

BAB 2

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian yang dilakukan oleh (Afifah et al., 2024) mengenai *rancang bangun website penjualan produk dan pemesanan jasa berbasis web pada PT. Emap Technology*. Penelitian ini bertujuan untuk mempermudah proses transaksi penjualan dan pemesanan jasa manufaktur yang sebelumnya masih dilakukan secara konvensional. Sistem dibangun menggunakan *framework* Laravel dengan metode pengembangan perangkat lunak *prototype*. Hasil dari penelitian ini menunjukkan bahwa website yang dirancang berhasil menyediakan fitur seperti katalog produk, form pemesanan jasa, manajemen transaksi, hingga dashboard admin. Pengujian alpha dan beta menunjukkan sistem dapat digunakan dengan baik, dengan tingkat kepuasan pengguna mencapai 93,33%, serta mampu meningkatkan efisiensi layanan dan memperluas jangkauan pelanggan di luar daerah.

Penelitian yang dilakukan oleh (Sriyeni et al., 2024) membahas mengenai *rancang bangun sistem informasi penjualan motor dan bengkel berbasis web pada PD Panca Motor KM 14 Palembang*. Penelitian ini bertujuan untuk mengatasi masalah proses penjualan motor dan *spare part* yang masih dilakukan secara manual, serta layanan bengkel yang kurang efisien. Pengembangan sistem menggunakan metode *prototype*, dengan tahapan komunikasi, perencanaan cepat, desain awal, konstruksi *prototype*, hingga pengujian dan umpan balik pengguna. Hasil penelitian menunjukkan bahwa sistem yang dikembangkan berhasil membantu dalam pengelolaan data penjualan, data *spare part*, dan layanan servis motor. Sistem juga mendapat tingkat penerimaan pengguna sebesar 81% berdasarkan evaluasi *User Acceptance Testing*, yang menandakan bahwa sistem dapat diterima sepenuhnya dan memberikan dampak positif terhadap efisiensi operasional perusahaan.

Penelitian yang dilakukan oleh (Hapsah et al., 2025) mengenai *perancangan sistem informasi e-commerce berbasis website* pada toko Ruang Style. Penelitian ini bertujuan untuk mengatasi berbagai permasalahan yang dihadapi toko, seperti keterbatasan media promosi, belum tersedianya *platform* penjualan *online*, dan pengelolaan data penjualan serta inventaris yang masih dilakukan secara manual. Peneliti menggunakan metode *prototype* untuk merancang sistem, memungkinkan pengguna memberikan evaluasi langsung terhadap tampilan dan fungsionalitas sebelum sistem sepenuhnya dikembangkan. Hasil dari penelitian ini menunjukkan bahwa sistem yang dirancang mampu mempermudah proses transaksi secara *online*, mendukung proses pemasaran digital, serta mengotomatisasi pengelolaan data penjualan dan inventaris melalui antarmuka berbasis web yang dapat diakses melalui *desktop browser*. Sistem ini juga memperluas jangkauan pemasaran toko dan meningkatkan efisiensi operasionalnya.

Penelitian yang dilakukan oleh (Sidiq et al., 2023) membahas tentang *perancangan aplikasi penjualan berbasis web dengan metode prototype* pada UMKM Sinar Terang di Desa Pusakasari, Kecamatan Cipaku. Tujuan dari penelitian ini adalah membantu UMKM dalam meningkatkan efektivitas promosi dan penjualan produknya secara *online* agar dapat menjangkau pasar yang lebih luas. Penelitian ini menggunakan bahasa pemrograman PHP dan MySQL sebagai basis data, serta memanfaatkan metode *prototype* untuk memungkinkan interaksi langsung antara pengembang dan pengguna selama proses pengembangan sistem. Hasil dari penelitian ini berupa rancangan sistem berbasis web yang memudahkan pelanggan dalam melakukan pemesanan produk serta memberikan kemudahan bagi pemilik usaha dalam mengelola data transaksi dan laporan penjualan secara digital. Sistem yang dibangun diharapkan mampu meningkatkan efisiensi dan efektivitas proses bisnis UMKM.

Penelitian yang dilakukan oleh (Daffa et al., 2022) mengenai digitalisasi *UMKM melalui perancangan aplikasi e-commerce berbasis website dengan model Business to Consumer (B2C)* pada Toko Balgis Cake. Penelitian ini bertujuan untuk mengatasi masalah proses jual beli yang masih dilakukan secara

manual serta promosi yang kurang efektif akibat ketergantungan pada *marketplace*. Dengan menggunakan metode *prototype*, penulis membangun sistem *e-commerce* berbasis web menggunakan *framework* Laravel dan PHP, serta melakukan pengujian sistem menggunakan metode *black box*. Hasil dari penelitian ini menunjukkan bahwa sistem yang dibangun mampu memberikan kemudahan transaksi bagi pelanggan, meningkatkan efektivitas promosi, memperluas jangkauan pasar, dan mengurangi kesalahan dalam pencatatan pesanan melalui digitalisasi proses penjualan.

