

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Teori Umum**

Dalam penulisan tugas akhir skripsi terdapat beberapa teori umum yang penulis jadikan sebagai landasan. Di bawah ini adalah penjelasan teori tersebut:

##### **2.1.1. Sistem**

Menurut Astuti (2013), terdapat dua kelompok pendekatan dalam mendefinisikan sistem, yaitu pendekatan pada prosedur dan pendekatan pada komponen/elemen.

Pendekatan sistem pada prosedurnya mendefinisikan sistem sebagai berikut: “Suatu sistem merupakan suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan untuk melakukan suatu kegiatan atau tujuan tertentu”.

Sedangkan pendekatan sistem pada komponen atau elemennya mendefinisikan sistem sebagai berikut: “Suatu sistem merupakan bagian-bagian elemen yang saling berinteraksi dan saling berhubungan untuk mencapai membentuk satu kesatuan”.

##### **2.1.2. Informasi**

Informasi merupakan hasil pemrosesan data (fakta) menjadi sesuatu yang bermakna dan bernilai untuk pengambilan keputusan (Wongso, 2016).

Informasi adalah data yang telah diolah menjadi suatu bentuk yang penting dan mempunyai nilai nyata yang dapat dirasakan dalam pengambilan keputusan-keputusan yang sekarang atau keputusan-keputusan yang akan datang (Andi Muh. Lukman, 2016).

Informasi merupakan hasil pengolahan data sehingga menjadi bentuk yang penting untuk penerima dan memiliki kegunaan dalam pengambilan keputusan yang dapat dirasakan akibatnya secara langsung maupun secara tidak langsung. Untuk memperoleh informasi, diperlukan adanya data yang akan diolah dan unit pengolah (Astuti, 2013).

### **2.1.3. Sistem Informasi**

Sistem informasi adalah sebuah sistem di dalam sebuah organisasi/perusahaan yang berguna untuk mempertemukan kebutuhan pengolahan transaksi yang mendukung fungsi operasi manajerial organisasi dengan kegiatan strategi dari suatu organisasi untuk menghasilkan laporan-laporan yang dapat disediakan kepada pihak luar tertentu (Khairullah, Soedijono, & Hanif Al Fatta, 2017).

Menurut Riswaya (2013) Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi yang terjadi, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu sistem organisasi dan menyediakan pihak luar tertentu akan laporan-laporan yang diperlukan.

Sistem informasi adalah kumpulan elemen yang saling berhubungan dan berfungsi untuk memproses, mengumpulkan, mendistribusikan serta menyimpan informasi yang berguna sebagai pendukung dalam pembuatan keputusan juga pengawasan dalam organisasi (Anthony et al., 2017).

#### **2.1.4. Penjualan**

Penjualan adalah suatu usaha yang terpadu untuk mengembangkan rencana-rencana strategis yang diarahkan pada usaha pemuasan kebutuhan dan keinginan pembeli, guna mendapatkan penjualan yang menghasilkan laba. Penjualan merupakan sumber hidup suatu perusahaan, karena dari penjualan dapat diperoleh laba serta suatu usaha memikat konsumen yang diusahakan untuk mengetahui daya tarik mereka sehingga dapat mengetahui hasil produk yang dihasilkan (S. Christopel M. A. H, Tutut Wuriyanto, 2013).

Menurut Sri, Rejeki, Utomo, & Susanti (2011), penjualan adalah suatu sistem keseluruhan dari suatu kegiatan usaha yang bertujuan untuk merencanakan, menentukan harga, mempromosikan dan mendistribusikan barang, jasa, ide kepada pasar sasaran agar dapat mencapai tujuan organisasi.

