*library*) dalam *Node.js* yang digunakan untuk mengelola dan berkomunikasi dengan *database*, seperti *PostgreSQL*. Dengan *Sequelize*, kita bisa mengatur data seperti menambahkan, mengubah, menghapus, atau mengambil data dari database tanpa harus menulis perintah SQL secara manual. Cara kerja *Sequelize* lebih sederhana karena menggunakan kode JavaScript, sehingga memudahkan proses pengembangan aplikasi. Dalam proyek ini, *Sequelize* digunakan untuk menyimpan dan mengatur data seperti informasi produk biji kopi, data pelanggan, riwayat pemesanan, dan status pembayaran. *Sequelize* juga membantu menghubungkan antar data, misalnya antara pesanan dan pelanggan, sehingga sistem lebih terstruktur dan mudah dikelola. Penggunaannya sangat membantu dalam mempercepat pembangunan *backend*, terutama saat sistem masih dalam tahap pengujian dan sering mengalami perubahan.



Gambar 2.6 *Sequelize ORM*

2.1.8 *PostgreSQL*

Menurut Srinivasa (2020), *PostgreSQL* digunakan sebagai sistem manajemen basis data dalam pengembangan backend untuk *Jokanti Coffee Roasters* karena mampu menangani pengelolaan data produk biji kopi, data pengguna, transaksi pemesanan, serta integrasi dengan layanan pihak ketiga seperti *Midtrans* dan *Shipper*. Dengan dukungan fitur relasional yang kuat dan kepatuhan terhadap standar ACID, *PostgreSQL* memungkinkan pencatatan transaksi secara konsisten dan aman, sekaligus memberikan fleksibilitas dalam pengelolaan data yang kompleks. Selain itu, *PostgreSQL* juga mendukung skalabilitas dan keandalan, sehingga cocok untuk mengakomodasi kebutuhan sistem seiring pertumbuhan bisnis *Jokanti Coffee Roasters*.



Gambar 2.7 PostgreSQL

2.1.9 Midtrans

Menurut pendapat Mishra (2019) Midtrans adalah sebuah platform pembayaran yang menawarkan solusi lengkap untuk memproses transaksi online. Didirikan pada tahun 2012, *Midtrans* kini telah menjadi salah satu penyedia layanan pembayaran terkemuka di Indonesia. Platform ini mendukung berbagai metode pembayaran, mulai dari kartu kredit, transfer bank, *e-wallet*, hingga pembayaran melalui minimarket.



Gambar 2.8 Midtrans

2.1.10 shipper

Menurut Rajesh (2024) Shipper adalah pihak (individu, UMKM, atau perusahaan) yang bertanggung jawab mengirimkan barang atau kargo dari satu tempat ke tempat lain, biasanya melalui jasa ekspedisi atau logistik. Shipper bertindak sebagai pengirim yang menyiapkan barang, mengemas, dan memberikan instruksi pengiriman kepada kurir atau kurir.



Gambar 2.8 shipper

2.1.11 Usability Testing

Usability Testing adalah metode pengujian yang bertujuan untuk mengevaluasi tingkat kemudahan, kenyamanan, serta kepuasan pengguna terhadap suatu sistem dengan melibatkan pengguna secara langsung. Melalui interaksi nyata

ini, pengembang dapat mengidentifikasi hambatan yang dihadapi pengguna dan melakukan perbaikan yang berbasis pada pengalaman pengguna (*user experience*).

Menurut Lai & Crassidis (2021) fokus utama dari *usability testing* adalah mengamati bagaimana pengguna menyelesaikan tugas-tugas spesifik serta menilai sejauh mana sistem dapat beroperasi secara efektif, efisien, dan memuaskan.