Dari penelitian-penelitian sebelumnya, terdapat beberapa *gap* yang perlu diatasi. Penelitian oleh (Afifah et al., 2024) belum membahas sistem pembayaran *online*, sementara penelitian (Sriyeni et al., 2024) belum mendukung pembelian *online* penuh dengan fitur seperti keranjang belanja. Penelitian (Hapsah et al., 2025) tidak mencakup integrasi media sosial atau pemasaran digital, sedangkan (Sidiq et al., 2023) masih mengandalkan pengelolaan stok manual. Terakhir, penelitian (Daffa et al., 2022) belum mengoptimalkan pengelolaan produk dan promosi secara otomatis. Penelitian ini bertujuan mengatasi *gap-gap* tersebut dengan mengembangkan sistem *e-commerce* pada Toko Takashimura yang mencakup pembayaran *online*, manajemen produk, dan integrasi pemasaran digital.

Tabel 2. 1 Penelitian Terdahulu

Penulis	Judul Penelitian	Masalah	Hasil	GAP
Afifah et al., 2024	Rancang Bangun Website Penjualan Produk dan Pemesanan Jasa <i>Service Manufaktur</i> Menggunakan <i>Framework</i> Laravel	Penjualan dan pemesanan jasa masih dilakukan secara manual, tidak efisien dan menyulitkan pelanggan	Menghasilkan <i>website</i> berbasis Laravel dengan fitur katalog produk, pemesanan jasa, dan pengelolaan transaksi.	Belum membahas sistem pembayaran <i>online</i>
Sriyeni et al., 2024	Rancang Bangun Sistem Informasi Penjualan Motor dan Bengkel Menggunakan Metode <i>Prototype</i>	Proses penjualan dan pencatatan servis masih manual dan rawan kesalahan	Sistem berhasil mengelola data penjualan, servis motor, dan data transaksi.	Belum mendukung pembelian online secara penuh seperti fitur keranjang belanja, dan checkout barang, dsb
Hapsah et al., 2025	Prototype Pengembangan Sistem Informasi <i>E-Commerce</i> Berbasis <i>Website</i>	Belum tersedia sistem penjualan <i>online</i> , pemasaran hanya <i>offline</i>	Sistem memfasilitasi transaksi online, penyajian katalog produk, dan pengelolaan inventaris dan penjualan	Belum membahas pengembangan fitur seperti integrasi media sosial
Sidiq et al., 2023	Perancangan Aplikasi Penjualan	Kurangnya promosi dan keterbatasan	Aplikasi mempermudah pemesanan	Pengelolaan stok masih

Penulis	Judul Penelitian	Masalah	Hasil	GAP
	Berbasis Web dengan Metode <i>Prototype</i> pada UMKM Sinar Terang	jangkauan pasar UMKM secara manual.	produk dan pencatatan transaksi secara digital	dilakukan secara manual
Daffa et al., 2022	Digitalisasi UMKM dengan Membangun Aplikasi <i>E-Commerce Model Business to Consumer (B2C)</i>	Transaksi dan promosi masih manual, ketergantungan pada <i>marketplace</i> .	Sistem <i>e-commerce</i> memudahkan transaksi, meningkatkan promosi dan jangkauan pasar	Belum adanya sistem untuk mengelola produk, seperti katalog produk.

2.2 Landasan Teori

2.2.1 Toko Takashimura

Toko Takashimura merupakan salah satu usaha mikro, kecil, dan menengah (UMKM) yang terletak di Pasar Atas, Bukittinggi. Bisnis ini fokus pada penjualan pakaian, baik pakaian sehari-hari maupun pakaian khas Bukittinggi yang sering dijadikan oleh-oleh. Produk khas yang dijual di Toko Takashimura mencerminkan budaya dan identitas lokal Bukittinggi, yang menjadikannya menarik bagi wisatawan lokal maupun mancanegara. Menurut hasil wawancara penulis dengan pemilik toko, proses bisnis yang berjalan saat ini di Toko Takashimura melibatkan beberapa aktivitas utama seperti :

1) Penjualan Langsung di Toko Fisik

Pelanggan datang langsung ke toko untuk memilih dan membeli produk yang tersedia.

