

LAPORAN UAS
REKAYASA PERANGKAT LUNAK



KELOMPOK 2

Riva Dian Ardiansyah (22031554043)

Rizky Yanuar Kristiyanto (22031554017)

Ahmad Hilmy Rakha Alcander (22031554052)

PROGRAM STUDI S1 SAINS DATA
FAKULTAS PENGETAHUAN DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI SURABAYA

2025

Daftar Isi

I. PENDAHULUAN	3
1. Latar Belakang	3
2. Tujuan	3
3. Rumusan Masalah	3
4. Metodologi Perancangan	3
II. PEMBAHASAN	5
1. Perancangan Sistem	5
1. Use Case Diagram	5
2. State Diagram	7
3. Activity swimlane Diagram	9
4. Class Diagram	10
5. Spesifikasi Design Pattern	13
2. Pemodelan Waterfall	13
2.3.1 Requirement Analysis (Analisis Kebutuhan)	13
2.3.2 System Design (Perancangan Sistem)	13
2.3.3 Design System (Perancangan Sistem)	13
2.3.4 Implementation System	14
2.3.5 Testing (Penerapan)	14
2.3.6 Maintenance (Pemeliharaan)	14
3. Implementasi Pseudocode	14
III. KESIMPULAN	19
IV. DAFTAR PUSTAKA	20

I. PENDAHULUAN

1. Latar Belakang

Perusahaan *Harmon Corp.* membutuhkan sistem informasi untuk mendukung proses bisnisnya, termasuk penjualan barang secara *online* dan pembelian barang dari *supplier*. Sistem ini bertujuan untuk meningkatkan efisiensi operasional, mengurangi kesalahan manual, dan memberikan pelayanan optimal kepada pelanggan.

Maka dari itu, pengembangan sistem informasi penjualan *online* menjadi langkah strategis yang sangat penting bagi *Harmon Corp.* Sistem ini tidak hanya akan mendukung pertumbuhan bisnis, tetapi juga memberikan pengalaman belanja yang lebih baik bagi pelanggan serta meningkatkan daya saing perusahaan di tengah pasar digital yang terus berkembang.

2. Tujuan

- 1.2.1. Merancang sistem informasi menggunakan metodologi SDLC berbasis *Waterfall*.
- 1.2.2. Menerapkan tiga jenis *Design Pattern* dalam implementasi sistem.
- 1.2.3. Menghasilkan diagram teknis seperti *Use Case Diagram*, *Activity Swimlane Diagram*, dan *Pseudocode*.

3. Rumusan Masalah

- 1.3.1. Bagaimana merancang sistem informasi penjualan online yang mendukung proses bisnis *Harmon Corp.* secara efisien?
- 1.3.2. Bagaimana penerapan metodologi *SDLC Waterfall* untuk pengembangan sistem yang terstruktur?
- 1.3.3. Bagaimana menghasilkan diagram teknis (*Use Case*, *Activity Swimlane*, dan *Pseudocode*) yang jelas untuk menggambarkan alur sistem?

4. Metodologi Perancangan

E-Commerce adalah proses jual beli produk secara elektronik, baik antara konsumen maupun antara perusahaan, dengan menggunakan komputer sebagai alat untuk memfasilitasi transaksi bisnis. Ada juga yang menyebutkan bahwa e-Commerce, atau yang lebih dikenal dengan istilah Ecom atau Emmerce (EC), merupakan pertukaran bisnis yang berlangsung secara rutin melalui media seperti Electronic Data Interchange, email, papan pengumuman elektronik, faksimile, dan transfer dana elektronik yang terkait dengan transaksi belanja di platform belanja online[1].

E-Commerce merujuk pada penggunaan internet untuk melakukan transaksi jual beli secara online dengan cakupan yang lebih terbatas. E-Commerce sendiri merupakan bagian dari e-Business yang lebih luas, dengan fokus utama pada mekanisme transaksi elektronik [2]. Saat ini, transaksi dalam e-Commerce sudah dilengkapi dengan sistem yang dapat mengurangi risiko dan memberikan jaminan keamanan bagi penggunanya.

