

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Dengan tujuan untuk mendukung gagasan bahwa penelitian ini mempersembahkan kontribusi baru dan signifikan dalam bidangnya, telah dilakukan penelusuran terhadap sejumlah penelitian terdahulu yang memiliki relevansi langsung dengan topik yang dibahas. Informasi rinci mengenai penelitian-penelitian tersebut dijabarkan secara terperinci dalam Tabel 2.1.

Tabel 2.1 Penelitian terdahulu

Penelitian 1	
Peneliti	Yokeshwaran, M., & Murugachandavel, J.
Jurnal	International Journal of Health Sciences, Vol.6, No.2, 3250-3260 (2022)
Judul Artikel	Enhancement of Business Using E-commerce with Point of Sale [16]
Metode yang Digunakan	Kualitatif
Masalah	Penelitian ini menghadapi permasalahan dalam mengadopsi integrasi <i>e-commerce</i> dan sistem <i>point of sale</i> (POS) untuk produk medis. Banyak bisnis belum menggunakan teknologi ini sepenuhnya, menyebabkan efisiensi, produktivitas, dan hubungan pelanggan yang kurang optimal, serta membuat proses penjualan, pelacakan stok, dan transaksi pembayaran lambat dan rentan terhadap kesalahan.
Hasil Penelitian	Makalah ini berfokus pada perancangan, pengembangan, dan pemeliharaan <i>E-commerce</i> khusus produk medis seperti masker oksigen, termometer, dan lain sebagainya. Penelitian ini menggunakan Angular sebagai kerangka kerja sisi klien dan PHP, MySQL, dan OpenCart di sisi server untuk pengembangan API.
Penelitian 2	
Peneliti	L. Ranjaliba Saragih, M. Dachyar, Teuku Yuri M. Zagloel
Jurnal	Heliyon, Vol.7, No.5 (2021)
Judul Artikel	Implementation of telecommunications cross-industry collaboration through agile project management [17]
Metode yang Digunakan	Agile Dynamic Systems Development Method
Masalah	Penelitian ini menghadapi tantangan utama bagi penyedia layanan telekomunikasi (TSP) dalam beralih dari melayani pelanggan manusia ke basis pelanggan IoT non-manusia di tengah gangguan digital. Transformasi bisnis sering terhambat oleh manajemen perubahan yang buruk, resistensi internal, masalah teknis, dan konflik antara kerangka kerja strategis dan kebutuhan bisnis.

Hasil Penelitian	Penelitian ini memusatkan perhatian pada layanan telekomunikasi antar industri dengan pendekatan <i>Agile Dynamic Systems Development Method</i> (DSDM), dengan tujuan mengembangkan metode yang dapat diimplementasikan pada proyek kerjasama lintas industri telekomunikasi berbasis IoT.
Penelitian 3	
Peneliti	Jessica Lourenco, Aparna S. Varde
Jurnal	2020 IEEE International Conference on Big Data (Big Data) (2020)
Judul Artikel	Item-Based Collaborative Filtering and Association Rules for a Baseline Recommender in E-Commerce [18]
Metode yang Digunakan	Item-Based Collaborative Filtering
Masalah	Penelitian ini menghadapi masalah membantu pengguna menemukan produk relevan di tengah volume data <i>e-commerce</i> yang besar, menggunakan recommender systems untuk menyarankan item berdasarkan popularitas dan perilaku pembelian.
Hasil Penelitian	Penelitian ini berfokus pada implementasi <i>item-based collaborative filtering</i> dan <i>association rule mining</i> terhadap data produk seperti ponsel dan aksesoris elektronik di situs web Amazon pada masa pandemi. Pada tahapan implementasinya, penelitian ini menggunakan <i>Pearson correlation-based similarity</i> dalam memetakan produk-produk yang akan direkomendasikan kepada pengguna.
Penelitian 4	
Peneliti	Isnaini Hayati, Ayuningtyas, Erwin Sutomo
Jurnal	Jurnal Informatika Universitas Pamulang, Vol.7, No.2, 374-381 (2022)
Judul Artikel	Penerapan User Centered Design Pada Perancangan Web E-Commerce Alat Kesehatan [19]
Metode yang Digunakan	User Centered Design
Masalah	Penelitian ini berfokus pada pengembangan website CV Andi Jaya yang mengalami kendala, seperti proses penjualan barang yang tidak lancar dan tampilan yang kurang menarik. Solusinya adalah merancang <i>prototype</i> desain website baru dengan metode <i>User-Centered Design</i> (UCD) untuk memenuhi kebutuhan pengguna.
Hasil Penelitian	Penelitian ini berfokus pada pengembangan desain antarmuka pengguna <i>e-commerce</i> berbasis web bagi sebuah badan usaha yang bergerak di bidang alat kesehatan menggunakan pendekatan <i>User Centered Design</i> .
Penelitian 5	
Peneliti	Pragati Budhe, Rohit Wasnik, Yash Lilhare, Hitanshu Pande, Rohit Motghare, Sahil Shinde
Jurnal	SSGM Journal of Science and Engineering, Vol.1, No.1, 136-141 (2023)
Judul Artikel	E-commerce With Auction – Web Application Using MERN Technology [20]
Metode yang Digunakan	SDLC Waterfall Model
Masalah	Masalah yang dihadapi adalah kebutuhan bisnis untuk meningkatkan pengalaman belanja pelanggan dengan cara yang unik dan menarik. Salah satu pendekatannya

	adalah dengan mengimplementasikan sistem lelang dalam platform <i>e-commerce</i> .
Hasil Penelitian	Penelitian ini berfokus pada pengembangan web <i>e-commerce</i> menggunakan <i>MERN stack framework</i> dengan mempertimbangkan kelebihan kerangka kerja ini dalam melakukan skalabilitas dan cocok di berbagai platform.