2) Pencatatan Pesanan Secara Manual

Seluruh transaksi dicatat menggunakan metode konvensional menggunakan buku catatan, yang berisiko terjadi kesalahan pencatatan.

3) Strategi Promosi Sederhana

Promosi dilakukan melalui spanduk atau dari mulut ke mulut, tanpa menggunakan media digital untuk memperluas jangkauan pasar.

2.2.2 E-Commerce

Menurut pendapat (Jabat, 2022), *e-commerce* merupakan aktivitas transaksi jual beli yang dilakukan secara elektronik, dengan menggunakan internet sebagai media utama penghubungnya. Hal ini berarti menandakan bahwa proses pertukaran barang atau jasa antara penjual dan pembeli tidak terjadi secara fisik, melainkan melalui *platform* dan jaringan internet. Konsumen dapat melihat produk, melakukan pemesanan, dan melakukan pembayaran secara daring, sementara penjual dapat menawarkan produknya kepada khalayak yang lebih luas tanpa adanya batasan geografis.

Menurut (Rahmadani et al., 2020) *e-commerce* terdiri dari enam komponen utama yang saling mendukung untuk menciptakan pengalaman belanja online yang

optimal. Pertama yaitu produk, tempat jualan produk, metode pembayaran, metode pengiriman, cara terima pesanan, dan yang terakhir yaitu, pelayanan. Enam komponen utama *e-commerce* ini membentuk sebuah ekosistem yang saling terhubung dan saling mendukung. Keenam komponen ini menciptakan pengalaman belanja *online* yang lancar, aman, dan menyenangkan, sekaligus menjadi fondasi kesuksesan bisnis *e-commerce* di era digital.

2.2.3 Website

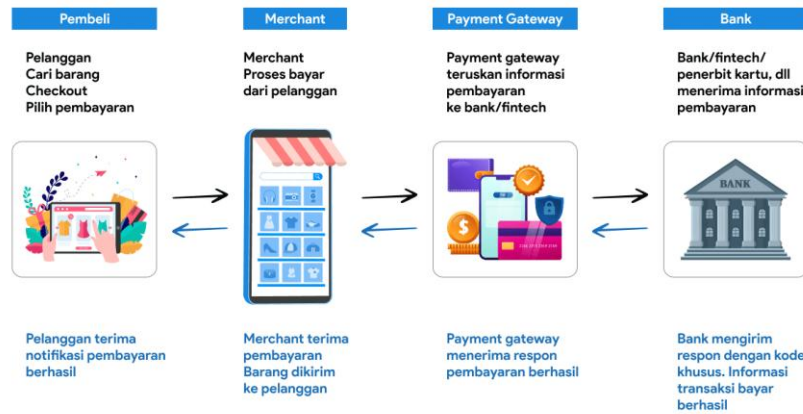
Menurut (Abdur Rochman, 2020) *website* adalah halaman informasi yang disediakan melalui jalur internet, sehingga bisa diakses dimanapun selama terkoneksi dengan jaringan internet. *Website* merupakan kumpulan komponen yang terdiri dari teks, gambar, suara, animasi, sehingga lebih merupakan media informasi yang menarik untuk dikunjungi. *Website* merupakan bagian dari layanan yang dapat berjalan di atas teknologi internet. Halaman *website* dapat diakses dengan adanya teknologi web server sebagai penyedia halaman *website*, HTML sebagai bahasa baku dan HTTP sebagai jalur pengiriman dokumen *website*.

Website tidak hanya berfungsi sebagai media informasi, namun juga sebagai sarana interaksi antara penyedia layanan dengan pengguna. *Website* dapat dikategorikan menjadi 3 golongan. Pertama, *website statis* adalah jenis *website* yang konten halamannya cenderung tetap dan tidak mengalami perubahan secara berkala tanpa adanya campur tangan langsung pada kode sumbernya. Kedua, *website dinamis* menawarkan fleksibilitas yang lebih besar dalam pengelolaan konten. Ketiga, *website interaktif* melampaui sekadar penyajian informasi satu arah, memungkinkan pengguna untuk berpartisipasi aktif dan berinteraksi dengan konten maupun dengan pengguna lain.

2.2.4 Payment Gateway

Menurut buku *Konsep Dasar Paymen Gateway* (Rizky Wicaksono, 2021), *payment gateway* bisa didefinisikan dari sudut pandang yang berbeda-beda, seperti sudut pandang dari sistem informasi, manajemen, perbankan, dan keuangan. Secara umum *payment gateway* adalah komponen penting dalam ekosistem pembayaran *online* yang memungkinkan transaksi keuangan cepat, aman, dan mudah bagi semua pihak yang terlibat, baik itu bisnis, konsumen, atau

institusi keuangan. Secara keseluruhan, *payment gateway* sangat dibutuhkan dalam dunia bisnis saat ini, terutama dalam sektor *e-commerce* dan layanan berbasis aplikasi.



