

BAB II

TINJAUAN STUDI

2.1 Kajian Teori

2.1.1 Sistem

Sistem merupakan suatu objek dunia nyata yang tersusun dari berbagai komponen atau elemen yang saling terkait, saling bergantung, saling mendukung, dan bersatu sebagai satu kesatuan untuk mencapai suatu tujuan tertentu secara efektif dan efisien. (Ginting et al., 2022). Sedangkan menurut Rahmanuddin Tomalili (2019) menyatakan bahwa sistem merupakan jaringan dari berbagai proses yang saling berkaitan dalam mencapai tujuan, yaitu menggerakkan fungsi utama dari perusahaan. Informasi merupakan kumpulan data atau fakta yang diolah dan dikelola dengan cara yang dapat dimengerti dan berguna bagi penerimanya. Dapat dipahami dari definisi bahwa kata "informasi" memiliki arti yang berbeda dari kata "data". Data merupakan fakta mentah atau belum diolah, lalu setelah data diproses atau diolah, data tersebut dapat menjadi informasi yang berguna dan dapat digunakan untuk tujuan tertentu. (Ginting et al., 2022). Sedangkan menurut Tukino (2020) menyatakan bahwa informasi adalah data yang diproses sehingga menjadi sesuatu yang lebih bernilai bagi penerima untuk membantu mereka membuat keputusan.

Sistem informasi sendiri merupakan suatu sistem yang memberikan berbagai informasi bagi manajemen untuk mengambil keputusan dalam hal kebijakan manajemen suatu perusahaan. Sistem adalah kumpulan orang, teknologi informasi, dan proses yang sudah diatur untuk tujuan tertentu. (Ginting et al., 2022). Sedangkan menurut Piccoli dan Pigni (2018) menyatakan bahwa sistem informasi terdiri dari dua kata yaitu sistem dan informasi. Sistem merupakan sekumpulan orang yang bekerjasama secara teratur dengan fungsi yang sama dalam menentukan keputusan dan mencapai tujuan yang diinginkan. Sedangkan informasi merupakan hasil dari data yang telah diolah dalam proses pengambilan keputusan dalam suatu keadaan untuk mengurangi ketidakpastian. Sehingga sistem informasi dapat didefinisikan sebagai sebuah sistem yang akan mengatur dan menganalisa data sehingga dapat diolah menjadi sebuah informasi

2.1.2 Persediaan Bahan Baku

Dalam membangun suatu perusahaan tentunya terdapat kegiatan produksi yang membutuhkan persediaan bahan baku, karena dengan ketersediaan bahan baku diharapkan perusahaan dapat melakukan proses produksi sesuai dengan kebutuhan dan keinginan konsumen. Dengan memiliki persediaan bahan baku yang cukup dapat menghindari terjadinya kekurangan bahan baku, dan diharapkan dapat mendorong kegiatan produksi perusahaan. Persediaan bahan baku merupakan salah satu bagian terpenting dalam operasional suatu perusahaan. (Vikaliana et al., 2020)

2.1.3 Persediaan

Definisi persediaan merupakan barang atau bahan yang digunakan oleh suatu organisasi atau perusahaan untuk menjalankan bisnisnya. Barang ini digunakan untuk mendukung atau menyediakan kebutuhan produksi ketika sebuah perusahaan memproduksi suatu produk atau jasa (Musiafa, 2019). Persediaan dapat dikelompokkan menjadi lima jenis, antara lain:

- a. Persediaan bahan baku (*raw material*)
- b. Komponen rakitan (*part/components*)
- c. Bahan pembantu (*supplies*)
- d. Barang dalam proses (*work in process*)
- e. Barang jadi (*finished goods*)

2.1.4 MVC (*Model, View, Controller*)

Model, View, dan Controller atau biasa disebut dengan MVC merupakan konsep yang memudahkan pengembang untuk mengembangkan aplikasi web. Model View Controller ini menjadi pola arsitektur yang banyak dikenal dan digunakan oleh framework terkenal karena kinerjanya yang baik. MVC memisahkan kode dalam pembuatan aplikasi kedalam tiga komponen, yaitu :

- a. Model

Model adalah bagian yang bertanggung jawab untuk menyiapkan, mengorganisasikan, mengatur, dan memanipulasi data dalam basis data sesuai dengan instruksi dari controller. Pada bagian ini berisi data yang dilewatkan melalui komponen controller. Misalnya, pada sebuah situs web, controller akan

mengambil input berupa info pengunjung yang memiliki akun. Komponen model kemudian memanipulasi data dan mengirimnya kembali.

b. View

Bagian kedua dalam MVC yaitu view. Bagian ini berfungsi untuk menampilkan informasi berupa penyajian data dalam bentuk GUI (Graphical User Interface). Apa yang terlihat di tampilan berasal dari data yang dibuat pada komponen model. Komponen model memberikan informasi tentang data tertentu. kemudian, View menampilkannya kepada pengguna. View menampilkan data dalam bentuk bagan, grafik, dan tabel. Misalnya, pengguna aplikasi melihat informasi dari komponen UI dalam bentuk text box, opsi menu drop down, dan lain sebagainya.

