

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Berikut merupakan tabel penelitian terdahulu yang berhubungan dengan pembuatan sistem *management inventory* dan *framework* yang akan menjadi referensi pada penelitian ini:

Tabel 2.1 Penelitian Terdahulu

Judul Jurnal	Nama Jurnal	Penulis / Tahun	Hasil	Kesimpulan
<i>Web-based Inventory Management System</i>	<i>International Journal of Science and Applied Information Technology</i> Volume 12, No.5	Maredel T. Tanaman, Jhon Lloyd A. Baylosis, Bhrnt Joshua A. Abiles, Mark Lester P. Catungal, Dr. Philipcris C. Encarnacion / 2023	Sistem ini berhasil mengimplementasikan manajemen inventori yang berbasis web, <i>framework</i> yang digunakan adalah PHP untuk pengembangan dan MySQL sebagai basis data, sistem yang dikembangkan dapat mengotomatisasi proses yang sebelumnya dilakukan secara manual. Sistem ini memperbaiki proses bisnis dengan mencatat dan memproses data secara elektronik [9].	Implementasi sistem berbasis web ini menunjukkan peningkatan signifikan dalam efisiensi dan kemudahan pengelolaan inventori. Sistem ini membuktikan bahwa teknologi web dapat secara substansial mengurangi beban kerja dan meningkatkan akurasi data [9].
<i>Web Based Online Inventory Information System</i>	<i>IOP Conf. Series: Materials Science and Engineering</i> 879	E S Soegoto, A F Palalungan / 2020	Sistem ini memanfaatkan teknologi web untuk memaksimalkan kinerja manusia dalam proses kerja di perusahaan, <i>framework</i> yang digunakan adalah ASP.NET dan Oracle 9 sebagai basis data. Sistem yang dikembangkan memudahkan	Implementasi sistem inventori <i>online</i> berbasis web ini berhasil meningkatkan efisiensi operasional dan memperbaiki pengelolaan inventori secara keseluruhan. Sistem ini menyediakan <i>platform</i> yang dapat diandalkan dan mudah diakses untuk mempercepat proses

Judul Jurnal	Nama Jurnal	Penulis Tahun /	Hasil	Kesimpulan
			pengelolaan barang dan menyederhanakan proses yang kompleks. Ini membuktikan bahwa sistem inventori berbasis web dapat mendukung pekerjaan perusahaan secara efektif [4].	kerja dan pengambilan Keputusan [4].
<i>Development of a Web Based Inventory Information System</i>	<i>International Journal of Engineering, Science & Information Technology (IJESTY). Volume 1, No. 2 (2021) pp. 24-31</i>	Johani S Pasaribu / 2021	Sistem yang dikembangkan menggunakan YII Framework dan metode MVC (Model View Controller) berhasil memfasilitasi proses layanan kepada konsumen dengan lebih cepat dan jelas. Sistem ini mendukung manajemen inventori dengan antarmuka yang efisien dan pengelolaan data yang lebih baik [10].	Sistem manajemen inventori berbasis web yang dikembangkan menunjukkan peningkatan substansial dalam pengelolaan dan aksesibilitas data inventori, menawarkan solusi yang dapat diadaptasi dan dinamis untuk kebutuhan manajemen modern [10].
<i>Web-based systems for inventory control in organizations: A Systematic Review</i>	<i>IEEE/ACIS 22nd International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)</i>	Gunther Misahuaman, Alfredo Daza, Emily Zavaleta / 2021	Jurnal ini menyajikan sebuah tinjauan sistematis terhadap penggunaan sistem berbasis web untuk kontrol inventori di berbagai organisasi. Penelitian menemukan bahwa banyak perusahaan masih mengandalkan inventarisasi manual yang tidak efisien. Sistem berbasis web menawarkan solusi dengan memperbaiki	Penerapan sistem kontrol inventori berbasis web sangat penting untuk meningkatkan efisiensi operasional di organisasi kecil, menengah, dan besar. Penelitian ini mengusulkan bahwa dengan menganalisis dan menerapkan teknologi web yang tepat, organisasi dapat mencerminkan perubahan substansial dalam pengelolaan inventori mereka, yang menghasilkan penghematan biaya dan