Gambar 2. 1 Alur Payment Gateway

Gambar 2.1 menjelaskan tentang alur kerja payment gateway. Alur kerja payment gateway dimulai saat pelanggan checkout dan memilih pembayaran, informasi datanya diteruskan dari merchant ke *payment gateway*. Selanjutnya, *payment gateway* meneruskan pemrosesan ke penerbit pembayaran terkait (seperti bank untuk *virtual account*, penerbit *e-wallet* untuk *QRIS*, atau penerbit kartu seperti *MasterCard/Visa* untuk kartu kredit). Institusi penerbit kemudian merespons dengan kode khusus dan informasi status transaksi. Terakhir, *payment gateway* menerima konfirmasi keberhasilan dan memberitahukannya kepada *merchant* dan pembeli.

2.2.5 Midtrans

Midtrans merupakan sistem gerbang pembayaran (*payment gateway*) yang berfungsi sebagai mediator transaksi digital antara pembeli dan penjual demi menunjang proses jual-beli serta menjamin keamanan dalam pembayaran pesanan. Sistem ini dirancang untuk memiliki fitur-fitur yang terintegrasi secara dinamis dengan platform *e-commerce*, sehingga mampu mengakomodasi seluruh jenis metode pembayaran yang umum digunakan di Indonesia. Layanan yang disediakan mencakup transaksi pembayaran secara online yang sangat bervariasi, mulai dari penggunaan kartu debit, kartu kredit, hingga memfasilitasi proses penarikan serta

pengiriman uang tunai sesuai dengan kebutuhan pasar digital saat ini.(Fatman et al., 2023)

Penerapan *payment gateway* seperti *midtrans* ini memberikan dampak signifikan terhadap efisiensi operasional dengan memudahkan kedua belah pihak, baik pembeli maupun penjual, dalam menyelesaikan proses transaksi secara *instan* dan *real-time*. Sinkronisasi sistem *midtrans* pada aplikasi penjualan berbasis *website* juga memberikan keuntungan praktis bagi pengguna, di mana pembeli dapat langsung mengakses halaman *website* dan menyelesaikan pembayaran tanpa perlu melewati proses mengunduh aplikasi tambahan terlebih dahulu. Fleksibilitas ini menjadikan *midtrans* sebagai solusi infrastruktur pembayaran yang dinamis, andal, dan efektif dalam mendukung ekosistem perdagangan *elektronik* modern yang menuntut kecepatan serta proteksi data finansial yang tinggi.

2.2.6 Raja Ongkir

RajaOngkir merupakan sebuah layanan penyedia informasi mengenai jasa ekspedisi dan sistem penghitungan ongkos kirim di Indonesia. Sebagai sebuah situs sekaligus penyedia layanan antarmuka *pemrograman* aplikasi atau *Application Programming Interface (API/Web Service)*, Raja Ongkir mengintegrasikan data tarif pengiriman dari berbagai kurir logistik ternama di Indonesia, seperti POS Indonesia, JNE, TIKI, PCP, ESL, dan RPX. Kehadiran teknologi API ini dirancang untuk menjembatani kebutuhan data logistik secara *real-time*, sehingga pengguna dapat dengan mudah mengetahui, menghitung, dan membandingkan tarif ongkos kirim dari berbagai penyedia jasa ekspedisi secara akurat dalam satu waktu tanpa harus memeriksa situs web masing-masing kurir secara manual.

