

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Teori Dasar

##### 2.1.1 Konsep Dasar Sistem

###### 1. Pengertian Sistem

Sistem adalah suatu elemen-elemen yang saling terkait dan bekerja secara bersama-sama berdasarkan masukan (*input*) yang diberikan ke sistem tersebut, selanjutnya masukan tersebut diolah sehingga menghasilkan keluaran (*output*) yang dikehendaki (Andri Kristanto, 2018).

###### 2. Karakteristik Sistem

Suatu hal dapat dikatakan sebagai suatu sistem jika memiliki karakteristik sebagai berikut (Hutahean, 2015):

###### a. Komponen

Sebuah sistem dapat terdiri dari interaksi antar komponen-komponen yang saling bekerja sama dan membentuk satu-kesatuan. Komponen sistem juga terdiri dari subsistem atau elemen-elemen dari sistem utama yang memiliki fungsionalitas yang berbeda di masing-masing subsistemnya.

###### b. Batasan (*Boundary*)

Batasan dalam sistem dapat diilustrasikan seperti suatu garis yang membatasi antara satu sistem dengan sistem lainnya atau dengan lingkungan eksternalnya. Batasan ini memperlihatkan sistem sebagai suatu kesatuan yang utuh. Batasan sistem juga mengindikasikan ruang lingkup dari sistem tersebut..

###### c. Lingkungan (*Environment*)

Lingkungan sendiri merujuk pada elemen yang berada di

luar batas sistem, tetapi memiliki potensi untuk memengaruhi operasi sistem tersebut. Pengaruh tersebut dapat berdampak positif atau negatif terhadap sistem.

d. Penghubung (*Interface*)

Setiap subsistem saling terkoneksi dengan subsistem lainnya melalui suatu media penghubung. Dengan adanya penghubung ini, memungkinkan subsistem untuk meneruskan sumber daya ke subsistem lainnya.

e. Masukan (*Input*)

Masukan dalam sistem adalah sumber daya yang akan diolah sistem. Masukan ini dapat berupa input perawatan (program) maupun input sinyal (data).

f. Keluaran (*Output*)

Keluaran adalah hasil dari masukan yang diproses dan diolah oleh sistem.. Keluaran dapat menjadi masukan bagi sistem atau subsistem lainnya.

g. Pengolahan (*Process*)

Pengolahan yang dimaksud di sini adalah sebuah proses yang dilakukan terhadap masukan (*input*) untuk diubah menjadi keluaran (*output*).

h. Sasaran (*Objective*)

Sasaran di dalam sistem merupakan sebuah tujuan atau sasaran yang ingin dicapai dari sebuah sistem. Berdasarkan tujuan dari sistem yang ada, maka dapat ditentukan apa yang akan menjadi masukan bagi sistem dan dapat memberikan keluaran (*output*) yang selaras dengan tujuan sistem.

3. Klasifikasi Sistem

Klasifikasi sistem terdiri dari beberapa perspektif pengguna, (Hutahean, 2015), di antaranya:

a. Sistem Abstrak dan Sistem Fisik

Sistem abstrak adalah sistem yang tidak terlihat secara langsung dan masih berada dalam bentuk pemikiran. Di

sisi lain, sistem fisik adalah sistem yang dapat dilihat secara nyata dan serta dapat dimanfaatkan fungsionalitasnya.

b. Sistem Alamiah dan Sistem Buatan Manusia

Sistem alamiah merujuk pada sistem yang terbentuk secara spontan tanpa adanya intervensi manusia dalam proses pembentukannya. Beberapa contohnya antara lain seperti sistem pernapasan dan juga sistem ekologi. Sedangkan sistem buatan manusia tentunya merupakan hasil rancangan dan buatan manusia dan memerlukan interaksi manusia itu sendiri di dalamnya. Contoh dari sistem buatan manusia antara lain seperti sistem informasi penjualan dan sistem manajemen pembelajaran.

c. Sistem Deterministik dan Sistem Probabilistik

Sistem deterministik adalah sistem yang dapat diukur dan diprediksi secara operasi dan tingkah lakunya. Contoh dari sistem deterministik ialah sistem komputer. Sistem probabilitas adalah sistem yang mengandung elemen probabilitas atau ketidakpastian, sehingga kondisinya di masa depan tidak dapat diprediksi dengan pasti. Contoh dari sistem probabilistik adalah sistem ramalan cuaca.

d. Sistem Terbuka dan Sistem Tertutup

Sistem terbuka merujuk pada sistem yang memiliki keterkaitan dan dapat diintervensi oleh lingkungan eksternalnya. Sistem terbuka dapat memanfaatkan masukan dan keluaran yang berasal dari lingkungan luar atau subsistem lain. Karena adanya pengaruh dari lingkungan luar tersebut, maka diperlukan kontrol atau kendali yang baik terhadap apa yang menjadi masukan dari sistem tersebut. Sedangkan sistem tertutup adalah

kebalikan dari sistem terbuka. Sistem ini tidak terhubung dan tidak dipengaruhi oleh lingkungan luarnya. Sistem tertutup ini dapat berjalan dengan sendirinya. Namun realitanya, tidak ada sistem yang sepenuhnya terisolasi dari lingkungan eksternalnya.

#### 4. Pengertian Data

Gordon B. Davis memaparkan bahwa data merupakan bahan mentah untuk informasi, dirumuskan sebagai kelompok lambang-lambang, tindakan dan hal-hal sebagainya (Hutahean, 2015). Data merujuk pada informasi yang memberikan deskripsi tentang objek, peristiwa, aktivitas, dan transaksi yang memiliki makna atau tidak secara langsung mempengaruhi pengguna (Anggraeni & Irviani, 2017).

#### 5. Pengertian Informasi

Menurut (Andri Kristanto, 2018), Informasi adalah rangkaian data yang telah diproses menjadi bentuk yang lebih bermanfaat dan memiliki makna yang lebih signifikan bagi penerimanya. Data yang telah diolah menjadi sesuatu yang memiliki nilai dan manfaat bagi penerimanya bertujuan untuk memberikan informasi atau pengetahuan. Dalam hal ini, data menjadi sumber informasi yang dapat memberikan pemahaman atau wawasan. Informasi juga dapat dianggap sebagai pengetahuan yang diperoleh melalui proses belajar, pengalaman, atau instruksi (Anggraeni & Irviani, 2017).

Menurut Parker, ada beberapa persyaratan yang harus dipenuhi untuk menentukan apakah suatu informasi berkualitas atau tidak antara lain (Tyoso, 2016).

##### a. Ketersediaan

Informasi yang berkualitas harus bisa diakses kapanpun dan di manapun oleh orang yang membutuhkannya. Maka dari itu, informasi harus selalu tersedia setiap saat di dalam basis data (*database*) yang dikelola dengan baik dan terorganisir.

b. Mudah dipahami

Informasi yang sulit untuk dipahami bahkan bersifat rumit, maka berakibat pada tertundanya pengambilan keputusan dari informasi tersebut yang disebabkan pembahasan akan informasi tersebut yang memakan waktu.

c. Relevan

Relevan yang dimaksud di sini adalah berkaitan dengan pengoperasian suatu organisasi. Informasi yang dibutuhkan adalah informasi yang relevan dengan permasalahan, misi serta tujuan organisasi.

d. Bermanfaat

Tentunya informasi yang berkualitas adalah informasi yang memiliki manfaat bagi khalayak banyak. Manfaat dari sebuah informasi dapat diperoleh jika disajikan dalam bentuk-bentuk yang mudah dilihat dan juga mudah dipahami.

e. Tepat waktu

Informasi itu sendiri harus tersedia tepat pada waktunya ketika seseorang membutuhkannya. Juga perlu diperhatikan kapan suatu informasi didapatkan dan momen atau peristiwa yang terjadi pada saat itu.