Judul Jurnal	Nama Jurnal	Penulis Tahun /	Hasil	Kesimpulan
			integrasi data dan efisiensi proses [11].	operasi yang lebih efisien [11].
<i>Towards inventory control excellence: An innovative approach based on a web-based platform</i>	<i>F1000Research</i>	Miguel Ramos-Miller, Alex Pacheco / 2023	Implementasi sistem inventori berbasis web di institusi pendidikan menunjukkan peningkatan efisiensi pencarian barang sebesar 85.51%, registrasi barang meningkat 90.31%, dan peningkatan generasi laporan tahunan sebesar 83.11% [12].	Peningkatan proses inventori melalui teknologi informasi menunjukkan bahwa solusi berbasis web dapat secara dramatis meningkatkan efisiensi dan kolaborasi, serta mendukung kebutuhan manajemen yang unik di institusi pendidikan [12].
<i>Web-Based Coffee Inventory Application</i>	Majalah Bisnis & IPTEK Vol 16, No. 2	Alifadillah, Haris Supriatna / 2023	Implementasi aplikasi inventori berbasis web dengan menggunakan PHP sebagai bahasa pemrograman dan MySQL sebagai basis data di sebuah kedai kopi menghasilkan peningkatan signifikan dalam manajemen inventori, termasuk pelacakan inventori waktu nyata dan pengurangan kesalahan dalam pencatatan manual [13].	Aplikasi ini mendemonstrasikan bagaimana transformasi digital dalam bisnis skala kecil dapat mengatasi hambatan operasional dan meningkatkan efisiensi, menyarankan integrasi lebih lanjut dari teknologi prediktif dan pengembangan mobile untuk meningkatkan fleksibilitas [13].
<i>System Development Method with The Prototype Method</i>	<i>International Journal of Scientific & Technology Research, Volume 7, Issue 9, pp. 141-144</i>	Azhar Susanto, Meiryani / 2019	Studi ini membahas penerapan metode <i>prototyping</i> dalam pengembangan sistem informasi. Metode ini digunakan untuk membuat <i>prototype</i> awal yang cepat guna mengidentifikasi umpan balik pengguna hingga sistem akhir sesuai	Metode <i>prototyping</i> sangat efektif dalam memahami dan menangkap kebutuhan pengguna dengan akurat. Proses iteratif yang melibatkan pembuatan dan pengujian <i>prototype</i> memungkinkan pengembang untuk secara bertahap memperbaiki sistem berdasarkan umpan

Judul Jurnal	Nama Jurnal	Penulis Tahun /	Hasil	Kesimpulan
			dengan kebutuhan dan harapan pengguna [14].	balik nyata dari pengguna. Hal ini meningkatkan kualitas akhir sistem dan memastikan bahwa sistem tersebut benar-benar memenuhi kebutuhan pengguna. Penggunaan <i>framework</i> seperti UML (<i>Unified Modeling Language</i>) untuk desain dan dokumentasi membantu dalam visualisasi dan komunikasi antara pengembang dan pengguna, membuat proses lebih terstruktur dan mudah dipahami oleh semua pihak yang terlibat [14].
<i>MERN Stack Unveiled: A Research Study on the Technology's Architecture and Benefits</i>	<i>Tuijin Jishu/Journal of Propulsion Technology ISSN: 1001-4055 Vol. 44 No. 1</i>	Neeraj Bhat, Rakesh Sharma, Anushka Kaushik, Teerthesh Jain / 2023	Penelitian ini menyelidiki arsitektur <i>MERN Stack</i> , menekankan pada keefisienan, fleksibilitas, dan kemudahan penggunaan <i>React JS</i> , <i>Express JS</i> , dan <i>MongoDB</i> . Ditekankan bagaimana <i>MERN Stack</i> memfasilitasi pengembangan yang cepat dan efisien dengan integrasi <i>front-end</i> dan <i>back-end</i> yang mulus [15].	<i>MERN Stack</i> sangat cocok untuk pengembangan aplikasi web yang dinamis dan responsif, memberikan solusi yang skalabel dan mudah dikelola [15].
<i>Review on Study and Usage of MERN Stack for Web Development</i>	<i>International Journal for Research in Applied Science & Engineering Technology (IJRASET)</i>	Sumangala A. Bafna, Pratiksha D. Dutonde, Shivani S. Mamidwar, Monali S. Korvate, Prof. Dhiraj Shirbhare / 2022	Tinjauan ini Jurnal ini mengulas penggunaan <i>MERN Stack</i> , memperlihatkan bagaimana <i>MongoDB</i> , <i>Express JS</i> , <i>React JS</i> , dan <i>Node JS</i> berkontribusi pada pengembangan web yang efisien dan terintegrasi [16].	Menggunakan <i>MERN Stack</i> mempercepat pengembangan dan <i>deployment</i> , memungkinkan pengembangan aplikasi yang lebih responsif dan interaktif [16].

Judul Jurnal	Nama Jurnal	Penulis / Tahun	Hasil	Kesimpulan
<i>Inventory Management and Performance of SMEs in the Manufacturing Sector of Harare</i>	<i>Procedia Manufacturing, Volume 38</i>	W Muchaendepi, C Mbohwa, T Hamandishe, J Kanyepe / 2019	Studi ini menemukan bahwa kebanyakan UKM di Harare menggunakan metode Just-In-Time untuk manajemen inventori dan kurangnya pengetahuan tentang sistem yang terkomputerisasi mengakibatkan tantangan dalam efisiensi rantai pasokan [17].	Penelitian mengungkapkan bahwa penggunaan sistem inventori yang terkomputerisasi dapat meningkatkan efisiensi dan mengurangi keterlambatan dalam pengiriman. Peneliti juga merekomendasikan studi lebih lanjut untuk mengeksplorasi pengaruh teknologi pada manajemen inventori di UKM [17].