c. Controller

Bagian terakhir yaitu bagian ketiga dari MVC adalah controller. Controller merupakan komponen yang dapat memperbaharui view dan model. Controller digunakan untuk menangani bagian interaksi pengguna. Controller akan memberikan ruang kepada pengguna untuk melakukan input agar sistem dapat diupdate. Misalnya, dalam sebuah aplikasi controller dapat mengubah model dengan memberikan akses kepada pengguna untuk mengubah informasi akun pengguna tersebut. Setelah itu, controller juga memperbarui view sehingga pengguna aplikasi dapat melihat informasi yang diperbaharui. (Mohamad, 2019)

2.1.5 Perancangan Basis Data

Perancangan basis data merupakan sebuah proses untuk menentukan isi dan penempatan data-data yang diperlukan untuk mendukung berbagai rancangan sistem. Defini basis data secara khusus adalah kumpulan dari berbagai macam Object data yang termasuk di dalamnya kumpulan *Form, Table, Image, Report, Query* dan lainnya. Data perlu disimpan, diolah, dan diproses di dalam database atau basis data sehingga informasi yang dihasilkan berkualitas dan efisien dalam penyimpanan data. (Aryanto, 2017)

Terdapat 5 jenis database yang beroperasi di dalam perangkat, antara lain :

a. *Operational Database (Json dan XML)*

Jenis ini memungkinkan pengguna untuk melakukan, melihat, dan mengedit data. Perubahan tersebut dapat berupa mengubah, menambah dan menghapus data secara langsung melalui perangkat keras yang digunakan.

b. *Database Warehouse (Microsoft SQL Server)*

Database Warehouse merupakan sistem basis data yang biasa digunakan untuk pelaporan dan analisis data. Sistem ini dianggap sebagai komponen inti dari intelijen bisnis. *Database Warehouse* juga merupakan pusat penyimpanan data terpadu dari satu atau lebih sumber yang berbeda. *Database* tersebut menyimpan data terkini dan historis di satu tempat yang digunakan untuk menghasilkan laporan analitik.

c. *Distributed Database (Microsoft Access (Office))*

Microsoft Access adalah sistem DBMS yang menggabungkan *Microsoft Jet Database Engine* dengan perangkat lunak. *Microsoft Access* menyimpan data dalam formatnya sendiri. Perangkat lunak ini memungkinkan pengguna untuk mengimpor data atau menghubungkan langsung ke data yang disimpan di *database* lain.

d. *Relational Database (MySQL, PostgreSQL, MariaDB, MongoDB, Oracle Database, SAP HANA, IBM Db2, MemSQL, Interbase, Firebird)*

Relational Database merupakan *database* yang diatur berdasarkan model relasi data. Banyak program menggunakan sistem ini untuk mengelola dan menyimpan basis data untuk informasi apa pun. Secara umum, semua sistem menggunakan *Structured Query Language (SQL)* sebagai bahasa pemrograman untuk pemeliharaan *database* dan query. *End-User Database (SQLite)*

e. *End-User Database*

SQLite adalah sistem manajemen basis data di library pemrograman C. Tidak seperti sistem lain, *SQLite* bukanlah mesin *database client-server*. *SQLite* tertanam dalam program akhir, sehingga cocok untuk digunakan dalam mendukung penyimpanan data pengguna akhir. *SQLite* adalah perangkat lunak basis data yang sangat populer untuk penyimpanan lokal/klien melalui perangkat lunak aplikasi seperti browser web. Sistem ini merupakan sistem yang paling banyak digunakan

dari sistem operasi, web browser dan lebih luas dari sistem embedded seperti ponsel. (Kadir, Abdul, 2020)

2.1.6 HTML

HTML (*Hyper Text Markup Language*) merupakan sekumpulan simbol-simbol atau tag-tag yang dituliskan dalam sebuah file yang digunakan untuk menampilkan halaman pada web browser. Tag dalam HTML diawali dengan <x> dan diakhiri dengan </x> dimana x tag HTML itu seperti b, i, u, dan lainnya. HTML sudah mengalami berbagai perkembangan dari tiap-tiap versinya. (Setyawan & Pratiwi, 2019).

2.1.7 Konsep OOAD

OOAD (Object Oriented Analysis and Design) memiliki konsep yang mencakup analisis dan desain sebuah sistem dengan pendekatan objek, yaitu analisis berorientasi objek (OOA) dan desain berorientasi objek (OOD). OOA (Object Oriented Analysis) merupakan metode analisis yang memeriksa requirement (persyaratan/keperluan yang harus dipenuhi oleh sistem) dari perspektif kelas dan objek yang terdapat dalam bisnis perusahaan. Sedangkan OOD merupakan metode pengendalian arsitektur perangkat lunak berdasarkan manipulasi objek sistem atau subsistem. (Zurfia, 2022)

1. Teknik Pemodelan *Object*

Teknik pemodelan objek menggunakan tiga macam model untuk menggambarkan sistem, diantaranya:

- a) Model Objek
 - 1) Model objek menggambarkan struktur statis dari suatu objek dalam sistem dan relasinya
 - 2) Model objek berisi diagram objek. Diagram objek adalah graph dimana nodenya adalah kelas yang mempunyai relasi antar kelas.
- b) Model Dinamik
 - 1) Model dinamik menggambarkan aspek dari sistem yang berubah setiap saat.
 - 2) Model dinamik dipergunakan untuk menyatakan aspek kontrol dari sistem.

