

## BAB II

### LANDASAN TEORI

#### 2.1 Presensi

Presensi karyawan adalah suatu kegiatan pencatatan terhadap setiap kehadiran karyawan dengan tujuan untuk mengetahui data yang berkaitan dengan kehadiran karyawan secara periodik baik harian maupun bulanan.

(Ratna Wulandari<sup>1</sup> , Danuri<sup>2</sup>, Jaroji<sup>3</sup> 2019)

#### 2.2 QR Code

QR Code adalah singkatan dari quick response code. Kode ini adalah barcode dua dimensi yang bisa memberikan beragam jenis informasi secara langsung. Untuk membukanya, dibutuhkan scan atau pemindaian dengan smartphone. QR Code biasanya mampu menyimpan 2089 digit atau 4289 karakter, termasuk tanda baca dan karakter spesial.

Hal ini membuat QR Code mampu menampilkan teks pada pengguna, membuka URL, menyimpan kontak ke buku telepon dan masih banyak lagi. QR Code dinilai lebih praktis dibanding barcode karena mampu menyimpan lebih banyak data. QR Code terdiri dari titik-titik hitam dan spasi putih yang disusun dalam bentuk kotak, dan setiap elemennya memiliki makna tersendiri. Hal tersebut membuatnya mampu di-scan oleh smartphone dan menampilkan data atau informasi yang dimuatnya.

Bagian-bagian dari QRCode yaitu :

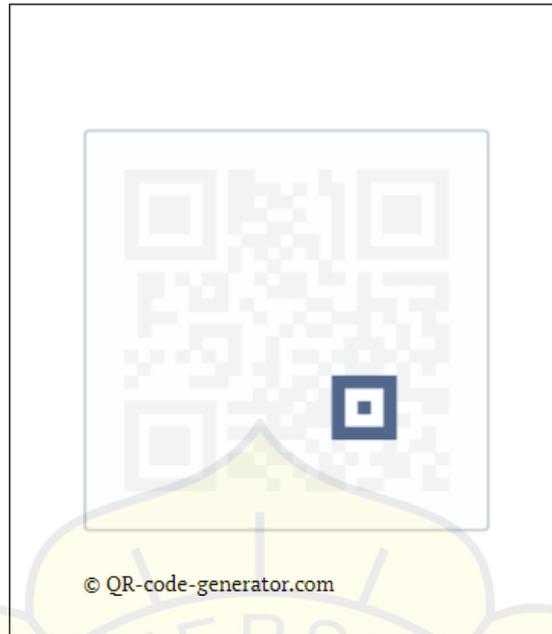
1. *Positioning Detection Markers* adalah bagian dalam QR Code yang berbentuk kotak dan jumlahnya ada tiga. Posisinya selalu ada di pojok QR

Code. Fungsinya adalah memastikan scanner mampu melakukan pembacaan kode secara cepat dan mengetahui orientasi atau posisi kode tersebut. Contohnya :



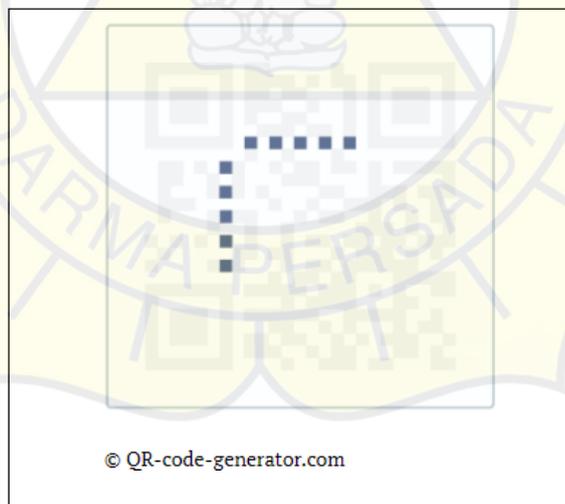
Gambar 2. 1 Contoh Positioning Detection Markers (*Glints.com*)

2. *Alignment Marking* yaitu penanda ini ukurannya lebih kecil dibandingkan *position detention markers* sama-sama berbentuk kotak. Fungsinya untuk menjaga permukaan QR Code meskipun di print di permukaan yang melengkung. Juga semakin banyak suatu data disimpan dalam QR Code ukurannya semakin besar dan jumlahnya pun bisa jadi lebih banyak. Contohnya :