f. Keterandalan

Informasi yang andal adalah informasi yang dapat dipertanggungjawabkan kebenarannya. Demikian pula dengan narasumber yang memberikan informasi haruslah sebagai seseorang yang dapat dipercaya.

g. Akurat

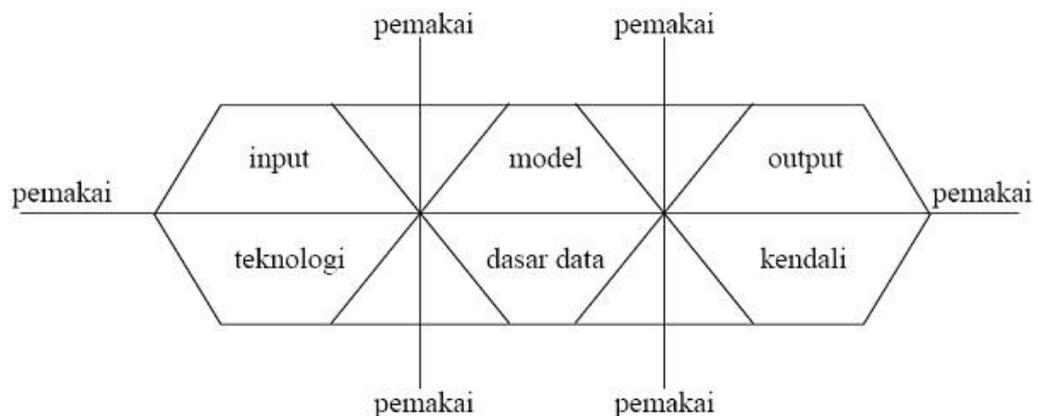
Akurat atau tidaknya sebuah informasi bisa dilihat dengan cara mengecek ada atau tidaknya kesalahan atau kekeliruan di dalam informasi tersebut. Maksudnya adalah, informasi yang diberikan haruslah jelas dan mencerminkan makna yang terkandung dari data.

h. Konsisten

Informasi yang baik dan berkualitas haruslah konsisten. Artinya adalah informasi yang ada tidak bertentangan dengan informasi lain yang ada atau dengan pengetahuan yang telah diterima sebelumnya. Konsistensi mengacu pada keberlanjutan, kesesuaian, dan keselarasan informasi dengan informasi lain yang ada atau dengan kerangka pengetahuan yang sudah ada.

**2.1.2 Definisi Sistem Informasi**

Sistem informasi merupakan suatu proses pengumpulan, penyimpanan dan analisis informasi untuk tujuan tertentu. Sistem informasi sendiri terdiri dari masukan (*input*) dan akan menghasilkan sebuah keluaran (*output*). Keluaran dari sistem tadi nantinya dapat digunakan atau dijadikan masukan oleh sistem lainnya serta kegiatan strategi di dalam organisasi yang bersifat strategis (Prehanto, 2020).



Gambar 2.1 Building Block Sistem Informasi (Prehanto, 2020, p. 21)

Mengacu pada gambar 2.1, sistem informasi terdiri dari elemen-elemen yang disebut sebagai *building block*, yang meliputi:

a. Blok masukan (*input block*)

Blok ini adalah *input* yang berupa data yang akan diolah di dalam sistem informasi. *Input* sendiri juga termasuk metode dan media yang dipakai untuk mengambil data.

b. Blok model (*model block*)

Blok ini terdiri dari gabungan langkah, logika, dan pola perhitungan matematis yang digunakan untuk memproses data dari blok input. Selain data dari blok masukan, blok model juga akan melakukan pemrosesan untuk data yang tersimpan di dalam basis data. Kemudian dari pemrosesan tersebut akan menghasilkan keluaran yang diinginkan.

c. Blok keluaran (*output block*)

Keluaran yang berkualitas merupakan sebuah produk dari sistem informasi. Tersedianya dokumentasi yang gamblang dan mudah dipahami juga berguna bagi segala tingkat manajemen.

d. Blok teknologi (*technology block*)

Teknologi memiliki peran penting dalam menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan serta mengirimkan keluaran, serta sebagai kontrol sistem secara keseluruhan. Teknologi terdiri dari tiga komponen utama, yaitu pengguna yang menjalankan (*brainware*), perangkat lunak (*software*), dan perangkat keras (*hardware*).

e. Blok basis data (*database block*)

Basis data sendiri memiliki tugas sebagai lokasi penyimpanan data.. Pengelolaan dan pengorganisasian data di dalam basis data perlu diperhatikan agar dapat menghasilkan informasi yang berkualitas dan juga relevan. Selain itu, basis data di dalam sistem informasi juga harus memiliki ketersediaan yang baik guna menunjang kebutuhan pengguna akan akses informasi tanpa adanya kendala berarti.

f. Blok kendali (*control block*)

Blok kendali di dalam sistem informasi berfungsi sebagai bagian yang mengendalikan dan mencegah segala hal yang akan merusak sistem. Di samping itu, diperlukan perancangan dan penerapan beberapa kontrol untuk memastikan bahwa masalah yang dapat merusak sistem dapat segera ditangani, bahkan jika terjadi kesalahan

di dalam sistem.

### 2.1.3 Pengertian Aplikasi

Perangkat lunak aplikasi merupakan sebuah program yang dibuat dengan memanfaatkan kemampuan komputer untuk dapat menyelesaikan tugas yang diberikan penggunanya (Sari, 2021). Jadi, aplikasi adalah sebuah program yang dibuat dalam perangkat lunak komputer untuk mempermudah pelaksanaan pekerjaan atau tugas tertentu, seperti menerapkan, menggunakan, dan menambahkan data yang diperlukan (Setyawan & Munari, 2020).

### 2.1.4 Konsep Aplikasi Berbasis Web

Menurut (Setyawan & Munari, 2020), aplikasi berbasis web ini menggunakan protokol HTTP, aplikasi di sisi server berkomunikasi dengan *client* melalui *web server*. Aplikasi di sisi *client* umumnya berupa *web browser*. Berikut ini karakteristik dari aplikasi berbasis web yaitu:

1. Keserempakan  
Serempak yang dimaksud di sini ialah merujuk pada waktu ketika mengakses aplikasi. Aplikasi berbasis web umumnya akan diakses di waktu yang bersamaan oleh pengguna.
2. Jumlah pengguna yang tidak dapat diprediksi  
Pengguna aplikasi berbasis web bisa saja meningkat secara signifikan di waktu tertentu. Akses terhadap aplikasi yang bisa dilakukan melalui peramban, membuat akses tidak terbatas pada perangkat tertentu seperti komputer. Akses ke aplikasi berbasis web bisa saja menggunakan perangkat *mobile* tanpa perlu adanya proses instalasi terlebih dahulu.
3. Kinerja  
Kinerja dari aplikasi berbasis web berbeda dengan aplikasi *desktop*. Hal ini lantaran proses eksekusi aplikasi bukan pada perangkat klien, tetapi harus melalui permintaan ke server yang mungkin membutuhkan waktu. Selain itu, karena aplikasi berbasis web berjalan di atas peramban, terdapat batasan

penggunaan sumber daya perangkat keras seperti RAM, CPU dan GPU.

4. Ketersediaan

Aplikasi berbasis web sangat bergantung dengan koneksi internet untuk dapat diakses. Hal ini dikarenakan setiap permintaan yang dilakukan oleh klien akan diteruskan melalui koneksi internet ke sebuah server tempat aplikasi berbasis web berada. Jika koneksi internet terputus, secara otomatis akan mengganggu aksesibilitas dan fungsionalitas dari aplikasi.

