

BAB III

METODOLOGI PENELITIAN

3.1 Gambaran Umum Objek Penelitian

Penelitian ini bertujuan untuk mengembangkan Sistem *Enterprise Resource Planning* (ERP) berbasis web dengan fokus pada Modul *Inventory Management* menggunakan teknologi *React JS* dan *Express JS* pada PT Polytech Indo Hausen. PT Polytech Indo Hausen, sebagai subjek penelitian, merupakan perusahaan yang bergerak di sektor manufaktur yang mendistribusikan produk kantong plastik.

Objek penelitian mencakup seluruh siklus pengembangan sistem, melibatkan analisis kebutuhan, desain, implementasi, pengujian, hingga penerapan pada lingkungan produksi perusahaan. Dalam implementasinya, *Visual Studio Code* (VS Code) digunakan sebagai lingkungan pengembangan utama, *Postman* sebagai alat pengujian API, *MongoDB* sebagai basis data, dan *GitHub* untuk manajemen kode sumber.

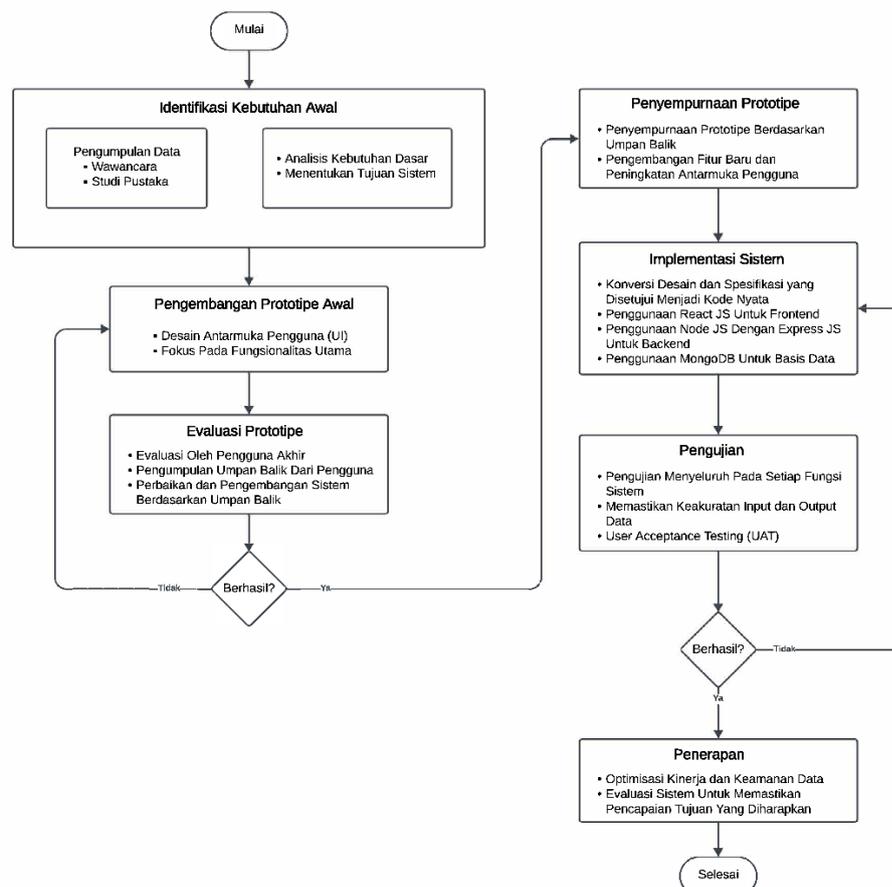
Proses pengembangan sistem difokuskan pada pemanfaatan *React JS* untuk menciptakan antarmuka pengguna interaktif dan *Express JS* sebagai teknologi *backend* untuk memastikan kinerja tinggi dan skalabilitas sistem. Solusi ini diharapkan dapat meningkatkan efisiensi operasional, memperkuat pengelolaan persediaan, dan mengoptimalkan proses *Inventory management*.

Melibatkan PT Polytech Indo Hausen sebagai lingkungan implementasi memberikan jaminan bahwa solusi yang dihasilkan dapat memenuhi kebutuhan spesifik perusahaan. Data yang digunakan dalam penelitian ini bersumber dari informasi yang diperoleh dari PT Polytech Indo Hausen. Analisis akan dilakukan untuk mengevaluasi dampak implementasi sistem ERP berbasis web terhadap efisiensi operasional dan pengelolaan persediaan di perusahaan tersebut.