- 3) Model dinamik berisi state diagram. State diagram adalah graph dimana nodenya adalah state dan arc adalah transisi antara state yang disebabkan oleh event.
- c) Model Fungsional
 - 1) Model fungsional menggambarkan transformasi nilai data di dalam sistem.
 - 2) Model fungsional berisi data flow diagram. DFD adalah suatu graph dimana nodenya menyatakan proses dan arcnya adalah aliran data.

2. Konsep *Object Oriented*

a) *Encapsulation* (Pengkapsulan)

- 1) *Encapsulation* merupakan dasar untuk pembatasan ruang lingkup program terhadap data yang diproses.
- 2) Data dan prosedur atau fungsi dikemas bersama-sama dalam suatu objek, sehingga prosedur atau fungsi lain dari luar tidak dapat mengaksesnya.
- 3) Data terlindung dari prosedur atau objek lain, kecuali prosedur yang berada dalam objek itu sendiri.

b) *Inheritance* (Pewarisan)

- 1) *Inheritance* adalah teknik yang menyatakan bahwa anak dari objek akan mewarisi data/atribut dan metode dari induknya langsung. Atribut dan metode dari objek dari objek induk diturunkan kepada anak objek, demikian seterusnya.
- 2) *Inheritance* mempunyai arti bahwa atribut dan operasi yang dimiliki bersama di antara kelas yang mempunyai hubungan secara hirarki.
- 3) Suatu kelas dapat ditentukan secara umum, kemudian ditentukan spesifik menjadi subkelas. Setiap subkelas mempunyai hubungan atau mewarisi semua sifat yang dimiliki oleh kelas induknya, dan ditambah dengan sifat unik yang dimilikinya.
- 4) Kelas Objek dapat didefinisikan atribut dan service dari kelas Objek lainnya.
- 5) *Inheritance* menggambarkan generalisasi sebuah kelas

- 6) Merupakan konsep yang menyatakan bahwa sesuatu yang sama dapat mempunyai bentuk dan perilaku berbeda. Contoh : kembar identik

3. Tema *Object Oriented*

Terdapat beberapa tema sebagai dasar teknologi berorientasi objek sebagai penunjang sistem berorientasi objek

a) Abstraksi

- 1) Abstraksi berarti fokus pada aspek yang melekat dari entitas dan mengabaikan sifat yang sementara.
- 2) Dalam pengembangan sistem, fokus pada apakah suatu objek dan apa yang dikerjakan oleh objek tersebut, sebelum menentukan implementasinya.
- 3) Penggunaan abstraksi yang sesuai memungkinkan model yang sama digunakan untuk analisa perancangan tingkat tinggi, struktur program, struktur basis data, dan dokumentasi.
- 4) Menggunakan abstraksi dalam analisa berarti hanya melakukan konsep domain aplikasi, tidak menentukan desain dan implementasi sebelum masalah dipahami.

b) Pengkapsulan

- 1) Pengkapsulan (information hiding) terdiri dari pemisahan aspek eksternal dari suatu objek, di mana dapat diakses oleh objek lain.
- 2) Pengkapsulan melindungi program dari saling keterkaitan sehingga kesempatan kecil mempunyai akibat penyimpangan.

c) Menggabungkan Data dengan Perilaku

- 1) Pemanggilan dari operasi tidak perlu dipikirkan berapa banyak implementasi yang diberikan oleh operasi yang ada.
- 2) Contoh, code yang bukan berorientasi pada objek menampilkan isi dari window harus membedakan tipe dari setiap bentuk, seperti poligon, lingkaran atau teks dan memanggil prosedur yang tepat untuk menampilkannya. Program berorientasi objek lebih sederhana memohon operasi draw pada setiap bentuk, penentuan di mana prosedur digunakan dibuat implisit oleh setiap objek berdasarkan kelasnya sehingga Tidak perlu mengulang pilihan prosedur setiap saat

operasi dilakukan dalam program aplikasi. Pemeliharaan lebih mudah, karena prosedur yang dipanggil tidak perlu dimodifikasi setiap kelas baru ditambahkan. Dalam sistem berorientasi objek hirarki struktur data identik dengan pewarisan hirarki operasi.

- d) Penggunaan Bersama (sharing)
 - 1) Teknik berorientasi objek menawarkan penggunaan bersama untuk beberapa tingkat yang berbeda. Pewarisan struktur data dan perilaku memungkinkan penggunaan bersama antara beberapa subkelas yang sama tanpa redundancy. Pemakaian bersama code adalah keuntungan utama dari bahasa berorientasi objek.
 - 2) Pengembangan berorientasi objek tidak hanya mengizinkan informasi dipergunakan bersama dalam aplikasi, tetapi memberikan prospek untuk menggunakan desain dan code bersama pada prospek mendatang. (Kusuma et al., 2022)

2.1.8 UML

Unified Modelling Language (UML) merupakan suatu bahasa untuk visualisasi, perancangan, dan juga pendokumentasian sistem perangkat lunak. Selain itu UML juga merupakan suatu metode pemodelan menggunakan sarana perancangan sistem berorientasi objek. Notasi-notasi yang dibutuhkan dalam pemodelan sistem dari berbagai sudut pandang sudah tersedia pada UML. Semua bidang yang membutuhkan pemodelan dapat menggunakan UML dengan artian UML tidak hanya digunakan untuk pemodelan perangkat lunak saja. (Nugroho et al., 2020).