Gambar 2. 2 Contoh Alignment Marking (*Glints.com*)

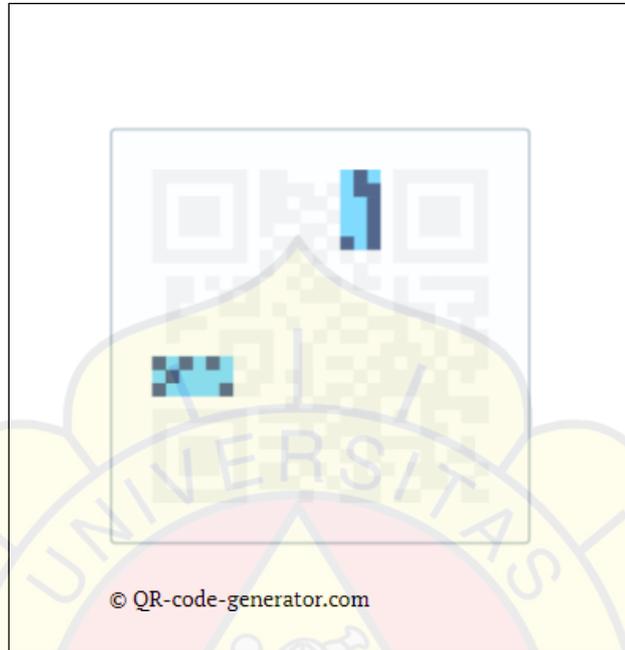
3. *Timing Pattern* adalah bagian yang tampak seperti kotak-kotak kecil berjejer, fungsinya untuk konfigurasi data grid. Scanner QR Code mampu mengetahui seberapa besar matriks data yang dimuat. Contohnya :



Gambar 2. 3 Contoh Timing Pattern (*Glints.com*)

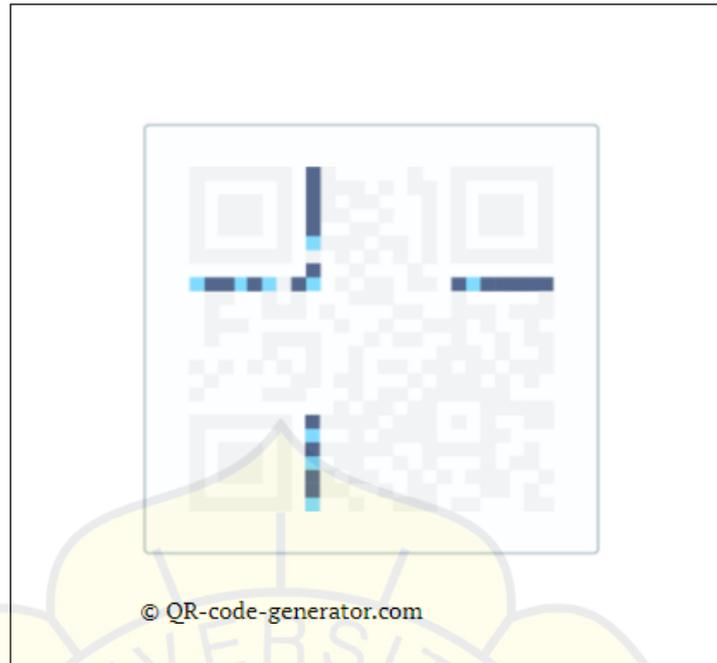
4. *Version Informasion* adalah bagian yang memberi informasi versi QR Code, pada saat ini ada 40 tipe QR Code yang berbeda dengan tanda ini *scanner*

bisa mengetahui versi QR Code mana yang dipindai, biasanya versi 1 sampai 7 adalah yang paling umum digunakan. Contohnya :



Gambar 2. 4 Contoh Version Informasi (*Glints.com*)

5. *Format Information* adalah bagian yang menjelaskan toleransi *error* dan pola *data mask*, *scanner* akan lebih mudah melakukan pemindaian QR Code untuk menampilkan data yang dimuatnya pada pengguna. Contohnya :



Gambar 2. 5 Contoh Format Information (*Glints.com*)

6. Data and error correction keys merupakan tempat dimana semua data disimpan dan meliputi bagian area *error correction block* yang menjaga data tetap dapat dipindai meski kode rusak sebanyak 30%. Contohnya :



Gambar 2. 6 Contoh Data and error correction keys (*Glints.com*)