Secara ringkas, e-Commerce adalah aktivitas perdagangan yang dilakukan secara elektronik melalui transaksi jual beli online yang memanfaatkan internet dan menyediakan layanan "get and deliver". E-Commerce telah mengubah cara pemasaran, mengubahnya dari transaksi tradisional menjadi digital, yang pada gilirannya dapat mengurangi biaya operasional perusahaan.

Pengembangan sistem informasi penjualan online untuk *Harmon Corp.* akan menggunakan metodologi *Waterfall SDLC*[3]. karena pendekatannya yang terstruktur dan linier, dimana sangat cocok untuk proyek dengan kebutuhan yang jelas dan spesifikasi yang sudah dapat ditentukan sejak awal. Dalam metodologi *Waterfall*, setiap tahap pengembangan diselesaikan sebelum melanjutkan ke tahap berikutnya, yang membantu meminimalkan risiko kesalahan dan memastikan bahwa semua persyaratan telah dipenuhi sebelum melanjutkan ke tahap berikutnya.

II. PEMBAHASAN

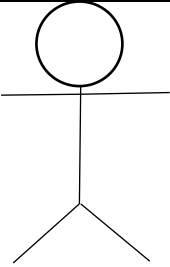


1. Perancangan Sistem


Perancangan sistem informasi untuk Harmon Corp. dilakukan dengan mengacu pada kebutuhan yang telah diidentifikasi[4]. Proses perancangan ini mencakup beberapa elemen utama, yaitu:

1. Use Case Diagram

Use Case Diagram menggambarkan interaksi antara aktor dan sistem. Aktor dalam sistem ini mencakup admin, pelanggan, sistem, dan kurir. Setiap aktor memiliki peran yang berbeda dalam mengakses fitur sistem, seperti mengelola pembelian barang jadi, penjualan daring dan luring, pengelolaan data, melakukan pemasaran

Simbol-simbol Use Case Diagram :

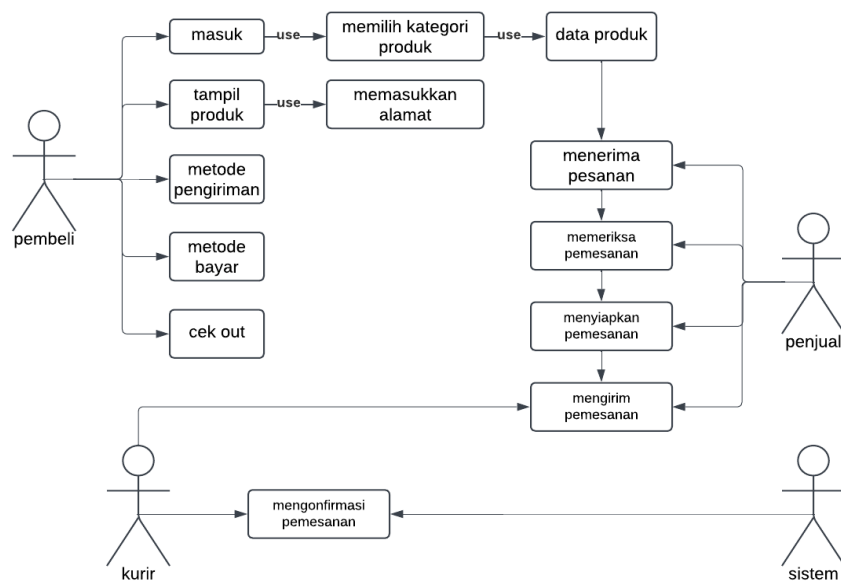
Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain atau alat ketika berkomunikasi dengan <i>use case</i>
	<i>Use Case</i> : Abstraksi dan interaksi antara sistem dan aktor
	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan use case

	<p><i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan use case</p>
---	--

Contoh dari Use Case Diagram

Berikut adalah Use Case Diagram untuk sistem informasi Harmon Corp

Usecase



1. Aktor:

- Pembeli
- Pelanggan
- Sistem
- Kurir

2. Use Cases:

- Mengelola pembelian barang jadi
- Mengelola penjualan daring dan luring


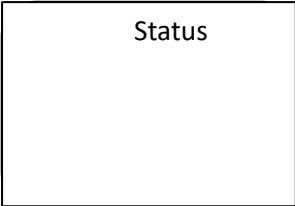
Alur :