UML memiliki beberapa diagram di dalamnya, diantaranya:

1. *Structural* Diagram

Dalam diagram ini memiliki beberapa diagram yang menggambarkan hubungan antar objek secara struktur pada sistem yang dimodelkan.

Adapun diagram yang terdiri, diantaranya:

- a. *Class* Diagram yaitu diagram yang menggambarkan hubungan antar kelas dalam sistem yang dibangun dan juga menjelaskan bagaimana mereka bekerja sama.

Simbol-simbol yang digunakan dalam *Class* Diagram.

Tabel 2. 1 Simbol-simbol Class Diagram (Atmaluhur.ac.id)

No	Gambar	Nama	Keterangan
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

- b. *Object* Diagram merupakan gambaran objek-objek dalam sistem pada satu titik waktu. *Object* diagram disebut dengan sebuah diagram perintah karena pada diagram ini lebih menonjolkan beberapa perintah dibandingkan kelas.

Simbol-simbol yang digunakan dalam *Object* Diagram:

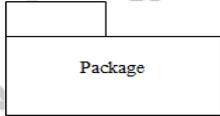
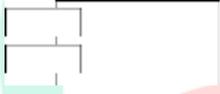
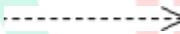
Tabel 2. 2 Simbol-simbol Object Diagram (Docplayer)

Simbol	Deskripsi
Objek 	Objek dari kelas yang berjalan saat sistem dijalankan
Link 	Relasi antar objek

- c. *Component Diagram* merupakan diagram yang menampilkan simbol-simbol struktur yang biasa digunakan dalam sistem dan hubungan antar komponen.

Simbol-simbol yang digunakan dalam *Component Diagram*

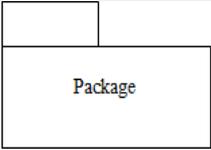
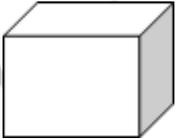
Tabel 2. 3 Simbol-simbol Component Diagram (Unikom.ac.id)

Simbol	.Keterangan
	PACKAGE <i>Package</i> merupakan sebuah bungkus dari satu atau lebih komponen
	KOMPONEN/COMPONENT Komponen merupakan komponen sistem.
	KEBERGANTUNGAN/DEPENDENCY Ketergantungan atau <i>dependency</i> atau kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai.
	ANTAR MUKA/INTERFACE Antar muka atau <i>interface</i> merupakan antarmuka sama dengan interface pada pemrograman berorientasi objek, yaitu sebagai antar muka komponen agar tidak mengakses langsung komponen.
	LINK Menggambarkan relasi antar komponen.

- d. *Deployment Diagram* merupakan sebuah gambaran struktur visualisasi sebuah sistem dari hubungan antara *software* dan *hardware*.

Simbol-simbol yang digunakan dalam *Deployment Diagram*

Tabel 2. 4 Simbol-simbol Deployment Diagram (Binus.ac.id)

Simbol	Keterangan
	PACKAGE <i>Package</i> merupakan sebuah bungkus dari satu atau lebih komponen.
	NODE <i>Node</i> biasa mengacu pada perangkat keras (<i>hardware</i>), perangkat lunak yang tidak dibuat sendiri (<i>software</i>), jika di dalam node disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang telah didefinisikan sebelumnya pada diagram komponen.
	KEBERGANTUNGAN/DEPENDENCY Ketergantungan atau <i>dependency</i> atau kebergantungan antar <i>node</i> , arah panah mengarah pada <i>node</i> yang dipakai.
	LINK Menggambarkan relasi antar node.

2. Behavioral Diagram

Dalam diagram ini menggambarkan bagaimana sebuah objek dapat berinteraksi satu sama lain untuk membuat sistem yang berfungsi, serta dalam diagram ini menunjukkan apa yang harus terjadi dalam suatu sistem tersebut.

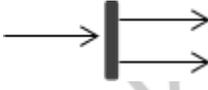
Adapun diagram yang terdiri, diantaranya:

- a. *Activity Diagram*, diagram dibuat untuk menggambarkan alur kerja yang berisi aktivitas pengguna atau sistem.

Simbol-simbol yang digunakan dalam *Activity Diagram*:

Tabel 2. 5 Simbol-simbol Activity Diagram (DosenIT.com)

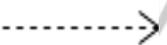
Simbol	Keterangan
	STATUS AWAL/INITIAL Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.

	AKTIVITAS/ ACTIVITY Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	PERCABANGAN / DECISION Asosiasi percabangan dimana lebih dari satu aktivitas digabungkan menjadi satu.
	PENGGABUNGAN/ JOIN Asosiasi penggabungan dimana lebih dari satu aktivitas lebih dari satu.
	STATUS AKHIR/ FINAL Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status satu.
	SWIMLINE Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

- b. *Use Case Diagram* merupakan diagram yang digunakan untuk menggambarkan secara singkat siapa yang menggunakan sistem dan apa yang dapat mereka lakukan.

Simbol-simbol yang digunakan dalam *Use Case Diagram*:

Tabel 2. 6 Simbol-simbol *Use Case Diagram* (Widuri.Raharja)

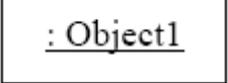
No	Gambar	Nama	Keterangan
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .

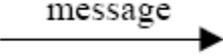
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

c. *Sequence Diagram* merupakan diagram yang menggambarkan hubungan pengguna dengan sistem secara berurutan.

Simbol-simbol yang digunakan dalam *Sequence Diagram*:

Tabel 2. 7 Simbol-simbol *Sequence Diagram* (Widuri.Raharja)

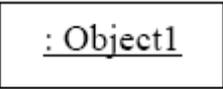
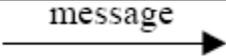
Simbol	Nama	Keterangan
	<i>Object</i>	Object merupakan instance dari sebuah class dan dituliskan tersusun secara horizontal. Digambarkan sebagai sebuah class (kotak) dengan nama obyek didalamnya yang diawali dengan sebuah titik koma
	<i>Actor</i>	Actor juga dapat berkomunikasi dengan object, maka actor juga dapat diurutkan sebagai kolom. Simbol Actor sama dengan simbol pada Actor Use Case Diagram.

	<i>Lifeline</i>	Lifeline mengindikasikan keberadaan sebuah object dalam basis waktu. Notasi untuk Lifeline adalah garis putus-putus vertikal yang ditarik dari sebuah obyek.
	<i>Activation</i>	Activation dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah lifeline. Activation mengindikasikan sebuah obyek yang akan melakukan sebuah aksi.
	<i>Message</i>	Message, digambarkan dengan anak panah horizontal antara Activation. Message mengindikasikan komunikasi antara object-object.

d. *Collaboration* Diagram merupakan diagram yang berbeda bentuknya dari diagram *sequence*, diagram ini menggambarkan struktur organisasi dari sistem termasuk pesan yang masuk dan keluar.

Simbol-simbol yang digunakan dalam *Collaboration* Diagram:

Tabel 2. 8 Simbol-simbol Collaboration Diagram (SlidePlayer)

Simbol	Nama	Keterangan
	<i>Object</i>	Object merupakan instance dari sebuah class dan dituliskan tersusun secara horizontal. Digambarkan sebagai sebuah class (kotak) dengan nama obyek didalamnya yang diawali dengan sebuah titik koma
	<i>Actor</i>	Actor juga dapat berkomunikasi dengan object, maka actor juga dapat diurutkan sebagai kolom. Simbol Actor sama dengan simbol pada Actor Use Case Diagram.
	<i>Message</i>	Message, digambarkan dengan anak panah yang mengarah antar obyek dan diberi label urutan nomor yang mengindikasikan urutan komunikasi yang terjadi antar obyek.

- e. *Statechart* Diagram, diagram ini memperlihatkan bagaimana sistem dapat merespon suatu kejadian internal atau eksternal, melalui event ini bertanggung jawab terhadap perubahan keadaan sistem. Simbol-simbol yang digunakan dalam *Statechart* Diagram:

Tabel 2. 9 Simbol-simbol Statechart Diagram (Widuri.Raharja)

No	Gambar	Nama	Keterangan
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

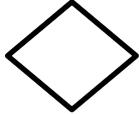
2.1.9 Business Process Model and Nation (BPMN)

BPMN merupakan pemodelan grafis untuk menentukan sebuah proses bisnis dalam suatu pemodelan proses bisnis yang akan terjadi dalam perusahaan berupa pesan yang disampaikan kepada pihak terkait. Tujuan diterapkannya BPMN untuk menyediakan sebuah notasi standar yang dapat dipahami dengan mudah oleh semua pihak yang berkepentingan dalam proses bisnis. BPMN juga digunakan sebagai tools atau alat untuk menjelaskan bagaimana proses bisnis dirancang serta untuk menggambarkan secara teknis bagaimana proses bisnis diimplementasikan untuk tujuan otomatisasi. (Nugroho et al., 2020)

BPMN memiliki beberapa dasar simbol-simbol, diantaranya:

Tabel 2. 10 Dasar Simbol BPMN (IlmuKomputer.com)

<i>Shape</i>	<i>Element/Object</i>
	<i>Event</i>

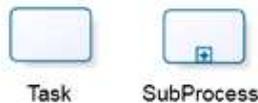
	<i>Task/Activity</i>
	<i>Gateway</i>
	<i>Sequence Flow</i>

1. *Event* adalah sesuatu yang terjadi selama Proses. Mempengaruhi aliran dari model dan biasanya memiliki pemicu atau hasil. Untuk membedakan fungsinya event digambarkan dalam sebuah lingkaran terbuka. Berdasarkan pengaruh aliran proses ada tiga jenis *event*: Awal, Menengah, dan Akhir.



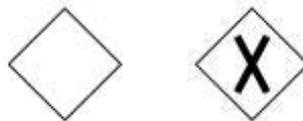
Gambar 2. 1 Simbol Event (IlmuKomputer.com)

2. Aktivitas merupakan istilah umum untuk suatu kegiatan yang memperlihatkan bahwa suatu organisasi sedang melakukan suatu Proses. Jenis aktivitas yang merupakan bagian dari Proses sebuah Model Digambarkan dengan gambar bulat persegi panjang.



Gambar 2. 2 Simbol Aktivitas (IlmuKomputer.com)

3. Gateway digunakan untuk memilih salah satu jalur dari cabang yang dibuat sesuai dengan kondisi yang diberikan. Gateway ini digambar dengan gambar bentuk belah ketupat yang memiliki tanda 'X' di dalamnya atau tidak ada tanda 'X' sama sekali.