Untuk mengukur kualitas antarmuka tersebut, terdapat lima kriteria utama yang digunakan:

1. *Learnability* (Kemudahan Dipelajari): Menilai seberapa mudah pengguna dapat memahami fungsi-fungsi dasar dan mulai mengoperasikan sistem saat pertama kali melihatnya.
2. *Efficiency* (Efisiensi): Mengukur kecepatan dan ketepatan pengguna dalam menyelesaikan tugas setelah mereka memahami cara penggunaan sistem.
3. *Memorability* (Kemampuan Mengingat): Menilai seberapa baik pengguna dapat mengingat cara menggunakan sistem kembali setelah tidak berinteraksi dengannya dalam jangka waktu tertentu.
4. *Errors* (Kesalahan): Mengamati jumlah, frekuensi, dan tingkat keparahan kesalahan yang dilakukan pengguna, serta seberapa mudah mereka dapat pulih dari kesalahan tersebut.
5. *Satisfaction* (Kepuasan): Mengukur aspek subjektif mengenai seberapa nyaman dan puas pengguna saat berinteraksi dengan sistem secara keseluruhan.

2.1.12 *Black Box Testing*

Black Box Testing merupakan metode pengujian perangkat lunak yang menitikberatkan pada fungsionalitas sistem berdasarkan kebutuhan pengguna tanpa melibatkan analisis struktur internal atau kode program. Fokus utama metode ini adalah mengevaluasi hasil (*output*) yang dihasilkan dari berbagai masukan (*input*) untuk memastikan bahwa setiap fungsi berjalan sesuai dengan spesifikasi yang telah dirancang. Menurut Kumari & Trakru (2020)), metode ini sangat efektif untuk memvalidasi fitur-fitur utama sistem dari perspektif pengguna tanpa memerlukan pemahaman teknis yang mendalam mengenai implementasi kodenya.

Dalam konteks pengembangan *sistem* pemesanan biji kopi berbasis *website*, *Black Box Testing* diterapkan untuk menguji alur kerja krusial pada sisi konsumen, yang meliputi:

- 1) Proses registrasi dan login pengguna.
- 2) Visualisasi katalog dan detail produk.
- 3) Mekanisme penambahan produk ke keranjang belanja.
- 4) Pengelolaan kuantitas pesanan.
- 5) Prosedur finalisasi atau pembuatan pesanan (*checkout*).

Pengujian dilakukan dengan memberikan berbagai skenario input, mencakup data *valid* (untuk menguji keberhasilan fungsi) dan tidak valid (untuk menguji ketahanan sistem terhadap kesalahan). Hal ini sejalan dengan penelitian Sonawane (2019), yang menunjukkan bahwa penerapan *Black Box Testing* pada sistem berbasis web mampu mengidentifikasi kesalahan fungsional sejak tahap awal pengembangan, sehingga memungkinkan penyempurnaan sistem secara optimal sebelum dirilis kepada pengguna akhir.

2.2 Penelitian Terdahulu

Berdasarkan hasil temuan referensi, penelitian terkait rancang bangun *system* informasi menggunakan metode *Rapid Application Development* (RAD) telah banyak dilakukan dalam berbagai bidang. Berikut adalah beberapa penelitian terdahulu yang menjadi acuan dalam penelitian ini:

Penelitian pertama dilakukan pada Puskesmas Sentajo Raya yang bertujuan untuk mengatasi kendala pengelolaan rekam medis manual seperti kerusakan dan kehilangan berkas. *Sistem* ini dikembangkan menggunakan metode RAD dengan dua kali iterasi dan diselesaikan dalam 90 hari. Hasil pengujian menunjukkan nilai *Black Box Testing* sebesar 100% dan *Usability Testing* mencapai 95,449%, menandakan bahwa sistem berfungsi sangat baik dan memuaskan pengguna SELLA (2021).