1. Pembeli masuk kemenu
2. Setelah masuk pembeli melihat tampilan produk dan memilih kategori produk
3. Saat produk sudah terpilih pembeli memasukkan alamat lalu pembeli memilih metode pengiriman
4. Sesudah terpilih selanjutnya memilih metode pembayaran ketika sudah terpilih pembeli melakukan check out
5. Lalu penjual menerima pesanan data produk dari pembeli, tahap selanjutnya penjual melakukan pemeriksaan pesanan lalu menyiapkan pesanan, setelah semuanya siap penjual mengirimnya kepada kurir
6. Pihak kurir menerima pesanan, lalu kurir mengkonfirmasi ke sistem, sistem menerima

2. State Diagram

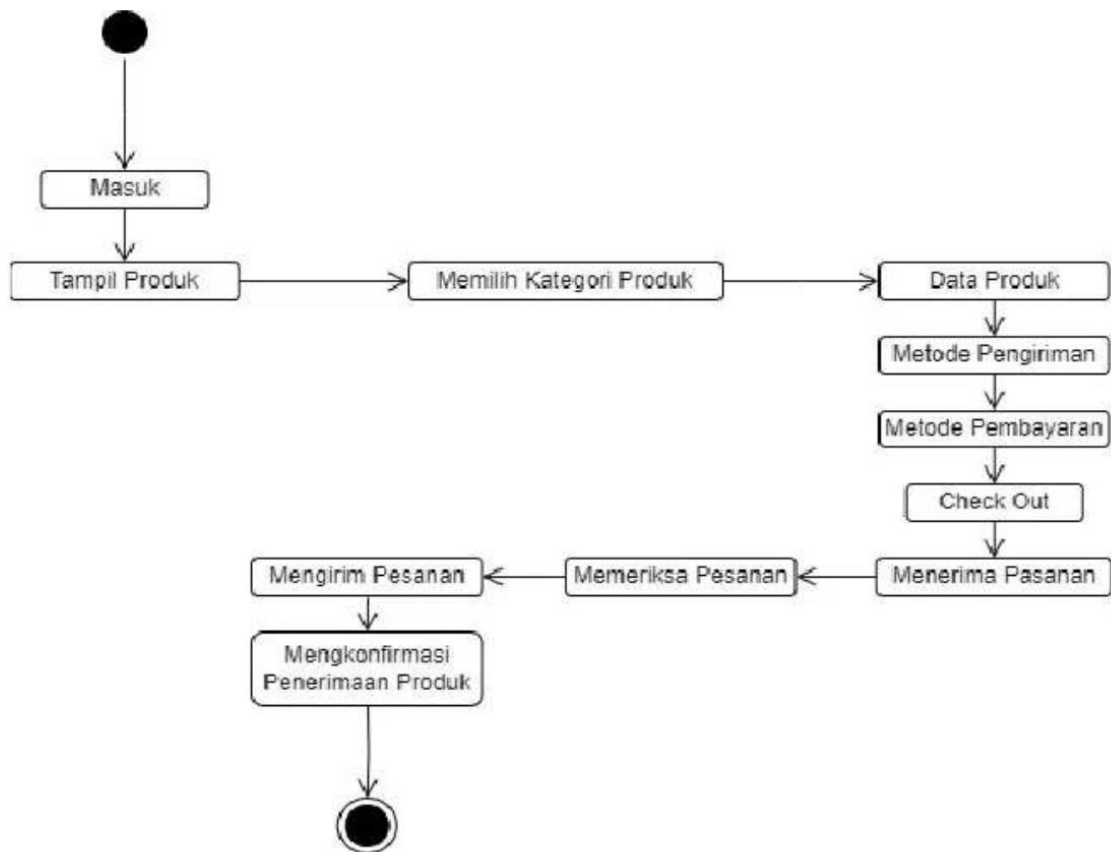
State diagram adalah diagram yang digunakan untuk mendeskripsikan perilaku sistem. State diagram mendeskripsikan semua kondisi yang mungkin muncul sebagai sebuah object begitu pula dengan event. Elemen-elemen didalam state diagram adalah kotak yang merepresentasikan state sebuah objek dan tanda panah untuk menunjukkan perpindahan ke state yang selanjutnya. Bagian aktivitas dari simbol state menggambarkan aktivitas object yang akan dilakukan selama didalam state.