Berdasarkan penelitian terdahulu pada tabel 2.1, pilihan untuk mengembangkan web *management inventory* didasarkan pada beberapa jurnal yang menunjukkan bahwa sistem berbasis web meningkatkan efisiensi operasional. Jurnal-jurnal yang ditinjau menunjukkan bahwa dengan sistem berbasis web, perusahaan dapat mengotomatisasi pengelolaan inventori yang sebelumnya dilakukan secara manual, yang mengurangi waktu dan kesalahan dalam pencatatan. Penelitian terdahulu ini menyelidiki arsitektur *MERN Stack*, menekankan pada keefisienan, fleksibilitas, dan kemudahan penggunaan *React JS*, *Express JS*, dan *MongoDB*. Ditekankan bagaimana *MERN Stack* memfasilitasi pengembangan yang cepat dan efisien dengan integrasi *front-end* dan *back-end* yang mulus. Selain itu, sistem ini memungkinkan akses *real-time* ke data inventori yang dapat diakses dari berbagai lokasi, meningkatkan fleksibilitas dan responsivitas perusahaan terhadap dinamika pasar [15] [16] [17].

Pemilihan teknologi atau *framework React JS* dipilih karena kemampuannya untuk membangun antarmuka pengguna yang dinamis. Jurnal terkait menunjukkan bahwa *React JS* memudahkan pembuatan komponen interaktif dan reaktif yang meningkatkan pengalaman pengguna pada aplikasi web. *Express JS* sebagai kerangka kerja *backend*, digunakan karena kemudahannya dalam membangun aplikasi *server* dengan *Node JS* yang ringan

dan efisien, seperti yang disebutkan didalam penelitian. *Express JS* mendukung pembuatan *RESTful API* yang merupakan bagian penting dalam arsitektur berbasis web untuk berkomunikasi dengan *frontend*.

MongoDB dipilih berdasarkan ulasan jurnal yang menunjukkan bahwa sebagai *database NoSQL*, *MongoDB* menawarkan fleksibilitas dalam penyimpanan data dan skema yang dinamis, yang ideal untuk aplikasi yang membutuhkan skalabilitas besar dan penanganan besar data yang tidak terstruktur. Setiap teknologi dan fitur yang akan dikembangkan, berusaha untuk menciptakan sistem yang tidak hanya efisien tetapi juga skalabel dan *user-friendly*, mengikuti tren terkini dalam pengembangan sistem.

2.2 Kajian Teori

2.1.1 Pentingnya Penerapan Sistem ERP

Penerapan Sistem *Enterprise Resource Planning* (ERP) pada suatu perusahaan memiliki peran krusial dalam mengoptimalkan efisiensi, mengelola sumber daya, dan meningkatkan daya saing di era bisnis yang kompetitif. Sistem ERP merupakan sebuah platform terintegrasi yang menyatukan berbagai fungsi operasional dan departemen perusahaan dalam satu sistem yang terpusat. Dalam konteks ini, fokus kajian teori akan dititikberatkan pada modul *Inventory Management*.

Modul *Inventory* dalam ERP memainkan peran sentral dalam mengelola persediaan barang. Dengan menggunakan modul ini, perusahaan dapat memantau dan mengendalikan setiap aspek dari siklus hidup produk, mulai dari pemesanan, penerimaan, penyimpanan, hingga distribusi. Keunggulan utama modul *Inventory* adalah pemantauan secara *real-time* terhadap tingkat persediaan, menghindari kekurangan stok atau kelebihan persediaan yang dapat mengakibatkan biaya tambahan. Selain itu, kemampuan analitik modul *Inventory*

memungkinkan perusahaan membuat keputusan yang lebih akurat berdasarkan data historis dan tren pasar [18].

Sementara itu, modul *Inventory Management* berfokus pada optimalisasi proses di gudang penyimpanan. Dengan penerapan modul ini, perusahaan dapat merencanakan, melacak, dan mengelola perpindahan barang di gudang secara efisien. Fasilitas seperti pemilihan lokasi penyimpanan yang optimal, pengelolaan perubahan dalam tingkat permintaan, dan pelacakan secara *real-time* memungkinkan perusahaan mengurangi waktu siklus dan meningkatkan produktivitas operasional.

Pentingnya penerapan modul *Inventory Management* pada sistem ERP bukan hanya terletak pada efisiensi operasional, tetapi juga pada kemampuan perusahaan untuk menghadapi dinamika pasar yang terus berubah. Dengan memiliki data yang akurat dan terkini, perusahaan dapat merespons perubahan pasar dengan lebih cepat dan mengoptimalkan rantai pasok secara keseluruhan. Dengan demikian, penerapan Sistem ERP dengan modul *Inventory Management* bukan hanya menjadi investasi strategis dalam meningkatkan efisiensi internal perusahaan, tetapi juga langkah cerdas untuk menjaga daya saing di tengah-tengah persaingan bisnis yang ketat [19].

2.1.2 *Module Inventory*

Dalam konteks pengembangan Sistem *Enterprise Resource Planning* (ERP), modul *Inventory* memegang peranan penting sebagai komponen utama yang mengelola seluruh aspek persediaan barang perusahaan. Modul ini tidak hanya terbatas pada aspek fisik penyimpanan, namun juga melibatkan proses pemesanan, pengiriman, dan pemantauan stok secara menyeluruh.