5. Keamanan

Keamanan pada aplikasi berbasis web tidak sebaik aplikasi *desktop*. Hal tersebut juga disebabkan keberadaan aplikasi berbasis web yang berada di server. Data yang ditransmisikan melalui jaringan internet selama aplikasi berbasis web dijalankan sangat rentan disusupi oleh peretas. Terlebih jika aplikasi tersebut banyak menggunakan API eksternal untuk mengakomodir kebutuhan pengguna. Diperlukan keahlian yang baik bagi pengembang aplikasi berbasis web dalam mengamankan data yang ditransmisikan antara klien dan server.

### 2.1.5 Definisi Sistem Basis Data

Basis data terdiri dari dua kata, yaitu basis dan data, yang dapat dimaknai sebagai lokasi penyimpanan atau pangkalan. Sementara itu, data adalah kumpulan informasi yang berisi fakta-fakta untuk merepresentasikan tentang suatu kondisi yang komprehensif. Dengan menggunakan data, seseorang dapat melakukan analisis, merepresentasikan, atau menjelaskan suatu kondisi. (Fergina et al., 2021).

Berikut ini beberapa komponen dalam basis data yaitu:

1. Perangkat Keras

Perangkat keras yang dimaksud di sini merupakan berbagai peralatan komputer yang harus ada dalam pemrosesan dan pengelolaan data. Perangkat keras ini dapat berupa komputer server, jaringan komputer untuk sarana komunikasi data dan

beberapa peralatan pendukung lainnya (*peripheral*).

## 2. Perangkat Lunak

Perangkat lunak juga tidak kalah penting untuk pengelolaan basis data. Adapun perangkat lunak yang dibutuhkan antara lain seperti sistem operasi, *Database Management System* (DBMS), bahasa pemrograman dan juga beberapa program penunjang seperti antivirus dan peramban.

## 3. Pengguna

Berdasarkan interaksinya dengan basis data, pengguna dapat dikelompokkan menjadi:

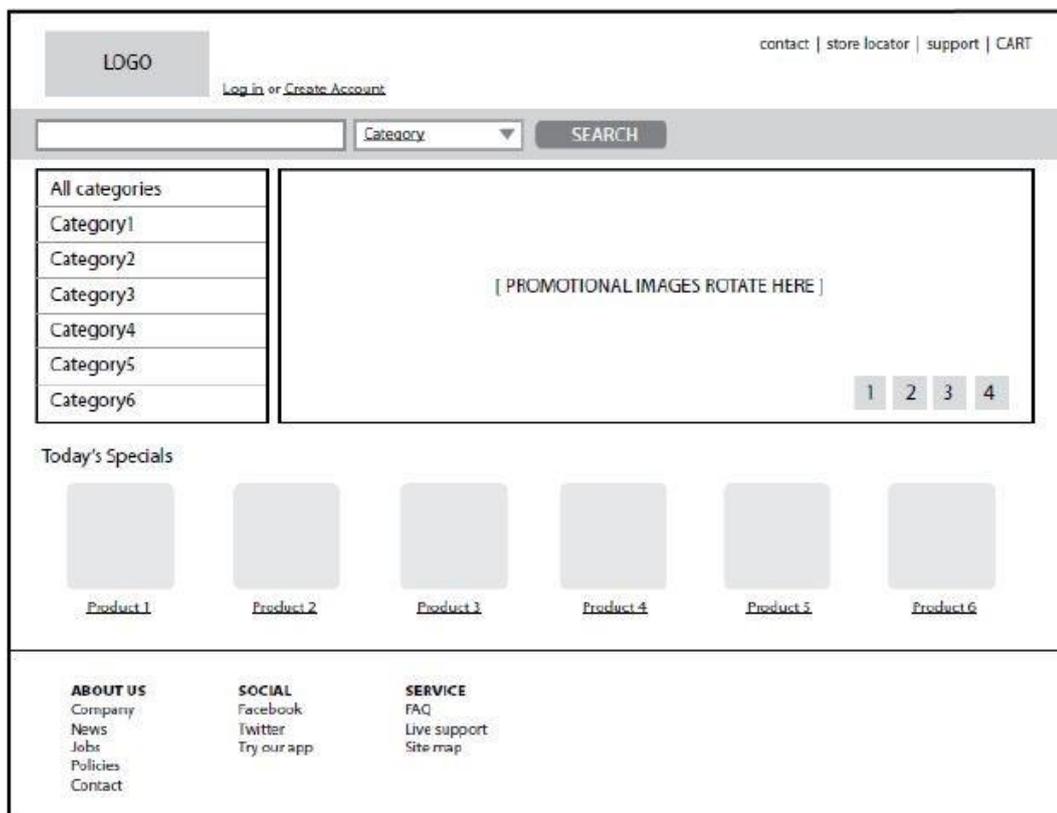
- a. Pemrogram aplikasi
- b. Pengguna mahir
- c. Pengguna umum
- d. Pengguna khusus

## 4. Data

Data sendiri merupakan unsur terkecil dari sistem basis data serta merupakan materi yang diproses atau dikelola.

### 2.1.6 Definisi Wireframe

*Wireframe* adalah sebuah kerangka dasar dari halaman sebuah web yang akan dibuat. *Wireframe* juga dapat dikatakan sebagai cetak biru dalam dunia arsitektur. Di dalam *wireframe* juga digunakan gambaran tata letak elemen-elemen penting dari halaman web. Tampilan dari *wireframe* hanya terdiri dari kotak dan garis yang digunakan sebagai elemen desain (Budiawan, 2021). Contoh *wireframe* dapat dilihat pada gambar x.x yang memperlihatkan rancangan dari antarmuka platform *YouTube*.



Gambar 2.2 Contoh Wireframe (Alexandra, 2017)

### 2.1.7 Definisi Sistem Repositori

Repositori sendiri memiliki tujuan memberikan ruang penyimpanan dokumen elektronik suatu lembaga sehingga mempermudah dalam proses pencarian kembali (Tello-Leal et al., 2015). Tujuan dari pemanfaatan tersebut adalah untuk menghemat waktu pengembangan dengan ketersediaan aset yang dibutuhkan. Kemudian repositori juga merupakan sebuah wadah yang juga dapat difungsikan untuk menyimpan produk intelektual yang tercipta dari aktivitas di dalam institusi (Kurniawan, 2016).

Berdasarkan paparan teori di atas, maka dapat disimpulkan bahwa sistem repositori merupakan sebuah sistem yang dibuat untuk menyimpan mengelola aset dan juga kekayaan intelektual sebuah organisasi untuk menjamin ketersediaannya serta dapat digunakan untuk berbagi pengetahuan di dalam organisasi tersebut.

### 2.1.8 Metode Pengembangan Sistem

1. *Rapid Application Development (RAD)*

RAD merupakan sebuah pengembangan sistem dengan pendekatan yang berorientasikan objek (*object oriented*). Metode RAD sebuah metodologi pengembangan sistem yang lebih baru yang muncul pada tahun 1990-an. Metodologi berbasis RAD berusaha mengatasi kelemahan metodologi desain terstruktur dengan mempercepat pengembangan sebagian sistem agar dapat segera digunakan oleh pengguna. Dengan demikian, pengguna dapat lebih memahami sistem dan memberikan saran perbaikan yang membuat sistem lebih sesuai dengan kebutuhan (Dennis et al., 2015).

Berikut ini adalah fase-fase yang terdapat dalam RAD:

1. Fase Perencanaan Kebutuhan

Fase ini merupakan fase pertama di dalam metode RAD. Pada fase ini dilakukan diskusi antara analis sistem bersama dengan pengguna untuk mengidentifikasi hasil akhir dari sistem yang akan dibuat. Selain tujuan, hal lain yang akan diidentifikasi pada fase ini adalah kebutuhan informasi terhadap sistem yang akan dikembangkan. Fase ini juga membutuhkan keterlibatan dua pihak yang intens antara tim pengembang dan juga pengguna yang akan menggunakan sistem. Orientasi dalam fase ini adalah pemecahan masalah yang timbul dari proses bisnis yang berjalan.