Simbol-simbol State Diagram :

Simbol	Keterangan
Status awal/kondisi awal 	Status awal alur sebuah objek, sebuah diagram status memiliki sebuah status awal
Status 	Status yang dialami objek selama hidupnya

Status akhir/kondisi akhir 	Kondisi akhir alur hidup objek, sebuah diagram status memiliki sebuah status akhir
Transisi	Garis transisi antar status pada daur hidup objek, transisi biasanya diberi nama pesan yang ada pada diagram

Activity Diagram untuk Penjualan Daring





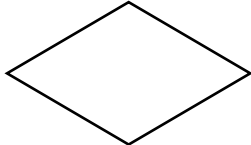



1. Pelanggan masuk lalu melihat tampilan produk.
2. Lalu pembeli memilih kategori produk setelah terpilih data produk akan masuk

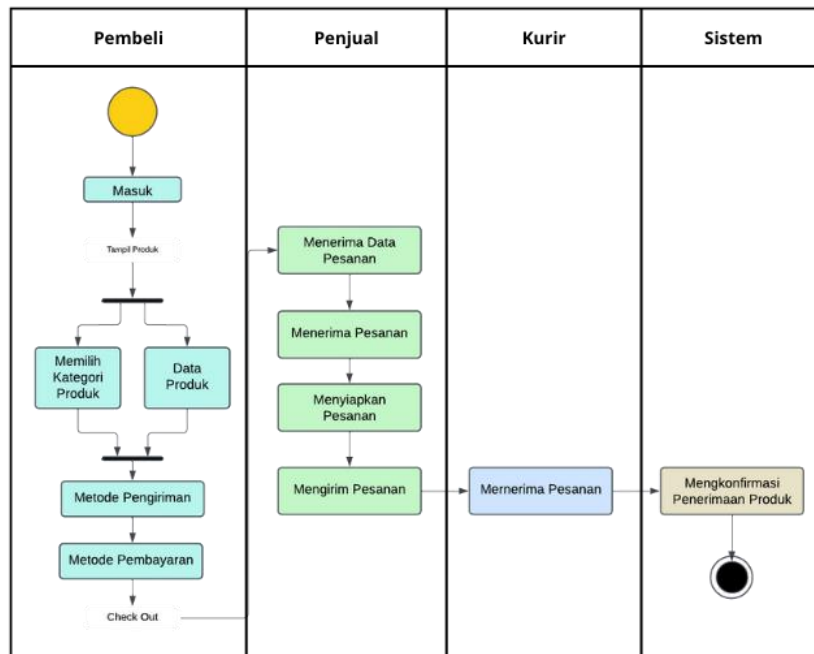
3. Setelah itu pembeli melakukan metode pengiriman ketika sudah terpilih maka pembeli akan memilih metode pembayaran dan selanjutnya pembeli checkout
4. Lalu penjual melihat dan menerima pesanan data produk dari pembeli
5. Ketika diterima penjual melakukan pemeriksaan data produk dari pembeli
6. Setelah itu maka penjual menyerahkan kepada pihak kurir dan kurir mengkonfirmasi
7. Lalu sistem menerima konfirmasi dari kurir

3. Activity swimlane Diagram

Activity Diagram digunakan untuk memvisualisasikan alur proses penjualan online, mulai dari pelanggan memilih produk, melakukan pembayaran, hingga pengiriman barang oleh kurir.

