

BAB II

TINJAUAN PUSTAKA

2.1 Teori-Teori Pendukung

Teori-teori pendukung adalah teori yang berhubungan dengan penelitian yang akan dilakukan, yang akan dijelaskan untuk mempermudah dalam memahami tentang aplikasi yang akan dibuat dan apa saja yang digunakan dalam membuat aplikasi tersebut.

2.1.1 *Quick Response code (QR Code)*

QR code adalah bentuk evolusi kode batang dari satu dimensi menjadi dua dimensi penggunaan *QR code* sudah sangat lazim di Jepang karena kemampuannya menyimpan data yang lebih besar dari kode batang, dan mampu mengkodekan informasi dengan bahasa Jepang karena mampu menampung huruf kanji. *QR code* adalah suatu jenis kode matriks atau kode batang yang dikembangkan oleh *Denso Wave*, sebuah divisi *Denso Corporation* yang merupakan perusahaan Jepang dan dipublikasikan pada tahun 1994 dengan fungsi utama yaitu dapat dengan mudah dibaca oleh pemindai. *QR code* berfungsi bagaikan hipertaut fisik yang dapat menyimpan alamat dan URL, nomor telepon, teks dan sms yang dapat digunakan pada majalah, surat harian, iklan, pada tanda-tanda bus, kartu nama ataupun media lainnya. Atau dengan kata lain sebagai penghubung secara cepat konten daring dan konten luring. Kehadiran kode ini memungkinkan audiens berinteraksi dengan media yang ditemelinya melalui ponsel secara efektif dan efisien. Pengguna juga dapat menghasilkan dan mencetak sendiri *QR code* untuk orang lain dengan mengunjungi salah satu dari beberapa ensiklopedia *QR code* (Lutfhi, 2016).



Gambar 2.1 *QR code*.

2.1.2 Android

Android adalah sistem operasi *mobile* berbasis *open source* yang dimiliki raksasa internet saat ini, Android dikembangkan dengan menggunakan kernel linux. Android memungkinkan untuk dimodifikasi secara bebas dan didistribusikan oleh pembuat perangkat tersebut dengan sifat *open source* tersebut telah banyak mendorong komunitas pengembang aplikasi untuk menggunakan *source code* Android sebagai dasar proyek pembuatan aplikasi. Android dimulai sebagai sebuah *start up* rahasia pada tahun 2003, dan dibeli oleh Google pada tahun 2005 dan sebagai jalan google untuk memasuki pasar perangkat lunak bergerak. *Handphone* komersil pertama yang menggunakan OS Android adalah *HTC Dream*, yang diluncurkan pada 22 Oktober 2008 (Lutfi, 2016). Dikutip dari kompas.com (2019), terungkap pula sebanyak 384 juta *Smartphone* dengan menggunakan OS Android yang berhasil terjual di Indonesia pada tahun 2018, untuk penjualan total semua jenis ponsel sebanyak 455 juta.



Gambar 2.2 Android

2.1.3 Sejarah Android

Pada awal mulanya Android Inc, merupakan sebuah perusahaan *software* kecil yang didirikan pada bulan Oktober 2003 di Palo Alto, California, USA. Didirikan oleh beberapa senior di beberapa perusahaan yang berbasis IT & Communication: Andy Rubin, Rich Miner, Nick Sears, dan Chris White. Menurut Rubin, Android Inc didirikan untuk mewujudkan *Mobile device* yang lebih mengerti pemiliknya(Chandra,2016). Android adalah sistem operasi pada *smarthphone*,

tablet dan perangkat elektronik pintar lainnya, bahkan kini Android sudah tersedia untuk versi desktop komputer/laptop. Android bisa kita peroleh secara gratis dan dapat dipasang/install berdampingan dengan sistem operasi lainnya pada desktop komputer. Ponsel atau smarthphone adalah benda yang tak dapat digunakan tanpa adanya perangkat lunak di dalamnya. Dalam hal ini ponsel kita sebut sebagai *hardware* (perangkat keras), sedangkan sistem operasi kita sebut sebagai *software* (perangkat lunak). Sistem operasi inilah yang menggerakkan ponsel atau smartphone yang kita miliki. Sistem operasi itu seperti nyawa dari makhluk hidup yang menggerakkan organ-organ di dalamnya dan Android menjadi semakin populer karena tersedia banyak aplikasi pada *Play Store* Android baik itu yang berbayar maupun yang gratis (Devi, 2019).