2. Fase Analisis

Fase selanjutnya adalah fase analisis. Selama tahap analisis, dilakukan identifikasi dan mengumpulkan kebutuhan pengguna sistem baru. Nantinya hasil dari analisis dalam fase ini akan di dokumentasikan dan digunakan rujukan untuk merancang sistem baru.

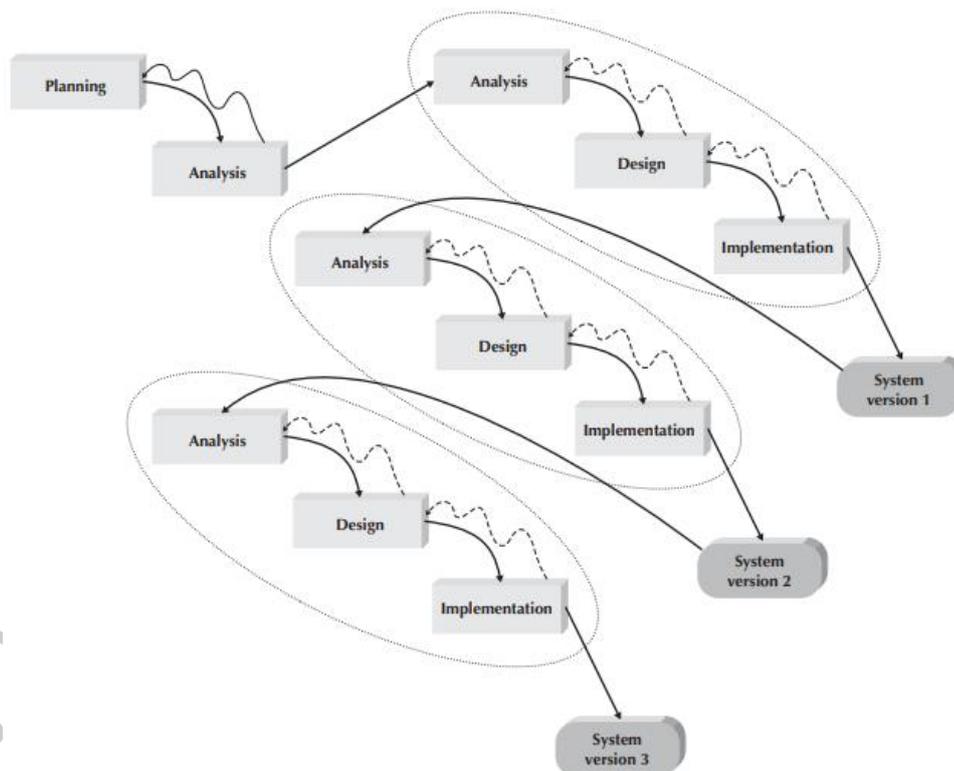
3. Fase Desain

Fase ini adalah fase ketiga di mana tim pengembang, analis bisnis dan para pemangku kepentingan berkumpul dalam sesi kolaboratif untuk merancang rancangan sistem yang

akan dikembangkan. Pada fase ini tim pengembang berusaha memahami secara mendalam terhadap proses bisnis yang berjalan. Selain proses pemahaman terkait proses bisnis yang ada, tim pengembang juga mulai merancang dan memodelkan sistem dengan berbagai teknik dan alat. Hasil dari desain dan rancangan sistem akan divalidasi oleh pemangku kepentingan yang terlibat dan kemudian menyusun dokumen desain yang mencakup rancangan sistem secara rinci. Dokumen desain tersebut akan digunakan sebagai panduan bagi tim pengembang selama fase pengembangan berikutnya.

#### 4. Fase Implementasi

Setelah proses mendesain dan merancang sistem di fase desain selesai, maka fase yang terakhir adalah fase implementasi. Pada fase ini, proses pengembangan sistem dimulai dengan melakukan pengkodean berdasarkan dokumen desain yang sudah dibuat pada fase sebelumnya. Tidak hanya proses pengkodean, pada fase ini juga akan dilakukan pengujian terhadap sistem. Proses pengujian ini tentunya untuk memastikan kinerja dari sistem sudah sesuai dan tidak ditemukan lagi *bug* ataupun *error*.



Gambar 2.3 Rapid Application Development Cycle (Dennis et al., 2015)

### 2.1.9 *Unified Modeling Language (UML)*

*Unified Modeling Language* atau lebih sering dikenal dengan sebutan UML, adalah salah satu metode dalam teknik rekayasa perangkat lunak yang digunakan untuk menggambarkan alur dan cara kerja sistem, fungsi, tujuan dan mekanisme kontrol sistem tersebut (Abdillah et al., 2019). UML juga berfungsi sebagai notasi untuk menggambarkan berbagai aktivitas, termasuk pemodelan kasus bisnis, analisis sistem saat ini, kebutuhan yang dibutuhkan, serta perencanaan desain arsitektur dan rincian pada berbagai tingkat (Rumpe, 2017).

Dalam rangka mengembangkan perangkat lunak, *Unified Modeling Language (UML)* merupakan metode dan notasi yang dapat digunakan untuk menggambarkan alur, cara kerja, fungsi, tujuan, mekanisme kontrol, arsitektur, dan desain sistem pada berbagai tingkat granularitas. UML digunakan untuk membantu pengembang sistem dalam memvisualisasikan dan merancang sistem dengan lebih terstruktur dan terdokumentasi dengan baik, sehingga memudahkan dalam pengembangan dan pemeliharaan sistem.

*Unified Modeling Language (UML)* adalah suatu acuan yang digunakan dalam pengembangan sistem berbasis objek, mulai dari tahap analisis hingga implementasi. Sejak diperkenalkan pada tahun 1997, UML telah mengalami beberapa perubahan hingga versi terakhirnya yaitu UML 2.0. Versi ini menyediakan 14 jenis diagram yang dibagi menjadi 2 kelompok, yaitu diagram yang menggambarkan struktur sistem dan diagram yang menggambarkan perilaku sistem. Berikut pembagian diagram-diagram tersebut dalam Tabel 2.1

Tabel 2.1 *Macam-macam Diagram UML*

Structured Diagrams	Behavioral Diagram
<i>Class</i>	<i>Activity</i>
<i>Object</i>	<i>Sequence</i>
<i>Package</i>	<i>Communication</i>
<i>Deployment</i>	<i>Interaction Overview</i>
<i>Component</i>	<i>Timing</i>
<i>Composite Structure</i>	<i>Behavioral State Machine</i>
	<i>Protocol State Machine</i>
	<i>Use Case</i>

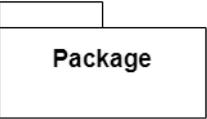
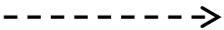
Berikut ini penjelasan dari masing-masing diagram:

1. *Package Diagram*

*Package* merupakan representasi *higher-level construct* yang mewakili *collaboration, partition* dan *layers*. Dalam penggunaannya, sebuah *package* ditujukan untuk mengelompokkan diagram-diagram agar menjadi lebih mudah untuk dibaca dan dipahami serta menjaga model perancangan dalam taraf kompleksitas yang wajar (Dennis et al., 2015).