#### **2.1.5. Smartphone**

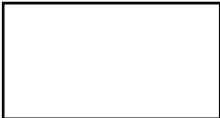
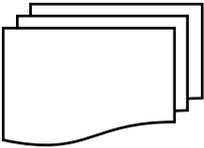
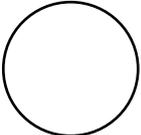
Menurut Marsal & Hidayati pada tahun 2017, *smartphone* adalah telepon yang bisa dipakai internetan yang biasanya menyediakan fungsi *Personal Digital Assistant* (PDA), seperti fungsi kalender, buku agenda, buku alamat, kalkulator, dan catatan.

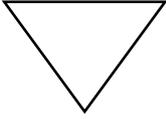
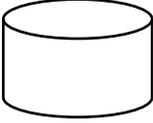
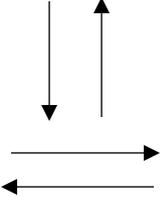
### 2.1.6. Aliran Sistem Informasi (ASI)

Merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya (Iswandy, 2015).

Aliran sistem informasi merupakan bagan alir yang menunjukan arus dari program dan formulir termasuk tembusan- tembusannya. Aliran sistem informasi menggunakan simbol-simbol dalam menggambarkan aliran sistem informasi (Putri, 2015).

**Table 2.1** Simbol-Simbol Pada Aliran Sistem Informasi

Gambar	Keterangan	Fungsi
	Simbol proses	Menunjukkan proses.
	Simbol <i>alternative</i>	Menunjukkan <i>alternative</i> .
	Simbol multi dokumen	Menunjukkan dokumen input dan output untuk proses manual, mekanik atau komputer.
	Simbol kegiatan manual	Menunjukkan perkerjaan manual.
	Simbol penghubung	Menunjukkan penghubung dalam satu halaman.

	Simbol simpanan <i>offline</i> atau simpanan mati	Menunjukkan simpan data secara manual.
	Simbol <i>harddisk</i> / <i>database</i>	Menunjukkan penyimpanan data secara elektronik.
	Simbol garis alir	Menunjukkan arus dari proses.

Sumber: Putri, 2015

### 2.1.7. Website

*Website* adalah sekumpulan dokumen yang berada pada *server* dan dapat dilihat oleh user dengan menggunakan browser. Dokumen itu bisa terdiri dari beberapa halaman. Setiap halaman memberi informasi maupun interaksi yang beragam sesuai isinya. Informasi dan interaksi tersebut dapat berupa tulisan, gambar atau bahkan dalam bentuk video yang dapat ditampilkan, suara, animasi, dan lain-lain (Hendrawan, Winardi, & Surbakti, 2014).

*Website* merupakan kumpulan halaman-halaman yang berisi informasi yang tersimpan di internet. Halaman-halaman tersebut dapat diakses atau dilihat melalui jaringan internet pada perangkat-perangkat yang bisa terhubung dengan internet seperti komputer (Puji Hastanti, Eka Purnama, & Uly Wardati, 2015).

### **2.1.8. Basis Data (*Database*)**

Basis data merupakan suatu susunan atau kumpulan data operasional lengkap dari suatu badan organisasi/perusahaan yang terorganisir dan disimpan secara terintegrasi dengan metode tertentu dan komputer sehingga mampu menyediakan informasi yang optimal yang diperlukan oleh pemakainya (Andi Muh. Lukman, 2016).

*Database* adalah sekumpulan data/informasi yang teratur berdasarkan kriteria tertentu yang saling berhubungan. Dalam dunia komputer bisa dikategorikan sangat spesial karena selalu menjadi hal utama dalam perancangan sistem komputer suatu perusahaan (Putri, 2015).

## **2.2 Teori Khusus**

### **2.2.1. Sistem Penjualan**

Sistem penjualan adalah sebuah sistem yang melibatkan antara sumber daya dalam suatu organisasi/perusahaan, prosedur, data, serta sarana pendukung lainnya untuk mengoperasikan sistem penjualan, sehingga menghasilkan informasi yang bermanfaat bagi pihak manajemen dalam pengambilan keputusan (S. Christopel M. A. H, Tutut Wuriyanto, 2013).