Sejak tahun 2008, Android mulai secara bertahap melakukan sejumlah pembaruan atau update untuk meningkatkan kinerja dari sistem operasi tersebut dengan menambahkan fitur baru, memperbaiki *bug* pada versi android yang sebelumnya. Setiap versi yang dirilis dinamakan secara *alfabeth* dengan berdasarkan nama sebuah makanan pencuci mulut, seperti *cupcake*, *donut*, dan sebagainya. Berikut nama-nama versi android.

1. Android 1.5 *Cupcake*



Gambar 2.3 Android versi *cupcake*.

Android 1.5 *Cupcake* dirilis pada tanggal 30 April 2009. *Cupcake* adalah versi android pertama yang menggunakan nama dari sebuah makanan. Konon katanya versi android *cupcake* seharusnya versi 1.2, namun Google telah memutuskan untuk membuat revisi yang besar serta membuatnya

menjadi versi 1.5. *Cupcake* adalah kue yang berbentuk kecil dan dipanggang dalam cetakan berbentuk *cup*(Devi, 2019)

2. Android 1.6 *Donut*



Gambar 2.4 Android versi *donut*.

Android 1.6 Donut dirilis pada tanggal 15 September 2009. pada versi ini telah diperbaiki beberapa kesalahan *reboot*, perubahan pada fitur foto dan video serta integrasi pencarian yang lebih baik. Donut merupakan makanan yang berbentuk cincin(Devi, 2019).

3. Android 2.0/2.1 *Éclair*



Gambar 2.5 Android versi *eclair*.

Android 2.0/2.1 *Eclair* dirilis pada tanggal 26 Oktober 2009. *Eclair* merupakan makanan penutup yang berupa kue berbentuk persegi panjang dan ada krim di tengah serta lapisan coklat di atasnya(Devi, 2019).

4. Android 2.2 *Froyo*



Gambar 2.6 Android versi *Froyo*.

Android 2.2 *Froyo* dirilis pada tanggal 20 Mei 2010. *Froyo* merupakan makanan penutup yang berasal dari sebuah nama merk produk yang terbuat dari yoghurt. *Froyo* adalah yoghurt yang dingin sehingga seperti es krim. *Froyo* merupakan singkatan dari *Frozen yoghurt*(Devi, 2019).

5. Android 2.3 *Gingerbread*



Gambar 2.7 Android versi *Gingerbread*.

Android 2.3 *Gingerbread* dirilis pada tanggal 6 Desember 2010. *Gingerbeard* merupakan sejenis kue kering yang memiliki rasa jahe. *Gingerbread* biasanya dibuat pada saat perayaan libur akhir tahun di benua Amerika(Devi, 2019).

6. Android 3.0 *Honeycomb*



Gambar 2.8 Android versi *Honeycomb*.

Android 3.0 *Honeycomb* dirilis pada tanggal 22 February 2011. *Honeycomb* merupakan sereal sarapan manis yang sudah pernah dibuat tahun 1965 oleh Posting Sereal. *Honeycomb* atau sarang lebah, sereal ini terbuat dari beberapa potongan jagung yang kemudian dibentuk seperti sarang lebah dengan rasa madu (Devi, 2019).

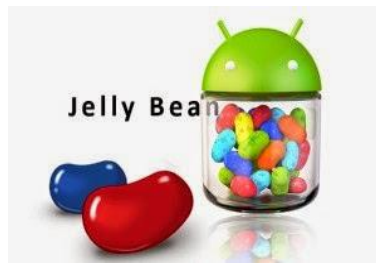
7. Android 4.0 *Ice Cream Sandwich*



Gambar 2.9 Android versi *Sandwich*.