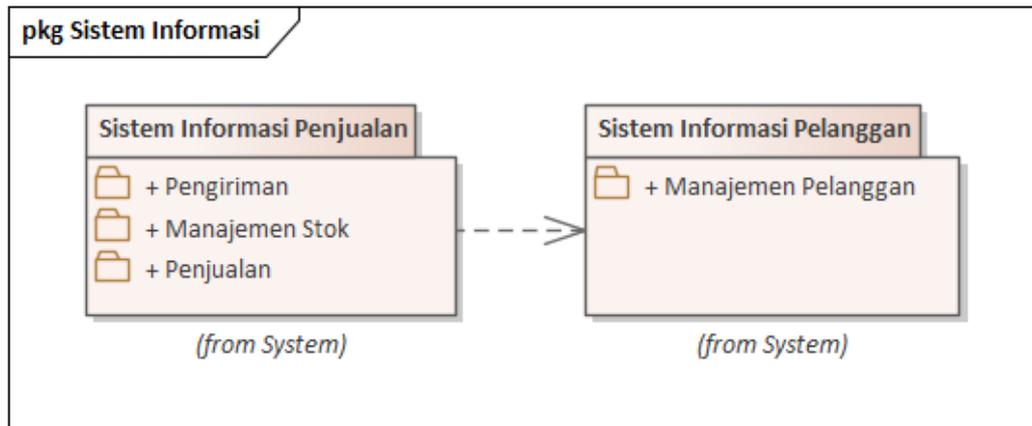
Berikut ini notasi yang digunakan dalam *package diagram*:

Tabel 2.2 *Notasi Package Diagram*

NOTASI	NAMA NOTASI	DESKRIPSI
	<i>Package</i>	Digunakan untuk menyederhanakan diagram UML dengan mengelompokkan elemen terkait menjadi satu elemen tingkat yang lebih tinggi
	<i>Dependency</i>	Merepresentasikan ketergantungan antar <i>package</i> . Jika sebuah <i>package</i> diubah, <i>package</i> yang bergantung juga harus dimodifikasi. Memiliki panah yang ditarik

		dari <i>package</i> yang bergantung menuju <i>package</i> yang digantungkan.
--	--	--

Berikut ini adalah contoh dari *package* diagram yang dapat dilihat pada gambar 2.4:



Gambar 2.4 Contoh Package Diagram

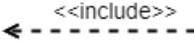
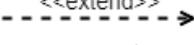
## 2. Use Case Diagram

Konsep *use-case driven* berarti bahwa *use case* menjadi alat pemodelan utama yang mendefinisikan perilaku sistem (Dennis et al., 2015). *Use case* digunakan untuk mengidentifikasi dan mengkomunikasikan kebutuhan sistem kepada para pemrogram yang akan menulis sistem tersebut. Dengan demikian, *use-case driven* menjadi landasan penting dalam pengembangan sistem untuk memastikan bahwa sistem yang dibangun sesuai dengan kebutuhan dan harapan pengguna.

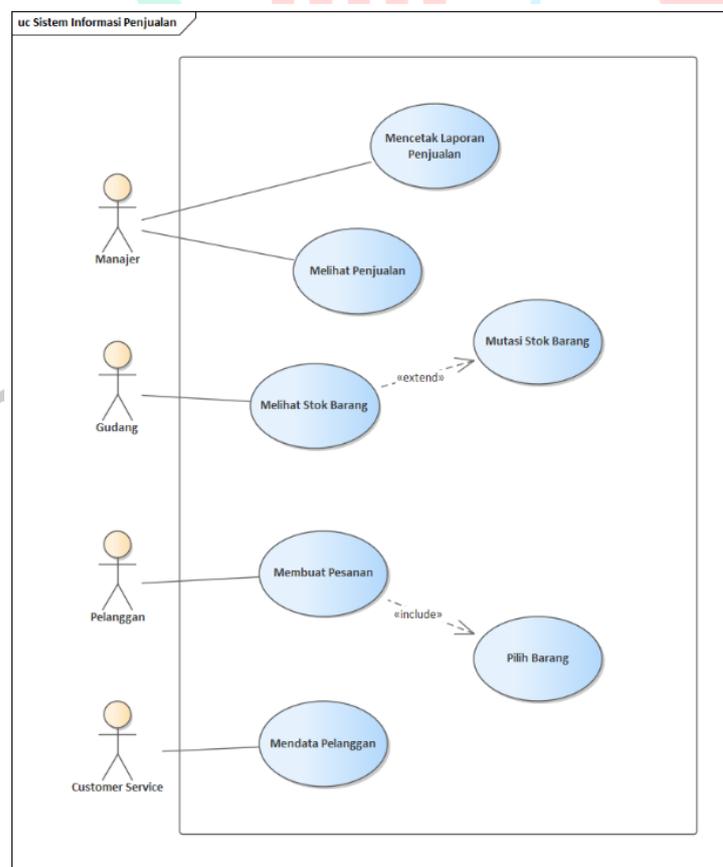
Berikut ini lambang yang dipakai dalam *use case* diagram:

Tabel 2.3 Notasi Use Case Diagram

NOTASI	NAMA NOTASI	DESKRIPSI
	<i>Actor</i>	<i>Actor</i> merupakan sebuah pengguna yang nantinya akan berinteraksi di dalam sistem dengan skenario tertentu. Pengguna ini tidak hanya orang, tetapi juga bisa sebuah sistem ataupun entitas lainnya yang terkait dengan sistem. Posisi <i>actor</i> berada di luar batas sistem.
	<i>Use case</i>	<i>Use case</i> merupakan representasi visual dari suatu tindakan atau fungsi yang akan dilakukan di dalam sistem. <i>Use case</i> juga menggambarkan interaksi antara aktor dan sistem dalam suatu skenario tertentu serta ditempatkan di dalam batas sistem

	<i>Association</i>	Merepresentasikan keterkaitan antara <i>actor</i> dan <i>use case</i> yang menunjukkan hubungan antara keduanya dalam suatu skenario tertentu.
	<i>Include</i>	Merupakan representasi visual dari hubungan antara dua <i>use case</i> yang menunjukkan bahwa satu <i>use case</i> membutuhkan perilaku dari <i>use case</i> yang lain dalam suatu skenario tertentu. <i>Include</i> menggambarkan hubungan yang bersifat wajib dan selalu terjadi dalam suatu skenario.
	<i>Extend</i>	Merupakan representasi visual dari hubungan antara dua <i>use case</i> yang menunjukkan bahwa satu <i>use case</i> dapat menambahkan perilaku ke <i>use case</i> yang lain dalam suatu skenario tertentu. <i>Extend</i> menggambarkan hubungan yang bersifat opsional dan bisa terjadi atau tidak terjadi dalam suatu skenario.
	<i>Generalization</i>	Merupakan representasi visual dari hubungan antara <i>use case</i> yang menunjukkan bahwa suatu <i>use case</i> merupakan varian dari <i>use case</i> yang lain. Hubungan ini menggambarkan bahwa <i>use case</i> yang lebih spesifik mewarisi perilaku dari <i>use case</i> yang lebih umum.

Berikut ini adalah contoh dari *use case* diagram yang dapat dilihat pada gambar 2.5:



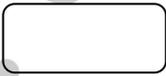
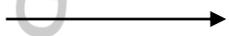
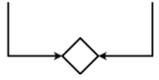
Gambar 2.5 Contoh Use Case Diagram

### 3. Activity Diagram

*Activity* diagram digunakan untuk menggambarkan perilaku dalam proses bisnis yang tidak tergantung pada objek. *Activity* diagram dapat digunakan untuk menggambarkan berbagai hal, mulai dari aliran bisnis tingkat tinggi yang melibatkan banyak kasus penggunaan yang berbeda, hingga rincian kasus penggunaan individu, termasuk detail spesifik dari metode individu. Secara sederhana, diagram aktivitas dapat digunakan untuk memodelkan berbagai jenis proses (Dennis et al., 2015).

