

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Teori yang Digunakan**

##### **2.1.1 Enterprise Resource Planning (ERP)**

Sistem ERP menangani fungsi perusahaan dari bisnis dasar mereka, seperti keuangan, sumber daya manusia, manajemen material, penjualan dan distribusi yang telah di support aplikasi paling populer di pasar Bisnis saat ini. *Enterprise Resource Planning*(ERP) adalah sistem bisnis yang mencakup fungsi yang keseluruhan dalam suatu perusahaan yang dikendalikan oleh modul perangkat lunak yang terintegrasi untuk mendukung proses bisnis dari proses dalam perseroannya langsung untuk perseroan tersebut. ERP memiliki beberapa macam modul seperti SCM (*Supply Chain Management*), CRM (*Customer Relationship Management*), MM (*Material Management*), HC (*Human Capital*), dan SD (*Sales and Distribution*). Penelitian ini akan lebih berfokus pada penggunaan SCM pada perusahaan, maka dari itu pembahasan akan lebih difokuskan pada SCM[2].

##### **2.1.2 Supply Chain Management**

Manajemen rantai pasok merupakan proses dalam mengelola berbagai macam kegiatan dengan tujuan untuk mendapatkan bahan yang belum jadi yang kemudian diubah dari barang yang belum jadi menjadi barang setengah jadi ataupun barang yang sudah jadi, kemudian hasil dari barang tersebut akan dikirimkan ke pelanggan atau konsumen yang telah diedarkan ke pasar-pasar[3]. Strategi ini dapat digunakan untuk menyelesaikan permasalahan terkait proses sourcing, manajemen pemasok, hubungan konsumen, hingga penentuan tingkat *outsourcing*. Apabila berbagai permasalahan tersebut dapat diatasi dengan tepat, maka bisnis perusahaan akan semakin kompetitif dan progresif di pasaran.

Dibawah ini merupakan tiga fungsi **SCM** yang perlu untuk diketahui.

### 1. Fungsi fisik

Proses produksi merupakan salah satu bentuk dari fungsi fisik rantai pasok. Tahapan pengubahan bahan mentah menjadi produk jadi dan biaya yang dikeluarkan selama prosesnya juga masuk dalam fungsi fisik SCM.

### 2. Fungsi mediasi

Dalam menyediakan bahan mentah, perusahaan perlu berhubungan dengan pemasok. Selain itu, dalam pengiriman produknya kepada para pelanggan, perusahaan juga membutuhkan seorang *distributor*. Dua alasan tersebut adalah gambaran dari fungsi *supply chain management* sebagai mediasi.

### 3. Fungsi biaya

Jika sebuah perusahaan mengatur jalannya rantai pasok, biaya yang berhubungan dengan riset pasar, desain produk, dan biaya operasionalisasi lainnya juga bisa dengan mudah dipantau dan ditinjau kembali. Konsep SCM meliputi cakupan kegiatan yang sangat luas.

Sementara untuk lebih memahami strategi yang tersedia, berikut ini ada beberapa komponen strategi SCM yang perlu untuk diketahui.

#### 1. Perencanaan (*Information*)

Menyusun sebuah perencanaan bisa membantu alur **SCM** agar lebih efisien. Proses ini berisikan kegiatan pengumpulan data, apa saja yang berkaitan dengan *supply chain*. Selain itu, tahap ini juga bisa dilakukan perusahaan untuk mengurangi resiko yang akan terjadi selama proses produksi hingga pengiriman produk kepada pelanggan.

#### 2. *Source and Inventory*

Strategi *supply chain management* adalah penjembutan antara perusahaan dan pemasok. Tentunya perusahaan ingin menghasilkan produk yang terbaik untuk pelanggan mereka, karena itu perlu adanya tindakan seleksi pemasok atau *supplier*. Sebuah *company* harus bisa memilih mana *source* yang kredibel atau tidak.

### 3. Produksi

Tidak hanya proses mengubah bahan mentah menjadi barang jadi, komponen satu ini meliputi uji coba produk hingga pengemasannya. Untuk meninggalkan impresi yang baik, Anda harus bisa memilih kualitas bahan mentah hingga kemasan layak untuk ditawarkan pada pelanggan.