Android 4.0 *Ice Cream Sandwich* dirilis pada tanggal 19 Oktober 2011. *Ice cream sandwich* merupakan lapisan es krim yang biasanya terdapat rasa vanila terjepit di antara dua kue cokelat, berbentuk persegi panjang (Devi, 2019).

8. Android 4.1 *Jelly Bean*



Gambar 2.10 Android versi *JellyBean*.

Android 4.1 *Jelly Bean* dirilis pada tanggal 9 Juli 2012. *Jelly bean* adalah nama sejenis permen dalam beraneka macam rasa buah-buahan. Ukurannya seperti kacang merah. Permen ini keras di luar namun lunak di dalam dan lengket apabila digigit (Devi, 2019).

9. Android 4.4 *KitKat*



Gambar 2.11 Android versi *Kitkat*.

Android 4.4 *KitKat* dirilis pada tanggal 31 Oktober 2013. *KitKat* merupakan merk cokelat yang dikeluarkan oleh Nestle (Devi, 2019).

10. Android 5.0 *Lollipop*



Gambar 2.12 Android versi Lollipop.

Android 5.0 *Lollipop* dirilis pada tanggal 15 Oktober 2014. *Lollipop* merupakan sebuah permen manis dalam *stickt* yang biasanya berbentuk lingkaran atau bulat(Devi, 2019).

11. Android 6.0 *Marshmallow*



Gambar 2.13 Android versi *Marshmallow*.

Pertama kali dikenalkan pada may 2015 dengan kode name ‘Android M’ yang kemudian dirilis secara resmi pada oktober 2015. *Marshmallow* merupakan pengembangan dari android *Lollipop* salah satu kelebihanya mampu menjaga konsumsi baterai yang lebih hemat(Devi, 2019).

12. Android 7.0 *Nougat*



Gambar 2.14 Android versi *Nougat*

Nougat adalah makanan ringan dari campuran gula dan putih telur dengan kacang di dalamnya. Makanan ringan tersebut berhasil menyingkirkan berbagai penganan manis dengan awalan huruf 'N' lain yang sempat diajukan oleh masyarakat ke Google(Devi, 2019).

13. Android Versi 8.0 *Oreo*



Gambar 2.15 Android Versi *Oreo*

Tampilan Android *Oreo* sangat berbeda dengan pendahulunya. Tampilan *UI* di versi *Oreo* kini lebih rapi dan segar. Tampilannya pun lebih difokuskan untuk memudahkan pengguna mengakses aplikasi dan mencari informasi yang dibutuhkan. Tersedia juga fitur baru seperti *Notification Dots*, *Picture in Picture*, *Autofill* untuk memudahkan isi formulir online, emoji, dan masih banyak lagi (Devi, 2019).

14. Android Versi 9.0 *Pie*



Gambar 2.16 Android Versi *Pie*

Versi Android terbaru dengan nama kue *Pie* membawa lompatan baru dalam sejarah sistem operasi Hp. Android *Pie* sudah didukung kemampuan kecerdasan buatan (AI). Berkat *AI*, Hp Android jadi semakin pintar karena bisa mempelajari pola penggunaan secara otomatis. Android *Pie* bisa menyesuaikan konsumsi baterai dengan rutinitasmu sehari-hari. Aplikasi yang ditampilkan di halaman utama pun bisa disesuaikan dengan aplikasi apa yang paling sering digunakan dan bermanfaat saat menjalankan aktivitas

sehari-hari. Android *Pie* juga bisa menganalisa waktu pemakaian Hp dan mengingatkanmu jika terlalu lama menggunakan Hp (Devi, 2019).

2.1.4 Pengertian Basis Data

Basis data terdiri dari 2 kata, yaitu basis dan data. Basis diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, mahasiswa, pembeli, pelanggan) barang, hewan, angka, symbol, huruf, teks, gambar, bunyi atau kombinasinya (Faizal Ari Prabowo, 2017).