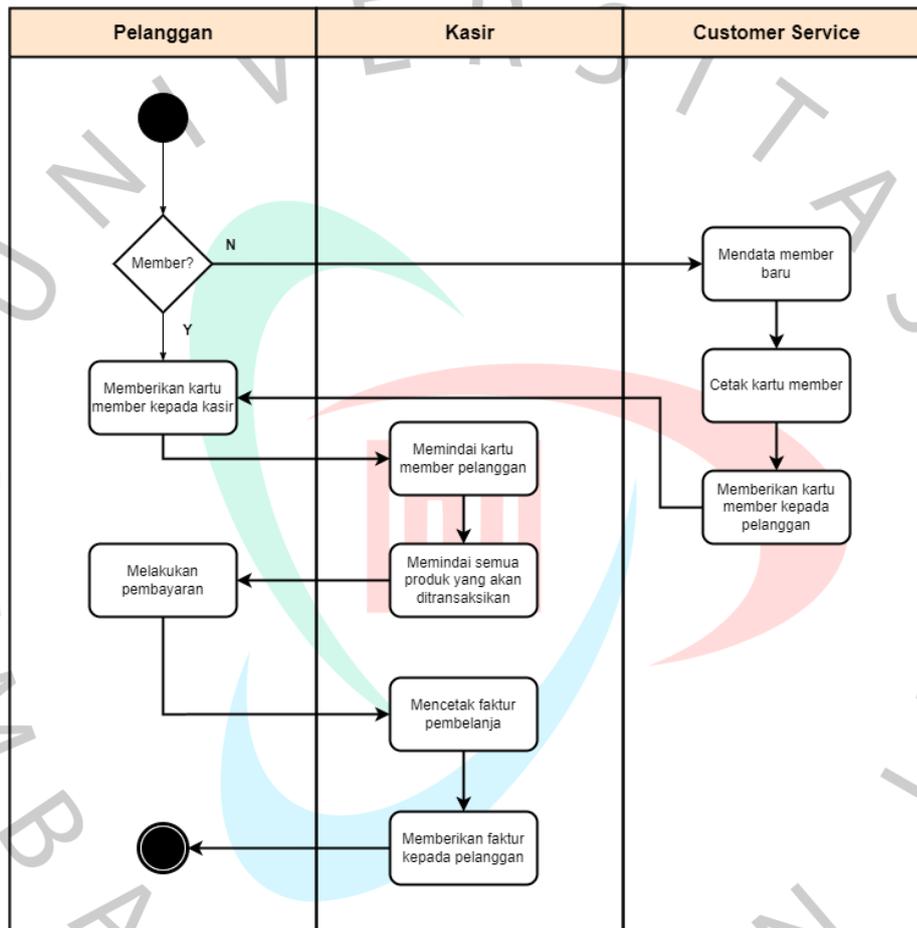
Berikut ini notasi yang digunakan dalam *activity* diagram:

Tabel 2.4 Notasi Activity Diagram

NOTASI	NAMA NOTASI	DESKRIPSI
	<i>Activity</i>	Merupakan representasi visual dari suatu aktivitas yang dilakukan dalam sistem atau proses bisnis. <i>Activity</i> biasanya diwakili oleh persegi panjang dan diberi label untuk menjelaskan aktivitas apa yang dilakukan.
	<i>Control Flow</i>	Menampilkan urutan eksekusi dari setiap aktivitas dalam sebuah <i>Activity Diagram</i> .
	<i>Initial Node</i>	Merupakan representasi visual yang menggambarkan awal dari serangkaian tindakan atau aktivitas.
	<i>Final-activity Node</i>	Merupakan representasi visual yang digunakan untuk menghentikan semua <i>control-flow</i> dalam aktivitas.
	<i>Decision Node</i>	Merupakan representasi visual yang digunakan untuk mewakili kondisi pengujian untuk memastikan bahwa <i>control-flow</i> hanya melewati satu jalur dan dilabeli dengan kriteria keputusan untuk melanjutkan ke aktivitas berikutnya.
	<i>Merge Node</i>	Merupakan notasi yang digunakan untuk menggabungkan jalur keputusan yang berbeda ke dalam satu aktivitas khusus.
	<i>Fork Node</i>	Merupakan notasi yang dipakai dengan tujuan memisahkan perilaku menjadi beberapa aliran aktivitas yang terjadi secara simultan.
	<i>Join Node</i>	Merupakan notasi yang dipakai dengan tujuan menggabungkan serangkaian aktivitas yang terjadi secara simultan.

<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">Swimlane</div>	<i>Swimlane</i>	Merupakan notasi yang dipakai untuk membagi diagram aktivitas menjadi baris dan kolom dengan tujuan menetapkan aktivitas individu atau objek yang bertanggung jawab dalam pelaksanaannya. Pada bagian atas kolom, terdapat label berisi nama individu atau objek yang bertanggung jawab.
--	-----------------	--

Berikut ini adalah contoh dari *activity* diagram yang dapat dilihat pada gambar 2.6:



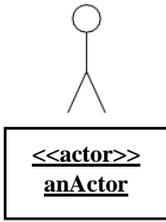
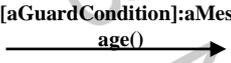
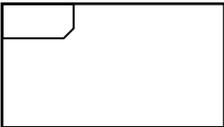
Gambar 2.6 Contoh Activity Diagram

#### 4. Sequence Diagram

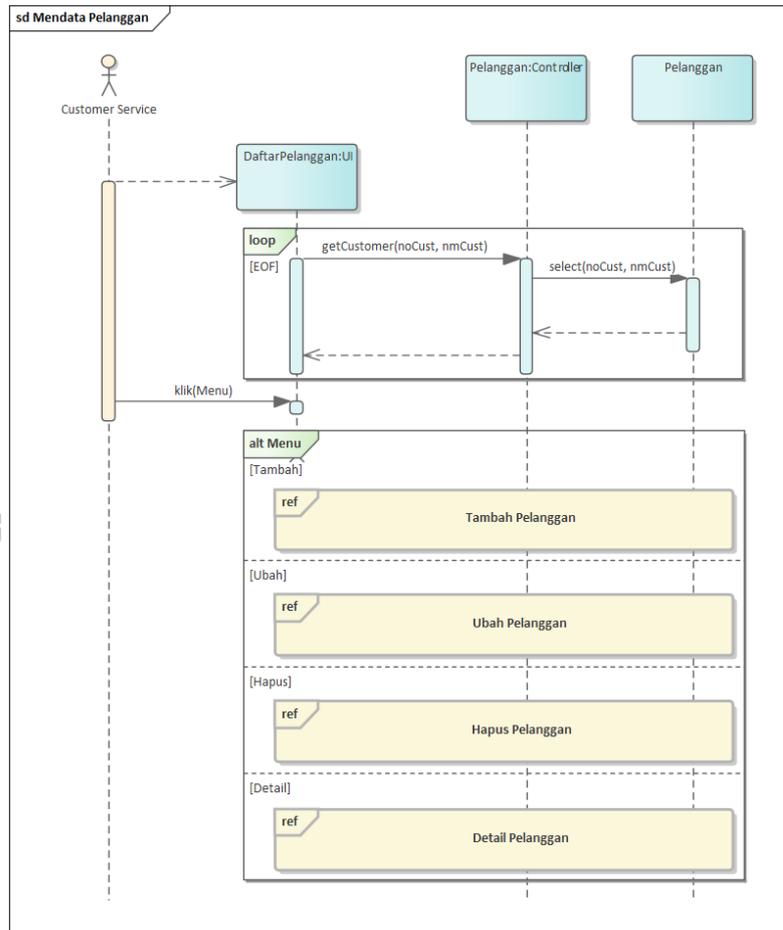
*Sequence* diagram merupakan diagram yang merepresentasikan perilaku sebuah objek yang ada pada *use case*. Representasi perilaku objek ini digambarkan dengan memperlihatkan masa hidup sebuah objek serta pesan apa yang dikirimkan dan diterima oleh objek tersebut (Hendini, 2016).