Gambar 2. 3 Simbol Aktivitas (IlmuKomputer.com)

4. *Sequence Flow* merupakan alur urutan digunakan untuk menunjukkan urutan sebuah aktivitas yang dilakukan dalam suatu proses.



Gambar 2. 4 Simbol Sequence Flow (IlmuKomputer.com)

5. *Message Flow* digunakan untuk menunjukkan aliran Pesan antara dua aktor yang dipersiapkan untuk mengirim dan menerima mereka. Dalam BPMN, dua Pools terpisah dalam Diagram Kolaborasi akan mewakili dua peserta. Misalnya, partner entitas atau partner roles.



Gambar 2. 5 Simbol Message Flow (IlmuKomputer.com)

6. *Association* digunakan untuk menghubungkan Data dan Objek dengan elemen grafis BPMN. Teks penjelasan dan Objek lain dapat terkait dengan elemen grafis. Semua mata panah dalam Asosiasi menunjukkan arah aliran. Misalnya, informasi.



Gambar 2. 6 Simbol Association (IlmuKomputer.com)

7. Pool adalah representasi grafis dari aktor/peserta dalam kolaborasi. Hal ini juga bertindak sebagai "kumpulan" dan wadah grafis untuk membagi serangkaian aktivitas ke dalam kumpulan lainnya, biasanya dalam konteks situasi B2B. Pool A mungkin memiliki internal yang terperinci, dalam bentuk proses yang berjalan.



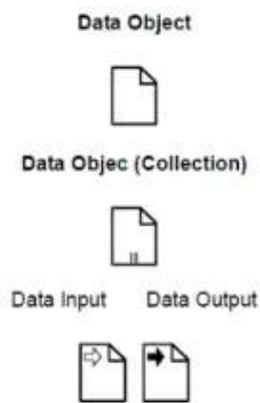
Gambar 2. 7 Simbol Pool (IlmuKomputer.com)

8. Lane adalah subdivisi dari suatu Proses, terkadang dalam Pool, akan memperpanjang seluruh Proses baik secara vertikal maupun horizontal. Jalur yang digunakan untuk mengatur dan mengkategorikan Kegiatan.



Gambar 2. 8 Simbol Lane (IlmuKomputer.com)

9. Data Object memberikan informasi tentang fungsi mana yang perlu diimplementasikan atau apa yang mereka lakukan. Data Object dapat mewakili satu objek atau kumpulan objek. Entri data dan notifikasi memberikan informasi yang sama ke proses.



Gambar 2. 9 Simbol Data Object (IlmuKomputercom)

10. *Group* adalah grup elemen grafis yang termasuk dalam kategori yang sama. Jenis grup tidak memengaruhi aliran pesan di dalam grup. Nama kategori muncul di bagan sebagai pengidentifikasi grup. Kelas dapat digunakan untuk tujuan dokumentasi atau analisis. Grup adalah cara untuk merepresentasikan kelas objek secara visual pada diagram



Gambar 2. 10 Simbol Group (IlmuKomputer.com)

2.10 Rapid Application Development (RAD)

RAD (*Rapid Application Development*) dikembangkan untuk para developer sebagai solusi teknik pengembangan yang cepat tanpa mengurangi kualitas sistem

yang dihasilkan. Metode pengembangan sistem *Rapid Application Development* (RAD) ini menekankan kecepatan dalam pengembangannya secara cepat, berulang dan inkremental melalui serangkaian prototype dari suatu sistem yang akan menjadi sistem akhir atau versi tertentu dari perkembangannya. *Rapid Application Development* (RAD) memiliki dua metodologi yaitu pengembangan bertahap dan prototype. Dalam metodologi tahap pengembangan seluruh sistem didekomposisi menjadi serangkaian tindakan sedangkan dalam metodologi prototyping semua tahap (analisis, desain dan implementasi) dilakukan secara bersamaan. (Habibi & Aprilian, 2019)



Gambar 2. 11 Metode Rapid Application Development Kendall

(Subianto, 2020, p. 47)

1. *Requirement Planning*

Pada tahap pertama, sistem yang dirancang akan fokus untuk memecahkan sebuah masalah yang dihadapi Kedai Minuman Iboba dengan menggunakan sistem yang berjalan saat ini. Dalam hal ini akan diberikan beberapa saran untuk dipertimbangkan oleh pemilik kedai dan pemilik kedai juga akan memberikan masukan mengenai sistem yang akan dikembangkan.

2. *Design Workshop*

Tahap kedua merupakan proses perancangan dan memperbaiki beberapa modul seperti:

Desain prototype dan beberapa fitur sistem dibangun dan diuji coba. Nantinya, aplikasi yang sudah selesai akan menjadi hasil dari proses iterasi akhir.

- a. Setelah setiap prototype dan sistem fungsional dikembangkan dan diuji, pengguna diberi kesempatan untuk mencobanya. Harapan dari pengguna

terhadap prototype akan menjadi kebutuhan sistem baru dan memberikan tanggapan atas gambaran proses bisnis.

- b. Setelah setiap prototype dan sistem fungsional dikembangkan dan diuji, *system analyst* dan *designers* akan meninjau untuk mendapatkan tanggapan teknis dan panduan pengembangan untuk proses selanjutnya.
- c. *Designer* akan mengidentifikasi tujuan atau arah perubahan desain berdasarkan tanggapan yang didapat.