Modul inventaris merujuk pada konsep manajemen persediaan yang digunakan untuk mengontrol stok produk di lokasi penyimpanan

tertentu. Dalam modul ini, rumah sakit memiliki kemampuan untuk mengelola informasi terkait aset, penerimaan barang, atau pesanan, serta pengeluaran barang. Modul inventori ini mencakup sistem pencatatan data mengenai aliran barang masuk dan keluar dari gudang, yang kemudian dapat disajikan melalui sistem pelaporan terintegrasi untuk digunakan oleh pihak lain dalam lingkungan inventori [20].

Modul *Inventory* pada Sistem ERP dirancang untuk memberikan kerangka kerja terintegrasi, memungkinkan perusahaan untuk memahami dan mengelola seluruh siklus hidup persediaan. Mulai dari penerimaan barang hingga distribusi, modul ini mencatat setiap perubahan stok secara akurat dan memberikan visibilitas yang diperlukan untuk pengambilan keputusan yang tepat waktu. Dengan optimalnya manajemen persediaan barang, akan berdampak positif pada kinerja dan hasil layanan yang disajikan kepada pelanggan dalam pengadaan stok barang [21].

Fungsi utama modul *Inventory* mencakup pencatatan persediaan, manajemen pemesanan, pemantauan stok secara *real-time*, dan penyusunan laporan persediaan. Pencatatan yang akurat membantu perusahaan menghindari kesalahan dalam stok, sedangkan manajemen pemesanan mengoptimalkan proses pengadaan. Pemantauan stok secara *real-time* memberikan visibilitas langsung terhadap perubahan stok, sedangkan laporan persediaan menyediakan informasi untuk analisis dan pengambilan keputusan.

Manajemen *layout* dan organisasi gudang juga menjadi fokus utama dari modul *Inventory Management*. Dengan pemetaan yang baik, modul ini memastikan bahwa setiap item ditempatkan secara strategis untuk meminimalkan waktu pencarian dan mempercepat proses pengambilan barang. Organisasi gudang yang efisien berkontribusi pada peningkatan produktivitas, mengurangi kesalahan pengambilan barang, dan mempercepat proses pemenuhan pesanan.

Keunggulan modul *Inventory* terletak pada integrasinya dengan modul lain seperti Pembelian, Penjualan, dan Produksi. Integrasi ini memastikan bahwa informasi stok otomatis disinkronkan di seluruh departemen, menciptakan keterkaitan yang efisien. Manfaat dari penggunaan modul *Inventory* pada Sistem ERP mencakup peningkatan efisiensi operasional, layanan pelanggan yang lebih baik, dan optimalisasi rantai pasok.

2.3 Framework dan Algoritma

2.2.1 *React JS*

React, sebuah perpustakaan *JavaScript front-end* yang terkenal dan bersifat *open source*, lahir dari tangan kreatif *Facebook*. Keistimewaan *React* sangat digemari dalam komunitas pengembang karena kemudahan dan efektivitas dalam proses pengembangannya. Perpustakaan ini mendorong terbentuknya komponen Antarmuka Pengguna yang dapat dipakai kembali, menyajikan informasi yang terus berkembang seiring berjalannya waktu [22].

Salah satu karakteristik utama *React JS* adalah penggunaan komponen (*components*) yang memecah antarmuka pengguna menjadi bagian-bagian kecil yang dikelola secara terpisah. Setiap komponen dapat dianggap sebagai modul mandiri yang dapat digunakan kembali, memfasilitasi pengembangan dan pemeliharaan kode secara efisien. Pendekatan ini juga mendukung konsep pemrograman berorientasi objek, di mana setiap komponen dapat memiliki properti dan metode mereka sendiri.

Virtual DOM (Document Object Model) adalah fitur lain yang membuat *React JS* unik. *React* menggunakan struktur data *virtual DOM* untuk merepresentasikan struktur DOM aktual pada halaman web. Saat ada perubahan dalam state atau data, *React* membangun *virtual DOM*

terlebih dahulu, membandingkannya dengan DOM yang ada, dan hanya memperbarui bagian-bagian yang berubah. Hal ini mengurangi *overhead* pengolahan yang tidak perlu, menjadikan aplikasi lebih responsif dan efisien.

React JS juga mendukung konsep *unidirectional data flow*, yang artinya data mengalir dalam satu arah. Ini memudahkan pelacakan perubahan data dan memahami aliran logika aplikasi. Selain itu, *React JS* dapat diintegrasikan dengan mudah dengan berbagai *library* atau *framework* lain dan dapat digunakan dalam pengembangan aplikasi berbasis SPA (*Single Page Application*) [22].

Keunggulan *React JS* dalam pengembangan antarmuka pengguna telah membuatnya menjadi pilihan utama dalam proyek-proyek besar dan kompleks. Penggunaan *React JS* pada pengembangan Sistem ERP berbasis web di PT Polytech Indo Hausen diharapkan dapat memberikan antarmuka yang dinamis, responsif, dan mudah dipelihara, meningkatkan pengalaman pengguna secara keseluruhan.

2.2.2 *Node JS*

Node.js adalah ruang pelaksanaan serba *platform*, dibangun di atas V8 yang merupakan mesin *JavaScript* andal dengan sumber terbuka dari Google. *Node.js* merubah fungsi *JavaScript* menjadi kode mesin dengan performa dan kecepatan yang mengagumkan. Dengan keunggulan tersebut, *Node.js* memberikan penampilan yang luar biasa dalam membangun aplikasi jaringan yang dapat diubah skala dan sangat efisien. Ini dicapai melalui pendekatan input-output yang berbasis pada peristiwa, tanpa pemblokiran, dan bersifat *asynchronous* [23].