Simbol-simbol Activity Diagram :

Simbol	Nama	Keterangan
	Status awal/kondisi awal	Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem aktivitas biasanya diawali dengan kata kerja
	Percabangan/ Decision	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan/ Join	Penggabungan yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
	Status akhir/kondisi akhir	Kondisi akhir alur hidup objek, sebuah diagram status memiliki sebuah status akhir
	Swimlane	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi


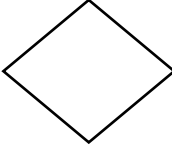

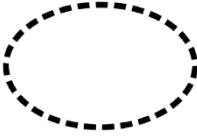





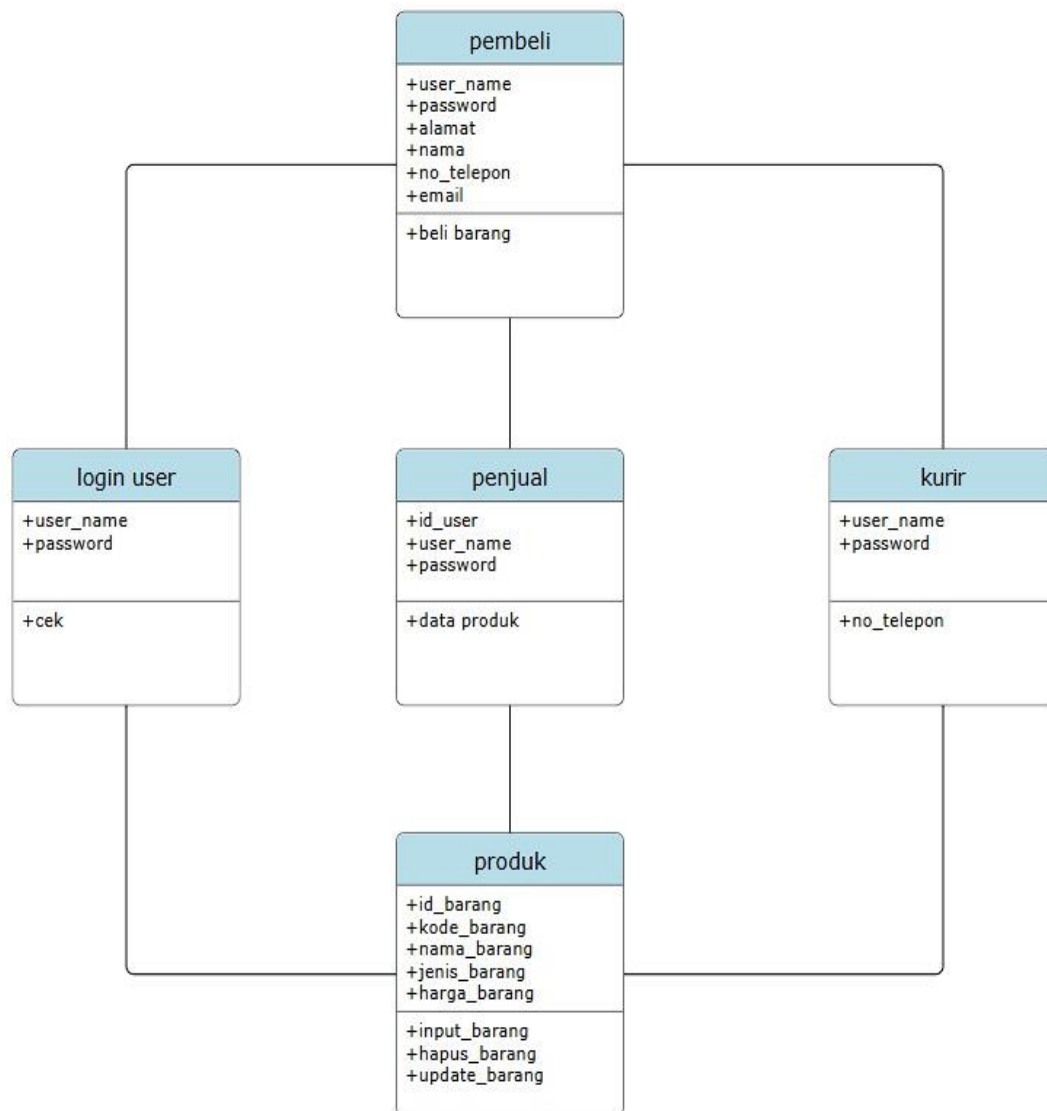
Alur:

1. Pembeli masuk ke apk lalu muncul tampilan produk
2. Kemudian pembeli memilih kategori produk setelah dipilih akan ada tampilan data produk
3. Setelah itu pembeli memilih metode pengiriman dilanjutkan dengan memilih metode pembayaran
4. Lalu pembeli melakukan checkout pesanan
5. Setelah pesanan siap maka data pesanan akan masuk ke penjual dan penjual menerima data pesanan lalu memeriksa pesanan, kemudian penjual menyiapkan pesanan
6. Pesanan yang siap akan dikirim ke pihak kurir lalu kurir menerima pesanan
7. Kemudian pihak kurir mengkonfirmasi pesanan kepada pihak Harmon Corp atau sistem
8. Setelah dikonfirmasi pihak Harmon Corp menerima konfirmasi pesanan dan data pesanan

4. Class Diagram

Diagram kelas atau diagram kelas adalah jenis diagram struktural dalam UML yang secara jelas menunjukkan struktur dan deskripsi kelas, atribut, metode, dan hubungan setiap objek[5]. Diagram kelas bersifat statis dalam arti menggambarkan hubungan apa yang terjadi, bukan apa yang terjadi ketika kelas-kelas dihubungkan satu sama lain.

Simbol	Nama	Keterangan
	Generalisasi	Hubungan dimana objek anak(<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Navy Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class</i>	Himpunan dari objek-objek yang berbagi ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independen</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri .
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.



Alur:

1. Dimulai dari class login user dimana memiliki username dan password dari masing-masing atribut, selanjutnya memiliki operation dimana login user memiliki cek
2. Selanjutnya bahwa tanda panah ini menunjukkan login user sub kelas dari pembeli yang artinya adalah login user saat ini memiliki 8 atribut dan dua operation yaitu cek dan beli barang
3. Selanjutnya adalah class penjual sama dengan login user menjadi sub kelas dari super class user sehingga keduanya mewarisin atribut dan operation dari super class
4. Class selanjutnya adalah penjual dengan beberapa atribut dan method, tanda panah yang menunjukkan bahwa login user sub kelas dari penjual yang memiliki 9 atribut dan dua method yaitu data produk dan beli barang.

5. Spesifikasi Design Pattern

Dalam implementasi sistem, digunakan tiga jenis design pattern:

- a. **Creational Pattern:** Factory Pattern digunakan untuk proses pembelian barang jadi (transaksi)
- b. **Structural Pattern:** Adapter Pattern digunakan untuk mengintegrasikan sistem dengan platform e-commerce pihak ketiga.
- c. **Behavioral Pattern:** Observer Pattern diterapkan untuk memberikan notifikasi otomatis kepada pelanggan terkait pembaruan stok atau diskon baru.

Perancangan sistem ini memastikan bahwa semua kebutuhan fungsional dan non-fungsional terpenuhi, sehingga sistem dapat berjalan secara efisien dan mendukung operasional Harmon Corp. secara menyeluruh.

2. Pemodelan Waterfall

Model *Waterfall* atau biasa disebut dengan model air terjun. adalah sebuah model siklus hidup klasik sistematis dalam mengembangkan sebuah *software* (atau yang biasa di kenal dengan nama perangkat lunak)[3]. Alur pengembangan sistem ini terstruktur dimulai dari perencanaan, analisis, desain, implementasi, pengoperasian dan pemeliharaan.

Model *Waterfall* ini memiliki beberapa langkah-langkah terstruktur dalam mengembangkan sistem sehingga menjadi produk yang siap pakai oleh pengguna[6]. Pada model ini tahapan yang harus dilalui oleh pengembang yaitu:

2.2.1 Requirement Analysis (Analisis Kebutuhan)

Di tahap ini melakukan perencanaan kebutuhan apa saja yang diperlukan pada penjualan *online* Harmon Corp ini. hasilnya terdiri dari Kurir, Penjual, Pembeli, dan Sistem.

2.2.2 System Design (Perancangan Sistem)

Pada tahap analisa sistem. Melakukan pengkajian sistem, software, metode, dan model apa yang cocok untuk digunakan pada Harmon Corp. Hasilnya penjualan *online* yang berbasis *website* dengan menggunakan sistem pengembangan Metodologi SDLC dengan Model *Waterfall*. Untuk *software* menggunakan *Python*.

2.2.3 Design System (Perancangan Sistem)

Tahapan ini disebut juga dengan tahap *blue print* atau cetak biru. Dimana tahapan ini menghasilkan purwarupa seperti design, pola, komponen, dan lain-lain. Hasilnya melakukan perancangan database, pembuatan Use Case Diagram Dan Activity Diagram Penjualan.