### 4. Transportasi dan Pengiriman

Perusahaan harus menentukan secara tepat transportasi apa yang digunakan untuk memindahkan pasokan bahan dan produk. Dalam hal ini seorang *supply chain* manajer harus mampu mengelola jaringan distribusi dengan baik agar prosesnya berjalan lancar.

### 5. Pengembalian

Sebuah *company* harus bisa menangani komplain berupa pengembalian produk secara responsif. Sebaiknya susun perencanaan layanan tambahan bagi pelanggan yang memiliki masalah dengan produk Anda.

## 2.2 Metode yang Digunakan

### 2.2.1 Prototype

*Prototyping* merupakan teknik pengembangan sistem yang menggunakan prinsip prototipe untuk mendeskripsikan sistem sehingga pelanggan atau pemilik sistem akan memiliki gambaran lebih jelas mengenai sistem yang akan dibangun oleh *developer*. Dengan menggunakan metode *prototyping*, maka penerapan sistem akan menjadi lebih mudah karena adanya peran aktif dari pengguna[4].

## 2.3 Tools yang Digunakan

### 2.3.1 PHP

**PHP** merupakan bahasa pemrograman untuk membuat web yang bersifat *server – side scripting*. **PHP** memungkinkan untuk membuat halaman web yang bersifat dinamis. Sistem manajemen basis data yang sering digunakan bersama **PHP** adalah **MySQL** namun **PHP** juga mendukung sistem manajemen *database Oracle, Microsoft Access, Interbase, d-base, PostgreSQL*[5].

**PHP** adalah singkatan dari *Hypertext Preprocessor*, yang sebelumnya dikenal dengan sebutan *Personal Home Pages* dan merupakan bahasa pemrograman yang biasa digunakan untuk mengembangkan situs web atau aplikasi web statis maupun dinamis.

### 2.3.2 phpMyAdmin

**phpMyAdmin** adalah aplikasi berbasis *graphical user interface (GUI)* yang ditulis dalam bahasa pemrograman **PHP** yang biasa digunakan untuk mengelola *database MySQL*, selain itu **phpMyAdmin** mendukung berbagai operasi *MySQL*, diantaranya (mengelola basis data, tabel-tabel, bidang (*fields*), relasi (*relations*), indeks, pengguna (*users*), perijinan (*permissions*))[6].

### 2.3.3 Visual Studio Code

**Visual Studio Code** merupakan aplikasi *editor* teks gratis yang fleksibel untuk digunakan di semua bahasa pemrograman yang didukung tanpa harus berpindah aplikasi pengeditan serta dapat dijalankan di banyak *platform* lain secara bersamaan. **Visual Studio Code** dapat digunakan untuk *JavaScript, TypeScript dan Node.js* serta memiliki ekosistem ekstensi yang kaya untuk bahasa lain (seperti *C ++, C #, Java, Python, PHP, Go*) dan *runtime* (seperti *.NET dan Unity*)[7].

### 2.3.4 XAMPP

**XAMPP** adalah sebuah paket perangkat lunak yang dirancang untuk memudahkan pengembangan dan pengujian situs web secara lokal. Nama "**XAMPP**" sendiri merupakan singkatan dari:

- a. **X**: Sistem operasi (*cross-platform*, dapat digunakan pada *Windows, Linux, Mac*).
- b. **A**: *Server web Apache*.
- c. **M**: *Database server MySQL*.
- d. **P**: Bahasa pemrograman **Perl**.
- e. **P**: Bahasa pemrograman **PHP**.

XAMPP menyediakan lingkungan pengembangan web yang komprehensif dan mudah dipasang yang memungkinkan pengembang dan perancang untuk membuat dan menguji situs web tanpa perlu terhubung ke server web langsung di Internet. Ini sangat berguna dalam fase pengembangan dan pengujian proyek web sebelum peluncuran resminya[1].