7. *Quiet Zone* adalah elemen penting dalam QR Code untuk menegaskan struktur dan membuatnya lebih mudah dipindai, quiet zone harus ada untuk

memisahkan QR Code dari lingkungan sekitarnya agar scanner dapat mengenalinya tanpa kesulitan walaupun kosong area ini adalah bagian vital dari sebuah QR Code. Contohnya :



Gambar 2. 7 Contoh Quiet Zone (*Glints.com*)

### 2.3 Webcam

Webcam alias 'web camera' merupakan perangkat yang berupa sebuah kamera digital yang dihubungkan ke komputer atau laptop. Layaknya kamera pada umumnya, sebuah webcam dapat merekam foto secara *realtime/live* dari manapun ia berada ke seluruh penjuru dunia dengan bantuan internet dan tidak untuk mendeteksi wajah.

Fungsi Webcam antara lain yaitu :

1. Berkomunikasi
2. Belajar Jarak Jauh
3. Konferensi Jarak Jauh

## 2.4 Algoritma

Algoritma merupakan fondasi yang harus dipahami atau dikuasai oleh seseorang yang akan menyelesaikan suatu masalah dengan komputer, dalam hal ini dengan membuat program.

( Drs. Lamhot Sitorus, M.Kom. 2015 )

### 2.4.1 Kriptografi

Kriptografi dapat didefinisikan sebagai seni maupun ilmu yang menghasilkan pesan yang rahasia. Sebuah pesan asli yang disebut sebagai plaintext disandikan menjadi pesan yang tersandi yang disebut sebagai ciphertext melalui proses enkripsi dan ciphertext dipulihkan menjadi plaintext kembali melalui proses dekripsi.

( Yusfrizal 2019 )

### 2.4.2 Viginere Cipher

Viginere Cipher termasuk dalam cipher abjad majemuk (Polyalphabetic Substitution Cipher) yang dipublikasikan oleh diplomat (sekaligus seorang kriptologis) Perancis, Blaise de Viginere pada tahun 1586. Viginere Cipher adalah metode menyandikan teks alfabet dengan menggunakan deretan sandi Caesar berdasarkan huruf-huruf pada kata kunci.

( Irham Mu'alimin Arrijal , Rusdi Efendi , Boko Susilo 2016 )

Teknik dari substitusi viginere cipher biasa dilakukan dengan 2 cara :

#### 1. Angka

Viginere Cipher dengan angka adalah metode menyandikan teks alfabet dengan menggunakan deretan sandi Caesar berdasarkan huruf-huruf pada kata kunci.

## 2. Huruf

Vigenere Cipher dengan huruf berisi alfabet yang dituliskan dalam 26 baris, masingmasing baris digeser ke kiri dari baris sebelumnya membentuk ke-26 kemungkinan sandi Caesar setiap huruf disediakan dengan menggunakan baris yang berbeda-beda sesuai kunci yang diulang.

Rumus dari enkripsi dan dekripsi data vigenere cipher adalah :

Enkripsi :

$$C_i = (P_i + K_i) \bmod 26$$

Dekripsi :

$$P_i = (C_i - K_i) \bmod 26; \text{ untuk } C_i \geq K_i$$

$$P_i = (C_i + 26 - K_i) \bmod 26; \text{ untuk } C_i < K_i .$$

( Irham Mu'alimin Arrijal , Rusdi Efendi , Boko Susilo 2016 )

Contoh Tabel *Vigènere cipher* menggunakan Bujursangkar *vigènere* untuk melakukan enkripsi (lihat tabel ). Kolom paling kiri menyatakan huruf-huruf kunci sedangkan baris paling atas menyatakan huruf-huruf *plainteks*. Setiap baris dalam bujursangkar menyatakan huruf-huruf cipheteks yang diperoleh dengan *Caesar Cipher*, yang mana jauh pergeseran huruf *plainteks* ditentukan oleh nilai decimal oleh huruf kunci tersebut ( $a = 0, b = 1, \dots, z = 25$ ).

Sebagai contoh, huruf kunci c menyatakan huruf *plainteks* digeser sejauh 2 huruf ke kanan dari susunan alfabetnya.