Berikut ini notasi yang dipakai dalam *sequence* diagram:

Tabel 2.5 Notasi *Sequence* Diagram

NOTASI	NAMA NOTASI	DESKRIPSI
	<i>Actor</i>	Merupakan individu atau sistem yang mendapatkan keuntungan dari dan berada di luar sistem. Aktor terlibat dalam rangkaian tindakan yang berurutan dengan mengirim dan/atau menerima pesan. Selanjutnya digambarkan sebagai figur tongkat ( <i>default</i> ) atau, jika melibatkan aktor bukan manusia, sebagai persegi panjang dengan <<actor>> di dalamnya (alternatif).
	<i>Object</i>	Merupakan representasi visual dari objek yang berinteraksi dalam <i>Sequence Diagram</i> . <i>Object</i> biasanya diwakili oleh sebuah kotak dengan nama objek di dalamnya. <i>Object</i> juga berpartisipasi dalam urutan dengan mengirim dan/atau menerima pesan dan ditempatkan di bagian atas diagram.
	<i>Lifeline</i>	Merupakan representasi visual yang menunjukkan waktu hidup suatu objek selama urutan. Berisi X pada titik di mana kelas tidak lagi berinteraksi.
	<i>Execution Occurrence</i>	Merupakan representasi visual yang menunjukkan interval waktu di mana objek sedang melakukan aktivitas. Notasi ini ditempatkan di atas <i>lifeline</i> dan diberi label untuk menjelaskan aktivitas apa yang dilakukan.
	<i>Message</i>	Merupakan representasi visual yang digunakan untuk menyampaikan informasi dari satu objek ke objek lainnya. Notasi ini dilabeli dengan pesan yang dikirim dan dalam sebuah garis penuh, sedangkan pengembalian pesan dilabeli dengan nilai yang dikembalikan dan ditampilkan dengan garis putus-putus.
	<i>Guard Condition</i>	Merepresentasikan pengujian yang harus dipenuhi agar pesan dapat dikirim.
	<i>Object Destruction</i>	Notasi ini digunakan untuk merepresentasikan bahwa sebuah objek tersebut akan hilang dan tidak digunakan lagi.
	<i>Fragment</i>	Notasi ini digunakan untuk merepresentasikan interaksi yang kompleks antara beberapa objek dalam <i>Sequence Diagram</i> . Notasi ini biasanya direpresentasikan oleh sebuah kotak dengan label yang menjelaskan bagaimana interaksi terjadi. Beberapa label di dalam sebuah <i>frame</i> antara lain <i>alt</i> , <i>loop</i> , <i>opt</i> , <i>break</i> , <i>par</i> , <i>strict</i> dan <i>seq</i> .

Berikut ini adalah contoh dari *sequence diagram* yang dapat dilihat pada gambar 2.7:



Gambar 2.7 Contoh Sequence Diagram

### 5. Class Diagram

Class Diagram adalah model statis yang menunjukkan kelas dan hubungan antar kelas yang tetap konstan dalam sistem dari waktu ke waktu. Diagram kelas menggambarkan kelas, yang mencakup perilaku dan status, dengan hubungan antar kelas (Dennis et al., 2015).

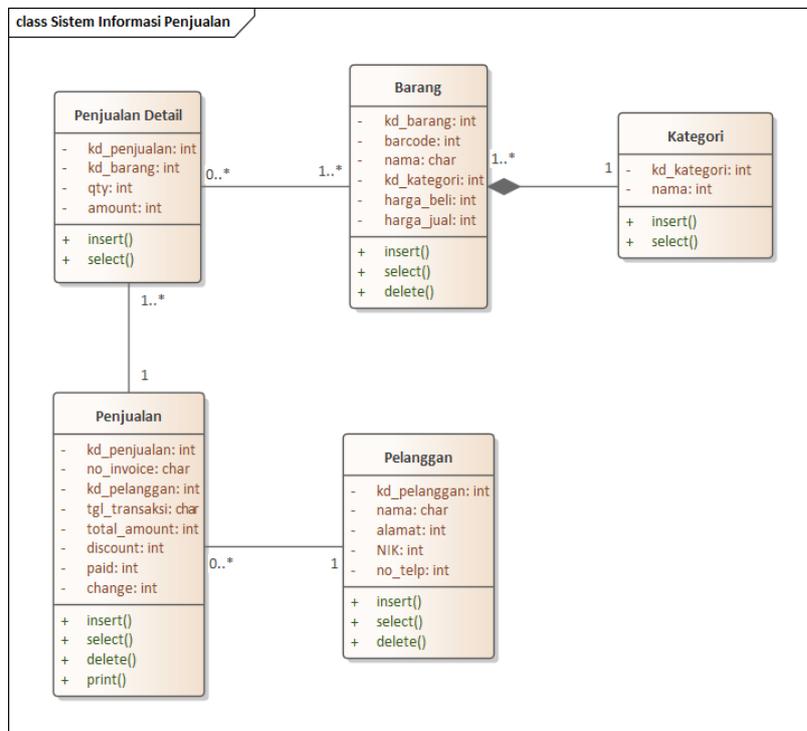
Berikut ini notasi yang digunakan dalam class diagram:

Tabel 2.6 Notasi Class Diagram

NOTASI	NAMA NOTASI	DESKRIPSI
<div style="border: 1px solid black; padding: 5px; width: fit-content;"> <p><b>Class1</b></p> <p>-Attribute-1</p> <p>+Operation-1()</p> </div>	Class	Merupakan representasi dari objek nyata atau konsep yang memiliki atribut dan metode. <i>Class</i> biasanya direpresentasikan oleh sebuah persegi panjang dengan nama <i>class</i> di dalamnya. Sebuah <i>class</i> mempunyai nama yang dicetak tebal dan ditempatkan di tengah kompartemen atas, mempunyai daftar atribut di kompartemen tengahnya dan mempunyai daftar operasi ( <i>method/operation</i> ) di kompartemen bawahnya.

<u>AssociatedWith</u> 0..*      1	<i>Association</i>	Merupakan representasi dari hubungan antara sebuah kelas dengan objek lainnya. Sebuah <i>association</i> diberi label menggunakan frasa kata kerja atau nama peran. Sebuah <i>association</i> juga berisi simbol multiplisitas, yang mewakili jumlah minimum dan maksimum instansiasi kelas dapat diasosiasikan dengan instansiasi kelas lainnya.
	<i>Generalization</i>	Merepresentasikan hubungan hierarki antara dua <i>class</i> di mana sebuah <i>class</i> merupakan sub- <i>class</i> dari <i>class</i> lain yang menjadi <i>superclass</i> -nya
	<i>Aggregation</i>	Merepresentasikan hubungan antara dua <i>class</i> yang menunjukkan bahwa sebuah <i>class</i> memiliki koleksi <i>object</i> lain yang dapat berdiri sendiri. <i>Aggregation</i> biasanya direpresentasikan oleh sebuah garis dengan sebuah <i>diamond</i> di ujungnya yang menghubungkan <i>class</i> yang memiliki koleksi dan <i>class</i> yang menjadi bagian dari koleksi tersebut.
	<i>Composition</i>	Hubungan antara dua <i>class</i> yang menunjukkan bahwa sebuah <i>class</i> memiliki <i>object</i> lain yang merupakan bagian dari dirinya dan tidak dapat berdiri sendiri. <i>Composition</i> biasanya direpresentasikan oleh sebuah garis dengan sebuah <i>diamond</i> hitam di ujungnya yang menghubungkan <i>class</i> yang memiliki <i>object</i> dan <i>class</i> yang menjadi bagian dari <i>object</i> tersebut.