Node.js menyediakan lingkungan yang memungkinkan pengembang menggunakan *JavaScript* untuk mengembangkan aplikasi *server-side*, yang sebelumnya terbatas pada penggunaan di sisi klien. Dengan

memanfaatkan model pemrograman *non-blocking*, *Node.js* dapat menangani operasi I/O tanpa memblokir eksekusi program, mengoptimalkan kinerja dan responsivitas aplikasi.

Kelebihan lain dari *Node.js* adalah manajemen paket menggunakan *npm (Node Package Manager)*, yang merupakan repositori paket *JavaScript* terbesar di dunia. Penggunaan *npm* memudahkan pengembang untuk mengelola dependensi proyek dan mengakses berbagai paket atau modul yang dapat meningkatkan fungsionalitas aplikasi. *Node.js* juga cocok untuk pengembangan aplikasi berbasis mikro dan kontainer, memungkinkan skalabilitas dan implementasi arsitektur berbasis layanan (*microservices*) dengan lebih efisien. Dengan dukungan komunitas yang besar dan pertumbuhan ekosistem yang pesat, *Node.js* menjadi pilihan yang tangguh untuk membangun backend aplikasi web *modern*.

2.2.3 *Express JS*

Express.js adalah framework untuk *Node.js* yang sangat populer di kalangan pengembang karena fleksibilitas dan efisiensinya dalam membangun aplikasi web dan API. Dalam pengembangan sistem ERP berbasis web, seperti yang sedang dikembangkan di PT Polytech Indo Hausen, *Express.js* memainkan peran kritis dalam memudahkan integrasi backend dan mengelola komunikasi antara *database* dan *frontend* [24]. Berikut ini adalah Fitur Utama dari *Express.js*:

1. **Simplifikasi *Routing***: *Express.js* menyederhanakan proses definisi *route* dan *handler* untuk berbagai permintaan HTTP. Fitur *routing* yang kuat ini memungkinkan pengembang untuk dengan mudah mengatur *endpoint* yang diperlukan untuk fungsi ERP seperti pengambilan, penambahan, dan perubahan data inventori.

2. *Middleware* yang Efisien: *Middleware* di *Express.js* bertindak sebagai blok bangunan utama dalam aplikasi *Node.js*. Fungsionalitasnya memungkinkan pengembang untuk menjalankan kode tertentu, mengubah objek permintaan dan respons, dan mengakhiri siklus permintaan-respons sebelum router menerima permintaan tersebut. Ini sangat berguna untuk otentikasi dan otorisasi pengguna, pemrosesan *log*, dan penanganan *error*.
3. Performa Tinggi dan Pengelolaan Sumber Daya yang Optimal: Menggunakan arsitektur *non-blocking* dan *event-driven* yang mendasari *Node.js*, *Express.js* memastikan bahwa aplikasi dapat menangani permintaan tinggi tanpa menghabiskan sumber daya *server* yang berlebihan. Fitur ini penting untuk sistem ERP di mana respons waktu nyata dan keandalan adalah kunci.
4. Kemudahan Integrasi dengan *Database*: *Express.js* mendukung integrasi dengan berbagai jenis database termasuk *MongoDB*, *MySQL*, dan *PostgreSQL*, melalui berbagai *library* seperti *Mongoose* untuk *MongoDB* atau *Sequelize* untuk *SQL databases*. Ini mempermudah penyimpanan, pengambilan, dan manipulasi data dalam aplikasi ERP.

Express.js menawarkan sejumlah keunggulan yang signifikan untuk pengembangan sistem ERP di PT Polytech Indo Hausen. Pertama, *framework* ini mendukung kecepatan pengembangan yang tinggi berkat struktur yang sederhana dan intuitif, memungkinkan pengembang untuk dengan mudah menyusun *middleware* dan *route* yang efisien. Keunggulan ini sangat penting untuk mempercepat pengerjaan proyek dan menghadirkan solusi teknis dalam waktu yang lebih singkat. Kedua, *Express.js* menyediakan dukungan yang kuat untuk *scalability*, sehingga memfasilitasi PT Polytech Indo Hausen dalam mengadaptasi dan memperluas sistem ERP mereka seiring dengan pertumbuhan dan evolusi kebutuhan bisnis [25].

Terakhir, kesederhanaan dan fleksibilitas yang ditawarkan oleh *Express.js* memungkinkan pengembang untuk menulis kode yang lebih bersih dan mudah dipelihara, yang berkontribusi pada pengurangan biaya pemeliharaan jangka panjang dan peningkatan kualitas kode secara keseluruhan. Keunggulan-keunggulan ini membuat *Express.js* menjadi pilihan yang tepat untuk mengembangkan aplikasi web yang adaptif di lingkungan korporat yang dinamis seperti PT Polytech Indo Hausen.