3.2 Metode Penelitian

3.2.1 Alur Penelitian

Alur penelitian adalah pendekatan sistematis yang digunakan untuk merancang, melaksanakan, dan mengevaluasi suatu penelitian. Ini membantu dalam mengorganisir langkah-langkah yang diperlukan untuk mencapai tujuan penelitian. Dalam konteks pengembangan Sistem ERP Berbasis Web dengan Modul *Inventory Management* menggunakan *React JS* dan *Express JS* pada PT Polytech Indo Hausen, penelitian ini menggunakan metode SDLC atau *Software Development Life Cycle* dengan model *prototyping* sebagai proses yang dipakai dalam pengimplementasian pada penelitian ini. Berikut tahapan-tahapan dalam alur penelitian.



Gambar 3.1 Alur Penelitian

1. Identifikasi Kebutuhan Awal

Tahap awal dalam metode *Prototyping* adalah identifikasi kebutuhan awal yang melibatkan penentuan kebutuhan dasar dan tujuan sistem melalui wawancara dan studi pustaka. Dalam konteks PT Polytech Indo Hausen, tahap ini melibatkan diskusi dengan pihak perwakilan perusahaan untuk memahami tantangan dan kebutuhan spesifik dalam manajemen inventori.

2. Pengembangan Prototipe Awal

Pada tahap ini, prototipe awal dari sistem dikembangkan berdasarkan kebutuhan yang telah diidentifikasi. Prototipe ini bersifat sederhana dan fokus pada fungsionalitas utama yang diharapkan. Pengembangan ini melibatkan pembuatan antarmuka pengguna yang intuitif menggunakan *Figma* sebagai alat untuk mengembangkan UI. Prototipe awal ini memungkinkan pihak perusahaan untuk melihat dan merasakan bagaimana sistem akan bekerja.

3. Evaluasi Prototipe

Setelah prototipe awal dibuat, dilakukan evaluasi oleh pengguna akhir dari pihak perusahaan. Umpan balik dari pengguna sangat penting untuk memperbaiki dan mengembangkan sistem lebih lanjut. Proses evaluasi ini bersifat iteratif, di mana prototipe diperbaiki dan diuji kembali hingga memenuhi kebutuhan dan harapan pengguna. Evaluasi ini membantu dalam mengidentifikasi area yang perlu ditingkatkan, seperti antarmuka pengguna atau integrasi data.

4. Penyempurnaan Prototipe

Berdasarkan umpan balik yang diterima dari tahap evaluasi, dilakukan penyempurnaan dan pengembangan lebih lanjut pada prototipe hingga mencapai bentuk final yang siap diimplementasikan. Penyempurnaan ini mencakup penambahan fitur baru, dan peningkatan

antarmuka pengguna. Dalam konteks ini, penyempurnaan mungkin mencakup fitur-fitur lanjutan seperti *dashboard*, dan notifikasi.

5. Implementasi Sistem

Pada tahap implementasi, proses dimulai dengan konversi desain dan spesifikasi yang telah disetujui menjadi kode nyata. *React JS* digunakan sebagai *frontend* dan *Node JS* dengan *Express JS* untuk *backend*, dan *MongoDB* untuk basis datanya. Setiap fitur sistem diimplementasikan sesuai dengan fungsi yang telah dirancang, termasuk pengembangan API untuk pertukaran data yang aman dan efisien.

6. Pengujian

Setelah implementasi selesai, dilakukan pengujian menyeluruh pada setiap fungsi sistem untuk memastikan keakuratan *input* dan *output* data serta kinerja fitur sesuai dengan skenario penggunaan sehari-hari. Tahap ini juga mencakup *User Acceptance Testing* (UAT) yang dilakukan oleh karyawan PT Polytech Indo Hausen untuk mengumpulkan *feedback* yang akan menjadi wawasan penting untuk perbaikan lebih lanjut.