Tabel 2. 1 Bujursangkar Vigènere

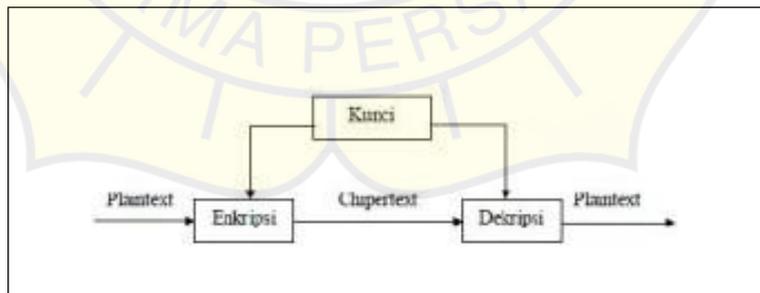
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z		
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z			
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z				
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z					
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z						
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z							
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z								
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z									
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z										
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z											
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z												
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z													
O	O	P	Q	R	S	T	U	V	W	X	Y	Z														
P	P	Q	R	S	T	U	V	W	X	Y	Z															
Q	Q	R	S	T	U	V	W	X	Y	Z																
R	R	S	T	U	V	W	X	Y	Z																	
S	S	T	U	V	W	X	Y	Z																		
T	T	U	V	W	X	Y	Z																			
U	U	V	W	X	Y	Z																				
V	V	W	X	Y	Z																					
W	W	X	Y	Z																						
X	X	Y	Z																							
Y	Y	Z																								
Z	Z																									

( Sumber : Modifikasi Algoritma Vigenere Cipher Menggunakan Metode Catatan Number Dan Double Columnar Transposition Vol.4, No.1, Mei 2015 )

### 2.4.3 Enkripsi

Enkripsi adalah proses penyandian plainteks menjadi cipherteks, sedangkan dekripsi adalah proses mengembalikan cipherteks menjadi plainteks semula. Enkripsi dan dekripsi membutuhkan kunci sebagai parameter yang digunakan untuk transformasi.

( Yusfrizal 2019 )



Gambar 2.8 Skema Enkripsi dan Deskripsi Kriptografi Type Symmetric Key

## 2.5 PHP

PHP singkatan dari Hypertext Preprocessor yang digunakan sebagai bahasa script server-side dalam pengembangan Web yang disisipkan pada dokumen HTML.

( Dani Ainur Rivai , Sukadi 2013 )

## 2.6 Javascript

JavaScript adalah bahasa pemrograman yang sederhana karena bahasa ini tidak dapat digunakan untuk membuat aplikasi ataupun applet. Dengan JavaScript kita dapat dengan mudah membuat sebuah halaman web yang interaktif.

( Dani Ainur Rivai , Sukadi 2013 )

## 2.7 DataBase

Database adalah sekumpulan data yang berisi informasi mengenai satu atau beberapa object. Data dalam database tersebut biasanya disimpan dalam tabel yang saling berhubungan antara satu dengan yang lain.

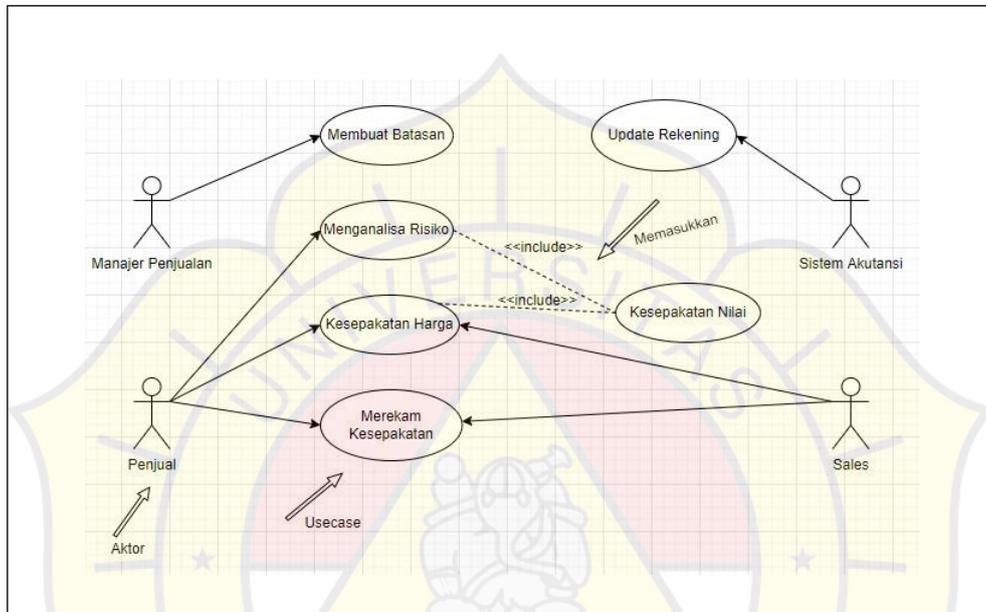
( Dani Ainur Rivai , Sukadi 2013 )

## 2.8 UML Diagram

Menurut Martin Fowler, 2004:1 *Unified Modeling Language (UML)* adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OO).