Komponen utama **XAMPP**:

1. **Apache HTTP Server**: Merupakan *web server* yang digunakan untuk menyajikan *website*. *Apache* adalah salah satu *web server* yang paling populer dan dapat menangani berbagai bahasa pemrograman seperti *PHP, Perl*, dan lain-lain.
2. **MySQL Database Server**: Merupakan sistem manajemen basis data relasional (*RDBMS*) yang memungkinkan penyimpanan dan pengambilan data. *MySQL* sering digunakan dengan *PHP* untuk membuat aplikasi web dinamis.
3. **PHP, Perl, dan Python**: merupakan bahasa-bahasa pemrograman yang digunakan untuk membuat skrip *server-side*. *PHP* adalah yang paling umum digunakan di dunia web.
4. **phpMyAdmin**: Merupakan web grafis antarmuka yang memungkinkan pengelolaan *database MySQL* melalui browser. Hal ini membuat administrasi pada database menjadi lebih mudah.

Kegunaan **XAMPP** :

1. Pengembangan Lokal: **XAMPP** memungkinkan pengembang untuk membuat dan menguji halaman web secara lokal di

komputer mereka sebelum mengunggahnya ke *server web* langsung.

2. Pengujian Aplikasi Web: Dengan *XAMPP*, Anda dapat menguji fungsionalitas dan interaksi aplikasi web sebelum aplikasi tersebut diterapkan secara resmi.
3. Pelatihan dan Pembelajaran: *XAMPP* adalah pilihan yang baik untuk pelajar dan instruktur yang ingin mempelajari atau mengajar pengembangan web tanpa harus mengakses ke *server web* langsung.
4. Pengelolaan Basis Data: *phpMyAdmin* mempermudah manajemen basis data *MySQL* melalui antarmuka web, termasuk membuat, mengedit, dan menghapus tabel serta menjalankan kueri *SQL*.
5. Eksperimen dan Pengembangan Proyek Pribadi: *XAMPP* menyediakan *platform* yang nyaman bagi pengembang dan pengguna dalam menguji web atau membuat proyek pribadi.

*XAMPP* menyediakan lingkungan pengembangan web yang komprehensif dan mudah dipasang yang memungkinkan pengembang dan perancang untuk membuat dan menguji situs web tanpa perlu terhubung ke server web langsung di Internet. Ini sangat berguna dalam fase pengembangan dan pengujian proyek web sebelum peluncuran resminya.

### 2.3.5 UML

*Unified Modeling Language (UML)* adalah bahasa standar yang digunakan untuk memodelkan, mendokumentasikan, dan merancang sistem perangkat lunak. *UML* menyediakan simbol dan aturan grafis yang kaya yang digunakan untuk menggambarkan struktur dan perilaku sistem, serta memfasilitasi komunikasi antara tim pengembangan dan pemangku kepentingan[8].

Beberapa poin penting tentang *UML* adalah sebagai berikut :



1. Standar Industri: *UML* adalah standar industri yang diakui secara luas dalam menggambarkan penerapan perangkat lunak. Dikembangkan oleh *Object Management Group (OMG)*, *UML* telah menjadi bahasa standar untuk pemodelan perangkat lunak.
2. Notasi Grafis: *UML* menggunakan notasi grafis untuk mendeskripsikan berbagai aspek sistem, termasuk struktur statis (seperti kelas dan objek) dan perilaku dinamis (seperti diagram aktivitas dan diagram urutan).
3. Pemodelan Berorientasi Objek: *UML* secara khusus cocok untuk pemodelan sistem berorientasi objek, di mana sistem dilihat sebagai kumpulan objek yang berinteraksi satu sama lain.
4. Beragam Jenis Diagram: *UML* mencakup berbagai jenis diagram, masing-masing dengan fokus pada aspek tertentu dari sistem. Beberapa jenis diagram *UML* yang umum meliputi diagram kelas, *diagram use case*, diagram aktivitas, diagram urutan, dan banyak lagi.
5. Komunikasi Tim: *UML* membantu tim pengembangan untuk berkomunikasi dengan jelas dan efektif. Dengan menggunakan notasi grafis yang dapat dipahami oleh orang teknis dan non-teknis, *UML* memfasilitasi pemahaman yang lebih baik tentang desain sistem.
6. Pertumbuhan dan Perubahan: *UML* dirancang untuk mendukung pertumbuhan dan perubahan sepanjang siklus hidup pengembangan perangkat lunak. Hal ini memungkinkan pengembang untuk merinci dan merancang sistem dari tahap awal hingga tahap implementasi dan pemeliharaan.