7. Penerapan

Setelah meninjau ulang mengenai sistem dan berjalan dengan baik. Penelitian ini mungkin dapat diterapkan secara nyata di perusahaan PT Polytech Indo Hausen. Apabila diterapkan, langkah pertama yang harus dilakukan adalah memastikan bahwa semua komponen sistem telah dioptimalkan untuk kinerja tinggi dan keamanan data. Kemudian proses migrasi data adalah kunci untuk transisi yang sukses ke sistem manajemen inventori baru. Ini melibatkan validasi data *pre-migrasi* dan *post-migrasi* yang insentif untuk memastikan integritas data terjaga. Selanjutnya adalah menyelenggarakan sesi pelatihan yang komprehensif untuk semua pengguna sistem, lalu saat hari peluncuran tiba, dilakukan monitoring kinerja dan keandalan sistem secara *real-*

time, dan sebagai bagian dari proses peninjauan berkelanjutan, harus dilakukan evaluasi sistem secara menyeluruh untuk memastikan tujuan yang diharapkan dapat tercapai.

Tabel 3.1 Perbandingan Model *Prototyping* dengan *Spiral*, dan *V-Shaped*

Aspek	<i>Prototyping</i>	<i>Spiral</i>	<i>V-Shaped</i>
Fokus Utama	Pengembangan iteratif berdasarkan umpan balik pengguna	Manajemen risiko melalui iterasi dan evaluasi	Pengujian yang ketat di setiap tahap pengembangan
Model Pengembangan	Iteratif dan inkremental	Iteratif dengan fokus pada analisis risiko	Linier dengan pengujian pada setiap tahap
Fleksibilitas dalam Perubahan	Sangat fleksibel, mudah menyesuaikan dengan perubahan	Fleksibel, perubahan diatur melalui <i>loop spiral</i>	Moderat, perubahan bisa dilakukan namun terbatas
Pengelolaan Risiko	Risiko dikelola berdasarkan umpan balik dan iterasi	Fokus utama adalah identifikasi dan mitigasi risiko	Risiko dikelola melalui pengujian pada setiap tahap
Waktu Implementasi	Relatif cepat karena produk dikembangkan secara bertahap	Variabel, tergantung pada jumlah iterasi spiral	Relatif cepat dengan fokus pada pengujian yang ketat
Cocok untuk Proyek	Proyek dengan persyaratan yang sering berubah	Proyek dengan tingkat risiko dan kompleksitas tinggi	Proyek yang membutuhkan validasi dan verifikasi
Output	Prototipe awal dan versi final berdasarkan iterasi	Prototipe, kemudian produk akhir	Produk yang divalidasi dan diverifikasi pada setiap tahap
Pengukuran Kinerja	Berdasarkan keberhasilan iterasi dan kepuasan pengguna [38].	Berdasarkan pengendalian risiko dan pencapaian tujuan iteratif.	Berdasarkan hasil pengujian pada setiap tahap [39].

Berdasarkan perbandingan pada Tabel 3.1, penelitian ini memilih model *Prototyping* untuk pengembangan sistem manajemen inventori dikarenakan terdapat aspek teknis yang mendukung kebutuhan spesifik

pada penelitian ini. Berikut penjelasan teknis mengapa *Prototyping* dipilih:

1. Responsif Terhadap Umpan Balik Pengguna

Tujuan utama dari penelitian ini adalah untuk mengembangkan sistem manajemen inventori yang efektif dan efisien, yang dapat memenuhi kebutuhan pengguna di PT Polytech Indo Hausen. Dengan menggunakan model *Prototyping*, pengembang dapat membuat prototipe awal yang memungkinkan pengguna untuk memberikan umpan balik langsung. Dalam konteks ini, pihak perusahaan memberikan saran mengenai fitur-fitur yang dibutuhkan seperti pelacakan pergerakan stok, integrasi pelanggan, dan integrasi distribusi. Umpan balik yang diterima dari pengguna digunakan untuk melakukan penyesuaian dan perbaikan pada prototipe, memastikan bahwa sistem yang dikembangkan benar-benar sesuai dengan kebutuhan operasional perusahaan.

2. Fleksibilitas dalam Pengembangan

Pengembangan sistem menggunakan *React JS* dan *Express JS* membutuhkan fleksibilitas tinggi untuk menyesuaikan dengan perubahan kebutuhan atau masalah yang teridentifikasi selama proses pengembangan. Model *Prototyping* memungkinkan pengembang untuk dengan cepat mengubah dan memperbaiki sistem berdasarkan umpan balik pengguna. Misalnya, jika setelah uji coba awal ditemukan bahwa antarmuka pengguna kurang intuitif, pengembang dapat segera melakukan iterasi untuk meningkatkan desain UI/UX. Fleksibilitas ini sangat penting dalam memastikan bahwa sistem yang dikembangkan dapat beradaptasi dengan permintaan pihak PT Polytech Indo Hausen.