### 2.8.1 Diagram Usecase

Menurut Martin Fowler, 2004:141 *Use Case* adalah teknik untuk merekam persyaratan fungsional sebuah sistem. *Use case* mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan. Contoh *Use case* :



Gambar 2. 9 Usecase Diagram

Komponen – komponen pembentuk use case diagram adalah :

- A. Aktor : Merupakan gambaran pelaku atau pengguna dari sistem atau produk.



Gambar 2. 10 Simbol Actor

- B. Use Case : Merupakan gambaran fungsi dari sistem yang diberi label dengan kata kerja yang deskriptif.



Gambar 2. 11 Simbol Use Case

C. Relasi : Merupakan menghubungkan antara Aktor dan Kasus Kegunaan dan menggambarkan hubungannya dengan sistem atau produk.



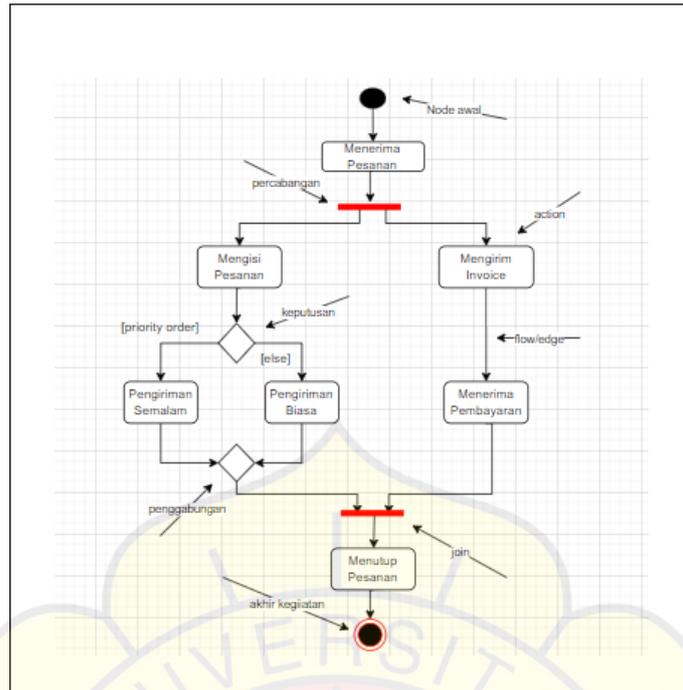
Gambar 2. 12 Simbol Relasi

Relasi memiliki tipe relasi yang mungkin terjadi pada diagram Use Case :

1. *<<uses>>* yaitu kelakuan yang harus terpenuhi agar sebuah *event* dapat terjadi, dimana pada kondisi ini sebuah *use case* adalah bagian dari *use case* lainnya.
2. *<<extends>>* yaitu kelakuan yang hanya berjalan di bawah kondisi tertentu seperti menggerakkan alarm.

### 2.8.2 Diagram Activity

Menurut Martin Fowler, 2004:163 Activity Diagram adalah teknik untuk menggambarkan logika prosedural, proses bisnis dan jalur kerja. Dalam beberapa hal, diagram ini memainkan peran mirip sebuah diagram alir, tetapi perbedaan prinsip antara diagram ini dan notasi diagram alir adalah diagram ini mendukung behavior paralel. Contoh Activity Diagram :

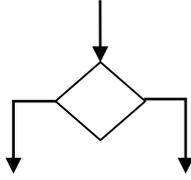


Gambar 2. 13 Contoh Activity Diagram

Komponen – komponen pembentuk activity diagram :