*UML* memiliki peran penting dalam rekayasa perangkat lunak modern dan digunakan dalam berbagai konteks seperti analisis dan desain sistem, dokumentasi, dan komunikasi antar anggota tim. *UML* mencakup beberapa jenis diagram, masing-masing memberikan pandangan yang berbeda

terhadap sistem[9]. Berikut adalah beberapa elemen utama *UML* dan jenis diagram yang biasa digunakan:

Elemen-Elemen *UML* Utama:

1. Diagram Kelas (*Class Diagram*): Menampilkan struktur statis sistem, termasuk kelas, atribut, metode, dan hubungan antar kelas.
2. *Diagram Use Case (Use Case Diagram)*: Mendeskripsikan fungsionalitas sistem dari sudut pandang pengguna atau aktor eksternal dan hubungan antar *use case*.
3. Diagram Aktivitas (*Activity Diagram*): Menampilkan langkah-langkah aktivitas atau alur kerja dalam suatu proses atau fungsi.
4. Diagram Urutan (*Sequence Diagram*): Menunjukkan urutan interaksi antar objek dalam skenario tertentu.
5. Diagram State (*State Diagram*): Memodelkan transisi objek antar keadaan dalam suatu siklus hidup.
6. Diagram Komunikasi (*Communication Diagram*): Menunjukkan interaksi antar objek dalam konteks tertentu.
7. Diagram Objek (*Object Diagram*): Memperlihatkan contoh objek dan hubungannya pada waktu tertentu.
8. Diagram Paket (*Package Diagram*): Menunjukkan bagaimana elemen-elemen *UML* dikelompokkan ke dalam paket atau *namespace*.
9. Diagram Komponen (*Component Diagram*): Memodelkan berbagai komponen fisik atau logis dan hubungannya dalam suatu sistem.
10. Diagram Implementasi (*Deployment Diagram*): Menunjukkan bagaimana komponen-komponen fisik dapat diterapkan dan implementasinya pada perangkat keras.



### Konsep-Konsep Penting *UML*:

1. Kelas (*Class*): Mewakili model suatu objek dan mengidentifikasi atribut serta metode yang dimilikinya.
2. Objek (*Object*): *Instance* dari suatu kelas.
3. *Use Case*: Menyajikan fungsi-fungsi yang dapat dilakukan oleh sistem yang dilihat dari sudut pandang pengguna.
4. Aktor: Pihak yang berinteraksi dengan sistem dan memiliki peran tertentu.
5. Relasi: Menunjukkan hubungan antar elemen, seperti asosiasi, pewarisan, dan ketergantungan.
6. *Diagram*: Memvisualisasikan aspek sistem.

### Langkah-Langkah Penggunaan *UML*:

1. Identifikasi Kebutuhan: Pahami kebutuhan sistem dan pengguna.
2. Pemodelan Struktur: Gunakan Diagram Kelas untuk memodelkan struktur statis.
3. Pemodelan Fungsionalitas: Gunakan Diagram *Use Case* untuk memodelkan fungsionalitas sistem.
4. Pemodelan Perilaku: Gunakan berbagai diagram (*Sequence*, *Activity*, *State*, *Communication*) untuk memodelkan perilaku sistem.
5. Pemodelan Implementasi: Gunakan *Diagram Component* dan *Deployment* untuk memodelkan aspek implementasi.

Penting untuk diingat bahwa UML adalah alat yang dapat disesuaikan dan tidak semua jenis diagram diperlukan untuk setiap proyek. Pemilihan jenis diagram tergantung pada kebutuhan analisis dan desain sistem yang sedang dikerjakan.

Class Diagram (Diagram Kelas) adalah salah satu jenis diagram di Unified Modeling Language (UML) yang digunakan untuk memodelkan struktur statis sistem perangkat lunak[10]. Class Diagram memberikan pandangan tentang kelas-kelas yang ada dalam sistem, hubungan antar kelas, atribut dan metode yang dimiliki oleh kelas, serta visibilitas dari atribut dan metode tersebut.