Berikut ini adalah contoh dari *class diagram* yang dapat dilihat pada gambar 2.8:



Gambar 2.8 Contoh Class Diagram

## 2.2 Tinjauan Studi

Tinjauan pustaka merupakan sebuah kegiatan untuk menganalisis dan mempelajari berbagai sumber kepustakaan seperti buku, jurnal, artikel, dan sumber-sumber yang tentunya memiliki topik yang relevan dan terkait dengan penelitian ini (Mahanum, 2021).

Adapun beberapa tinjauan pustaka yang dijadikan bahan analisis yaitu:

1. Penelitian yang pertama dilakukan oleh Raden Kurnia dengan judul **“Sistem Repository Perpustakaan Pada Universitas Banten Jaya Berbasis Web (Studi Kasus: Universitas Banten Jaya)”** pada 27 Agustus, 2021, vol. 4 no. 2. Penelitian ini mengajukan diskusi tentang cara merancang sistem repositori perpustakaan yang berbasis internet, dengan tujuan memberikan kemudahan bagi petugas perpustakaan dalam mengelola informasi perpustakaan.

Dalam penelitian ini digunakan metode pengembangan sistem dengan model air terjun yang terdiri dari tahapan analisis, desain, implementasi dan juga pengujian.

Dalam kaitannya dengan Tugas Akhir (TA) penulis, tugas akhir ini dijadikan referensi dan acuan untuk mengetahui proses perancangan sebuah sistem repositori yang dapat digunakan untuk mengelola sistem informasi..

2. Penelitian yang kedua dilakukan oleh Imadudin Abdan Syakuro yang dilakukan pada 29 Juni, 2020 dengan judul **“Rancang Bangun Knowledge Management System Berbasis Website Repositori”**. Penelitian ini dilakukan di BAPPEDA Provinsi Lampung. Peneliti melakukan observasi pada BAPPEDA Provinsi Lampung dan menemui masalah dalam penanganan dokumen di intitusi tersebut.

Dalam penelitian ini, peneliti mengidentifikasi permasalahan yang terjadi antara lain seperti:

1. Tidak adanya manajemen versi dari setiap dokumen. Pengguna yang ingin mengakses dan menggunakan dokumen tersebut tidak mengetahui proses dan *history* perubahan yang terjadi pada dokumen tersebut

2. Keterbatasan dalam mengakses dokumen. Para pengguna yang memiliki keperluan untuk mengakses dokumen harus berada di area kantor. Karena karyawan yang berada di luar jangkauan atau di luar area, tidak bisa mengakses komputer server di kantor.
3. Fitur pencarian dokumen masih mengandalkan basis pencarian yang dimiliki oleh sistem operasi *windows* sehingga membutuhkan waktu yang lama ketika mencari sebuah dokumen.

Hasil dari membaca dan memahami penelitian ini, penulis dapat mempelajari dan memahami permasalahan yang terjadi terkait pengelolaan dokumen. Lalu selanjutnya penulis dapat menjadikan penelitian ini sebagai referensi untuk dapat merancang dan membangun sistem repositori untuk dapat mengatasi permasalahan-permasalahan yang sama.

3. Penelitian yang ketiga dilakukan oleh Antonius Rahmat yang dilakukan pada tahun 2020 dengan judul “**Aplikasi Pengelolaan Dokumen dan Arsip berbasis Web untuk mengatur Sistem kearsipan dengan menggunakan Metode Waterfall**” vol. 11 no. 2. Penelitian ini berisi bahasan seputar permasalahan yang terjadi di dalam PT. POS Indonesia terkait pengelolaan arsip yang kurang baik karena jumlah arsip yang banyak dan kurang teratur dalam penataannya .

Hasil dari penelitian ini yaitu sebuah sistem pengelolaan dokumen dan arsip berbasis web dengan fitur utama adalah transaksi arsip. Tujuan dibuatnya arsip ini adalah untuk dapat membantu sistem kearsipan pada perusahaan-perusahaan besar dan memiliki banyak arsip supaya sistem kearsipannya berjalan dengan lebih teratur dan terkonsep.

Dalam kaitannya dengan Tugas Akhir ini, penelitian ini dijadikan acuan untuk mempelajari dan menganalisis bagaimana caranya membuat sebuah sistem repositori yang dapat digunakan untuk mengelola banyak dokumen agar lebih mudah digunakan kembali.

4. Penelitian yang keempat dilakukan oleh oleh Dicky Hariyanto, Ricki Sastra, Ferina Eka Putri yang dilakukan pada tahun 2021 dengan judul “**Implementasi Metode Rapid Application Development Pada Sistem**

**Informasi Perpustakaan**". Penelitian ini berisi bahasan mengenai implementasi metode *Rapid Application Development* dalam merancang dan membangun sebuah sistem informasi.

Dalam penelitian ini, berisi penjelasan bagaimana tahapan-tahapan dalam mengimplementasikan metode RAD ketika membangun sebuah sistem serta keuntungan yang dimiliki dari metode ini. Hasil dari penelitian ini yaitu dibangunnya sebuah Sistem Informasi Perpustakaan untuk SMK Merah Putih Bekasi.

Penelitian ini memiliki hubungan dengan penelitian ini dalam hal penggunaan metode pengembangan sistem. Dari penelitian ini dapat mengetahui dan mempelajari tahapan-tahapan yang ada di dalam metode RAD serta dapat menggunakannya metode tersebut dalam merancang dan membangun sistem repositori

- 5. Penelitian yang kelima dilakukan oleh Muh. Rais yang dilakukan pada tahun 2019 dengan judul "**Penerapan Konsep *Object Oriented Programming* Untuk Aplikasi Pembuat Surat**". Penelitian ini berisi bahasan terkait konsep pemrograman berorientasikan objek.

Penelitian ini menjelaskan konsep-konsep dasar yang harus dipahami ketika mengimplementasikan OOP. Selain itu, penelitian ini juga membahas keuntungan-keuntungan dari penggunaan paradigma pemrograman OOP. Berikut ini beberapa keuntungan yang didapatkan ketika menggunakan paradigma OOP:

1. Meningkatkan Daya Produksi

Kelas dan objek yang teridentifikasi dalam satu masalah dapat dipakai ulang untuk masalah lain dengan melibatkan objek yang sama (dapat digunakan ulang).

2. Percepatan pengembangan

Proses perancangan sistem yang dilakukan dengan tepat akan mengurangi kesalahan pada tahap pengkodean dan mempercepat proses pengembangan.

3. Mudah pemeliharaan

Penggunaan model objek, pola-pola yang cenderung konsisten dan

stabil dapat diisolasi, sementara pola-pola yang mungkin sering berubah dapat diidentifikasi secara terpisah.

4. Terciptanya konsistensi

Konsistensi yang terjadi disebabkan oleh penerapan pewarisan properti pada objek dan penggunaan notasi yang sama sepanjang fase analisis perancangan sistem hingga fase pengkodean.

5. Meningkatkan kualitas perangkat lunak

Dengan menggunakan pendekatan pengembangan yang lebih realistis dan menjaga konsistensi dalam prosesnya, perangkat lunak yang dihasilkan akan memenuhi kebutuhan pengguna dan meminimalisir tingkat kesalahan.

Dalam kaitannya dengan Tugas Akhir, penelitian ini dapat dijadikan sebagai acuan untuk mengimplementasikan konsep OOP pada fase pengkodean

- (*coding*). Terlebih lagi, di dalam penelitian ini menggunakan bahasa pemrograman yang sama yaitu PHP, sehingga dapat membantu penulis dalam mempelajari implementasi OOP dalam bahasa pemrograman PHP.