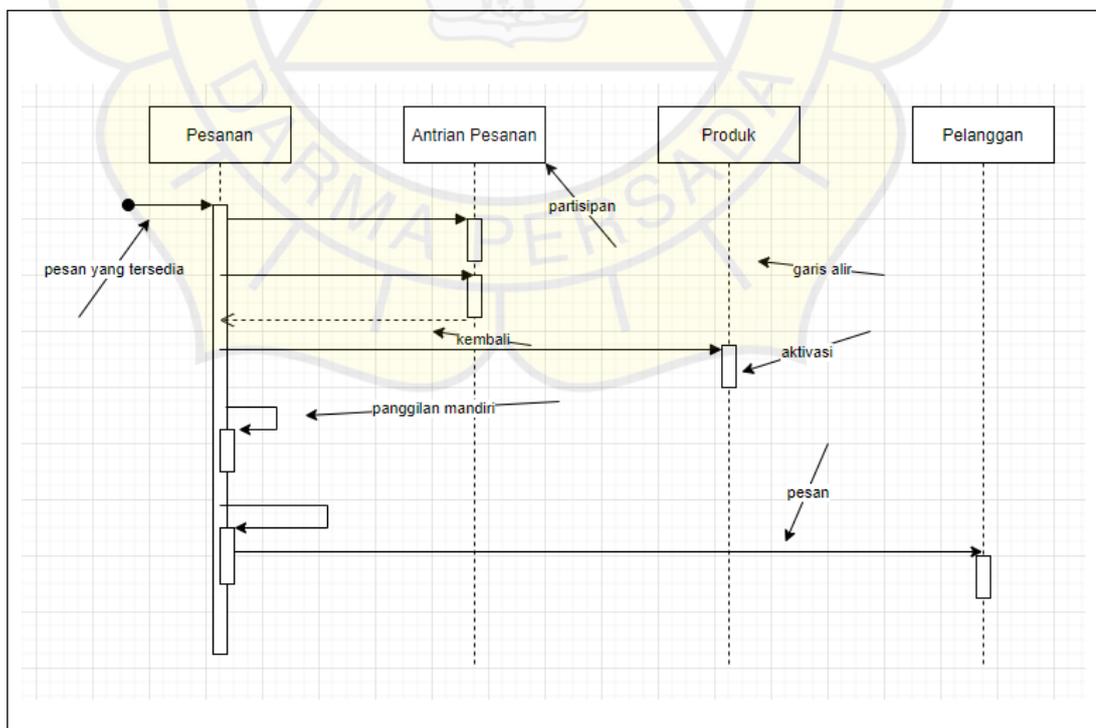
Tabel 2. 2 Simbol-simbol Activity Diagram

NO	SIMBOL	KETERANGAN
1.		Menunjukkan titik awal dari urutan tindakan/aktivitas yang ada.
2.		Digunakan untuk menyatakan suatu tindakan, dan juga menyatakan tahapan aksi dari pelaku.
3.		Menunjukkan urutan dari awal hingga akhir sebagaimana diagram alur pada umumnya.
4.		Digunakan untuk menunjukkan ruang lingkup peran. Dengan demikian, peran pembagian peran tiap aktivitas yang ada dapat diperjelas.

5.		<p>Sebagaimana umumnya untuk memfasilitasi percabangan pada diagram alir.</p>
6.		<p>Menunjukkan titik akhir dari urutan tindakan/aktivitas yang ada.</p>

### 2.8.3 Diagram Sequence

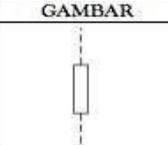
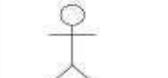
Menurut Martin Fowler, 2004:81 Sequence Diagram adalah *interaction* diagram menunjukkan bagaimana kelompok-kelompok objek saling berkolaborasi dalam beberapa behavior. Secara khusus, menjabarkan *behavior* sebuah skenario tunggal, diagram tersebut menunjukkan sejumlah objek contoh dan pesan-pesan yang melewati objek-objek ini didalam use case. Contoh Sequence Diagram :



Gambar 2. 14 Contoh Sequence Diagram

Komponen – komponen pembentuk sequence diagram :

Tabel 2. 3 Simbol-simbol Sequence Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		LifeLine	Objek entity, antarmuka yang saling berinteraksi.
		Actor	Digunakan untuk menggambarkan user / pengguna.
2		Message	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
3		Boundary	Digunakan untuk menggambarkan sebuah form.
4		Control Class	Digunakan untuk menghubungkan boundary dengan tabel.
5		Entity Clas	Digunakan untuk menggambarkan hubungan kegiatan yang akan dilakukan.