3. Pengurangan Risiko Kegagalan

Dengan pendekatan iteratif yang diterapkan dalam model *Prototyping*, risiko kegagalan proyek dapat diminimalisir. Setiap iterasi menghasilkan versi prototipe yang lebih baik dan lebih dekat dengan produk akhir. Proses ini memungkinkan identifikasi dan penyelesaian masalah sejak dini, seperti *bug* dalam kode atau ketidaksesuaian fitur dengan kebutuhan bisnis. Dalam penelitian ini, setiap iterasi pengembangan modul inventori diuji secara menyeluruh untuk memastikan fungsionalitas yang stabil dan sesuai dengan kebutuhan pengguna sebelum melanjutkan ke tahap berikutnya.

4. Waktu Implementasi yang Cepat

Dalam penelitian ini, waktu implementasi yang cepat sangat penting untuk memberikan solusi yang segera dapat digunakan oleh PT Polytech Indo Hausen. Model *Prototyping* memungkinkan pengembangan produk dalam iterasi kecil yang dapat diimplementasikan lebih cepat. Fitur-fitur utama seperti pelacakan pergerakan stok, integrasi produk, penjualan, pelanggan, dan distribusi dapat segera diuji dan digunakan oleh pengguna, memberikan nilai tambah secara cepat dan memungkinkan perusahaan untuk mulai melihat peningkatan dalam manajemen inventori.

5. Validasi dan Verifikasi Berkelanjutan

Proses validasi dan verifikasi berkelanjutan dalam model *Prototyping* memastikan bahwa setiap komponen sistem berfungsi dengan baik sebelum digabungkan ke dalam sistem keseluruhan. Dalam konteks penelitian ini, setiap iterasi prototipe diuji untuk memastikan bahwa fungsionalitas seperti integrasi data, kinerja sistem, dan keamanan telah memenuhi standar yang ditetapkan.

Pengujian berkelanjutan ini meningkatkan kualitas akhir dari sistem manajemen inventori dan memastikan bahwa sistem tersebut stabil dan andal.

6. Pengembangan Berpusat pada Pengguna

Dalam penelitian ini, pendekatan pengembangan berpusat pada pengguna sangat penting untuk memastikan bahwa sistem yang dikembangkan benar-benar memenuhi kebutuhan pengguna akhir. Dengan melibatkan pengguna dalam proses pengembangan melalui sesi umpan balik dan uji coba prototipe, sistem manajemen inventori yang dihasilkan lebih mungkin untuk diterima dan diadopsi oleh pengguna. Pendekatan ini juga membantu meningkatkan tingkat kepuasan pengguna, karena mereka merasa kebutuhan dan preferensi mereka diperhatikan dalam pengembangan sistem.

3.2.1.1 Validasi dan Iterasi

Untuk memastikan sistem yang dikembangkan memenuhi kebutuhan pengguna, proses validasi dan iterasi dilakukan melalui serangkaian *User Acceptance Tests* (UAT). Setiap fase implementasi diikuti dengan sesi umpan balik dari pengguna pihak PT Polytech Indo Hausen. *Feedback* ini kemudian dianalisis dan digunakan untuk melakukan perbaikan pada antarmuka pengguna dan fungsionalitas sistem. Protokol ini mencakup pengujian sistem dalam skenario penggunaan nyata untuk mengidentifikasi dan memperbaiki *bugs*, mengoptimalkan UX/UI, dan memperbaiki kekurangan fungsionalitas.