#### Komponen Utama dalam *Class Diagram*:

1. Kelas (*Class*): Representasi dari suatu objek atau entitas dalam sistem. Kelas memiliki atribut (*variabel*) dan metode (fungsi atau perilaku).
2. Atribut (*Attribute*): Variabel yang dimiliki oleh kelas. Atribut menggambarkan data yang dapat disimpan oleh suatu objek dari kelas tersebut.
3. Metode (*Method*): Fungsi atau perilaku yang dimiliki oleh kelas. Metode menggambarkan tindakan atau operasi yang dapat dilakukan oleh objek dari kelas tersebut.
4. Hubungan (*Relationships*): Menunjukkan keterkaitan antar kelas. Beberapa jenis hubungan melibatkan asosiasi, pewarisan, dan komposisi.

#### Jenis-Jenis Hubungan dalam *Class Diagram*:

1. Asosiasi (*Association*): Hubungan yang menunjukkan bahwa objek dari satu kelas berhubungan dengan objek dari kelas lain. Contohnya, hubungan antara kelas Mahasiswa dan Kelas.
2. Pewarisan (*Inheritance*): Hubungan yang menunjukkan bahwa suatu kelas (*subclass* atau *child class*) mewarisi sifat atau karakteristik dari kelas lain (*superclass* atau *parent class*).
3. Komposisi (*Composition*): Hubungan di mana objek dari kelas satu (komponen) merupakan bagian dari objek dari kelas lain (komposit). Jika objek komposit dihapus, objek komponen juga akan dihapus.

*Diagram Use Case* adalah jenis diagram di **Unified Modeling Language (UML)** yang digunakan untuk menggambarkan fungsionalitas sistem dari perspektif pengguna atau aktor eksternal. *Use Case Diagram* membantu dalam memahami interaksi antara pengguna (aktor) dengan sistem dan menjelaskan skenario penggunaan sistem tersebut. Diagram ini juga dapat membantu tim pengembangan untuk merancang dan mengkomunikasikan kebutuhan fungsional sistem[11].

Komponen Utama dalam *Use Case Diagram*:

1. **Aktor (Actor)**: Merupakan entitas eksternal yang berinteraksi dengan sistem. Aktor bisa berupa pengguna manusia, perangkat keras, atau sistem eksternal lainnya.
2. **Use Case**: Representasi dari suatu fungsionalitas atau tugas yang dapat dilakukan oleh sistem. *Use case* menggambarkan interaksi antara aktor dan sistem untuk mencapai tujuan tertentu.
3. **Asosiasi**: Garis yang menghubungkan aktor dengan *use case*, menunjukkan keterlibatan aktor dalam *use case* tertentu.

*Activity Diagram* adalah jenis diagram di **Unified Modeling Language (UML)** yang digunakan untuk memodelkan aktivitas atau langkah-langkah dalam suatu proses atau alur kerja. Diagram ini memberikan pandangan visual tentang alur kerja suatu sistem dan cara berbagai aktivitas saling terkait. *Activity Diagram* sangat berguna untuk memahami dan mendokumentasikan proses bisnis atau alur kerja dalam suatu aplikasi[12].

Komponen Utama dalam *Activity Diagram*:

1. **Aktivitas (Activity)**: Representasi dari suatu tindakan atau langkah dalam proses. Aktivitas dapat mencakup operasi atau aksi konkret, seperti mengirim email atau mengisi formulir.
2. **Garisan Alur (Control Flow)**: Menunjukkan alur dari satu aktivitas ke aktivitas lainnya. Panah atau garis alur menunjukkan arah dan urutan aktivitas.

3. Keputusan (*Decision*): Menunjukkan titik dalam alur di mana sistem harus membuat keputusan, dan jalur yang akan diambil bergantung pada kondisi tertentu.
4. Bifurkasi (*Fork*): Menunjukkan pembagian jalur eksekusi menjadi dua atau lebih jalur yang berjalan secara bersamaan.
5. Gabung (*Join*): Menunjukkan penggabungan jalur eksekusi yang telah bercabang kembali menjadi satu.
6. *Swimlanes*: Menunjukkan tanggung jawab atau pemilik aktivitas. *Swimlanes* dapat dibagi berdasarkan peran, departemen, atau unit organisasi.