2.2.4 *Mongo DB*

MongoDB merupakan sistem manajemen basis data dokumen berbasis *open-source* yang memberikan kinerja tinggi, tingkat ketersediaan yang optimal, dan kemampuan penskalaan otomatis. Setiap data dalam *MongoDB* disajikan sebagai sebuah dokumen, yang merupakan struktur data terdiri dari pasangan *field* dan nilai. Dokumen *MongoDB* memiliki kemiripan dengan objek JSON, dan nilai *field* dapat mencakup dokumen lain, *array*, dan *array* dokumen. Keunggulan penggunaan dokumen termasuk kemampuan untuk menyelaraskan dengan tipe data asli di berbagai bahasa pemrograman, mengurangi kebutuhan untuk join yang kompleks melalui penggunaan dokumen dan *array* bersarang, serta skema dinamis yang mendukung *polimorfisme* yang lebih fleksibel.

Dalam konteks pengembangan aplikasi *modern*, *MongoDB* sangat cocok untuk kebutuhan pengelolaan data yang bersifat dinamis dan kompleks. Struktur data yang disimpan dalam format BSON (*Binary JSON*) memungkinkan pengembang untuk menyimpan data dalam dokumen yang dapat berisi berbagai tipe data dan dapat berkembang seiring waktu [26].

Keunggulan *MongoDB* juga terletak pada kemampuan untuk menangani jumlah data yang besar dan skala horizontal dengan baik. Adopsi model replikasi dan sharding membuat *MongoDB* mampu mengelola data dengan distribusi dan ketersediaan yang tinggi, mendukung aplikasi yang membutuhkan kinerja tinggi dan toleransi terhadap kesalahan. *MongoDB* menawarkan keuntungan berupa desain tanpa hubungan, struktur objek tunggal yang jelas, ketersediaan *query* yang dalam, kemudahan dalam penskalaan, dan tidak perlu konversi/pemetaan objek aplikasi ke objek basis data. *MongoDB* juga menggunakan memori *internal* untuk menyimpan set kerja yang memungkinkan akses data lebih cepat [27].

2.2.5 *Tailwind CSS*

Tailwind CSS merupakan *framework* CSS yang sangat dapat dikustomisasi dan digunakan dalam membangun desain antarmuka pengguna tanpa meninggalkan *markup* HTML. *Framework Tailwind* memberikan fleksibilitas tinggi dengan pendekatan *utility-first*, memungkinkan pengembang untuk menggabungkan kelas-kelas kecil untuk menciptakan desain yang kompleks dengan kode yang lebih bersih dan lebih terorganisir. Kelebihan utama dari *Tailwind CSS* terletak pada kemampuannya untuk mengurangi waktu yang dibutuhkan untuk menulis CSS, dan mempercepat proses pengembangan.

Dalam konteks pengembangan sistem manajemen inventori, *Tailwind CSS* memberikan berbagai kemudahan dalam *styling*, *Tailwind* menyediakan kelas utilitas untuk membangun desain dengan mudah, memungkinkan pengembang untuk mengcustomisasi setiap aspek tampilan tanpa harus menimpa *style* yang ada, dan memfasilitasi pembuatan tampilan yang unik dan sesuai dengan kebutuhan spesifik sistem. Penggunaan *Tailwind CSS* pada penelitian ini, membantu dalam membuat antarmuka pengguna yang interaktif dan menarik. Fitur-fitur

seperti tabel, inventori, *product*, navigasi, dan lain-lain dirancang menggunakan *Tailwind CSS* untuk memastikan konsistensi dan estetika visual yang menarik serta memudahkan navigasi pengguna [28].

2.2.6 *Software Development Life Cycle*

Software Development Life Cycle (SDLC) adalah proses yang digunakan oleh industri perangkat lunak untuk merancang, mengembangkan, dan menguji perangkat lunak berkualitas tinggi. SDLC menyediakan kerangka kerja untuk mengelola proyek perangkat lunak dari awal hingga akhir, memastikan bahwa produk yang dihasilkan memenuhi ekspektasi pelanggan dan standar kualitas yang tinggi. SDLC terdiri dari beberapa tahap, yang masing-masing memiliki tujuan dan output spesifik. Tahap-tahap ini adalah: perencanaan, analisis, desain, implementasi, pengujian, dan *review*.

Tahap pertama dalam SDLC adalah perencanaan, di mana tujuan proyek ditetapkan, ruang lingkup didefinisikan, dan sumber daya yang diperlukan diidentifikasi. Dalam tahap ini juga pengembang bekerja sama dengan pihak yang berkepentingan dalam hal ini adalah pihak perusahaan PT Polytech Indo Hausen, untuk memahami kebutuhan dan harapan mereka, serta untuk menentukan strategi pengembangan yang akan digunakan [29]. Tahap berikutnya adalah analisis, di mana kebutuhan sistem dirinci lebih lanjut dan spesifikasi fungsional serta *non-fungsional* dibuat. Ini adalah tahap di mana masalah dan solusi potensial diidentifikasi dan dianalisis secara mendalam. Desain sistem kemudian dimulai, mencakup pembuatan arsitektur perangkat lunak, desain antarmuka pengguna, dan perencanaan basis data [30].