Penelitian yang dilakukan di PT Sinar Ogan Industri menggunakan metode-metode yang pernah digunakan sebelumnya namun dengan penerapan pada objek dan industri yang berbeda. Penelitian ini berfokus di industri alat kesehatan spesifik pada alat kesehatan sekali pakai dimana penelitian terdahulu melakukan penelitiannya di skala industri alat kesehatan yang lebih luas. Lalu, dari penelitian yang dilakukan oleh Isnaini, Ayuningtyas, dan Erwin, terdapat kebaruan di sisi antarmuka web dimana penelitian terdahulu hanya desain antarmuka sedangkan pada penelitian ini, perancangan antarmuka pengguna sampai pada tahap *deployment* dan dapat berfungsi sebagaimana web *e-commerce* diterapkan. Pemilihan kerangka kerja MERN dipilih berdasarkan hasil perbandingan pada penelitian [21] yang menunjukkan bahwa MERN memiliki tingkat fleksibilitas, skalabilitas, dan kemudahan pemeliharaan yang lebih tinggi dibandingkan kerangka kerja lain seperti LAMP. Juga pada penelitian yang dilakukan oleh Yokeshwaran dan Murugachandavel [16] masih menggunakan kerangka kerja Angular di sisi klien dan OpenCart pada pengembangan API dimana hal ini dirasa kurang efektif dari segi pemeliharaan sistem. Di sisi lain, kerangka kerja yang mirip dengan MERN yaitu MEAN kurang direkomendasikan untuk proyek yang berskala kecil dikarenakan tingkat kompleksitas yang kurang cocok bagi pengembangan dalam waktu singkat. Lebih daripada itu, penelitian ini juga menggabungkan dua metode pada pengembangan web *e-commerce* PT Sinar Ogan Industri yaitu *Agile Dynamic Systems Development Method* sebagai metode dalam pengembangan web dari sisi klien sampai dengan server dan *Cross-Industry Standard Process for Data Mining* (CRISP-DM) sebagai metode dalam pengembangan model *machine learning* pada sistem rekomendasi produk di web *e-commerce* PT Sinar Ogan Industri.

2.2 Teori tentang Topik Skripsi

2.2.1 E-Commerce

Berdasarkan penelitian yang diadakan oleh Fatty Rosyana Okatvia, *e-commerce* digambarkan sebagai penggunaan internet dan web untuk menjalankan bisnis di pasar global. *E-commerce* telah mengubah cara bisnis beroperasi di zaman modern [22]. Dengan *e-commerce*, platform ini memberikan kesempatan kepada penjual untuk mencapai audiens yang lebih besar dan memiliki potensi mendapatkan nilai jual yang lebih tinggi untuk produk mereka, sedangkan bagi pembeli, ada peluang untuk memperoleh barang dan layanan dengan potensi harga lebih murah melalui persaingan penawaran [20]. Terdapat beberapa jenis *e-commerce* antara lain B2B, B2C, C2C, *social e-commerce*, *retail e-commerce*, dan lainnya. Namun, terdapat empat jenis *e-commerce* utama yaitu sebagai berikut:

- B2B (*Business-to-Business*): model bisnis yang menawarkan layanan transaksi jual-beli dari satu bisnis kepada bisnis lainnya.
- B2C (*Business-to-Customer*): model bisnis yang menawarkan layanan transaksi jual-beli dari satu bisnis kepada konsumen individu.
- C2C (*Customer-to-Customer*): model bisnis yang menawarkan layanan transaksi jual-beli dari satu individu kepada individu lainnya.
- C2B (*Customer-to-Business*): model bisnis yang menawarkan layanan transaksi jual-beli dari satu individu kepada sebuah perusahaan atau bisnis [23].

2.2.2 Inventaris

Inventaris merupakan daftar barang atau aset yang dimiliki oleh suatu badan usaha dan biasanya berisikan nama, harga, dan terutama stok barang. Dan kegiatan pencatatan yang dilakukan oleh suatu badan usaha dalam proses pengelolaan barang kepemilikan disebut inventarisasi. Tujuan dari adanya inventaris ini salah satunya untuk memberi identifikasi khusus pada suatu unit produk maupun fasilitas yang terhubung pada suatu badan usaha. Oleh karenanya, inventaris gudang harus memiliki informasi yang dapat dipahami dengan cepat, mudah dan juga jelas sehingga dapat membantu efektivitas dan

efisiensi kegiatan di pergudangan. Adanya inventaris juga ditujukan untuk mendukung pengontrolan dan pemeliharaan barang [24], [25].

2.2.3 Website

Menurut Deborah E. Rosen dan Elizabeth Purinton (2004), “*By whatever measure used, the web is big.*” Dimana hal ini berarti tidak ada yang mampu mengukur seberapa besar suatu situs web yang telah dikembangkan sampai saat ini [26]. *Website* atau yang lebih dikenal sebagai *World Wide Web* (WWW) merupakan sekumpulan halaman yang menyimpan dan menyuguhkan berbagai informasi yang disematkan oleh pengembang dan informasi ini dapat diakses oleh siapapun melalui jaringan internet. Di dalam suatu situs web, terdapat beberapa unsur pendukung yang cukup penting diantaranya *domain*, *hosting*, konten, bahasa pemrograman, dan *user interface*. Di era teknologi yang selalu berkembang ini, situs web terus menjadi salah satu unsur esensial yang seharusnya dimiliki oleh suatu badan usaha perseorangan maupun perusahaan. Manfaat dari adanya *website* ini diantaranya membangun *branding*, menjadi platform berbisnis, sumber informasi yang kredibel, saran pengembangan bisnis, dan lainnya [27].

Sebuah *website* dikembangkan bukan semata-mata hanya untuk mendapatkan manfaat darinya namun terdapat beberapa unsur tekstual di dalamnya seperti GUI (*graphical user interface*) dan berbagai elemen signifikan seperti *icon* yang bergerak, laman yang bisa disesuaikan dengan skala atau ukuran jendela *browser*, dan lainnya. *Browser* juga berperan penting dalam menjalankan sebuah situs web dengan cara menggunakan *transfer protocols* (contohnya HTTP dan URL) dan menejemahkan apa yang ada di dalam *script code* (yang kebanyakan menggunakan HTML dan CSS) seperti teks, gambar, posisi objek, dll [28]. Secara umum terdapat beberapa bahasa pemrograman yang dapat digunakan untuk membuat sebuah *website*. Yang akan digunakan pada penelitian ini antara lain HTML (Hyper Text Markup Language), CSS (Cascading Style Sheets), dan JavaScript.