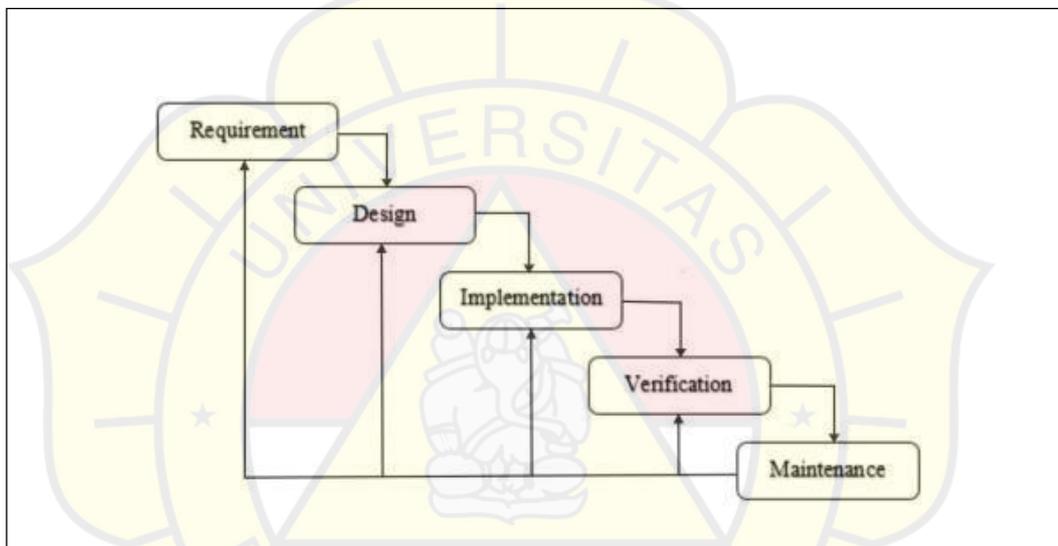
## 2.9 Waterfall

Model Waterfall merupakan salah satu model SDLC yang sering digunakan dalam pengembangan sistem informasi atau perangkat lunak. Model ini menggunakan pendekatan sistematis dan berurutan. Tahapan dalam model ini dimulai dari tahap perencanaan hingga tahap pengelolaan (*maintenance*) dan dilakukan secara bertahap. Pengembang perlu mengetahui lebih lanjut tentang bagaimana proses pengembangan sistem jika menggunakan model *waterfall* dan juga karakteristik dari model *waterfall* tersebut.

Model *waterfall* pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai didalam *Software Engineering (SE)*. saat ini model *waterfall* merupakan model pengembangan perangkat lunak yang sering digunakan. Model pengembangan ini melakukan pendekatan secara sistematis dan berurutan. Disebut

*waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Model pengembangan ini bersifat *linear* dari tahap awal pengembangan sistem yaitu tahap perencanaan sampai tahap akhir pengembangan sistem yaitu tahap pemeliharaan. Tahapan berikutnya tidak akan dilaksanakan sebelum tahapan sebelumnya selesai dilaksanakan dan tidak bisa kembali atau mengulang ke tahap sebelumnya.

Tahapan Metode Waterfall sebagai berikut :



Gambar 2. 15 Contoh Tahapan Metode Waterfall

1. *Requirement* Tahap ini pengembang sistem diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Informasi dapat diperoleh melalui wawancara, diskusi atau survei langsung. Informasi dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna.
2. *Design* Pada tahap ini, pengembang membuat desain sistem yang dapat membantu menentukan perangkat keras (*hardware*) dan sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

3. *Implementation* Pada tahap ini, sistem pertama kali dikembangkan di program kecil yang disebut unit, yang terintegrasi dalam tahap selanjutnya. Setiap unit dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai unit testing.
4. *Verification* Pada tahap ini, sistem dilakukan *verifikasi* dan pengujian apakah sistem sepenuhnya atau sebagian memenuhi persyaratan sistem, pengujian dapat dikategorikan ke dalam unit testing (dilakukan pada modul tertentu kode), sistem pengujian (untuk melihat bagaimana sistem bereaksi ketika semua modul yang terintegrasi) dan penerimaan pengujian (dilakukan dengan atau nama pelanggan untuk melihat apakah semua kebutuhan pelanggan puas).
5. *Maintenance* Ini adalah tahap akhir dari metode *waterfall*. Perangkat lunak yang sudah jadi dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya.



**TEKNOLOGI INFORMASI**  
**UNIVERSITAS DARMA PERSADA**