Berikutnya adalah tahap implementasi yang dimana kode sumber ditulis berdasarkan desain yang telah dibuat. Semua komponen sistem dikembangkan dan diintegrasikan untuk membentuk produk perangkat lunak yang berfungsi penuh. Tahap ini diikuti oleh pengujian, di mana

sistem diuji untuk memastikan bahwa semua fungsionalitas bekerja dengan benar dan sesuai dengan spesifikasi. Pengujian mencakup uji unit, uji integrasi, uji sistem, dan *user acceptance testing* (UAT) [31]. Tahap akhir dalam SDLC adalah *review*, di mana perangkat lunak yang telah dirilis terus diperbarui dan diperbaiki berdasarkan umpan balik pengguna dan perubahan dalam kebutuhan bisnis. Proses *review* melibatkan perbaikan *bug*, peningkatan performa, dan penambahan fitur baru [32].

2.2.7 Model *Prototyping* dalam *Software Development Life Cycle*

Metode *Prototyping* bertujuan untuk meningkatkan pemahaman terhadap kebutuhan pengguna dan memastikan bahwa sistem yang dikembangkan sesuai dengan harapan dan kebutuhan tersebut. *Prototyping* adalah salah satu metode dalam pengembangan perangkat lunak yang memungkinkan pengembang untuk membuat, menguji, dan menyempurnakan prototipe sebelum sistem akhir dikembangkan. Dalam konteks pengembangan modul *Inventory Management* berbasis web menggunakan *React JS* dan *Express JS* di PT Polytech Indo Hausen, metode ini sangat relevan untuk memastikan bahwa sistem yang dikembangkan dapat memenuhi kebutuhan operasional perusahaan dengan efektif. Berikut penjelasan terperinci tentang cara integrasi ini dilakukan:

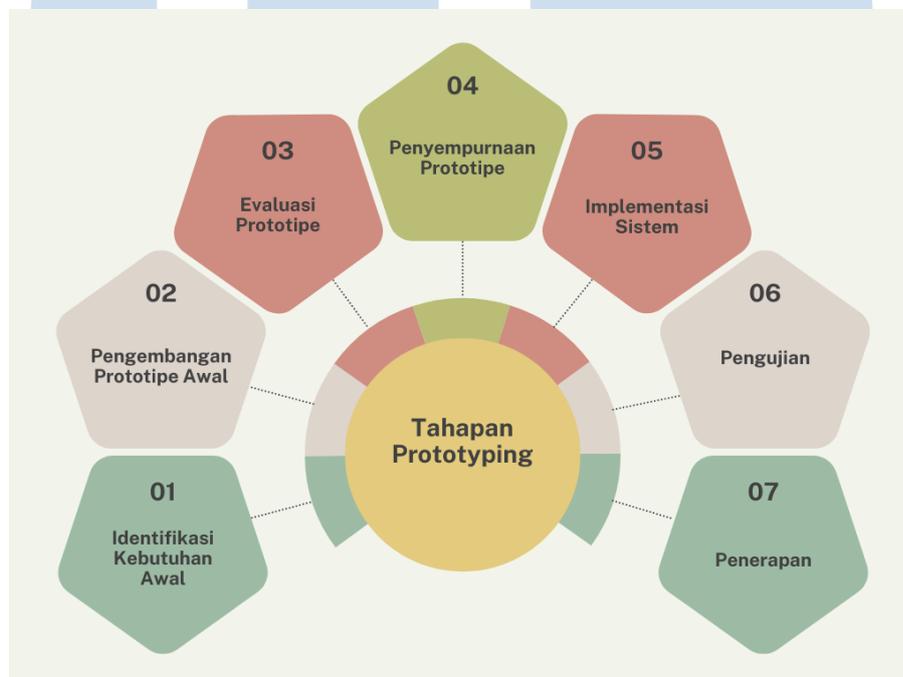
1. Definisi dan Tujuan *Prototyping*

Prototyping adalah proses iteratif dalam pengembangan sistem di mana model awal atau prototipe dari sistem yang akan dikembangkan dibuat, diuji, dan diperbaiki berdasarkan umpan balik dari pengguna. Tujuan utama dari *prototyping* adalah untuk:

- a. Meningkatkan pemahaman kebutuhan pengguna. Dengan membuat prototipe, pengembang dapat lebih memahami kebutuhan dan ekspektasi pengguna.

- b. Mempercepat pengembangan. Prototipe memungkinkan identifikasi dini terhadap masalah dan perbaikan yang diperlukan, sehingga mengurangi waktu dan biaya pengembangan.
- c. Meminimalisir risiko kegagalan. Prototipe memungkinkan pengujian dan validasi konsep sebelum investasi besar dilakukan untuk pengembangan penuh [33].