2.2.4 *Implementation System* (Penerapan Sistem)

Setelah itu di tahap ini. Maka penerapan Sistem dilakukan. Ditahap ini dilakukan Pengcodingan, dan pembuatan aplikasi berdasarkan Analysis System, dan Desain System. Hasilnya Aplikasi telah dibuat sesuai dengan Analysis System, Desain System. Menggunakan *Python*.

2.2.5 *Testing* (Penerapan)

Setelah aplikasi dibuat dengan menggunakan sistem yang diinginkan. Sistem Tersebut Diuji. Apakah sistem tersebut terjadi error atau rusak, tidak berjalan sesuai keinginan ataupun terjadi *bug*. Pada sistem yang dibuat. Hasilnya melakukan perbaikan pada sistem yang rusak atau error.

2.2.6 *Maintenance* (Pemeliharaan)

Tahapan terakhir dari metode *waterfall* adalah *Maintanance*. Pada tahapan ini sistem akan dilakukan pemeliharaan. Dan apabila ada *update* penambahan fitur dalam program yang di buat maka diperlukan *maintance*. Atau pada saat berjalannya sistem user menemukan sebuah kesalahan atau *bug* pada *website* yang di buat. Agar lebih optimal *maintenance* dapat dilakukan secara berkala.

3. Implementasi Pseudocode

Creational Pattern: Factory Pattern

1. Create an ActorFactory class with a create_actor method.
2. Check the actor_type input.
 - If "Pembeli", return a Pembeli object.
 - If "Penjual", return a Penjual object.
 - If "Kurir", return a Kurir object.
 - If "HarmonCorp", return a HarmonCorp object.
3. Raise an error for unknown actor_type.

```
1. class ActorFactory:

    def create_actor(self, actor_type):

        if actor_type == "Pembeli":

            return Pembeli()

        elif actor_type == "Penjual":

            return Penjual()

        elif actor_type == "Kurir":

            return Kurir()

        elif actor_type == "HarmonCorp":

            return HarmonCorp()

        else:

            raise ValueError("Unknown actor type")

class Pembeli:

    def action(self):

        return "Pembeli melakukan aksi membeli produk."

class Penjual:

    def action(self):

        return "Penjual mengelola produk dan pesanan."

class Kurir:

    def action(self):

        return "Kurir mengantarkan barang."

class HarmonCorp:

    def action(self):

        return "HarmonCorp memproses pesanan pelanggan."
```

```
# Penggunaan

factory = ActorFactory()

actors = ["Pembeli", "Penjual", "Kurir", "HarmonCorp"]

for actor_type in actors:

    actor = factory.create_actor(actor_type)

    print(f"{actor_type}: {actor.action()}")
```

Structural Pattern : Adapter Pattern

Pseudocode

1. Define an old system (e.g., OldPaymentSystem).
2. Create an adapter class (e.g., PaymentAdapter).
3. The adapter class translates the old interface into the new one.
4. Use the adapter to integrate old systems with the new one.

```
1. # Old system

class OldPaymentSystem:

    def process_payment(self):

        return "Memproses pembayaran lama"

# Adapter

class PaymentAdapter:

    def __init__(self, old_payment_system):

        self.old_payment_system = old_payment_system

    def make_payment(self):

        return self.old_payment_system.process_payment()

# Usage

old_payment = OldPaymentSystem()

payment_adapter = PaymentAdapter(old_payment)
```



```
print(payment_adapter.make_payment())
```