3.2.2 Metode Pengembangan Sistem

Dalam konteks pengembangan sistem ERP berbasis web dengan modul *Inventory Management* menggunakan *React JS* dan *Express JS* pada PT Polytech Indo Hausen, metode pengembangan sistem yang digunakan melibatkan beberapa teknologi kunci, yaitu *React JS* untuk pengembangan *frontend*, *Express JS* untuk pengembangan *backend*, dan *MongoDB* sebagai basis data. Berikut adalah penjelasan lebih rinci mengenai penggunaan *React JS*, *Express JS*, dan *MongoDB*, serta kelebihan dan kekurangannya:

1. *React JS*

React JS digunakan untuk mengembangkan tampilan pengguna (UI) atau *frontend* dari sistem ERP. Keunggulan *React JS* terletak pada kemampuan untuk membuat antarmuka pengguna yang responsif dan dinamis. Komponen-komponen yang dapat digunakan ulang memudahkan pengembangan, dan *virtual DOM (Document Object Model)* *React* mempercepat proses *rendering*, meningkatkan kinerja aplikasi [40]. *React JS* dipilih sebagai teknologi utama untuk membangun UI atau *frontend* modul *Inventory Management* berbasis web ini karena beberapa keunggulannya, sebagai berikut:

a. Komponen yang Dapat Digunakan Kembali

React JS menggunakan konsep komponen yang dapat digunakan kembali, sehingga pengembang dapat membuat UI yang kompleks dengan mudah dan efisien. Hal ini memungkinkan untuk melakukan pengembangan dengan lebih cepat dan terstruktur, serta memudahkan pemeliharaan kode di masa yang akan datang.

b. Ekosistem yang Kuat

React JS memiliki komunitas dan ekosistem yang besar, dengan banyak Pustaka dan alat bantu yang tersedia untuk mempermudah pengembangan. Hal ini memungkinkan

pengembang untuk memanfaatkan berbagai sumber data dan dukungan untuk menyelesaikan proses pengembangan dengan lebih mudah dan efisien.

c. *Virtual DOM*

React JS menggunakan *virtual DOM* untuk meningkatkan performa *rendering* dan memperbarui UI dengan lebih cepat. Hal ini meningkatkan kinerja sistem dan memberikan pengalaman pengguna yang lebih mulus [41].

2. *Express JS*

Express JS digunakan sebagai *framework* yang dirancang untuk aplikasi web dan *mobile* dengan menggunakan *Node JS*. *Express JS* menyediakan alat-alat yang sangat bermanfaat untuk mengembangkan *server-side logic* dan API dengan cepat dan efisien, yang tentunya berguna juga dalam sistem ERP yang kompleks [42]. Dalam penelitian ini *Express JS* dipilih didasarkan beberapa keunggulan yang dimilikinya, yaitu:

a. Kemudahan Penggunaan *Middleware*

Express JS memungkinkan pengembang untuk menggunakan *middleware* yang membantu dalam berbagai fungsi seperti parsing *body request*, *cookie handling*, dan lebih banyak lagi. *Middleware* ini sangat berguna dalam penanganan sesi dan autentikasi pengguna dalam sistem ERP, memastikan keamanan dan akses data yang terkontrol.

b. *Routing* yang Fleksibel

Express JS menyediakan fitur *routing* yang sangat kuat dan fleksibel yang memungkinkan pengembang untuk membangun *route* aplikasi yang lebih bersih dan mudah dipahami. *Route* ini penting untuk mengarahkan permintaan ke fungsi-fungsi

backend yang sesuai, seperti menangani *queries database* atau operasi CRUD (*Create, Read, Update, Delete*) pada data inventaris.

c. Integrasi API

Express JS sangat ideal untuk mengembangkan API *RESTful* yang efisien. Hal ini memudahkan *frontend* yang dibangun dengan *React JS* untuk berkomunikasi dengan *database* dan logika *server* melalui *HTTP requests*. API *RESTful* memungkinkan interaksi data yang terstruktur dan mudah diintegrasikan dengan berbagai aplikasi dan *platform*.

d. Penanganan *Error* yang Lebih Baik

Express JS menyediakan mekanisme yang kuat untuk penanganan *error*. Hal ini memastikan bahwa aplikasi dapat menangani dan merespons terhadap kegagalan operasional atau masukan yang tidak valid dengan cara yang efisien. Penanganan *error* yang baik sangat penting untuk sistem ERP yang kompleks, di mana stabilitas dan keandalan aplikasi sangat penting.