3. *Implementation*

Pada tahap ketiga, programmer menerapkan desain yang telah disepakati pada tahap sebelumnya dari satu sistem ke sistem lainnya. Terdapat proses pengujian program untuk mendeteksi kesalahan sebelum menerapkan sistem.

2.1.10 Observasi

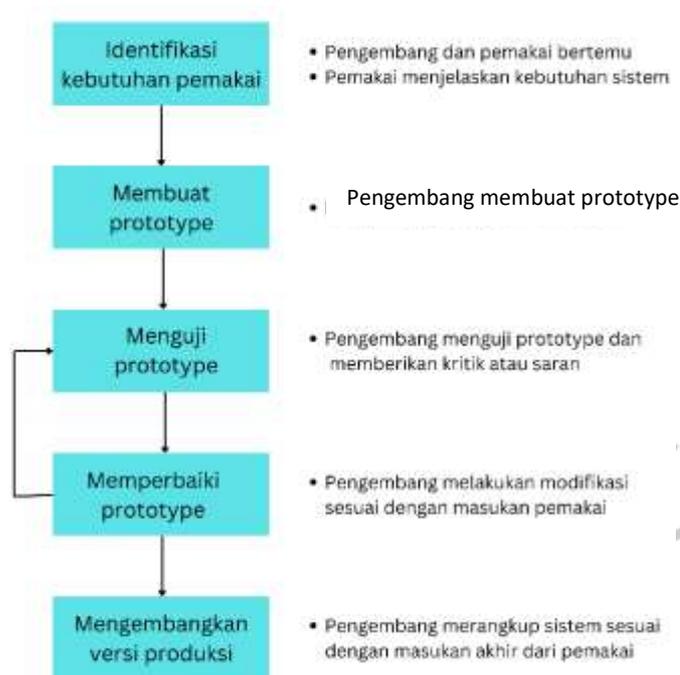
- Menurut Sugiyono (2018), observasi adalah sebuah teknik pengumpulan data yang mempunyai ciri yang detail atau perincian dengan teknik lainnya.

2.1.11 Wawancara

Menurut Kriyantono (2018), wawancara adalah kegiatan tanya jawab untuk mengumpulkan data-data berupa informasi dari seorang narasumber.

2.1.12 Prototype

Prototype dalam metode RAD digunakan untuk mendapatkan versi dari pemodelan aplikasi yang nantinya akan dibuat. Perancangan awal aplikasi dilakukan dalam bentuk mockup dan akan dievaluasi oleh pengguna. Setelah mockup dievaluasi oleh pengguna, pada tahap selanjutnya mockup tersebut menjadi bahan acuan bagi *software developer* untuk merancang aplikasi. (Agustia et al., 2021)



Gambar 2. 12 Mekanisme Perkembangan Sistem dengan Prototype (Elib.unikom.ac)

2.2 Penelitian Terdahulu

2.2.1 Sistem Informasi *Inventory* Data Barang pada UD. Mutiara Meubel Berbasis Web

Penelitian yang dilakukan oleh Guslan dan Rodianto pada tahun 2019 dengan judul "Sistem Informasi *Inventory* Data Barang pada UD. Mutiara Meubel Berbasis Web" membahas mengenai bidang penjualan meuble, UD. Mutiara Meubel adalah salah satu toko meubel terbesar di wilayah Kecamatan Plampang namun masih menggunakan sistem manual untuk pencatatan bahan baku atau pengolahan data. Dibuatnya sistem informasi *inventory* data barang dengan tujuan untuk memudahkan proses transaksi dalam hal menginput data barang agar tidak lagi dilakukan secara manual sehingga dengan sistem yang akan dibuat dapat lebih efektif dan efisien serta mengurangi kesalahan yang mungkin bisa terjadi. Sistem ini telah berhasil dibangun dengan menggunakan bahasa pemrograman PHP dan database MySQL. Dengan sistem yang telah dibangun tentunya akan lebih mudah melakukan pendataan barang masuk, barang keluar, dan mempermudah dalam mencari ketersediaan barang, serta laporan yang didapat lebih akurat dari sistem manual. (Guslan & Rodianto, 2019)

2.2.2 Rancang Bangun Sistem Inventori Pengendalian Stok Barang Berbasis Java pada PT Kalibesar Artah Perkasa

Penelitian yang dilakukan oleh Tuti Handayani, Ahmad Hapip Furqon dan Supriyono pada tahun 2020 dengan judul "Rancang Bangun Sistem Inventori Pengendalian Stok Barang Berbasis Java pada PT Kalibesar Artah Perkasa" membahas tentang karyawan PT Kalibesar Artah Perkasa merasa sulit ketika ingin mendapatkan laporan update persediaan barang karena masih memakai sistem manual. Admin masih harus mencari satu persatu untuk mengetahui data persediaan barang atau biasanya menggunakan *Control Find* untuk mencari barang yang akan dicari agar prosesnya lebih cepat. Dengan masalah yang dihadapi saat ini, agar perusahaan dapat menyimpan *file* persediaan barang dan mengolahnya menjadi sebuah informasi yang lebih baik bagi pihak manajemen perusahaan maka dibutuhkan sebuah rancangan sistem *inventory* yang bertujuan untuk menghasilkan rancangan fisik. Dengan merancang sebuah sistem *inventory* dapat memberi kemudahan untuk admin dalam membuat laporan sehingga proses dalam keputusan yang akan diambil semakin cepat dan tepat. Program aplikasi pengendalian ini dibuat menggunakan bahasa pemrograman *Java NeatBeans* dengan database *MySQL*.(Handayani et al., 2020)