2. Tahap-tahap dalam *Prototyping*



Gambar 2.1 Model *Prototyping*
(Sumber: [34])

a. Identifikas Kebutuhan Awal

Tahap pertama dalam metode *prototyping* adalah identifikasi kebutuhan awal yang melibatkan penentuan kebutuhan dasar dan tujuan sistem. Pada tahap ini, informasi mengenai kebutuhan pengguna dan spesifikasi sistem dikumpulkan melalui wawancara, dan studi pustaka. Tujuan dari tahap ini adalah untuk mendapatkan gambaran yang jelas mengenai apa yang diharapkan dari sistem yang akan dikembangkan.

b. Pengembangan Prototipe Awal

Setelah kebutuhan awal diidentifikasi, langkah berikutnya adalah pengembangan prototipe awal. Prototipe ini biasanya bersifat sederhana dan fokus pada fungsionalitas utama yang diharapkan. Prototipe awal memungkinkan pihak perusahaan untuk melihat dan merasakan bagaimana sistem akan bekerja, sehingga mereka dapat memberikan umpan balik yang konstruktif.

c. Evaluasi Prototipe

Berdasarkan umpan balik yang diterima dari tahap evaluasi, dilakukan penyempurnaan dan pengembangan lebih lanjut pada prototipe. Penyempurnaan ini mencakup penambahan fitur baru, peningkatan antarmuka pengguna, dan optimasi performa sistem. Tujuannya adalah untuk mengembangkan prototipe yang semakin mendekati sistem akhir yang diinginkan.

d. Penyempurnaan Prototipe

Berdasarkan umpan balik yang diterima, dilakukan penyempurnaan dan pengembangan lebih lanjut pada prototipe hingga mencapai bentuk final yang siap diimplementasikan. Penyempurnaan ini mencakup penambahan fitur baru, peningkatan antarmuka pengguna, dan optimasi performa sistem.

e. Implementasi Sistem

Setelah penyempurnaan selesai, sistem yang telah disempurnakan siap untuk diimplementasikan, proses dimulai dengan konversi desain dan spesifikasi yang telah disetujui menjadi kode nyata. Dalam proses ini juga memastikan bahwa sistem dapat beroperasi sesuai dengan yang diharapkan.

f. Pengujian

Pengujian menyeluruh terhadap sistem yang telah diimplementasikan dilakukan untuk memastikan bahwa semua

fungsionalitas berjalan dengan baik, mencakup uji fungsional, uji performa, dan *user acceptance testing* (UAT).

g. Penerapan

Setelah sistem lulus uji, dilakukan penerapan penuh dan pelatihan pengguna, serta *monitoring* dan evaluasi kinerja sistem dalam lingkungan operasional.

Prototyping memiliki beberapa keuntungan utama, termasuk meningkatkan pemahaman terhadap kebutuhan pengguna, mempercepat waktu pengembangan, dan mengurangi risiko kegagalan. Namun, metode ini juga memiliki tantangan, termasuk kebutuhan akan iterasi yang berulang dan potensi peningkatan biaya pengembangan jika tidak dikelola dengan baik [35].

2.4 Tools / Software

2.3.1 Visual Studio Code

Visual Studio Code (VS Code) adalah sebuah editor teks sumber terbuka yang dikembangkan oleh *Microsoft*. Dirancang untuk mendukung berbagai bahasa pemrograman dan memfasilitasi pengembangan perangkat lunak, VS Code telah menjadi pilihan populer di kalangan pengembang perangkat lunak. Kelebihan utama dari *Visual Studio Code* adalah kesederhanaan antarmukanya yang bersih dan ringan, sambil menyediakan fitur-fitur kaya yang mendukung pengembangan aplikasi yang kompleks.

Salah satu fitur unggulan dari VS Code adalah dukungan untuk ekosistem ekstensi yang luas. Pengguna dapat dengan mudah memperluas fungsionalitas editor dengan menginstal ekstensi yang sesuai dengan kebutuhan proyek. Selain itu, VS Code mendukung kontrol versi terintegrasi, memungkinkan pengguna untuk berkolaborasi dalam pengembangan perangkat lunak dengan mudah.

VS Code juga menawarkan penyorotan sintaksis yang kuat, saran kode otomatis, dan penyusunan kode yang efisien. Kemampuan ini sangat mendukung produktivitas pengembang, terutama dalam proyek-proyek yang melibatkan bahasa pemrograman seperti *JavaScript*, *TypeScript*, dan *HTML*, yang sering digunakan dalam pengembangan aplikasi berbasis web [36].

Kemudahan integrasi dengan Git membuat proses kontrol versi menjadi lebih mudah dielola langsung dari antarmuka editor. Dengan ini, pengembang dapat melacak perubahan kode, bekerja sama dengan tim, dan mengelola proyek secara efisien.

2.3.2 *Mongo DB Compass*

Mongo DB Compass merupakan antarmuka grafis yang digunakan untuk mengelola *database* dengan lebih intuitif. Dalam pengembangan sistem di PT Polytech Indo Hausen, *database NoSQL MongoDB* digunakan sebagai platform utama dalam pengelolaan data. Untuk meningkatkan efisiensi dan efektivitas dalam manajemen *database* tersebut, *MongoDB Compass* diadopsi sebagai alat bantu utama. *Mongo DB compass*

MongoDB Compass menyediakan berbagai fitur yang mendukung pengembangan dan pengelolaan *database*, termasuk kemampuan untuk memvisualisasikan dan mengedit dokumen secara langsung, mengimpor dan mengekspor data, serta memantau kinerja *query*. Alat ini dirancang untuk meningkatkan produktivitas pengembang dengan meminimalkan kebutuhan akan sintaks *query MongoDB* yang kompleks, sehingga memungkinkan pengguna untuk fokus pada peningkatan fungsi dan kinerja aplikasi [37].