e. *Model I/O Non-blocking*

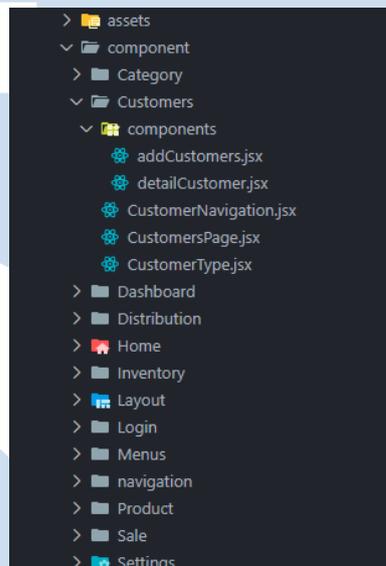
Dengan mengandalkan *model I/O* yang *non-blocking*, *Express JS* memastikan bahwa *server* dapat menangani jumlah permintaan yang besar secara simultan tanpa menghentikan operasi server untuk operasi tunggal. Hal ini sangat penting untuk aplikasi ERP di mana waktu *respons* cepat sangat dihargai.

f. Scalability

Express JS memudahkan penyesuaian aplikasi untuk menangani beban kerja yang lebih tinggi melalui teknik-teknik seperti *clustering* dan *load balancing*. Hal ini memastikan bahwa sistem ERP dapat tumbuh sesuai dengan kebutuhan perusahaan [43].

3.2.2.1 Metode Pengembangan *React JS*

Pengembangan antarmuka dengan *React JS* di PT Polytech Indo Hausen mengikuti beberapa metode berikut:



Gambar 3.2 Contoh Component *React JS* pada Sistem *Management Inventory* PT Polytech Indo Hausen

a. *Component-Based Development*

UI dibagi menjadi komponen-komponen kecil yang independent dan dapat digunakan kembali. Hal ini memungkinkan pengembangan yang modular dan terstruktur, serta memudahkan pemeliharaan kode.

b. *Data Flow Management*

Aliran data antar komponen dikelola menggunakan *state* dan *props*. *State* adalah data yang disimpan di dalam komponen, sedangkan *props* adalah data yang diteruskan dari komponen parent ke *child component*. Hal ini memungkinkan komponen untuk berkomunikasi satu sama lain dan berbagi informasi.

c. JSX

JSX adalah ekstensi sintaks *JavaScript* yang memungkinkan pengembang untuk menulis HTML secara langsung dalam *file JavaScript*. Hal ini memudahkan pengembang untuk membangun UI yang kompleks dengan cara yang lebih deklaratif dan mudah dipahami.

b. *Unidirectional Data Flow*

React JS mengikuti prinsip alur data satu arah, di mana data mengalir dari parent ke *child component*. Hal ini membuat kode lebih mudah dipahami dan dipelihara, serta membantu mencegah bug dan masalah sinkronisasi data.

c. *Tools dan Libraries*

Berbagai *tools* dan *library React JS* digunakan untuk mempermudah pengembangan, seperti *Create React App*, *Redux*, dan *React Router*. *Tools* dan *library* ini menyediakan fungsionalitas dan dukungan tambahan yang membantu pengembang membangun aplikasi *React JS* yang lebih kompleks dan skalabel [44].

Pemanfaatan *React JS* dalam pengembangan sistem ERP di PT Polytech Indo Hausen terbukti menjadi pilihan yang tepat. Keunggulan *React JS*, seperti kemampuannya dalam membuat UI yang responsif, komponen yang dapat digunakan kembali, *virtual DOM*, dan ekosistem yang kuat, memberikan manfaat

yang signifikan dalam pengembangan modul *Inventory Management*. Penerapan metode pengembangan *React JS* yang tepat, seperti *Component-Based Development*, *Data Flow Management*, *JSX*, *Unidirectional Data Flow*, dan penggunaan *tools* dan *library* yang sesuai, telah menghasilkan UI yang modular, terstruktur, mudah dipelihara, dan memberikan pengalaman pengguna yang optimal.