2.3 Teori Tentang Framework dan Bahasa Pemrograman yang Digunakan

2.3.1 MERN Stack Framework

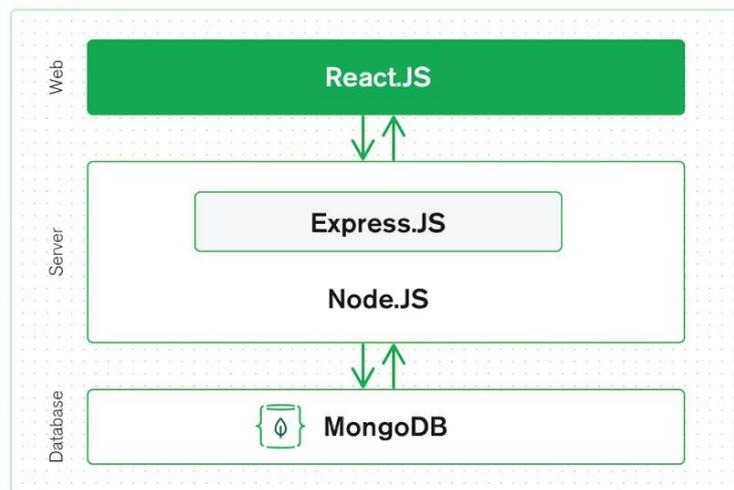


Gambar 2.1 MERN Stack Framework

MERN *stack framework* merupakan kerangka kerja dalam pengembangan situs web yang terdiri dari empat teknologi yaitu MongoDB, Express.js, React.js, dan Node.js. Di dalam terminologi pengembangan web, stack merujuk pada kumpulan teknologi yang digunakan untuk membuat sebuah web. Dalam pengembangan suatu web terdapat dua sisi pengembangan yaitu *front end* dan *back end*. Pengembangan *front end* berfokus pada apa yang dapat dilihat dan ditampilkan di sisi klien seperti *user interface*. Sedangkan pengembangan *back end* berfokus pada bagian yang tidak dapat dilihat oleh klien atau pengguna seperti *database*, *server*, dan lainnya. Dalam lingkup MERN, sisi *front end* terdiri dari React.js, sedangkan MongoDB, Express.js, dan Node.js berada di sisi *back-end* [29], [30]. MERN *stack framework* memiliki beberapa keunggulan antara lain:

- Dikembangkan sepenuhnya menggunakan bahasa pemrograman tunggal, JavaScript di sisi *front-end* dan *back-end* yang juga akan mempermudah proses pengembangan.
- Menggunakan Node.js di *server-side* yang membuat *tech stack* ini memiliki fleksibilitas, skalabilitas, dan tingkat performa yang tinggi.
- Memiliki komunitas *developer* aktif yang luas dan secara tidak langsung menyediakan akses ke banyak sumber daya terkait.
- Menggunakan arsitektur modular yang memungkinkan *developer* untuk lebih mudah menggunakan kode dan komponen berulang kali yang juga berpengaruh pada proses pengembangan yang lebih singkat [12].

MERN *architecture* merupakan arsitektur yang memungkinkan *developer* untuk mengembangkan *three-tier architecture* (*front end*, *back end*, *database*) menggunakan JavaScript dan JSON. Tiga lapisan ini masing-masing terdiri dari React.js di lapisan teratas sebagai *front end*, Express.js yang berjalan di dalam Node.js di lapisan berikutnya sebagai *back end*, dan MongoDB di lapisan terakhir sebagai *database* [31].



Gambar 2.2 MERN *architecture*

Berdasarkan pada Gambar 2.2, alur kerja dari MERN *architecture* yang pertama yaitu pengguna melakukan interaksi dengan komponen yang dikembangkan dengan React.js di sisi *front end*, lalu dari interaksi yang dilakukan pengguna, request untuk perubahan data dikirim ke Express.js yang dikembangkan dengan Node.js. Saat diperlukan, dari sisi server akan mengambil data dari MongoDB dan mengirimnya ke sisi *front end* yang ditampilkan kepada pengguna [32].

2.3.1.1 MongoDB

MongoDB adalah sebuah *database* NoSQL. MongoDB yang juga merupakan *document-oriented database* menggunakan koleksi dan dokumen, bukan tabel dan baris selayaknya *database* SQL pada umumnya. MongoDB menyimpan data dalam format JSON (*JavaScript Object Notation*) yang memungkinkan pertukaran data yang cepat antara klien dan server [33]. MongoDB memiliki beberapa fitur unik yaitu *indexing*, *aggregation*, *special collection types*, dan *file storage*. MongoDB juga dapat

mempermudah *developer* untuk fokus pada proses pengembangan aplikasi karena dengan model data yang berorientasi pada dokumen mempermudah data untuk berpisah melintasi lebih dari satu server didukung dengan kemampuannya dalam melakukan penyeimbangan pada data dan load secara otomatis [34]. *Library* yang digunakan pada pengembangan web *e-commerce* ini yaitu Mongoose yang merupakan kerangka kerja JavaScript yang berfungsi untuk memudahkan pemodelan data di MongoDB dan memfasilitasi interaksi antara aplikasi Node.js dan *database* MongoDB [35].

2.3.1.2 Express.js

Express.js adalah kerangka kerja Node.js yang berperan sebagai *middleware* pada sebuah *browser* dan digunakan untuk mengatur URL *routes*. Pengaturan model URL *routes* ini diimplementasi menggunakan metode RESTful yang memiliki beberapa metode HTTP *request* yaitu GET, POST, DELETE, PUT, PATCH, dan lainnya [29]. Di sisi pengembangan web, Express.js bertindak sebagai *back-end* yang bekerja secara asinkron. Fitur yang dimiliki oleh Express.js yaitu dapat bekerja di web dengan pendekatan *single* maupun *multi-page* dan fleksibilitas yang tinggi [21].

2.3.1.3 React.js

React.js merupakan *library* JavaScript yang dikembangkan oleh Facebook untuk memecahkan masalah yang ada saat mengembangkan tampilan antarmuka yang kompleks dengan data yang silih berganti di tiap interaksi. React.js umumnya digunakan dalam pendekatan antarmuka *single page application* [33], [36]. React.js memiliki beberapa fitur dan poin yang membedakannya dari yang lain antara lain:

- Mendukung penggunaan JSX, *markup syntax* yang mirip dengan XML yang memungkinkan *developer* untuk menulis JavaScript yang lebih mudah dibaca.
- Menggunakan Virtual DOM untuk menghindari operasi DOM yang tidak diperlukan saat memperbarui tampilan antarmuka.