### **2.2.2. API (*Application Programming Interface*)**

API adalah sebuah aplikasi pemrograman yang secara khusus dikembangkan untuk digunakan sebagai perantara komunikasi antara komponen-komponen

perangkat lunak. Manfaat dan kelebihan API adalah membuat semuanya menjadi lebih cepat, tepat, fleksibel (Prasetyo, Triyono, & Andayati, 2016).

*Application Programming Interface* (API) merupakan seperangkat instruksi pemrograman untuk mengakses suatu aplikasi berbasis *web software* atau *web tool*. Sebuah organisasi/perusahaan perangkat lunak dapat merilis sebuah API kepada publik sehingga pengembang perangkat lunak lain dapat merancang/membangun produk yang didukung oleh layanannya. API menentukan bagaimana komponen perangkat lunak tersebut saling berinteraksi. API juga dapat didefinisikan sebagai sebuah bahasa yang digunakan oleh sebuah program aplikasi dalam berkomunikasi dengan OS (*Operating System*)/sistem operasi atau program kontrol lainnya seperti *Database Management System* atau protokol komunikasi (Rachmadi, Hamzah, & Raharjo, 2017).

### **2.2.3. PHP**

*Script* PHP adalah bahasa pemrograman yang berjalan pada sebuah *webserver*, atau sering disebut *server-side*. PHP dapat mengolah data dengan tipe data apapun, menciptakan sebuah halaman web yang dinamis, serta menerima dan juga menciptakan *cookies*, dan bahkan lebih dari itu. Pengertian *script server-side* adalah untuk dapat menjalankan *script* ini dibutuhkan tiga program utama, yaitu *webserver* (dapat berupa IIS dari windows atau apache), modul PHP dan juga *web browser*. PHP bisa dijalankan di banyak jenis sistem operasi, antara lain pada Linux dan varian Unix (HP-UX, Solaris dan OpenBSD), pada Microsoft Windows, Mac OS dan banyak lagi. Selain itu PHP juga mampu berjalan pada beberapa jenis

*webserver* antara lain Apache, Microsoft IIS, personal *webserver*, Netscape dan Iplanet *Server*, Caudium, Xitami, Omnihttpd dan lain-lain (Irsan, 2015).

#### **2.2.4. HTML (*Hypertext Markup Language*)**

HTML adalah kependekan dari *Hyper Text Markup Language*. Dokumen HTML merupakan file teks yang murni dapat dibuat dengan sembarang editor teks dan dikenal sebagai *web page*. Dokumen HTML disajikan dalam suatu *browser web surfer*, pada umumnya berisi informasi ataupun *interface* aplikasi di dalam internet (Prasetyo et al., 2016).

HTML merupakan Bahasa pemrograman yang fleksibel dimana seperti *script* dari bahasa pemrograman lain seperti JAVA, Visual Basic, C dan lain-lain dapat diletakkan didalamnya. Ketika HTML tidak mendukung suatu perintah atau bahasa pemrograman tertentu, *browser* tidak akan menampilkan kotak *dialog* “*Syntax Error*” apabila terdapat kesalahan penulisan kode pada *script* HTML sepanjang kode-kode yang dituliskan merupakan kode-kode HTML tanpa penambahan kode-kode dari luar seperti Java. Maka itu, apabila terjadi *syntax error* pada *script* HTML, efek yang sangat jelas adalah HTML tersebut tidak akan ditampilkan pada halaman jendela *browser* (Lestanti & Susana, 2016).

#### **2.2.5. CSS (*Cascading Style Sheet*)**

*Cascading Style Sheet* (CSS) adalah *template* yang mengontrol pemformatan *tag* HTML pada sebuah halaman web. Konsep CSS mirip dengan *template* pada Microsoft Word. CSS mampu mengubah tampilan dokumen Microsoft Word

dengan mengubah format pada *style* dokumen. Begitu juga CSS dapat mengubah tampilan halaman web dengan mengubah format pada *tag* HTML tertentu melalui CSS, untuk selanjutnya mengganti spesifikasi *default* dari *browser* web untuk *tag-tag* tersebut (Raharjo, 2015).