Behavioral Pattern : Observer

```
1. # Subject

class Order:

    def __init__(self):

        self.observers = []

        self.status = None

    def attach(self, observer):

        self.observers.append(observer)

    def detach(self, observer):

        self.observers.remove(observer)

    def notify(self):

        for observer in self.observers:

            observer.update(self.status)

    def update_status(self, status):

        self.status = status

        self.notify()

# Observers

class PembeliObserver:

    def update(self, status):

        print(f"Pembeli diberi tahu: Status pesanan sekarang '{status}'.")

class PenjualObserver:

    def update(self, status):

        print(f"Penjual diberi tahu: Status pesanan sekarang '{status}'.")

class KurirObserver:
```

```
def update(self, status):  
  
    print(f"Kurir diberi tahu: Status pesanan sekarang '{status}'.")  
  
# Penggunaan  
order = Order()  
  
# Menambahkan observer  
pembeli = PembeliObserver()  
penjual = PenjualObserver()  
kurir = KurirObserver()  
  
order.attach(pembeli)  
order.attach(penjual)  
order.attach(kurir)  
  
# Memperbarui status pesanan  
order.update_status("Pesanan sedang diproses")  
  
order.update_status("Pesanan telah dikirim")
```

III. KESIMPULAN

Kesimpulan dari laporan ini adalah bahwa pengembangan sistem informasi penjualan online untuk perusahaan Harmon Corp. berhasil dilakukan dengan pendekatan metodologi Waterfall. Tujuan utama pengembangan ini adalah meningkatkan efisiensi operasional, mengurangi kesalahan manual, dan memberikan pelayanan optimal kepada pelanggan.

Proses pengembangan dimulai dari analisis kebutuhan, perancangan sistem, implementasi pengkodean, pengujian, hingga tahap pemeliharaan. Sistem ini dirancang dengan memanfaatkan berbagai diagram teknis seperti Use Case Diagram, State Diagram, Activity Diagram, dan Class Diagram untuk memberikan gambaran yang jelas tentang alur dan struktur sistem. Selain itu, tiga pola desain diterapkan untuk meningkatkan modularitas dan efisiensi, yaitu Factory Pattern (Creational) untuk proses pembelian barang jadi, Adapter Pattern (Structural) untuk integrasi dengan platform pihak ketiga, dan Observer Pattern (Behavioral) untuk memberikan notifikasi otomatis kepada pelanggan.

Implementasi sistem menggunakan Python sebagai bahasa pemrograman utama. Hasil akhirnya adalah sebuah sistem yang mendukung pengelolaan penjualan daring dan luring, pengelolaan data, serta aktivitas pemasaran dengan baik. Sistem ini juga memiliki kemampuan untuk diintegrasikan dengan platform e-commerce lain, sehingga meningkatkan daya saing perusahaan.

Secara keseluruhan, pengembangan sistem ini telah memenuhi kebutuhan fungsional dan non-fungsional perusahaan. Untuk keberlanjutan, sistem ini memerlukan pemeliharaan dan pengembangan secara berkala agar tetap relevan dengan teknologi terbaru dan kebutuhan pelanggan di masa depan.

IV. DAFTAR PUSTAKA

- [1] Erlanie Sufarnap, Mirza Ilhami, and Jefri Junifer Pangaribuan, “Analisis dan Perancangan Sistem Informasi Penjualan pada Toko XYZ,” *SATESI: Jurnal Sains Teknologi dan Sistem Informasi*, vol. 2, no. 2, pp. 170–176, Oct. 2022, doi: 10.54259/satesi.v2i2.1181.
- [2] V. Apriana and U. Nurhasanah, “Implementasi Metode Waterfall Pada Sistem Informasi Penjualan Berbasis Web,” 2021. [Online]. Available: <https://jurnal.bsi.ac.id/index.php/akasia>
- [3] Rony Setiawan, “Metode SDLC Dalam Pengembangan Software,” <https://www.dicoding.com/blog/metode-sdlc/>.
- [4] S. H. Bariah and D. Pradina, “Implementasi SDLC Model Prototype Pada Sistem Informasi Company Profile SMP PGRI Bungbulang Berbasis Website,” vol. 10, no. 1, doi: 10.31980/jpetik.v10i1.1030.
- [5] B. Alturas, “Connection between UML use case diagrams and UML class diagrams: a matrix proposal,” *International Journal of Computer Applications in Technology*, vol. 72, no. 3, pp. 161–168, 2023, doi: 10.1504/IJCAT.2023.133294.
- [6] R. Awan Sagita, H. Sugiarto, A. BSI Bekasi, and A. BSI Jakarta, “Penerapan Metode Waterfall Pada Sistem Informasi Penjualan Furniture Berbasis Web,” Online, 2016.