3.2.2.2 Metode Pengembangan *Express JS*

Pengembangan *server-side logic* dan API dengan *Express JS* di PT Polytech Indo Hausen menggunakan metode berikut:

a. *Middleware*

Middleware digunakan untuk berbagai fungsi, seperti *parsing body request*, *cookie handling*, autentikasi pengguna, dan otorisasi akses data. *Middleware* membantu menjaga kode *server-side logic* tetap terstruktur dan modular.

```
1  const jwt = require('jsonwebtoken');
2
3  // Middleware Autentikasi User
4  const authMiddleware = (req, res, next) => {
5    const authHeader = req.headers.authorization;
6
7    if (authHeader && authHeader.startsWith('Bearer ')) {
8      const token = authHeader.split(' ')[1];
9
10     try {
11       const decoded = jwt.verify(token, process.env.JWT_SECRET);
12       req.user = decoded;
13
14       next();
15     } catch (error) {
16       console.error('Token verification failed:', error.message);
17       res.status(401).json({ message: "Not authorized, token failed" });
18     }
19   } else {
20     console.error("No token provided");
21     res.status(401).json({ message: "No token, authorization denied" });
22   }
23 }
```

Gambar 3.3 Contoh *Code Middleware* Autentikasi Pengguna

```

25 // Middleware untuk memeriksa role admin
26 const adminCheck = (req, res, next) => {
27   if (req.user && req.user.role === 'admin') {
28     next();
29   } else {
30     res.status(403).json({ message: "Akses ditolak: hanya role admin yang
31       diperbolehkan" });
32   }
33 };
34 module.exports = { authMiddleware, adminCheck };

```

Gambar 3.4 Contoh *Code Middleware* Untuk Mengecek *Role Admin*

b. Routing

Route didefinisikan untuk mengarahkan permintaan HTTP ke fungsi-fungsi *backend* yang sesuai. *Route* yang jelas dan terstruktur memudahkan pengembang untuk memahami dan memelihara kode *server-side*.

c. Controller

Controller merupakan fungsi yang menangani permintaan HTTP dan melakukan operasi bisnis yang sesuai. *Controller* didefinisikan secara modular dan terstruktur, memungkinkan pengembang untuk fokus pada logika bisnis yang spesifik.

d. Model

Model mewakili data dan interaksi dengan *database*. Model didefinisikan secara terpisah dari *controller*, memastikan pemisahan tanggung jawab yang jelas dan mengurangi redundansi kode [45].

e. Testing

Pengujian *unit* dan integrasi digunakan untuk memastikan fungsionalitas dan stabilitas *server-side logic*

dan API. Pengujian yang menyeluruh membantu mencegah bug dan masalah di masa depan.

3.3 Teknik Pengumpulan Data

3.3.1 Studi Pustaka

Studi pustaka dilakukan untuk mengumpulkan informasi dan data dari berbagai referensi seperti buku, jurnal, artikel penelitian terdahulu, dan dokumentasi resmi yang relevan. Teknik ini digunakan untuk mendapatkan pemahaman yang mendalam tentang teknologi yang digunakan dan metodologi yang relevan. Dalam penelitian ini, studi pustaka membantu dalam memahami konsep dan implementasi dari framework yang digunakan, yaitu React JS, Express JS, dan MongoDB, serta membantu dalam mencari informasi mengenai Unified Modeling Language (UML) seperti class diagram, use case diagram, activity diagram. Informasi yang didapatkan dari hasil studi pustaka ini digunakan untuk membantu membuat sistem manajemen inventori yang lebih baik dan sesuai dengan kebutuhan.

3.3.2 Wawancara

Wawancara dilakukan untuk mendapatkan informasi mendalam dan pemahaman yang lebih baik tentang persepsi responden. Wawancara dilakukan dengan Ibu Nadya selaku perwakilan dari PT Polytech Indo Hausen dengan tujuan mengidentifikasi kelemahan dalam manajemen inventori saat ini, memahami kebutuhan pengguna, dan mendapatkan masukan langsung tentang fungsi dan proses yang diinginkan. Informasi yang diperoleh dari wawancara akan digunakan untuk memastikan bahwa sistem manajemen inventori yang sedang dikembangkan dapat mengatasi masalah-masalah yang ada dan meningkatkan efisiensi dari proses bisnis perusahaan.

Tabel 3.2 Teknik Pengumpulan Data

Teknik Pengumpulan Data	
Studi Pustaka	Wawancara
1. Mengumpulkan informasi dari buku, jurnal, artikel, dan dokumentasi resmi.	1. Wawancara dengan Ibu Nadya sebagai perwakilan dari PT Polytech Indo Hausen
2. Fokus pada framework yang digunakan (React JS, Express JS, MongoDB), dan Unified Modeling Language (UML).	2. Mengidentifikasi kelemahan dalam manajemen inventori saat ini.
3. Mendapatkan pemahaman mendalam tentang teknologi dan metodologi.	3. Memahami kebutuhan sistem.
	4. Mendapatkan masukan langsung tentang fungsi dan proses yang diinginkan.