Selanjutnya, penelitian kedua diterapkan pada Kelompok Masyarakat Nanas Berduri (Ponari) di Kabupaten Kampar. Pengembangan *website* pemasaran berbasis RAD ini difokuskan pada katalog produk dan integrasi pemesanan

langsung ke *WhatsApp* Bisnis. Penelitian terkait *system* pemasaran digital menunjukkan bahwa *website* mampu memperluas jangkauan dan diterima dengan baik oleh pengguna, dengan hasil SUS sebesar 79 yang berada pada kategori *Good* Givana (2023)

Penelitian ketiga membahas perancangan *system* informasi pendaftaran *online* pada *English Camp Pare AEC* Man 2 Kuansing untuk mengatasi proses pendaftaran *konvensional* yang memakan waktu dan biaya. Dengan menerapkan metode RAD, sistem yang dikembangkan mampu meningkatkan efisiensi, memperbarui informasi lebih cepat, serta memberikan kemudahan administrasi. Penelitian tentang *system* pendaftaran *online* menunjukkan bahwa *usability* mencapai 90,6% yang termasuk kategori sangat memuaskan Amirul (2021).

Penelitian keempat dilakukan pada *Cotton Patisserie*, yaitu pengembangan *sistem e-commerce* berbasis RAD yang mencakup fitur riwayat pesanan dan payment gateway. Pengujian *System Usability Scale* (SUS) menghasilkan skor 76 atau *Grade B*, menunjukkan bahwa *system* cukup baik dari sisi kemudahan penggunaan dan dapat membantu optimalisasi proses pemesanan Cahyani (2024).

Terakhir, penelitian kelima dilakukan pada PT. Andalas Karya Mulia (AKM) yang fokus pada pengembangan sistem pelaporan kendaraan berat dan alat berat berbasis *Laravel* dan *Flutter* menggunakan metode RAD. Sistem ini dirancang untuk mempermudah proses pelaporan secara *real-time* dari lokasi proyek. Hasil *Usability Testing* menunjukkan nilai rata-rata 68,24%, di mana responden menyatakan bahwa sistem sudah memenuhi kriteria fungsional dalam meningkatkan koordinasi antara kantor dan lapangan Oktavian (2023).

Melalui berbagai studi tersebut, dapat disimpulkan bahwa pengembangan *system* informasi berbasis *website*, baik untuk kebutuhan *Jokanti Coffee Roasters*, toko, maupun usaha kopi, memiliki kesamaan kebutuhan utama: pencatatan data yang akurat, manajemen produk yang terstruktur, dan integrasi dengan teknologi pendukung seperti notifikasi atau laporan otomatis.

Tabel 2.1 Perbandingan Penelitian Terdahulu

Uraian	Sella, S. S. (2024)	Audia Angel Givana (2024)	Amirul Mukminin (2024)	Ardiah Pramesty Cahyani (2024)
Judul / Studi Kasus	Sistem Informasi Rekam Medis Berbasis <i>Website</i>	Rancang Bangun <i>System</i> Informasi Pengelolaan Persediaan Barang Berbasis Web (Toko Bangunan Berkah Jaya)	Rancang Bangun <i>System</i> Informasi Penjualan & Persediaan Barang Berbasis Web (Toko Bangunan Berkah Jaya)	Pengembangan <i>E-Commerce</i> Penjualan Roti dan Kue Berbasis Web (Toko Kue Mamma's Kitchen)
Metode	RAD	RAD	RAD	RAD
Output / Hasil Utama	Digitalisasi rekam medis dan pengelolaan data klinis	Pengelolaan persediaan barang masuk/keluar dan laporan stok	<i>System</i> penjualan, kontrol stok, dan laporan penjualan	<i>Platform</i> penjualan online lengkap dengan pembayaran digital
Fitur	• Login & Register • <i>Dashboard Multi - user</i> • Manajemen Rekam Medis • Notifikasi Resep • Manajemen Obat • Cetak Laporan (<i>Excel</i>)	• Login User • Manajemen Data Master • Transaksi Barang Masuk • Transaksi Barang Keluar • Laporan Stok Barang • Laporan Bulanan	• Manajemen Data User • Manajemen Data Master • Transaksi Penjualan • Transaksi Barang Masuk • Laporan Penjualan • Laporan Stok	• Registrasi & Login • Katalog Produk • Keranjang & Checkout • Payment <i>Gateway</i> • Riwayat Pesanan • <i>Dashboard</i> Admin