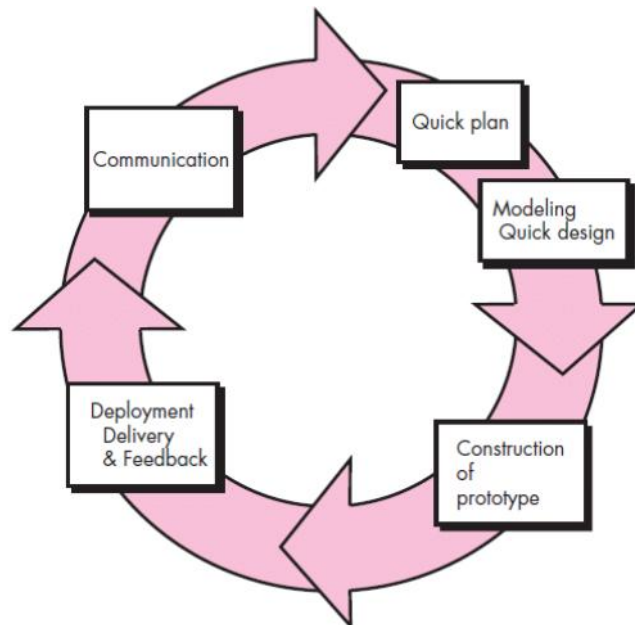
Secara fungsional, implementasi API Raja Ongkir ditujukan untuk memberikan kemudahan akses, baik bagi pengguna umum yang sering melakukan aktivitas belanja daring maupun secara khusus bagi para pemilik toko online (*e-commerce*). Bagi pemilik toko online, integrasi sistem ini ke dalam platform penjualan mereka sangat krusial untuk otomatisasi kalkulasi biaya kirim saat proses pemesanan barang (*checkout*). Dengan demikian, penerapan teknologi dari Raja Ongkir ini terbukti mampu menyederhanakan dan mempermudah proses pengecekan tarif pengiriman barang ke seluruh wilayah Indonesia, yang pada

akhirnya meningkatkan efisiensi operasional bisnis, transparansi biaya bagi konsumen, serta mendukung kelancaran ekosistem perdagangan elektronik secara keseluruhan(Maulana et al., 2024).

2.2.7 Metode Prototype

Dalam buku *Software Engineering: A Practitioner's Approach*, (Roger S. Pressman, n.d.) menjelaskan bahwa metode *Prototype* adalah pendekatan evolusioner yang menekankan pada perencanaan dan pemodelan cepat, diikuti dengan pembangunan *prototype* yang kemudian diserahkan kepada pemangku kepentingan untuk evaluasi dan umpan balik.

Metode ini sangat berguna ketika kebutuhan pengguna belum sepenuhnya dipahami. Dengan membangun *prototype* awal, pengembang dan pengguna dapat berinteraksi untuk memperjelas kebutuhan dan harapan terhadap sistem yang akan dikembangkan. Proses ini bersifat iteratif hingga *prototype* berkembang menjadi sistem yang memenuhi kebutuhan pengguna.



Gambar 2. 2 Alur Prototype (Roger S. Pressman, n.d.)

Gambar 2.2 mengilustrasikan tentang alur tahapan inti dalam metode *prototype*, berikut merupakan penjelasan tentang alur metode *prototype* berdasarkan gambar 2.2 :

1) *Communication*

Pada tahap ini, pengembang berinteraksi langsung dengan pengguna untuk memahami kebutuhan sistem secara umum. Komunikasi yang baik di awal proses sangat penting untuk memastikan arah pengembangan sistem sesuai dengan ekspektasi pengguna.

2) *Quick Plan*

Tahap perencanaan awal dilakukan dengan cepat untuk menentukan ruang lingkup proyek dan mengidentifikasi kebutuhan fungsional utama yang akan dimasukkan dalam *prototype* awal.

3) *Modeling Quick Design*

Desain cepat atau *quick design* dibuat sebagai representasi kasar dari antarmuka pengguna dan alur sistem. Desain ini tidak bersifat detail, melainkan menjadi dasar bagi pembuatan *prototype* pertama. Pada tahap ini, desain awal sistem dibuat dengan menggunakan notasi UML dan pemodelan berbasis objek.

4) *Construction of Prototype*

Pengembang membangun *prototype* berdasarkan desain cepat menggunakan alat bantu atau teknologi yang memungkinkan iterasi cepat. *Prototype* ini akan dievaluasi oleh pengguna untuk memberikan umpan balik terhadap desain dan fungsionalitasnya.

5) *Deployment, Delivery & Feedback*

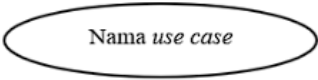
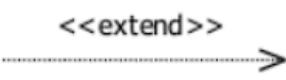
Prototype diberikan kepada pengguna untuk diuji. Pengguna memberikan masukan terkait apa yang perlu diperbaiki atau ditambahkan. Proses ini dapat berlangsung beberapa siklus hingga sistem memenuhi kebutuhan pengguna secara menyeluruh.

2.2.8 Use Case Diagram

Use case diagram adalah alat yang digunakan untuk memodelkan interaksi antara aktor dan sistem dalam pengembangan perangkat lunak, yang membantu dalam menganalisis dan merancang sistem dengan menggambarkan fungsionalitas yang tersedia dalam sistem tersebut (Fauzan et al., 2021).