- Memisahkan kompleksitas komponen antarmuka dalam bentuk komponen yang dapat digunakan kembali saat diperlukan dan ada pembaharuan di sisi antarmuka pengguna [37].

Fitur-fitur React.js membuatnya cocok untuk tampilan antarmuka berbasis pada data dengan struktur kode yang ramping dan efektif. Pada penelitian ini, ada beberapa dependensi React dan *library* yang digunakan yaitu React Hooks, React Router DOM, dan Axios. React Hooks adalah sebuah fungsi yang memungkinkan *developer* untuk mengakses *state* dan siklus hidup komponen tanpa perlu menggunakan *class* [38]. React Router DOM adalah *package* yang mengimplementasikan navigasi dinamis di web dengan menggunakan komponen [39]. Axios merupakan *package* yang memungkinkan pengguna untuk mengirim *request* HTTP berbasis Promise dari *browser* dan Node.js [40].

2.3.1.4 Node.js

Node.js merupakan *runtime* JavaScript yang bersifat asinkron dan menggunakan mesin V8 dari Google dalam pengeksekusiannya. Node.js bertindak sebagai *back end* atau *server-side* dan menggunakan arsitektur *event-driven* untuk menangani banyak *request* dari pengguna secara bersamaan dan akan menaruh semua *request* ini di dalam *event queue* [41], [42]. *Runtime* JavaScript ini juga memiliki koleksi modul yang banyak seperti modul HTTP, utilitas, dan lainnya yang dapat meningkatkan efisiensi durasi perancangan *back end*. Node.js juga memiliki Node Package Manager atau NPM yang merupakan *package manager* JavaScript dari Node.js [43]. Salah satu *package* yang berkorelasi dengan pengembangan Node.js dan digunakan dalam penelitian ini yaitu Nodemon yang berfungsi untuk menjalankan ulang aplikasi Node.js bila terdapat perubahan pada *file* sistem secara otomatis saat proses pengembangan [44].

2.3.2 Flask

Flask merupakan *microframework* sekaligus *library* yang ditulis dalam bahasa pemrograman Python. Walaupun dikenal sebagai kerangka kerja mikro,

Flask adalah kerangka kerja yang sangat sederhana namun sangat mungkin untuk diperluas penggunaannya. Flask terdiri dari dua komponen utama, yaitu Werkzeug dan Jinja2. Werkzeug bertanggung jawab atas penyediaan *routing*, *debugging*, dan Web Server Gateway Interface (WSGI), sementara Jinja2 digunakan sebagai *template machine*. Dengan kombinasi ini, Flask dapat memberikan kerangka kerja yang kuat untuk pengembangan aplikasi web [45]. Pada penelitian ini, Flask digunakan dalam pembuatan API dari model *machine learning* rekomendasi produk di situs web *e-commerce* perusahaan.

2.3.3 HTML

HyperText Markup Language atau yang umumnya disingkat HTML merupakan fondasi dari semua konten yang ditampilkan di *World Wide Web* (WWW). Dikutip dari buku karya David R. Brooks, HTML terdiri dari dua bagian penting yaitu konten yang berisi informasi dan sekumpulan instruksi yang memberi perintah pada komputer bagaimana cara menampilkan konten tersebut. Pada implementasinya, dokumen HTML memerlukan setidaknya empat elemen yaitu dokumen itu sendiri, bagian *heading*, *title*, dan *body* [46]. Setiap elemen HTML dimulai dan ditutup dengan sebuah *tag* “<...>” [47].

2.3.4 CSS

Cascading Style Sheet (CSS) adalah bahasa yang digunakan untuk menentukan tampilan dan format pada laman situs web. CSS memiliki keterkaitan dengan HTML dari segi visual web karena CSS menawarkan lebih banyak variasi tampilan [48]. *Style sheet* sendiri merupakan file yang terdiri dari sejumlah definisi yang mengubah tampilan halaman web. Cara kerja CSS pada dasarnya yaitu dengan mengambil *tag* HTML dan menetapkan beberapa properti seperti warna, ukuran, dan properti CSS lainnya [49].

2.3.5 JavaScript

JavaScript atau ECMAScript merupakan skrip atau bahasa pemrograman yang berorientasi pada objek (*object-oriented programming*) dan umumnya digunakan untuk memberi esensi pada suatu laman web agar

lebih interaktif. Bahasa pemrograman ini termasuk dalam lapisan standar teknologi pada web di sisi klien (*client-side*) yaitu HTML, CSS, dan JavaScript [46]. JavaScript dikembangkan pertama kali oleh Netscape yang berkolaborasi dengan Sun Microsystem. Sebelum JavaScript hadir, yang ditampilkan pada suatu web yaitu hanyalah *hyper-text document*. Selaku bahasa pemrograman dengan basis skrip dalam penulisannya, cara kerja JavaScript dapat diumpamakan dengan memberi tahu apa yang harus dilakukan oleh sebuah *web browser*. Sebagai contoh, bahasa pemrograman ini dapat digunakan untuk memanipulasi properti-properti HTML seperti tinggi, lebar, dan posisi properti tersebut [50].

2.3.6 Python

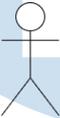
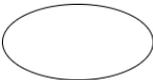
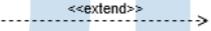
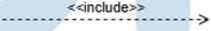
Python merupakan salah satu bahasa pemrograman yang paling banyak digunakan dan cocok untuk pemula. Python juga diklasifikasikan sebagai bahasa pemrograman yang ekspresif dengan artian *developer* dapat menuliskan jauh lebih sedikit baris kode untuk dapat menghasilkan aplikasi yang ditulis dalam bahasa pemrograman lain seperti C++ atau Java. Python hadir dengan standar *library* yang lengkap yang memungkinkan pengembang untuk mengunduh file dari internet, membuat server web, dan fitur lainnya. Lebih dari pada itu, Python juga memiliki ribuan *third-party library* seperti NumPy, SimPy, lxml, dll. Bahasa pemrograman ini dapat digunakan secara prosedural, berorientasi pada objek sampai dengan gaya fungsional. Namun pada hakikatnya, Python sendiri merupakan bahasa pemrograman berbasis *object-oriented* [51].