2.2.3 Perancangan Sistem Informasi *Inventory* Barang Berbasis Web pada Universitas Budi Luhur

Pada penelitian yang dilakukan oleh Alfath Dioni dan Bullion Dragon Andah pada tahun 2019 dengan judul "Perancangan Sistem Informasi *Inventory* Barang Berbasis Web pada Universitas Budi Luhur" membahas tentang penelitian, dimana terdapat masalah yang sudah teridentifikasi dalam melakukan sebuah pengadaan setiap unit yang ada di Universitas Budi Luhur harus mengisi formulir yang berisi deskripsi barang yang dibutuhkan oleh unit tersebut. Karena formulir ini masih berbentuk kertas sering terjadi kasus formulir terselip dan tidak adanya notifikasi pengingat yang berakibat lamanya birokrasi pengadaan barang. Agar mempermudah bagian pengadaan untuk melakukan pengadaan barang sesuai dengan barang yang paling sering diminta dan untuk meminimalisir kesalahan pada saat melakukan pembelian barang maka

dibuat sistem *inventory* barang berbasis web yang dikembangkan oleh metode waterfall. Dengan sistem yang akan dibuat proses bisnisnya akan lebih detail dari sebelumnya. Lalu, Jika ada permintaan barang yang diminta oleh staff tidak akan terselip lagi karena telah muncul notifikasi sebagai pengingat. (Diano & Andah, 2021)

2.2.4 Penerapan Metode RAD Dalam Sistem Persediaan Barang Berbasis Web Pada PT. Agree Progress International di Jakarta Barat

Penelitian yang dilakukan oleh Anisa Puji Ikawati¹ dan Veri Arinal pada tahun 2021 dengan judul "Penerapan Metode RAD Dalam Sistem Persediaan Barang Berbasis Web Pada PT. Agree Progress International di Jakarta Barat" membahas tentang PT. Agree Progress International merupakan perusahaan yang bergerak di bidang distributor garmen yang menjual *underwear*. Berdasarkan informasi yang didapat sistem ini masih menggunakan sistem manual dimana admin gudang mencatat data persediaan stok masuk dan keluar menggunakan buku stok barang sehingga dengan sistem manual ini pihak perusahaan menjadi kesulitan dalam mengetahui sisa persediaan barang digudang, terlebih banyaknya permintaan produk yang beragam.

Sehingga muncul lah masalah yang terjadi berupa kesalahan dalam perhitungan barang, kesulitan dalam pencatatan barang, pembuatan laporan barang masuk dan barang keluar dari banyaknya surat jalan. Tentu saja ini menimbulkan masalah baru karena sulitnya mencari data barang yang diperlukan akibat banyaknya penumpukan berkas. Dengan perancangan sistem *inventory* menjadi dasar yang paling penting untuk memenuhi kebutuhan konsumen dalam waktu yang tepat. Data yang ada juga akan terkomputerisasi sehingga dapat mengatasi kesulitan dalam pembukuan, pengelolaan data, dan meminimalisir kecurangan serta menghasilkan laporan yang akurat. Metode pengembangan yang diterapkan dalam penelitian ini adalah metode RAD dimana metode ini tidak memerlukan waktu yang lama dalam pengerjaannya. (Puji Ikawati & Arinal, 2021)

2.2.5 Sistem Informasi *Inventory* Menggunakan Framework Laravel Pada CV. Grace Bhakti Utama Bogor

Penelitian ini dilakukan oleh Marshel Christian, Suparni, dan Lilyani Asri Utami pada tahun 2021 dengan judul "Sistem Informasi *Inventory* Menggunakan Framework Laravel Pada CV. Grace Bhakti Utama Bogor" membahas tentang waktu yang terbuang untuk menyelesaikan laporan stok barang karena CV. Grace Bhakti Utama masih menggunakan sistem manual. Dengan sistem manual segala proses mencatat barang masuk dan barang keluar memerlukan waktu yang cukup lama untuk menyelesaikannya, tidak jarang terjadi kesalahan dalam penginputan yang mengakibatkan laporan tidak akurat. Tentu saja ini menjadi permasalahan yang harus diselesaikan agar CV. Grace Bhakti Utama dapat berjalan dengan efektif dan efisien. Pada penelitian ini diusulkan membuat perancangan *inventory* dengan menggunakan *Framework* Laravel agar dapat memberi kemudahan kepada user dalam mengelola data *inventory* yang ada di CV. Grace Bhakti Utama. Dalam penelitian ini menerapkan metode waterfall karena metode ini seperti air terjun atau terurut mulai dari analisis, desain, pengkodean, pengujian, serta tahap pendukung. Dengan adanya sistem *inventory* yang akan dibuat diharapkan dapat mempercepat proses pencatatan laporan, transaksi penjualan maupun pembelian. Selain itu, pengguna dapat dengan mudah melihat aktivitas perusahaan dalam sistem informasi, sehingga lebih mudah untuk memantau penjualan. (Christian et al., 2021)