### 2.3.6 HTML

**HTML**, atau **Hypertext Markup Language**, adalah bahasa markup standar yang digunakan untuk membuat dan merancang halaman web. **HTML** memberikan cara untuk mendefinisikan struktur dan konten suatu halaman web dengan menggunakan elemen dan *tag markup*. Setiap elemen **HTML** memberikan informasi tentang bagaimana kontennya harus ditampilkan atau diinterpretasikan oleh *browser*[13].

Beberapa poin penting tentang HTML:

1. *Markup Language*: **HTML** menggunakan tag atau penanda untuk mengelompokkan dan menandai elemen-elemen dalam dokumen. *Tag* tersebut memberi petunjuk kepada *browser* bagaimana cara menampilkan konten.
2. Struktur Dokumen: **HTML** mendefinisikan struktur dasar suatu dokumen web. Hal ini melibatkan elemen-elemen seperti `<html>`, `<head>`, dan `<body>`.
3. *Elemen dan Tag*: *Elemen HTML* dibuat dengan menggunakan *tag*, seperti `<p>` untuk paragraf, `<h1>` untuk *heading level 1*, `<a>` untuk tautan, dan banyak lagi.
4. Atribut: *Elemen HTML* dapat memiliki atribut yang memberikan informasi tambahan tentang elemen tersebut.

Contohnya, atribut href pada tag <a> digunakan untuk menentukan URL tautan.

5. Versi **HTML**: **HTML** memiliki beberapa versi, dan versi terbaru saat ini adalah **HTML5**. Setiap versi mungkin menambahkan fitur baru atau perbaikan dari versi sebelumnya.

## 2.4 Penelitian Terdahulu

Tabel 2.1 Penelitian Terdahulu

<b>Penelitian 1</b>	
Judul	Desain dan Metodologi Teknologi Blockchain Untuk Monitoring Manajemen Rantai Pasokan Makanan yang Terdesentralisasi.
Penulis	Ninda Lutfiani, Fitra Putri Oganda, Chandra Lukita, Qurotul Aini, Untung Rahardja(2020)[14]
Indikator	-
Definisi	Meningkatkan efisiensi, transparansi, dan biaya rendah dalam penanganan kasus COVID-19. Teknologi blockchain digunakan dalam rantai pasokan untuk lalu lintas barang dan menciptakan transparansi dalam pasokan barang.
Metode	Metode yang digunakan pada penelitian tersebut adalah metode Blockchain.
Future Research	Pada penelitian berikutnya, Implementasi dan evaluasi model yang diusulkan sebaiknya dilakukan secara menyeluruh untuk memastikan kerapian dan efektifitas dalam penanganan kasus Virus Corona. Selain itu, melakukan penyederhanaan dan peningkatan akurasi model itu disarankan agar lebih mudah untuk diadopsi oleh berbagai pihak dalam rantai pasokan. Sebagai tambahan, penelitian juga dapat memeriksa keterbatasan yang dapat diidentifikasi dalam penelitian ini serta berupaya untuk mengurangnya melalui inovasi ataupun penyesuaian model. Peningkatan standarisasi data dan antarmuka dalam rantai pasokan serta pemeriksaan aspek hukum dan etika terkait dengan penggunaan teknologi blockchain juga menjadi fokus potensial yang dapat digunakan untuk penelitian mendatang.
<b>Penelitian 2</b>	
Judul	Kinerja Manajemen Rantai Pasok Kelapa di Provinsi Sulawesi Tengah
Penulis	Nur Rahmi Suud, Ria Indriani, Yuliana Bakari(2021)[15]
Indikator	Pemenuhan pesanan sempurna, Kinerja pengiriman, Siklus produksi, Lead time, Fleksibilitas volume, Total biaya, Siklus Cash to Cash, dan Persediaan harian.