2.3.7 Unified Modelling Language (UML)

Unified Modeling Language (UML) adalah bahasa grafis standar yang digunakan untuk memvisualisasikan, menentukan, membangun, dan mendokumentasikan artefak dari sistem perangkat lunak yang kompleks. UML memungkinkan pengembang dan pemangku kepentingan lainnya untuk membuat cetak biru sistem yang mencakup berbagai aspek, mulai dari konsep abstrak seperti proses bisnis dan fungsi sistem hingga elemen konkret seperti

kelas dalam bahasa pemrograman tertentu, skema basis data, dan komponen perangkat lunak yang dapat digunakan kembali. Dengan UML, semua pihak yang terlibat dalam pengembangan perangkat lunak dapat memiliki pemahaman yang konsisten dan terstruktur mengenai desain dan fungsi sistem. Bahasa pemodelan ini diciptakan dan dikembangkan dalam upaya menyederhanakan beberapa metode pengembangan *object-oriented* terdahulu. Unified Modelling Language terdiri dari banyak jenis diagram seperti *use case diagram*, *class diagram*, *sequence diagram*, *object diagram*, *state diagram*, dan diagram lainnya [52]. Diagram yang akan digunakan pada penelitian ini yaitu *use case diagram* untuk memodelkan proses bisnis, *activity diagram* untuk memodelkan *use cases behavior* dan objek pada sistem, dan *class diagram* untuk memodelkan struktur kelas.

Tabel 2.2 *Use Case Diagram*

Simbol	Keterangan
	<i>Actor</i>
	<i>Use Case</i>
	<i>Association</i>
	<i>Inheritance</i>
	<i>Extend</i>
	<i>Include</i>

Tabel 2.2 merupakan ilustrasi dari simbol-simbol yang terdapat pada *use case diagram*. *Use case diagram* merupakan diagram dalam UML yang digunakan untuk memodelkan perilaku dinamis dari sistem. Diagram ini menampilkan *use cases* (fungsi atau layanan yang disediakan oleh sistem) dan *actors* (entitas eksternal yang berinteraksi dengan sistem), serta hubungan di antara mereka. Fungsi utamanya adalah untuk memodelkan konteks dan

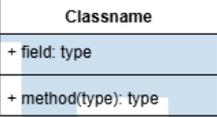
kebutuhan perilaku sistem, subsistem, atau kelas. Diagram ini membantu dalam visualisasi, spesifikasi, dan dokumentasi perilaku sistem, serta dalam pengujian dan pemahaman sistem yang telah dieksekusi. *Use case diagram* membuat elemen sistem lebih mudah dipahami dengan menunjukkan bagaimana mereka digunakan dalam konteks nyata [52].

Tabel 2.3 *Activity Diagram*

Simbol	Keterangan
	<i>Activity</i>
	<i>Initiate Activities</i>
	<i>Start Process</i>
	<i>Final Process</i>
	<i>Concurrent Bar</i>
	<i>Sequential Branch</i>

Tabel 2.3 merupakan ilustrasi dari simbol-simbol yang terdapat pada *activity diagram*. *Activity diagram* adalah diagram yang mirip dengan *flowchart*, menampilkan alur kontrol dari satu aktivitas ke aktivitas lainnya. Fungsi utama *activity diagram* adalah untuk memodelkan aspek dinamis sistem, khususnya langkah-langkah berurutan (dan mungkin bersamaan) dalam sebuah proses komputasi. Diagram ini juga dapat memodelkan aliran objek saat berpindah dari satu status ke status lain di berbagai titik dalam alur kontrol. *Activity diagram* bisa berdiri sendiri untuk memvisualisasikan, menentukan, membangun, dan mendokumentasikan dinamika sekumpulan objek, atau digunakan untuk memodelkan alur kontrol suatu operasi. Fokus dari *activity diagram* adalah pada alur kontrol dari aktivitas ke aktivitas, dimana diagram ini penting untuk memodelkan aspek dinamis sistem dan juga untuk membangun sistem yang hendak dieksekusi [52].

Tabel 2.4 *Class Diagram*

Simbol	Keterangan
	<p><i>Class</i> (<i>Classname, Attribute, Method</i>)</p>
	<p><i>Association</i></p>
	<p><i>Aggregation</i></p>
	<p><i>Generalization</i></p>
	<p><i>Interface</i></p>

Tabel 2.4 merupakan ilustrasi dari simbol-simbol yang terdapat pada *class diagram*. *Class diagram* merupakan diagram yang paling umum digunakan dalam pemodelan sistem berorientasi objek. Diagram ini menunjukkan sekumpulan kelas, antarmuka, kolaborasi, dan hubungan di antara mereka. Fungsi utama *class diagram* adalah untuk memodelkan pandangan desain statis dari sebuah sistem. Ini melibatkan pemodelan kosakata sistem, pemodelan kolaborasi, atau pemodelan skema. *Class diagram* penting untuk memvisualisasikan, menentukan, dan mendokumentasikan model struktural [52].

2.3.8 RESTful API

RESTful API (atau REST API) merupakan API yang menggunakan REST (REpresentational State Transfer) *architectural style*. Saat ini, RESTful API merupakan salah satu jenis API yang saat ini banyak digunakan dibandingkan dengan tipe arsitektur API terdahulu untuk mengelola sumber daya, memungkinkan sisi klien untuk mengakses dan memanipulasinya menggunakan *stateless operation* [53]. REST API memanfaatkan prinsip-prinsip dan kendala-kendala tertentu untuk mendorong kesimpulan, skalabilitas, dan *statelessness* dalam perancangannya. Komunikasi yang

dilakukan antar klien dan server dengan RESTful API menggunakan protokol HTTP seperti GET, POST, PUT, PATCH, dan DELETE dalam operasinya. Adapun API eksternal yang juga digunakan dalam penelitian ini yaitu Midtrans sebagai *payment gateway* dan *shipping aggregator*, RajaOngkir sebagai kalkulator ongkos pengiriman dari PT Sinar Ogan Industri ke tempat tujuan [54].

2.4 Teori Tentang Tools dan Software yang Digunakan

2.4.1 Visual Studio Code

Visual Studio Code (VS Code) adalah *text atau code editor* yang dikembangkan oleh Microsoft untuk membantu proses perancangan struktur kode di berbagai sistem operasi. Beberapa sistem operasi populer yang didukung oleh VS Code yaitu Microsoft, Windows, Linux, dan macOS. Fitur-fitur yang ada di dalam VS Code antara lain *debugging*, sintaksis, *IntelliSense code completion*, *snippets code*, *refactoring*, indentasi otomatis, pencocokan tanda kurung, penyorotan, dan lainnya. Visual Studio Code juga mendukung penulisan kode dalam banyak bahasa pemrograman antara lain, HTML, CSS, JavaScript, C++, Go, dan bahasa pemrograman lainnya [55].