3.4 Populasi dan Sampel

Dalam penelitian ini, penentuan populasi dan sampel sangat penting untuk memastikan bahwa data yang dikumpulkan dapat mewakili kebutuhan dan operasi inventori yang realistis di PT Polytech Indo Hausen.

3.4.1 Populasi

Populasi dalam penelitian ini adalah seluruh transaksi inventori yang mungkin terjadi dalam sistem manajemen inventori di PT Polytech Indo Hausen. Ini mencakup semua jenis transaksi yang berkaitan dengan penambahan, pengurangan, pergerakan stok barang, pelanggan, dan distribusi barang.

3.4.2 Sampel

Sampel diambil dari populasi tersebut dengan memilih transaksi inventori yang terjadi selama periode tiga bulan, yaitu dari Februari hingga April 2023. Pemilihan periode ini didasarkan pada:

1. Representativitas, data simulatif dibuat untuk mencerminkan aktivitas inventori yang realistis dan representatif.
2. Ketersediaan, data Data ini dibuat oleh peneliti berdasarkan saran dan masukan dari pihak perusahaan selama wawancara.
3. Relevansi Waktu, memilih periode waktu ini untuk memberikan gambaran yang konsisten dan relevan tentang pola dan masalah dalam manajemen inventori.

Data yang digunakan dalam penelitian ini disusun berdasarkan saran dan masukan dari pihak perusahaan mengenai fitur-fitur yang dibutuhkan. Prototipe UI kemudian dikembangkan dan diuji ulang dengan pihak perusahaan untuk memastikan kesesuaiannya dengan kebutuhan operasional.

3.5 Variabel Penelitian

3.5.1 Variabel Independen

Variabel independen adalah faktor-faktor yang dianggap sebagai penyebab atau pendorong perubahan dalam penelitian ini. Dalam konteks ini, variabel independen melibatkan unsur-unsur utama yang memengaruhi pengembangan sistem ERP dan implementasi modul *Inventory Management*. Beberapa contoh variabel independen melibatkan:

1. Penggunaan *React JS*, variabel ini mengukur sejauh mana penggunaan *React JS* memengaruhi kemudahan aksesibilitas dan responsivitas antarmuka pengguna pada sistem ERP.
2. Penggunaan *Express JS*, variabel ini mencerminkan dampak penggunaan *Express JS* terhadap performa *backend* sistem ERP, terutama dalam hal kecepatan dan efisiensi pengolahan data.
3. Peningkatan Kinerja Modul *Inventory Management*, variabel ini mengevaluasi sejauh mana peningkatan pada modul *Inventory Management* berkontribusi terhadap peningkatan efisiensi dan efektivitas dalam manajemen stok dan informasi persediaan.

3.5.2 Variabel Dependen

Variabel dependen mencakup hasil atau dampak dari implementasi sistem ERP berbasis web dengan modul *Inventory Management*. Dalam konteks ini, variabel dependen dapat diidentifikasi sebagai berikut:

1. Efisiensi Pengelolaan Persediaan, variabel ini mencerminkan tingkat efisiensi dalam manajemen persediaan, termasuk optimalisasi stok, penanganan pesanan, dan minimisasi kekurangan atau kelebihan persediaan.
2. Responsivitas Sistem, variabel ini mengevaluasi sejauh mana sistem merespons dengan cepat terhadap permintaan pengguna, khususnya dalam hal penambahan, pembaruan, atau penghapusan data inventori.
3. Performa Operasional Gudang, variabel ini mencakup hasil dari integrasi modul *Inventory Management*, termasuk efisiensi pengelolaan ruang gudang, peningkatan akurasi pengiriman, dan penurunan waktu pemrosesan.

Penyusunan variabel ini berdasarkan pada kebutuhan khusus PT Polytech Indo Hausen untuk mengintegrasikan teknologi terkini dalam manajemen inventori. Variabel-variabel ini dirancang untuk menggambarkan secara langsung hubungan sebab-akibat antara penggunaan teknologi spesifik dan peningkatan operasional di perusahaan.