2.1.5 Pengertian Android Studio

Android Studio adalah *Integrated Development Environment (IDE)* untuk sistem operasi Android, yang dibangun diatas perangkat lunak *JetBrains IntelliJ IDEA* dan didesain khusus untuk pengembangan Android. *IDE* ini merupakan pengganti dari *Eclipse Android Development Tools (ADT)* yang sebelumnya merupakan *IDE* utama untuk pengembangan aplikasi android, Android studio sendiri pertama kali diumumkan di *Google I/O conference* pada tanggal 16 Mei 2013.



Gambar 2.18 Logo Android Studio

2.1.6 Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Java merupakan bahasa pemrograman yang sangat sensitif dimana huruf besar dan kecil pada sintak java harus dibedakan. Saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi.

2.1.7 PHP (Hypertext Preprocessor)

PHP adalah bahasa pemrograman yang sering disisipkan kedalam bahasa pemrograman *HTML*, bahasa pemrograman ini menggunakan sistem *server-side*. *Server-side programming* adalah jenis bahasa pemrograman yang nantinya script/program tersebut akan dijalankan/diproses oleh server. Kelebihannya adalah mudah digunakan, sederhana, dan mudah untuk dimengerti dan dipelajari. Penulisan *script PHP* selalu diawali dengan tanda `<?php` dan di akhiri dengan tanda `?>` (Chandra,2016).

```
<html>
<head>
<title>Menampilkan Data Buku</title>
</head>
<body>
<table width="450" border="1" cellspacing="1" cellpadding="2">
<tr>
<td width="31"><strong>NO</strong></td>
<td width="289"><strong>JUDUL BUKU </strong></td>
<td width="106"><strong>PENERBIT</strong></td>
</tr>
<?php
include "inc.koneksidb.php";
$sql = "SELECT * FROM databuku ORDER BY kd_buku";
$qry = mysql_query($sql, $koneksi)
or die ("SQL Error: ".mysql_error());
while ($data=mysql_fetch_array($qry)) {
?>
<tr>
<td> <?php echo $data['kd_buku']; ?> </td>
<td> <?php echo $data['judul']; ?> </td>
<td> <?php echo $data['penerbit']; ?> </td>
</tr>
<?php
}
?>
</table>
</body>
</html>
```

Gambar 2.19 Script PHP

2.1.8 Sublime Text

Sublime text adalah sebuah aplikasi yang digunakan untuk menulis kode-kode, seperti *HTML*, *PHP*. Ketika kita menuliskan kode-kode *HTML* pada *sublime text* maka dapat menghasilkan output berupa website.

2.1.9 My SQL

My sql adalah salah satu jenis *database server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi yang menggunakan *database* sebagai

sumber dan pengolahan datanya. *My sql* merupakan database yang pertama kali didukung oleh bahasa pemrograman script untuk internet (*PHP dan Perl*). *My sql* lebih sering digunakan untuk membangun aplikasi berbasis web. Umumnya pengembangan aplikasinya menggunakan bahasa pemrograman script PHP. *XAMPP, Apache, My Sql, PHP dan Perl* merupakan paket *server web PHP* dan *database My Sql* yang paling populer (Agustina, 2018).

2.1.10 Rational Rosse.

Rational Rose adalah *software* yang memiliki perangkat-perangkat pemodelan secara visual untuk membangun suatu solusi dalam rekayasa software dan pemodelan bisnis. *design proses, generate code, reverse engineering*, serta banyak fungsi-fungsi yang lain. *Rational Rose* merupakan *tools* yang sangat mudah karena menyediakan contoh-contoh *design* dari beberapa *software*.