2.4.2 Postman

Postman adalah platform yang dirancang untuk mengembangkan, membangun, menguji, dan memodifikasi *Application Programming Interface* (API). Platform ini memiliki kemampuan untuk membuat berbagai jenis *request* HTTP yaitu GET, POST, PUT, PATCH, DELETE, dan lainnya [56]. Selain itu, Postman menyediakan *environment* yang terisolasi untuk mensimulasikan berbagai kondisi dan skenario, memungkinkan *developer* untuk menguji API mereka di berbagai situasi. Dengan dukungan untuk otomatisasi pengujian dan kolaborasi tim, Postman menjadi alat yang sangat berguna dalam siklus pengembangan perangkat lunak.

2.4.3 Git

Git merupakan sistem *version control* terdistribusi yang cepat, terukur, dengan rangkaian *command* yang menyediakan operasi tingkat tinggi dan akses penuh ke internal yang diciptakan oleh Linus Torvalds. Git memiliki beberapa *command line* yang sering digunakan dalam mengelola *repository*-nya antara lain *git init*, *git push*, *git commit*, *git pull*, *git clone*, *git checkout*, dan *command line* lainnya. Fungsi Git secara umumnya yaitu memantau seluruh perubahan yang terjadi pada file proyek disimpan sebuah *repository* yang juga secara tidak langsung akan memudahkan sebuah kolaborasi antar *developer* di dalam suatu proyek [57]. Platform kolaboratif berbasis web yang menggunakan sistem *version control* Git yang juga digunakan sebagai repositori pada penelitian ini adalah GitHub.

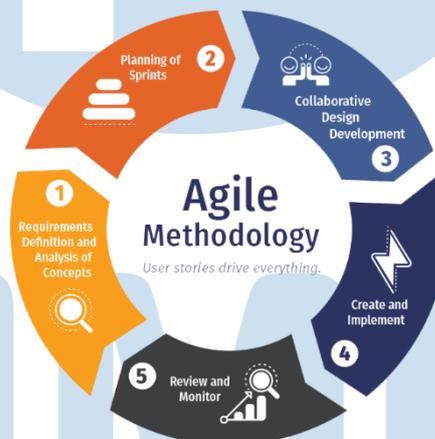
2.4.4 Amazon S3

Amazon Simple Storage Service atau yang umumnya diketahui sebagai Amazon S3, merupakan layanan penyimpanan objek berbasis *cloud*. Layanan ini ditawarkan oleh Amazon Web Services (AWS). Amazon S3 menawarkan skalabilitas dan keamanan yang tinggi. Hal ini didukung dengan kemampuan penyimpanan ini untuk memberikan akses kepada pengguna dalam menentukan dimana wilayah penyimpanan dan siapa saja yang dapat mengakses data yang disimpan di dalamnya. Tujuan dari hal ini yaitu untuk mengurangi periode waktu pada penyimpanan dan penarikan data. Konsep umum Amazon S3 terbagi menjadi dua yaitu *buckets* dan *objects*. *Buckets* merupakan kontainer tempat file disimpan. Di sisi lain, *objects* merupakan file yang disimpan di dalam Amazon S3 dan tiap objek memiliki pengidentifikasi unik. Pada penelitian ini, *buckets* digunakan untuk menyimpan file-file gambar produk dan aset-aset media lainnya [58].

2.5 Teori Mengenai Rancangan Sistem

2.5.1 Agile Software Development Method

Di pertengahan tahun 1990-an, beberapa konsultan dan pengembang secara independen membuat dan menemukan metode yang kini dikenal dengan *agile software development methodologies*. Poin utama pada metode ini yaitu pengembangan dalam waktu yang singkat dan responsif pada berbagai kemungkinan dalam berubahnya kebutuhan dalam proses pengembangan. Metodologi pengembangan Agile memiliki beberapa metode di dalamnya yang salah satunya adalah *Dynamic Systems Development Method (DSDM)* yang juga merupakan metode yang akan digunakan pada penelitian ini. Metode Agile juga memiliki beberapa 4 nilai penting di dalamnya yaitu *individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, dan responding to change over following a plan.*

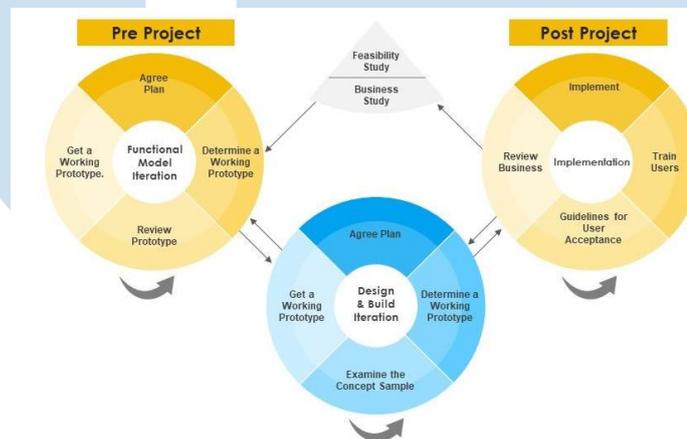


Gambar 2.3 Agile Methodology

Pada penelitian dan studi pustaka terdahulu, terdapat satu poin yang menjadi poin penting dalam pengembangan sistem ini yaitu *iteration demonstration* dimana poin ini berarti akan berulang kali memberikan hasil sementara dari proses pengembangan sistem informasi untuk mendapatkan *feedback* dari audiens atau objek penelitian. *Iteration demonstration* dimulai dengan MVP (*minimal viable product*) yang akan menjadi titik mula proses *feedback* berkala dan proses pengembangan sistem diadakan [14]. Metode

Agile memiliki beberapa kelebihan yang dapat menunjang suatu proses pengembangan sistem informasi pada pengguna antara lain:

- Adaptif pada kebutuhan yang silih berganti seiring berjalannya waktu.
- Pengembangan yang berfokus pada kepuasan pengguna.
- Berfokus pada desain dan keunggulan teknis.
- Diskusi berulang dengan pihak pengguna untuk memastikan *requirement* dengan baik.
- Tidak memakan biaya yang banyak.
- Perkembangan kinerja *developer* dapat dilihat melalui kinerja dalam pengembangan perangkat lunak, dll [59].