Menurut (Ramdany et al., 2024) ada 2 elemen penting yang harus digambarkan dalam *use case diagram*, yaitu *aktor* dan *use case*. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang. Sedangkan, *use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. Untuk simbol-simbol yang ada pada *diagram use case* akan dijelaskan pada tabel 2.2.

Tabel 2. 2 Simbol Use Case (Ramdany et al., 2024)

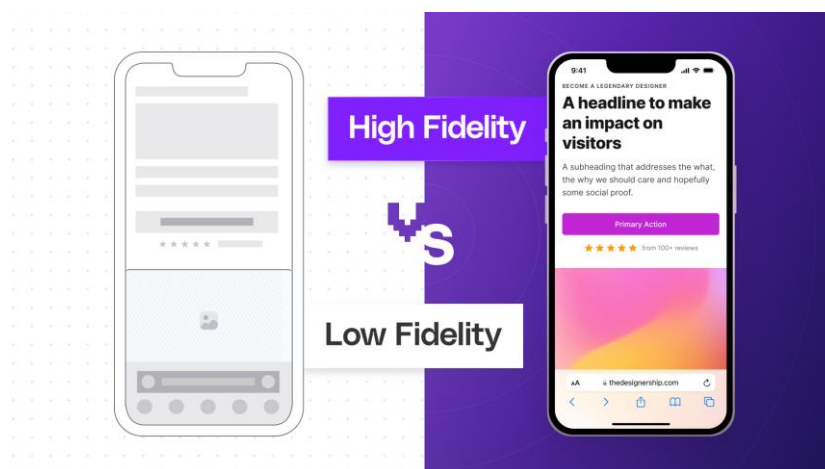
Simbol	Keterangan
Use Case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>.
Aktor 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri. Aktor merupakan peran yang dimainkan oleh pemakai ketika berinteraksi dengan sistem.
Asosiasi 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
Extend 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dan <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.

Include <<include>> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dan <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.
---	--

2.2.9 Wireframe

Wireframe adalah kerangka kerja desain yang digunakan untuk menyusun elemen-elemen pada halaman aplikasi sebelum proses desain utama dimulai. Pembuatan *wireframe* umumnya dilakukan dengan menggunakan alat desain seperti Figma. Secara visual, *wireframe* terdiri dari garis dan kotak yang menata tata letak elemen-elemen yang akan ada di dalam aplikasi (Rayhaan Yusri et al., 2024).

Wireframe terbagi menjadi dua jenis, yaitu *low fidelity* dan *high fidelity*. *Low fidelity wireframe* biasanya berbentuk sketsa sederhana yang hanya menampilkan struktur dasar tanpa elemen visual atau teks yang mendetail. Jenis ini digunakan pada tahap awal untuk mendiskusikan ide atau konsep. Sebaliknya, *high fidelity wireframe* memiliki detail yang lebih rinci dan mendekati desain akhir, termasuk penggunaan warna, tipografi, dan elemen visual lainnya (Rayhaan Yusri et al., 2024). Gambar 2.3 dibawah ini merupakan contoh gambaran perbedaan *wireframe low fidelity* dan *high fidelity*.



Gambar 2. 3 Low Fidelity VS Hight Fidelity

2.2.10 Framework Laravel

Menurut (Saefudin et al., 2023) laravel merupakan *framework* PHP bersifat open source yang dirilis di bawah lisensi MIT dan dibangun dengan konsep *MVC* (*Model View Controller*). Laravel juga dapat diartikan sebagai pengembangan *website* berbasis *MVC* yang ditulis dalam PHP yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pemeliharaan. Laravel merupakan *framework* PHP terbaik yang dikembangkan oleh Taylor Otwell. Sebagai sebuah *framework* PHP, laravel hadir sebagai platform pengembangan *website open source*. Ekspresi laravel dan sintaks yang elegan juga sangat menarik karena laravel ini dirancang khusus untuk menyederhanakan dan mempercepat proses pengembangan *website*.

Menurut (Subecz, 2021) laravel sebagai sebuah *framework* PHP, mengadopsi arsitektur *Model-View-Controller* (*MVC*), sebuah pola desain yang membagi pengembangan aplikasi menjadi tiga komponen utama dengan tanggung jawab yang berbeda. Komponen *view* berfokus pada logika antarmuka pengguna, bertanggung jawab atas presentasi data kepada pengguna. Komponen *controller* bertindak sebagai perantara, menerima input dari pengguna, memprosesnya, dan kemudian berinteraksi dengan model untuk mengambil atau memanipulasi data sebelum akhirnya memilih tampilan yang sesuai untuk merespons pengguna. Sementara itu, komponen *model* merupakan inti logika aplikasi yang secara langsung berhubungan dengan data, termasuk interaksi dengan basis data dan representasi data yang ditransfer antara *view* dan *controller*. Konfigurasi ketiga komponen ini secara terpisah bertujuan untuk mengelola kompleksitas pengembangan aplikasi dengan memisahkan aspek presentasi, logika bisnis, dan manajemen data.

2.2.11 MYSQL

Menurut buku *Manajemen Database MYSQL* (Diapoldo et al.), MySQL adalah sistem manajemen *database* yang sangat populer pada server *website*. Saat ini, banyak *website* yang populer di internet menggunakan MySQL karena memiliki

teknologi yang cukup baik. Salah satu alasan kesuksesan MySQL adalah dapat digunakan secara gratis, seperti penggunaan PHP.

MySQL adalah sistem manajemen *database* relasional (RDBMS) yang mampu mengelola banyak *database* secara bersamaan. Dalam konteks ini, satu *server* MySQL dapat mengelola beberapa *database* yang mungkin dimiliki oleh berbagai orang. Setiap *database* memiliki struktur untuk menyimpan dan mengelola data, termasuk struktur data itu sendiri.

MySQL bekerja dengan jantungnya berupa *server* MySQL yang bertindak sebagai pengelola sistem *database*, menangani semua instruksi *database*. Ketika ada permintaan seperti membuat *database* baru, kita mengirim pesan ke *server*, yang kemudian membuat *direktori* dan file yang diperlukan. Untuk menambahkan data, kita juga mengirim pesan ke *server* dengan data dan lokasi yang diinginkan. Agar dapat menerima instruksi, *server* MySQL harus berjalan dan terus mendengarkan permintaan. Biasanya, *server* ini diatur untuk otomatis berjalan saat komputer dinyalakan, terutama untuk kebutuhan situs *website*, namun juga bisa dijalankan secara manual saat dibutuhkan.

2.2.12 Black Box Testing

Menurut (Nur Ichsanudin et al., 2022) metode *Black Box Testing* merupakan pengujian untuk menunjukkan kesalahan pada sistem aplikasi seperti kesalahan pada fungsi sistem aplikasi, serta menu aplikasi yang hilang. Jadi *Black Box testing* merupakan metode uji fungsionalitas sistem aplikasi. Dalam melakukan pengujian menggunakan masukan data acak dengan tujuan untuk mendapatkan hasil yang pasti. Dikatakan pasti artinya bila salah, maka ditolak oleh sistem atau data input tersebut tidak dapat disimpan dalam *database*, sedangkan bila data input benar maka dapat diterima atau masuk di *database* sistem.

Teknik pengujian pada *Black Box Testing* ada banyak macamnya, berikut merupakan teknik pengujian menggunakan *Black Box Testing*:

- 1) Teknik *Equivalence Partitioning* yaitu dengan cara melakukan *partition* atau pembagian menjadi beberapa partisi dari input data.
- 2) Teknik *Boundary Value Analysis* yaitu dengan cara mencari adakah

error dari luar atau sisi dalam software, minimum maupun maximum nilai dari error yang ditemukan.

- 3) Teknik *Fuzzing* yaitu merupakan teknik untuk mencari *bug*/gangguan dari *software* dengan menggunakan injeksi data yang terbilang cacat .
- 4) Teknik *Cause-Effect Graph* adalah suatu teknik *testing* dimana menggunakan *graphic* sebagai acuannya. Dimana dalam grafik ini menggambarkan relasi antara efek dan penyebabnya.
- 5) Teknik *Orthogonal Array Testing* adalah jenis teknik yang digunakan jika input domain yang *relative* terbilang kecil ukurannya, tetapi cukup berat untuk digunakan dalam skala besar.
- 6) Teknik *All Pair Testing* yaitu semua pasangan dari test case di desain sedemikian rupa agar dapat dieksekusi semua kemungkinan kombinasi *diskrit* dari seluruh pasangan berdasar input parameternya, Tujuan testing ini adalah memiliki pasangan test case yang mencakup semua pasangan tersebut.
- 7) Teknik *State Transition* yaitu teknik ini berguna untuk melakukan pengetesan terhadap kondisi dari mesin dan navigasi dalam bentuk grafik.

Pada penelitian ini teknik Pengujian yang dilakukan pada Rancang Bangun Sistem Penjualan Online Toko Takashimura adalah Teknik *Equivalence Partitioning* & Teknik *Boundary Value Analysis*.

2.2.13 Usability Testing

Usability Testing adalah metode evaluasi yang digunakan untuk mengukur sejauh mana suatu produk atau sistem dapat digunakan oleh pengguna untuk mencapai tujuan tertentu dengan efektif, efisien, dan memuaskan. Dalam praktiknya, metode ini melibatkan pengamatan langsung terhadap pengguna saat mereka berinteraksi dengan produk atau sistem, diikuti dengan pengumpulan umpan balik untuk mengidentifikasi area yang memerlukan perbaikan. Tujuan utamanya adalah untuk meningkatkan pengalaman pengguna dan memastikan bahwa produk atau sistem memenuhi kebutuhan dan harapan pengguna secara optimal.

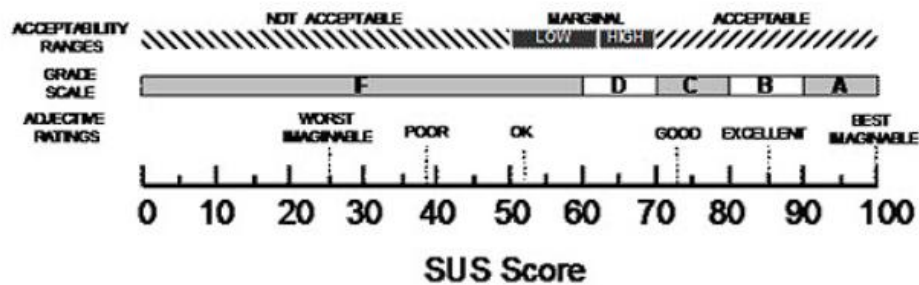
Salah satu alat yang sering digunakan dalam metode ini adalah *System Usability Scale (SUS)*. SUS adalah instrumen yang dikembangkan oleh John Brooke pada tahun 1986 untuk memberikan penilaian cepat terhadap sistem. SUS terdiri dari 10 pertanyaan dimana setiap pertanyaan dinilai menggunakan skala Likert (Brooke, 1995). Adapun 10 pertanyaan yang digunakan akan berdasarkan adaptasi Indonesia berdasarkan penelitian yang dilakukan oleh (Sharfina & Santoso, 2016). *Skala Likert* adalah alat pengukuran yang digunakan untuk mengukur sikap, pendapat, atau persepsi seseorang. Skala ini terdiri dari serangkaian pertanyaan yang diikuti oleh pilihan jawaban dengan tingkat persetujuan tertentu. Setiap pilihan jawaban diberikan skor numerik yang kemudian dapat dianalisis untuk menentukan kecenderungan sikap atau persepsi responden terhadap pernyataan yang diberikan. Berikut adalah tabel yang berisi contoh 5 butir penilaian dengan *skala likert* beserta dengan masing-masing nilainya.

Tabel 2. 3 Penilaian Skala Likert

SS	Sangat Setuju
S	Setuju
N	Netral
TS	Tidak Setuju
STS	Sangat Tidak Setuju

Adapun perhitungan skor SUS dilakukan dengan mengurangi skor dengan 1 untuk setiap pertanyaan bernomor ganjil (1, 3, 5, 7, 9) atau pertanyaan yang bersifat positif. Sementara, untuk pertanyaan bernomor genap (2, 4, 6, 8, 10) atau pertanyaan yang bersifat negatif, maka nilai 5 dikurangi dengan skor yang diberikan responden untuk menghitung nilai kontribusinya. Kemudian, semua skor dijumlahkan dan dikalikan dengan 2.5 untuk mendapatkan nilai akhir yang berada pada skala 0 hingga 100. Hasil akhir skor SUS yang didapatkan kemudian dikategorikan berdasarkan kategorisasi skor seperti yang ditunjukkan pada Gambar 2.4. *Grading score* atau kategorisasi ini dibuat oleh (Bangor, 2009) ,terdiri dari 3

kategori yaitu *adjective ratings*, *grade scale*, *acceptability ranges*. Adapun tabel 2.4 merupakan tabel yang dibuat berdasarkan Gambar 2.4, yang terdiri dari kategori penilaian SUS.



Gambar 2. 4 Nilai SUS(Bangor, 2009)

Tabel 2. 4 Keterangan Penilaian SUS

Skor SUS	Acceptability	Grade	Skor SUS	Keterangan
0-50	<i>Not Acceptable</i>	F	0-25	<i>Worst Imaginable</i>
50-60	<i>Marginal (Low)</i>		26-39	<i>Poor</i>
60-70	<i>Marginal (High)</i>	D	40-51	OK
70-80	<i>Acceptable</i>	C	52-73	<i>Good</i>
80-90		B	74-84	<i>Excellent</i>
90-100		A	85-100	<i>Best Imaginable</i>