2.2 Rational Unified Process (RUP)

Unified process atau dikenal dengan proses *iteratif* dan *incremental* merupakan sebuah proses pengembangan perangkat lunak yang dilakukan secara iteratif (berulang) dan inkremental (bertahap dengan proses membaik). *Iteratif* bisa dilakukan didalam setiap tahap, atau *iteratif* tahap pada proses pengembangan perangkat lunak untuk menghasilkan perbaikan fungsi yang inkremental (bertambah menaik) dimana setiap iterasi akan memperbaiki iterasi berikutnya. Salah satu *Unified process* yang terkenal adalah RUP (*Rational Unified Process*)(Shalahudin, Rosa(2015,p.124)).

RUP (*Rational Unified Process*) adalah pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang (*iterative*), fokus pada arsitektur, lebih diarahkan berdasarkan penggunaan kasus. RUP merupakan proses rekayasa perangkat lunak dengan pendefinisian yang baik dan perstrukturannya yang baik (Chandra,2016). Proses pengulangan / *iteratif* pada RUP secara global dapat dilihat pada gambar berikut:



Gambar 2.20 Proses *Iteratif* RUP

2.2.1 Fase RUP

RUP memiliki empat tahap atau *fase* yang dapat dilakukan pula secara *iteratif*. Berikut ini adalah fase dan alur hidup RUP:



Gambar 2.21 *fase* dan alur hidup RUP

Berikut ini penjelasan untuk setiap fase pada RUP:

1. *Inception* (Permulaan)

Tahap ini lebih pada memodelkan proses bisnis yang dibutuhkan (*business modeling*) dan mendefinisikan kebutuhan sistem yang akan dibuat (*requirements*). Hal yang dibutuhkan pada tahapan ini adalah:

- a. Memahami ruang lingkup dari proyek (biaya, waktu, kebutuhan, resiko, dan lain sebagainya).
- b. Membangun kasus bisnis yang dibutuhkan.

2. *Elaboration* (Perluasaan/ Perencanaan)

Tahapan ini lebih difokuskan pada perencanaan arsitektur sistem. Tahapan ini juga harus dapat mendeteksi apakah arsitektur sistem yang diinginkan dapat dibuat atau tidak. Tahapan ini lebih pada analisis dan desain sistem serta implementasi yang fokus pada sistem *prototype* (Chandra, 2016).

3. *Construction* (konstruksi)

Tahapan ini fokus pada pengembangan komponen dan fitur-fitur sistem. Tahapan ini lebih pada implementasi dan pengujian sistem yang fokus pada implementasi perangkat lunak pada kode program. Tahap ini menghasilkan sebuah produk perangkat lunak dimana menjadi syarat dari batas/tongkat kemampuan operasional(Chandra,2016).

4. *Transition* (transisi)

Tahap ini lebih pada *deployment* atau instalasi sistem agar dapat dimengerti oleh user. Tahap ini menghasilkan sebuah perangkat lunak dimana menjadi syarat atau batas atau tongkat kemampuan operasional awal. Aktifitas pada tahap ini termasuk pelatihan user, pemeliharaan dan pengujian sistem apakah sudah memenuhi harapan user(Chandra,2016).

2.2.2 Kelebihan RUP

Pendekatan iteratif /pengulangan dari RUP dapat mengakomodir beberapa kelemahan pengembangan perangkat lunak tanpa menggunakan konsep

pengulangan, misalnya pada pengembangan perangkat lunak *waterfall*, berikut adalah hal-hal yang dapat diatasi oleh RUP dibandingkan *waterfall*:

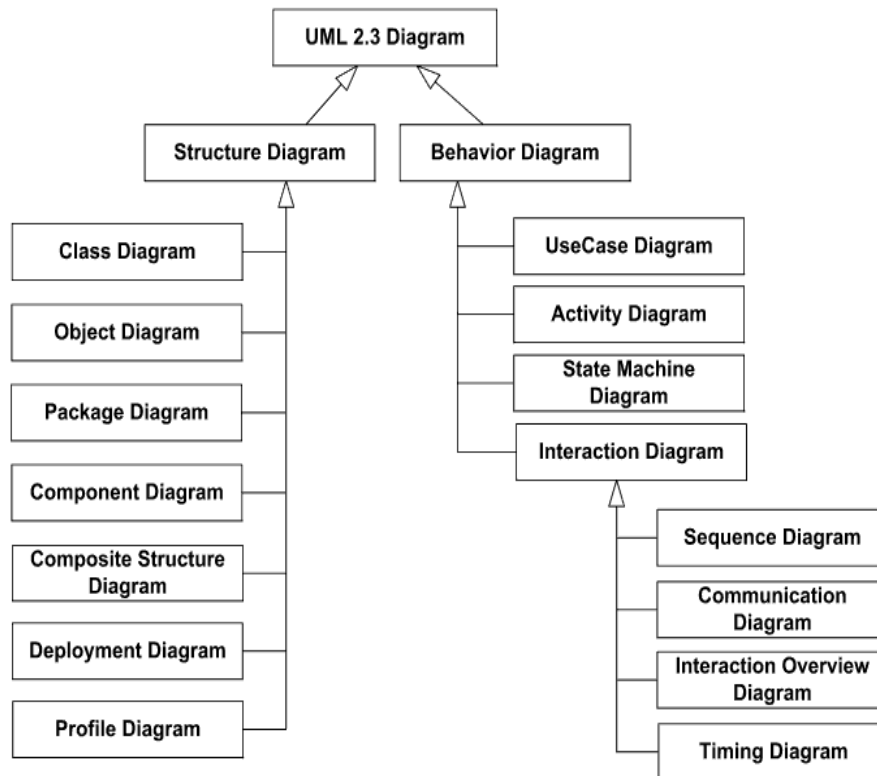
1. RUP mengakomodasikan perubahan kebutuhan perangkat lunak.
2. Integrasi bukanlah sebuah proses besar dan cepat (“*big bang*”) diakhir proyek.
3. Resiko biasanya ditemukan atau dialamatkan selama integrasi awal.
4. Manajemen berarti merubah taktik pada produk.
5. Mendukung fasilitas penggunaan kembali.
6. Kecacatan dapat ditemukan dan diperbaiki pada iterasi menghasilkan arsitektur yang baik dan aplikasi berkualitas tinggi.
7. Lebih baik menggunakan “anggota proyek” dibandingkan susunan secara seri pada tim proyek.
8. Anggota tim belajar selama proyek berjalan.
9. Pengembangan perangkat lunak dapat diperbaiki seiring proses pengembangan perangkat lunak.

2.3 Unified Modeling Language(UML)

UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. *UML* hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan *UML* tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya *UML* terbaru adalah *UML 2.3* yang terdiri dari 4 macam spesifikasi, yaitu *Diagram Interchange Specification*, *UML superstructure*, dan *Object Constraint Language (OCL)*(Chandra,2016).

2.3.1 Diagram UML

Pada *UML 2.3* terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar berikut.



Gambar 2.22 Diagram UML

Berikut ini penjelasan singkat dari pembagian kategori tersebut:

1. *Structure diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan .
2. *Behavior diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem yang lain maupun interaksi antar subsistem pada suatu sistem(Chandra,2016).

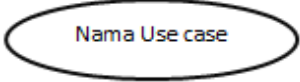
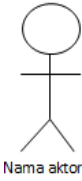

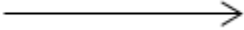
Berikut penjelasan dari diagram yang sering dipakai pada pemodelan:

1. Use Case Diagram

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih dari aktor sistem yang akan dibuat. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

- a. Aktor merupakan orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi itu sendiri.
- b. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor simbol-simbol yang ada pada use case dapat dilihat pada tabel 2.2(Chandra,2016):

Tabel 2.1 Simbol-simbol *Use Case Diagram*

Simbol	Deskripsi
Use Case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.
Aktor / <i>Actor</i> 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri (Chandra,2016).
Asosiasi / <i>Association</i> 	Komunikasi antara aktor dan <i>use case</i> yang saling berpartisipasi, atau <i>use case</i> memiliki interaksi dengan aktor(Chandra,2016).
Ekstensi/ <i>Extend</i> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri walau tanpa tambahan <i>use case</i> tambahan itu(Chandra,2016).

Tabel 2.1 Simbol-simbol *Use Case Diagram* lanjutan

Simbol	Deskripsi
<p>Menggunakan <i>/include/uses</i></p> <p style="text-align: center;"> </p>	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya. di <i>use case</i>:</p> <ul style="list-style-type: none"> • Include bearti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan, • Include bearti <i>use case</i> yang ditambah akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan sudah dijalankan sebelum <i>use case</i> tambahan dijalankan (Chandra,2016).

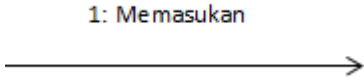
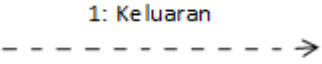
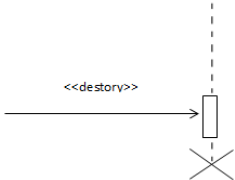

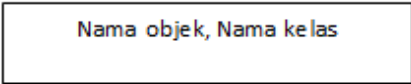
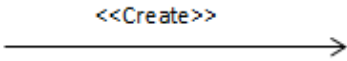
2. Sequence diagram

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan diterima antar objek. Oleh karena itu untuk menggambarkan diagram skuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki dan diinstansiasi menjadi objek itu (Chandra,2016). simbol-simbol yang ada pada diagram sekuen dapat dilihat pada tabel 2.3 berikut:

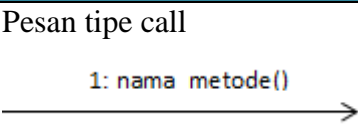
Tabel 2.2 Simbol-simbol pada *sequence diagram*

Simbol	Deskripsi
<p>Aktor</p> <p style="text-align: center;"> </p>	<p>Orang, proses atau sistem yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri(Chandra,2016).</p>

Tabel 2.2 Simbol-simbol pada *sequence diagram* lanjutan

Simbol	Deskripsi
<p>Pesan tipe <i>send</i></p> 	<p>Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim (Chandra,2016).</p>
<p>Pesan tipe <i>return</i></p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan sesuatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian (Chandra,2016).</p>
<p>Pesan tipe <i>destory</i></p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, (Chandra,2016).</p>
<p><i>Garis hidup/ Lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek</p>
<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi pesan</p>
<p>Pesan tipe <i>create</i></p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat</p>

Tabel 2.2 Simbol-simbol pada *sequence diagram* lanjutan

Simbol	Deskripsi
	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri, (Chandra,2016).

2.4 Pengujian Perangkat Lunak

Pengujian perangkat lunak perlu dilakukan tidak hanya untuk meminimalisasi kesalahan secara teknis tetapi juga kesalahan non teknis (contohnya: pengujian pesan kesalahan sehingga user tidak bingung atau tidak mengerti dengan pesan kesalahan yang muncul, atau juga jika masukan dan keluaran yang diperlukan berkapasitas sangat besar (Chandra,2016).

2.4.1 Teknik Pengujian Perangkat lunak

Ada dua macam pendekatan kasus uji coba perangkat lunak yaitu *white-box* dan *black-box* pendekatan *white-box* adalah pengujian dari segi desain dan kode program apakah mampu menghasilkan fungsi-fungsi, masukan dan keluaran yang sesuai dengan kebutuhan. Pengujian *white-box* dilakukan dengan spesifikasi memeriksa logik dari kode program. Pembuatan kasus uji bisa mengikuti standar pengujian dari standar program yang seharusnya(Chandra,2016).

Metode *black-box testing* yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk apakah fungsi-fungsi masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Pengujian *black-box* dilakukan dengan membuat kasus uji yang bersipat mencoba semua fungsi dengan mencoba dengan perangkat lunak sesuai dengan spesifikasi yang dibutuhkan(Chandra, 2016)