Definisi	Penelitian ini bertujuan untuk menganalisis kondisi dan kinerja manajemen rantai pasok kelapa di perusahaan tersebut, dengan kinerja yang diukur menggunakan Analisis Food Supply Chain Network (FSCN) dan Analisis Supply Chain Operation Reference (SCOR). Hasil penelitian menunjukkan kondisi dan kinerja rantai pasok berjalan sangat baik.
Metode	Metode penelitian yang digunakan adalah metode survei dengan pengumpulan data primer dan sekunder. Selain itu, terdapat analisis yang dilakukan dengan menggunakan Analisis Food Supply Chain Network (FSCN) dan Analisis Supply Chain Operation Reference (SCOR).
Future Research	-
<b>Penelitian 3</b>	
Judul	Kinerja Organisasi Pada Manajemen Rantai Pasokan Pariwisata : Bagaimana Peran Manajemen Sumber Daya Manusia, Kepuasan Pelanggan ?
Penulis	Rayyan Aqila Praditya(2022)[16]
Indikator	-
Definisi	Penelitian ini mengeksplorasi hubungan antara sumber daya manusia dan rantai pasok serta menguji dampak SDM pada rantai pasok, kepuasan pelanggan, dan kinerja organisasi.
Metode	Metode yang digunakan adalah survei.
Future Research	Penelitian selanjutnya dapat difokuskan pada identifikasi elemen-elemen spesifik yang dapat digunakan dalam meningkatkan kinerja organisasi pada konteks manajemen rantai pasok pariwisata.
<b>Penelitian 4</b>	
Judul	Manajemen Rantai Pasok Pada Industri Pertahanan di Era Industri 4.0 dan Digital
Penulis	Thomas Dolfinus Fanulene, Dwi Soediantono(2022)[17]
Indikator	Manajemen Rantai Pasok dan Kinerja Perusahaan.
Definisi	Hasil analisis data menunjukkan bahwa Supply Chain Management berpengaruh signifikan terhadap kinerja industri pertahanan. Keunggulan bersaing juga berpengaruh positif terhadap kinerja industri pertahanan. Praktik manajemen rantai pasok dan responsivitas rantai pasok tidak berpengaruh pada keunggulan kompetitif.
Metode	Penelitian ini menggunakan metode kuantitatif dan teknik analisis data Structural Equation Modeling Equation Modeling. Selain itu, metode pemilihan sampel yang digunakan adalah metode snowball sampling.



Future Research	Hasil dan keterbatasan yang ditemukan dalam penelitian ini dapat menjadi inspirasi untuk pengembangan penelitian berikutnya. Perluasan penelitian yang diusulkan yakni melibatkan penambahan variabel bebas yang dapat memengaruhi keunggulan dalam bersaing. Untuk lebih lanjutnya, diperlukan sampel yang lebih luas terkait dengan hubungan variabel-variabel tersebut, terutama dalam konteks sektor industri lainnya. Adapun kendala dalam penelitian ini, yakni jumlah responden kurang dari target awal yang diharapkan dapat diwawancarai atau mengisi kuesioner. Selain itu, daerah penelitian yang luas juga dapat mempengaruhi lamanya penelitian berlangsung, sehingga penelitian dapat berlangsung cukup lama.
<b>Penelitian 5</b>	
Judul	Pengembangan Model Konseptual Manajemen Rantai Pasok Halal Studi Kasus Indonesia
Penulis	Yeni Rohaeni, Ahmad Hidayat Sutawijaya(2020)[18]
Indikator	-
Definisi	Model konseptual ini menggunakan perspektif pemangku kepentingan yang mencakup pemerintah, pelaku usaha/industri halal, perguruan tinggi/lembaga riset, pelaku usaha penunjang infrastruktur rantai pasok halal, dan masyarakat. Setiap pemangku kepentingan memiliki faktor-faktor kritis untuk keberhasilan implementasi manajemen rantai pasok halal. Pendalaman pada faktor-faktor dan peran di setiap pemangku kepentingan dilakukan melalui wawancara mendalam.
Metode	Metode penelitian yang digunakan merupakan metode studi pustaka dan wawancara narasumber ahli.
Future Research	-

Dengan merujuk pada penelitian-penelitian sebelumnya sebagaimana tergambar pada Tabel 2.1, penelitian ini bertujuan untuk mengembangkan sebuah modul Supply Chain Management (SCM) yang terintegrasi. Pendekatan yang akan digunakan dalam pengembangan ini adalah metode prototipe, diharapkan mampu memberikan kontribusi positif berdasarkan temuan-temuan dari penelitian-penelitian terdahulu.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A