Gambar 2.4 Agile Dynamic Systems Development Method

Agile yang memiliki metode yang banyak di dalamnya juga memiliki beberapa klasifikasi tertentu dalam penggunaan metodenya. Salah satu metode agile yang memiliki keterlibatan langsung pada proses bisnis yaitu *Dynamic Systems Development Method (DSDM)* yang memiliki beberapa tahap atau fase dalam siklusnya yaitu *feasibility*, *business study*, *functional model iteration*, *design and build iteration*, sampai dengan fase implementasi sistem di PT Sinar Ogan Industri. DSDM sendiri memiliki kerangka kerja yang ditujukan untuk merancang dan pemeliharaan sistem dalam waktu tertentu. Pemeliharaan pada sistem ini didukung dengan adanya prototype yang selalu diperbaharui dari tiap iterasi atau perulangan pada siklus *agile* dalam menerima *feedback* klien terkait sistem yang dirancang [60]. Agile

Dynamic Systems Development Method ini juga memiliki beberapa prinsip dalam prosesnya antara lain:

- *Focus on the business needs*

Prinsip ini ada untuk dapat merancang sistem yang dapat memenuhi kebutuhan bisnis dalam tenggat waktu yang tepat. Adanya prinsip ini juga bertujuan untuk memastikan komitmen dari klien dan membangun pemahaman tentang tujuan dari perusahaan.

- *Deliver on time*

Prinsip ini memiliki tujuan dan intisari bagi pengembang atau peneliti untuk mempertimbangkan waktu, biaya, dan kualitas tetap dari sistem yang dihasilkan merupakan satu – satunya variabel dari proyek ini.

- *Collaborate*

Keterlibatan bersama antar anggota tim yang terlibat dalam proses perancangan dan pengembangan sistem.

- *Never compromise on quality*

Kualitas merupakan variabel tetap yang tidak dapat dipengaruhi oleh variabel proyek lainnya seperti biaya dan waktu. Prinsip ini menekankan pada penetapan ekspektasi kualitas, persyaratan & kebutuhan, dan validasi dalam keseluruhan sistem.

- *Incrementally build from the company's foundations*

Dengan prinsip ini, siklus model Agile DSDM ini mengharuskan peneliti untuk dapat melakukan studi mendalami untuk mendapatkan kebutuhan penelitian untuk dijadikan landasan yang kokoh dalam proses perancangan sistem.

- *Develop iteratively*

Prinsip ini menekankan poin dalam melakukan pengembangan secara iteratif dimana spesifikasi tidak dibuat berlebihan dari *feedback* klien yang dilakukan berkala. Hal ini diimplementasi agar proses perancangan sistem tetap bersifat adaptif pada perubahan bila diperlukan.

- *Communicate continuously and clearly*

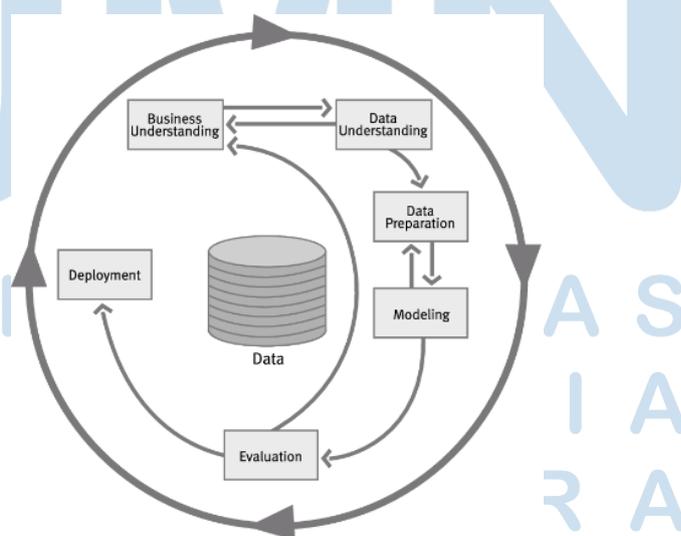
Melakukan komunikasi secara berkala dengan melakukan *modelling*, visualisasi, atau hal lainnya yang dapat memberikan penekanan besar pada *user experience* klien dengan sistem yang dikembangkan.

- *Demonstrate control (use appropriate techniques)*

Menunjukkan perencanaan yang bersifat adaptif dan berorientasi pada kemajuan yang telah dilakukan oleh pengembang. Prinsip ini bermaksud untuk tetap menjalankan penelitian sesuai dengan tenggat waktu yang telah ditentukan tanpa meninggalkan proses komunikasi dua pihak pada klien [17].

2.5.2 Cross-Industry Standard Process for Data Mining

Cross-Industry Standard Process for Data Mining atau yang biasa disingkat CRISP-DM merupakan kerangka kerja atau model proses independen di industri untuk *data mining*. CRISP-DM awalnya diusulkan pada tahun 1996 oleh konsorsium empat perusahaan: SPSS (penyedia solusi Data Mining komersial), NCR (penyedia *database*), Daimler Chrysler dan OHRA (perusahaan asuransi) [61]. Sebagai metodologi, model ini mencakup deskripsi fase-fase suatu proyek, tugas-tugas yang terlibat dalam setiap fase, dan penjelasan tentang hubungan antara tugas-tugas di dalam model ini [62]. Model ini memiliki enam fase iteratif dari pendalaman pada bisnis sampai dengan tahap *deployment* [63], [64].



Gambar 2.5 *Cross-Industry Standard Process for Data Mining Lifecycle*

2.5.2.1 Business Understanding

Fase ini melibatkan pemahaman konteks bisnis proyek, menentukan tujuan bisnis, dan mengidentifikasi tujuan dilakukannya *data mining*. Fase ini juga mencakup pembentukan tim proyek, pemilihan alat dan infrastruktur, dan penulisan rencana proyek.

2.5.2.2 Data Understanding

Pengumpulan, eksplorasi, mendeskripsikan data, dan dilakukan pemeriksaan dimana tahap-tahap ini merupakan hal yang penting pada fase ini. Untuk mendapatkan hasil yang lebih konkrit, biasanya dilakukan pendeskripsian perihal data dengan cara melakukan analisa statistik dan menentukan atribut.