CSS (*Cascading Style Sheet*) merupakan salah satu bahasa pemrograman yang digunakan untuk membangun sebuah web dan juga mengendalikan berbagai komponen dalam sebuah web sehingga tampilan web akan terlihat lebih rapi, terstruktur dan seragam.

CSS saat ini masih dikembangkan oleh *World Wide Web Consortium* (W3C) yang menjadi Bahasa standar dalam pembuatan web. CSS difungsikan sebagai penopang atau pendukung, dan pelengkap dari file html yang berperan dalam penataan kerangka dan *layout*.

CSS mampu berjalan di berbagai macam sistem operasi dan *web browser*. CSS secara umum melakukan pengaturan pada *layout*, kerangka, teks, warna, gambar, tabel, spasi, dan lain sebagainya (Chulkamdi & Purnomo, 2016).

#### **2.2.6. JavaScript**

JavaScript adalah bahasa pemrograman yang berbasis prototipe yang berjalan di *client-side*. JavaScript merupakan sebuah bahasa *scripting* yang dikembangkan oleh perusahaan ternama Netscape. Untuk menjalankan script yang ditulis dengan JavaScript kita membutuhkan *JavaScript-enabled browser* yaitu *browser* yang mampu menjalankan JavaScript. Secara fungsional, JavaScript berguna dalam menyediakan akses *script* pada objek yang dibenamkan (*embedded*). Contoh

sederhana dari penggunaan JavaScript adalah membuka halaman *pop up*, fungsi validasi pada suatu *form* HTML sebelum data dikirimkan ke *server*, merubah *image* kursor ketika melewati objek tertentu, dan lain-lain (Chulkamdi & Purnomo, 2016).

JavaScript adalah kumpulan kode-kode program kecil yang dapat digunakan untuk membuat halaman web terlihat lebih dinamis. Dengan menggunakan JavaScript, banyak fitur yang dapat ditambahkan yang dapat membuat tampilan lebih menarik dan juga dapat membatasi aksi dari pengguna. Menu navigasi yang lebih canggih dan efek grafis sederhana dapat dilakukan menggunakan JavaScript (Raharjo, 2015).

### **2.2.7. MySQL**

MySQL adalah sebuah *server database open source* terkenal yang banyak digunakan berbagai aplikasi terutama untuk *server* atau membuat *webservice*. MySQL berfungsi sebagai *Structured Query Language (SQL)* yang dimiliki sendiri dan sudah diperluas oleh MySQL yang umumnya digunakan bersamaan dengan PHP untuk membuat aplikasi *server* yang dinamis dan *powerfull*. Kehandalan suatu sistem basis data (DBMS) dapat diketahui dari cara kerja dalam menjalankan proses perintah-perintah SQL yang telah dibuat. Sebagai pengoperasi basis data, MySQL mendukung operasi basis data transaksional maupun operasi basis data non-transaksional. Pada modus operasi non-transaksional, MySQL dapat dikatakan unggul dalam hal unjuk kerja dibanding dengan perangkat lunak pengoperasi basis data kompetitor lainnya. Namun demikian pada modus non-transaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan. Oleh karena itu, modus non-

transaksional lebih cocok untuk jenis aplikasi yang tidak membutuhkan data sensitif atau reliabilitas data seperti aplikasi blogging berbasis web (Wordpress), CMS, dan sejenisnya. Untuk sistem yang bertujuan bisnis sangat disarankan untuk menggunakan modus basis data transaksional, hanya saja kinerja modus transaksional tidak secepat kinerja pada modus non-transaksional (Irsan, 2015).

### **2.2.8. Android**

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, middleware dan aplikasi. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka (Harianto & Khery, 2017).

Android 6: Marshmallow kini sudah tersedia untuk perangkat-perangkat Nexus dan diprediksi akan tersedia untuk perangkat-perangkat utama sebelum penghujung musim serta untuk perangkat-perangkat lainnya pada pertengahan 2016. Marshmallow menyingkap beberapa modifikasi yang memiliki pengaruh besar. Model otorisasi App kini berupa *opt-in* (memberikan otorisasi yang spesifik sesuai permintaan), berbalik dengan *opt-out* (semuanya diberi otorisasi lalu menggunakan App Ops untuk memberikan izin bagi pihak tertentu). Mode doze memungkinkan perangkat untuk melakukan hibernasi saat sedang tidak aktif, memutus daya listrik secara bertahap hingga tidak ada sama sekali. Unit pendukung sensor sidik jari kini telah tersemat didalam sistem operasi tanpa campur tangan dari *vendor* dan USB C kini telah didukung sepenuhnya. Selain itu, Marshmallow memungkinkan pengguna untuk melakukan format micro SD card selayaknya

internal storage dan berbagi tingkat keamanan internal yang sama persis (Khanna & Singh, 2016).

### **2.2.9. QR Code**

Menurut Nugraha & Munir (2011), *QR Code* adalah image dua dimensi yang merepresentasikan suatu data, terutama data berbentuk teks. *QR Code* merupakan evolusi dari *barcode* yang awalnya satu dimensi menjadi dua dimensi. *QR Code* memiliki kemampuan menyimpan data yang lebih jauh besar daripada *barcode*.

*QR Code* merupakan teknik yang mengubah data tertulis menjadi kode-kode 2-dimensi yang tercetak ke dalam suatu media yang lebih ringkas. *QR Code* adalah *barcode* 2-dimensi yang diperkenalkan pertama kali oleh perusahaan Jepang Denso-Wave pada tahun 1994 (Julizal, Lukman, & Sunoto, 2018).

QR Code disingkat dari *Quick Response Code* merupakan merek dagang untuk jenis *barcode* matriks (atau kode batang dua dimensi) yang pertama dirancang untuk industri otomotif di Jepang. Kode batang adalah label yang dapat dibaca mesin yang mencatat informasi terkait dengan item tersebut (Masalha & Hirzallah, 2014).

### **2.2.10. Java**

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995 (Irsan, 2015).

Menurut Nyura (2010), java adalah bahasa pemrograman yang disusun oleh James Gosling yang dibantu oleh rekan- rekannya seperti Patrick Naughton, Chris Warth, Ed Frank dan Mike Sheridan di suatu perusahaan perangkat lunak yang bernama Sun Microsystem pada tahun 1991. Kemudian pada perkembangan selanjutnya adalah Java dibagi menjadi tiga buah yaitu: J2SE, J2EE dan J2ME. J2ME merupakan sebuah kombinasi yang terbentuk antara sekumpulan interface Java yang sering disebut dengan Java API dengan JVM yang didesain khusus untuk alat (dalam hal ini *mobile device*).

#### **2.2.11. UML**

Pada perkembangan teknologi perangkat lunak, diperlukan adanya bahasa yang digunakan untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standarisasi agar orang di berbagai negara dapat dapat mengerti pemodelan perangkat lunak. Seperti yang kita ketahui bahwa menyatukan banyak kepala untuk menceritakan sebuah ide dengan tujuan untuk memahami hal yang sama tidaklah mudah, oleh karena itu diperlukan sebuah bahasa pemodelan perangkat lunak yang dapat dimengerti oleh banyak orang (Rosa A.S dan M. Shalahudin, 2011).

Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML). UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk

pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung (Rosa A.S dan M. Shalahudin, 2011).

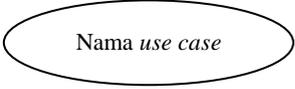
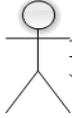
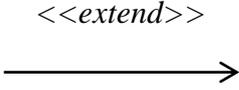
#### **2.1.11.1 Use Case Diagram**

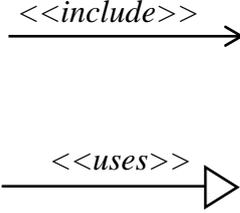
Menurut Rosa A.S dan M. Shalahudin (2011), *use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. Diagram *use case* digunakan untuk mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, diagram *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor itu orang, tapi aktor belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antarunit atau aktor.

**Table 2.2** Simbol-Simbol Pada *Use Case* Diagram

Simbol	Deskripsi
<p data-bbox="316 383 435 416"><i>Use case</i></p> 	<p data-bbox="810 383 1356 712">Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>.</p>
<p data-bbox="316 752 488 786">Aktor / <i>actor</i></p>  <p data-bbox="501 898 632 931">Nama aktor</p>	<p data-bbox="810 752 1356 1010">Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.</p>
<p data-bbox="316 1046 600 1079">Asosiasi / <i>association</i></p> 	<p data-bbox="810 1046 1356 1227">Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>
<p data-bbox="316 1270 539 1303">Ekstensi / <i>extend</i></p> 	<p data-bbox="810 1270 1356 1675">Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek.</p>
<p data-bbox="316 1711 691 1744">Generalisasi / <i>generalization</i></p> 	<p data-bbox="810 1711 1356 1968">Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p>

Simbol	Deskripsi
<p>Menggunakan / <i>include</i> / <i>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> ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini</p> <p>Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i>:</p> <p><i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan</p> <p><i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan.</p> <p>Kedua interpretasi diatas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>

**Sumber:** Rosa A.S dan M. Shalahudin, 2011

### 2.1.11.1 Activity Diagram

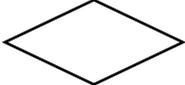
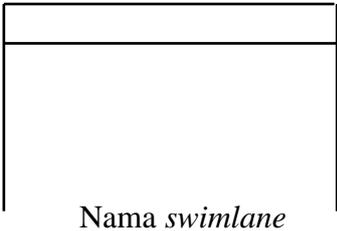
Diagram aktivitas atau *activity* diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Hal yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dilakukan oleh sistem (Rosa A.S dan M. Shalahudin, 2011).

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut (Rosa A.S dan M. Shalahudin, 2011):

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokan tampilan dari sistem/*user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.

**Table 2.3** Simbol-Simbol Pada *Activity* Diagram

Simbol	Deskripsi
Status awal  	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.

Simbol	Deskripsi
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

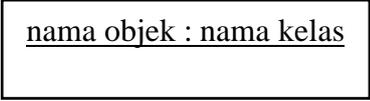
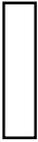
Sumber: Rosa A.S dan M. Shalahudin, 2011

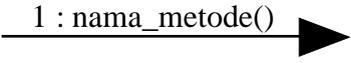
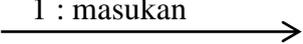
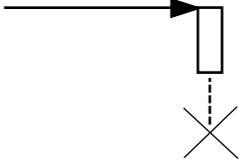
### 2.1.11.2 Sequence Diagram

Menurut Rosa A.S dan M. Shalahudin (2011), diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena

itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

**Table 2.4** Simbol-Simbol Pada *Sequence* Diagram

Simbol	Deskripsi
Aktor  Nama aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri
Garis hidup / <i>lifeline</i> 	Menyatakan kehidupan suatu objek
Objek  <u>nama objek : nama kelas</u>	Menyatakan objek yang berinteraksi pesan
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan.
pesan tipe <i>create</i> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.

Simbol	Deskripsi
<p data-bbox="312 387 507 421">pesan tipe <i>call</i></p> 	<p data-bbox="855 387 1410 566">Menyatakan suatu objek operasi memanggil/metode yang ada pada objek lain atau dirinya sendiri.</p>
<p data-bbox="312 611 520 645">Pesan tipe <i>send</i></p> 	<p data-bbox="855 611 1410 857">Menyatakan bahwa suatu objek mengirim data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.</p>
<p data-bbox="312 907 539 940">Pesan tipe <i>return</i></p> 	<p data-bbox="855 907 1410 1238">Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.</p>
<p data-bbox="312 1272 555 1305">Pesan tipe <i>destroy</i></p> 	<p data-bbox="855 1272 1410 1597">Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>.</p>

Sumber: Rosa A.S dan M. Shalahudin, 2011

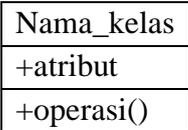
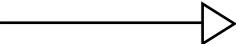
### 2.1.11.3 Class Diagram

Diagram kelas atau *class* diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas

memiliki apa yang disebut atribut dan metode atau operasi (Rosa A.S dan M. Shalahudin, 2011).

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

**Table 2.5** Simbol-Simbol Pada *Class* Diagram

Simbol	Deskripsi
Kelas  	Kelas pada struktur sistem.
Antarmuka / <i>interface</i>  	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
Asosiasi / <i>association</i>  	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Asosiasi beralih / <i>directed association</i>  	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
generalisasi  	Relasi antar kelas dengan makna generalisasi – spesialisasi (umum – khusus).

Simbol	Deskripsi
Kebergantungan / <i>dependency</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas.
Agregasi / <i>aggregation</i> 	Relasi antar kelas dengan makna semua-bagian ( <i>whole-part</i> ).

Sumber: Rosa A.S dan M. Shalahudin, 2011

### 2.3 Penelitian Terdahulu

Berikut penelitian terdahulu yang penulis gunakan sebagai dasar dalam penelitian ini.

1. Penelitian Rahayu, Nurhaeni, & Rohmah (2015) dengan judul “**Sistem Persediaan Alat Tulis Kantor Sebagai Penunjang Pengambilan Keputusan Bagian Logistik Di Perguruan Tinggi Raharja**” menggunakan metode penelitian survei dan wawancara yang menghasilkan web yang dapat memberikan pelayanan informasi yang akurat, tepat waktu, dan sesuai dengan kebutuhan pengguna.
2. Penelitian Meimaharani & Laily (2014) dengan judul “***E-Commerce Goody Bag Spunbound Menggunakan QR Code Berbasis Web Responsif***” menggunakan metode pengembangan *waterfall* dan menghasilkan sebuah *website* penjualan yang dilengkapi dengan *QR Code*.
3. Penelitian Kusuma & Kurniawan (2017) dengan judul “**Perancangan dan Impelemntasi *E-Commerce* untuk Penjualan Baju *Online* Berbasis**

**Android**” dengan metode *white box* untuk menguji *software* mengetahui cara kerja suatu perangkat lunak secara internal.

4. Penelitian yang dilakukan oleh Wongso (2016) dengan judul “**Perancangan Sistem Informasi Penjualan Berbasis Java Studi Kasus pada Toko Karya Gemilang Pekanbaru**” untuk mengatasi pencatatan transaksi penjualan, pencatatan transaksi pembelian, pencarian barang, perhitungan persediaan, dan pembuatan laporan-laporan yang memerlukan waktu yang cukup lama dan resiko kesalahan manusia yang cukup besar. Berdasarkan hal itu maka diusulkan sebuah sistem terkomputerisasi penjualan.
5. Penelitian Rachmawati, Laila, & Susana (2016) dengan judul “**Sistem Informasi Penjualan Alat Tulis Kantor Berbasis Webb Pada CV. Sumber Rezeki Jakarta**” menggunakan pemodelan UML dan ERD (*Entity Relationship Diagram*) yang menghasilkan sebuah sistem informasi berbasis web yang dapat diakses oleh orang-orang yang mempunyai akses internet dan dapat dilakukannya transaksi online.