2.5.2.3 Data Preparation

Fase ini melibatkan seluruh aktivitas untuk menyusun dataset akhir yang akan digunakan dalam fase pemodelan data. Fase ini mencakup pemilihan tabel, baris, dan kolom untuk analisis, membersihkan masalah pada kualitas data, membuat atribut atau baris baru, mengintegrasikan baris dan kolom dari beberapa tabel, dan memformat data ke dalam format yang memenuhi persyaratan teknik pemodelan data.

2.5.2.4 Modeling

Fase *modeling* melibatkan pemodelan aktual pada data yang mungkin didapatkan dari pengujian hipotesis statistik menggunakan algoritma yang ditentukan. Hal ini sangat bergantung pada jenis masalah dan teknik pemodelan, dan mencakup pemilihan teknik pemodelan, menghasilkan desain pengujian, membangun model aktual, dan mengevaluasi hasilnya. Pada fase ini, diperlukan juga kemampuan pemodelan data yang sesuai dengan kriteria dan kebutuhan bisnis.

2.5.2.5 Evaluation

Fase evaluasi ditujukan untuk mengevaluasi seberapa jauh dan baik tujuan bisnis telah dicapai. Hal ini memerlukan peninjauan yang cermat baik dari sudut pandang teknis maupun bisnis, dan melibatkan evaluasi pekerjaan yang diselesaikan, peninjauan proses, dan penentuan langkah selanjutnya.

2.5.2.6 Deployment

Selama fase ini, hasil dari fase *modeling* dan evaluasi yang telah dilakukan di-*deploy* untuk dijadikan komponen pada perangkat lunak nantinya. Hal ini bergantung pada sifat proyek dan melibatkan penerapan model, perencanaan pemantauan dan pemeliharaan, pembuatan laporan akhir, dan pelaksanaan retrospektif proyek untuk perbaikan di masa mendatang.

2.5.3 Item-Based Collaborative Filtering

Collaborative filtering merupakan metode pengambilan informasi yang merekomendasikan item kepada pengguna berdasarkan interaksi pengguna lain yang memiliki preferensi dan perilaku serupa terhadap item tersebut. Metode ini memiliki beberapa jenis dalam implementasinya dan salah satunya yaitu *item-based collaborative filtering*. *Item-based collaborative filtering* adalah satu jenis metode rekomendasi yang mencari item serupa berdasarkan item yang disukai atau berinteraksi secara positif dengan pengguna. Metode ini dikembangkan oleh Amazon pada tahun 1998 dan memiliki peran penting dalam keberhasilan Amazon. Cara kerja *item-based collaborative filtering* adalah menyarankan suatu item berdasarkan item yang telah dikonsumsi pengguna sebelumnya. Namun perlu diperhatikan bahwa dalam membandingkan item, sistem kolaboratif tidak membandingkan fitur item (seperti dalam pemfilteran berbasis konten) melainkan bagaimana pengguna berinteraksi dengan item tersebut [65], [66]. Pada penelitian ini, sistem kolaboratif ini dapat dianalogikan sebagai sistem rekomendasi produk. Algoritme yang dikembangkan dapat mengidentifikasi produk serupa berdasarkan korelasi antara jumlah pembelian pengguna yang diklasifikasikan

dalam bentuk *rating* untuk setiap produk. Sistem kemudian akan merekomendasikan produk baru kepada target pengguna berdasarkan peringkat yang berkorelasi. Artinya, jika pengguna target memberi peringkat tinggi pada produk a dan b tetapi belum pernah membeli produk c, sistem akan merekomendasikan produk c kepada target pengguna. Dengan cara ini, *item-based collaborative filtering* menghitung kesamaan item melalui similaritas perilaku pengguna. Implementasi *item-based collaborative filtering* memiliki beberapa konsep vital di dalamnya [67].

2.5.3.1 Pearson Correlation-Based Similarity

Pearson correlation-based similarity merupakan ukuran hubungan linier antara dua variabel atau fitur. Ini adalah metode yang digunakan untuk mengukur kekuatan dan arah hubungan antara dua variabel. Koefisien korelasi Pearson berkisar antara -1 sampai 1, dimana -1 menunjukkan korelasi negatif sempurna, 1 menunjukkan korelasi positif sempurna, dan 0 menunjukkan tidak adanya korelasi. Dengan kata lain, skor *Pearson Correlation* mengukur seberapa cocok dua objek data dalam satu garis. Di bawah ini merupakan rumus *Pearson correlation-based similarity*.

$$sim(u, v) = \frac{\sum_{i \in I_{uv}} (R_{u,i} - \bar{R}_u)(R_{v,i} - \bar{R}_v)}{\sqrt{\sum_{i \in I_{uv}} (R_{u,i} - \bar{R}_u)^2} \sqrt{\sum_{i \in I_{uv}} (R_{v,i} - \bar{R}_v)^2}}$$

Rumus 2.1 *Pearson Correlation-Based Similarity*

Ada beberapa manfaat menggunakan jenis metrik ini. Yang pertama adalah keakuratan skor meningkat ketika data tidak dinormalisasi. Hasilnya, metrik ini dapat digunakan ketika kuantitas bervariasi. Keuntungan lainnya yaitu skor *Pearson Correlation* dapat mengoreksi penskalaan apapun dalam suatu atribut, sementara skor akhir masih ditabulasi. Dengan demikian, objek yang mendeskripsikan data yang sama tetapi menggunakan nilai yang berbeda tetap dapat digunakan [68].

2.5.4 Black Box Testing

Black box testing adalah metode pengujian perangkat lunak tanpa pengetahuan sebelumnya tentang struktur kode internal, detail sistem, dan hal internal lainnya. Sederhananya, cara kerja dari metode pengujian ini yaitu *tester* akan memberikan *input* atau melakukan suatu interaksi pada sistem dan mengamati *output* yang dihasilkan oleh sistem. Pengujian ini dilakukan untuk mengidentifikasi apakah respon yang diberikan oleh sistem sudah sesuai dengan harapan pihak penguji. Dalam prosesnya, penguji dalam *black box testing* tidak diperbolehkan untuk mengetahui mekanisme internal sistem dalam proses pengujian. Secara singkat, *black box testing* ini disebut juga *functional testing* karena metode pengujian ini hanya berfokus pada fungsionalitas suatu sistem [69].

UMMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA