

BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Sistem

Pengertian sistem menurut (Iswandy, 2015: 72), Sebuah sistem terdiri dari berbagai unsur yang saling melengkapi dalam mencapai tujuan dan sasaran. Unsur-unsur yang terdapat dalam sistem itulah yang disebut dengan subsistem. Subsistem-subsistem tersebut harus saling berhubungan dan berinteraksi melalui komunikasi yang relevan sehingga sistem dapat bekerja secara efektif dan efisien.

Sistem juga di definisikan sebagai kumpulan atau himpunan dari unsur, komponen atau variabel yang terorganisir, saling berinteraksi, saling bergantung satu sama lain, dan terpadu. Sebuah sistem terdiri dari bagian-bagian atau komponen yang terpadu untuk satu tujuan (Andrianof, 2018: 13).

Sedangkan pengertian sistem menurut Hamim Tohari dalam jurnal (Andrianof, 2018: 13), sistem adalah kumpulan objek-objek yang saling berelasi dan berinteraksi, serta hubungan antara objek bisa dilihat sebagai satu kesatuan yang dirancang untuk mencapai satu tujuan yang telah ditetapkan.

Menurut (Husda, 2012: 111) dalam buku yang berjudul pengantar teknologi informasi, sistem juga merupakan kesatuan bagian-bagian yang saling berhubungan yang berada dalam suatu wilayah serta memiliki item-item penggerak, contoh umum misalnya seperti negara. Negara merupakan suatu kumpulan dari beberapa elemen kesatuan lain seperti provinsi yang saling

berhubungan sehingga membentuk suatu negara dimana yang berperan sebagai penggeraknya yaitu rakyat yang berada di negara tersebut. Contoh lainnya adalah sistem komputer terdiri dari: *Software*, *Hardware*, *Brainware*, Sistem Akuntansi, dll.

2.1.1.1. Karakteristik Sistem

Sistem memiliki beberapa karakteristik, berikut ini merupakan karakteristik sistem menurut (Husda, 2012: 112) diantaranya:

a. Komponen (*component*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk satu kesatuan. Komponen-komponen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap sistem tidak peduli betapapun kecilnya, selalu mengandung komponen-komponen atau subsistem-subsistem.

b. Batas Sistem (*boundary*)

Batas sistem (*boundary*) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan, karena dengan batas sistem ini fungsi dan tugas dari subsistem yang satu dengan lainnya berbeda tetapi tetap saling berinteraksi.

c. Lingkungan Luar Sistem (*environment*)

Segala sesuatu diluar dari batas sistem yang mempengaruhi operasi dari suatu sistem. Lingkungan luar sistem ini dapat bersifat

menguntungkan atau merugikan. Lingkungan luar yang menguntungkan harus dipelihara dan dijaga agar tidak hilang pengaruhnya, sedangkan lingkungan luar yang merugikan harus dimusnahkan dikendalikan agar tidak mengganggu operasi sistem.

d. Penghubung Sistem (*interface*)

Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Untuk membentuk satu kesatuan, sehingga sumber-sumber daya mengalir dari subsistem yang satu ke subsistem yang lainnya. Dengan kata lain *output* dari suatu subsistem akan menjadi *input* dari subsistem yang lainnya.

e. Masukan Sistem (*input*)

Merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*Maintenance Input*) dan masukan sinyal (*Signal Input*).

f. Keluaran Sistem (*output*)

Merupakan hasil dari energi yang diolah oleh sistem. Meliputi keluaran yang berguna dan keluaran yang tidak berguna yang dikenal sebagai sisa pembuangan.

g. Pengolah Sistem (*process*)

Merupakan bagian yang memproses masukan untuk menjadi keluaran yang diinginkan. Contoh CPU pada komputer, bagian produksi yang mengubah bahan baku menjadi barang jadi.

h. Tujuan Sistem (*goal*)

Setiap sistem pasti mempunyai tujuan ataupun sasaran yang mempengaruhi input yang dibutuhkan dan *output* yang dihasilkan. Dengan kata lain suatu sistem akan dikatakan berhasil kalau pengoperasian sistem itu mengenai sasaran atau tujuannya. Sistem yang tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya.

2.1.2. Pengertian Informasi

Menurut Hamim Tohari dalam jurnal (Andrianof, 2018: 13), Sumber dari informasi adalah data. Data adalah fakta atau kenyataan yang menggambarkan suatu kejadian-kejadian yang mempunyai arti tersendiri. Informasi adalah data yang telah diproses sedemikian rupa, sehingga memiliki arti yang lebih bermanfaat bagi penggunanya.

Sedangkan menurut (Husda, 2012: 117), secara umum informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata yang digunakan untuk mengambil keputusan.

Menurut (Lipursari, 2013: 28), Data yang telah diklasifikasi atau diolah atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan adalah informasi. Sistem pengolahan mengolah data menjadi informasi atau tepatnya mengolah data dari bentuk tak berguna menjadi berguna bagi penerimanya. Nilai informasi berhubungan dengan keputusan. Nilai informasi dilukiskan paling berarti dalam konteks sebuah keputusan. Bila tidak ada keputusan, maka informasi menjadi tidak diperlukan. Keputusan dapat berkisar dari keputusan berulang sederhana sampai keputusan strategis jangka panjang.

Pendapat lain menurut Raymond Mcleod dalam buku (Husda, 2012: 117), Informasi merupakan data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat bagi pengambilan keputusan saat ini atau saat mendatang. Selain itu menurut Gordon B. Davis dalam buku (Husda, 2012: 117), informasi adalah data yang telah diolah menjadi suatu bentuk yang penting bagi si penerima dan mempunyai nilai yang nyata yang dapat dirasakan dalam keputusan-keputusan yang sekarang atau keputusan-keputusan yang akan datang.

Terdapat 3 kualitas informasi menurut (Husda, 2012: 118), yaitu sebagai berikut:

a. Akurat

Berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai ke penerima informasi kemungkinan terjadi gangguan yang dapat merubah atau merusak informasi tersebut.

b. Tepat Waktu

Berarti informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi. Karena informasi merupakan landasan di dalam pengambilan keputusan. Bila pengambilan keputusan terlambat, maka dapat berakibat fatal bagi organisasi.

c. Relevan

Berarti informasi tersebut mempunyai manfaat untuk pemakainya.

Relevansi informasi untuk tiap-tiap orang berbeda-beda

2.1.3. Pengertian Sistem Informasi

Menurut (Anisah, 2017: 508), Sistem Informasi adalah kombinasi teratur apa pun dari orang-orang, *hardware*, *software*, jaringan komunikasi, dan sumber daya data yang mengumpulkan, mengubah, dan menyebarkan informasi dalam sebuah organisasi.

Sedangkan menurut (Husda, 2012: 119), Sistem informasi dapat didefinisikan sebagai suatu sistem di dalam suatu organisasi yang merupakan kombinasi dari orang-orang, fasilitas, teknologi, media prosedur-prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan eksternal yang penting dan menyediakan suatu dasar informasi untuk pengambilan keputusan.

Pendapat lain menurut (Tujni, 2018: 14), Sistem informasi yaitu kombinasi terorganisasi dari manusia, perangkat keras, perangkat lunak, jaringan komunikasi, sumber data, dan kebijakan serta prosedur yang terorganisasi yang menyimpan, mengambil, mengubah dan memisahkan informasi dalam sebuah organisasi.

Terdapat 6 buah komponen sistem informasi menurut (Husda, 2012: 120) dalam buku pengantar teknologi informasi, sebagai berikut:

a. Blok Masukan (*Input Block*)

Input merupakan data yang masuk ke dalam sistem informasi.

b. Blok Model (*Model Block*)

Kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data *input* dan data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

c. Blok Keluaran (*Output Block*)

Keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

d. Blok Teknologi (*Technology Block*)

Teknologi merupakan kotak alat (*tool box*) dalam sistem informasi, teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara menyeluruh.

e. Blok Basis Data (*Database Block*)

Merupakan kumpulan dari data yang saling berhubungan satu sama lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.

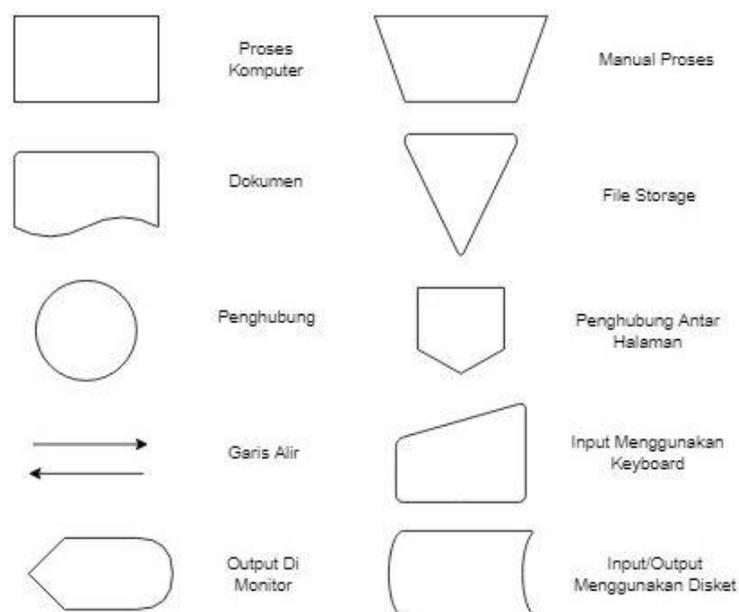
f. Blok Kendali (*Control Block*)

Beberapa pengendalian yang dirancang khusus untuk menanggulangi gangguan-gangguan terhadap sistem.

2.1.4. Aliran Sistem Informasi

Menurut (Amri Jaya, 2016: 184), *System flow* atau bagan alir sistem merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. *System flow* menunjukkan urutan-urutan dari prosedur yang ada didalam sistem dan menunjukkan apa yang dikerjakan sistem.

Pendapat lain menurut (Ismael, 2017: 149), Aliran Sistem Informasi (ASI) merupakan bagan yang menunjukkan arus pekerjaan serta keseluruhan dari sistem. Bagan ini menjelaskan urutan dari prosedur-prosedur yang ada dalam sistem. Gambar 2.1 dibawah ini merupakan daftar simbol-simbol yang digunakan dalam Aliran Sistem Informasi (ASI).



Gambar 2.1 Simbol dalam Aliran Sistem Informasi

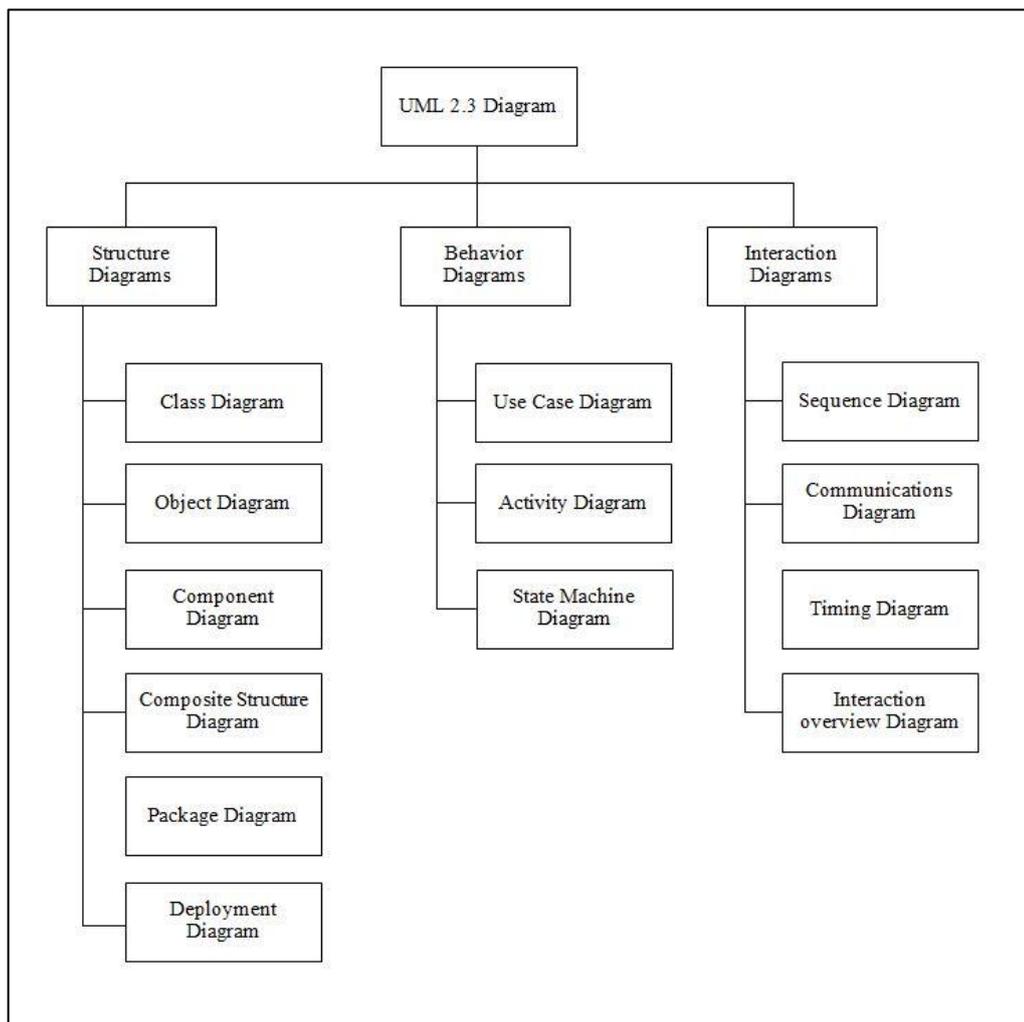
2.1.5. UML

Menurut (Prasetyo & Rahel, 2016:2), UML (*Unified Modelling Language*) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek. Dengan kata lain UML adalah satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek.

Unified Modeling Language, salah satu alat bantu yang dapat digunakan dalam bahasa pemrograman yang berorientasi objek, saat ini UML akan mulai menjadi standar masa depan bagi industri pengembangan sistem/perangkat lunak yang berorientasi objek sebab pada dasarnya UML digunakan oleh banyak perusahaan raksasa seperti IBM, Microsoft, dan sebagainya (Ferdika & Kuswara, 2017: 177).

Menurut (Rosa & Shalahuddin, 2018: 137), UML merupakan 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.

Dalam buku (Rosa & Shalahuddin, 2018: 140) 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 dibawah.



Gambar 2.2 Diagram UML 2.3

Berikut ini penjelasan singkat dari pembagian kategori tersebut.

- a. *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
- b. *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.

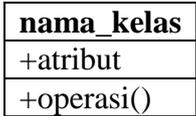
- c. *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

1. *Class Diagram*

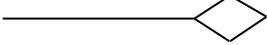
Menurut (Rosa & Shalahuddin, 2018: 141), Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas, sedangkan metode atau operasi adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Berikut adalah simbol-simbol yang ada pada diagram kelas:

Tabel 2.1 Simbol *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 antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi berarah / <i>directed association</i> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>

Tabel 2.1 Lanjutan

Generalisasi 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan / <i>dependency</i> 	Relasi antarkelas dengan makna kebergantungan antarkelas
Agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>)

2. Use Case Diagram

Menurut (Rosa & Shalahuddin, 2018: 155), *Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat. Secara kasar, *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* (Rosa & Shalahuddin, 2018: 155).

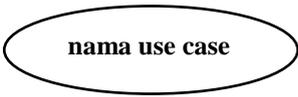
- a. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun

simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.

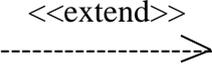
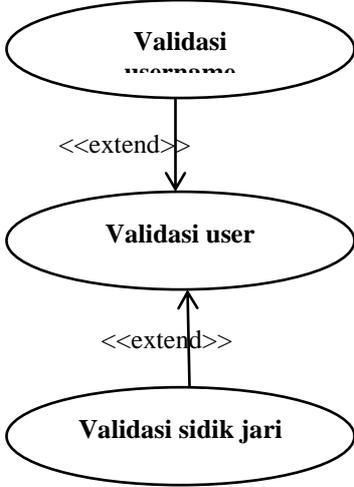
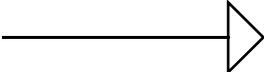
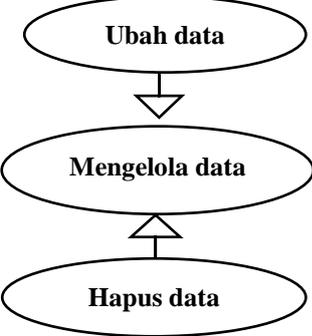
- b. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Berikut adalah simbol-simbol yang ada pada diagram *use case*:

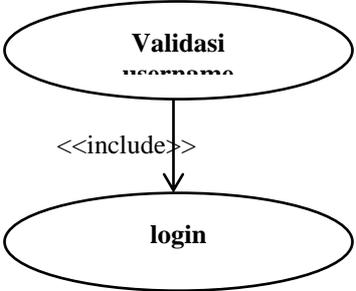
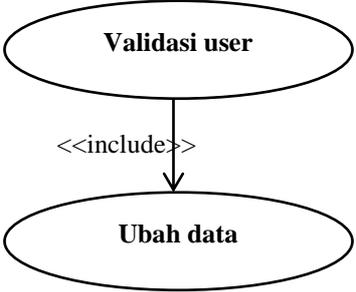
Tabel 2.2 Simbol *Use Case*

Simbol	Deskripsi
<p><i>Use case</i></p>  <p>nama use case</p>	<p>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>Aktor / <i>actor</i></p>  <p>nama aktor</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor</p>
<p>Asosiasi / <i>association</i></p> 	<p>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>

Tabel 2.2 Lanjutan

<p>Ekstensi / <i>extend</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 dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misal</p> <div style="text-align: center;">  </div> <p>Arah panah mengarah pada <i>use case</i> yang ditambahkan; biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> menjadi induknya</p>
<p>Generalisasi / <i>generalization</i></p> <p style="text-align: center;">  </p>	<p>Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya:</p> <div style="text-align: center;">  </div> <p>Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)</p>

Tabel 2.2 Lanjutan

<p>Menggunakan / <i>include</i> / <i>uses</i></p> <p style="text-align: center;"> <<include>> -----> </p> <p style="text-align: center;"> <<uses>> -----> </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 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> <ul style="list-style-type: none"> • <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan, misal pada kasus berikut: <div style="text-align: center;">  <pre> graph TD UC1([Validasi username]) -- <<include>> --> UC2([login]) </pre> </div> <ul style="list-style-type: none"> • <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, misal pada kasus berikut: <div style="text-align: center;">  <pre> graph TD UC1([Validasi user]) -- <<include>> --> UC2([Ubah data]) </pre> </div> <p>Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>
--	--

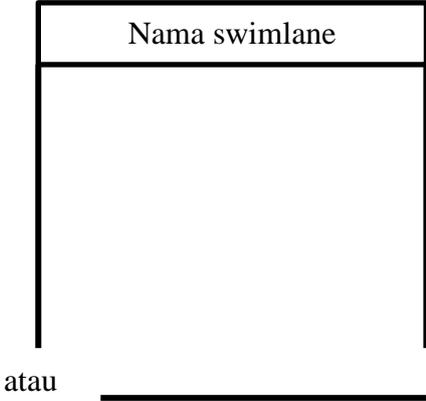
3. Activity Diagram

Menurut (Rosa & Shalahuddin, 2018: 161), Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

- a. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan
- b. Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan
- c. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
- d. Rancangan menu yang ditampilkan pada perangkat lunak.

Berikut adalah simbol-simbol yang ada pada diagram aktivitas:

Tabel 2.3 Simbol *Activity Diagram*

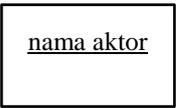
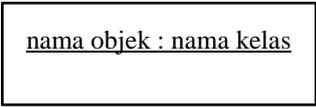
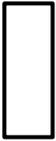
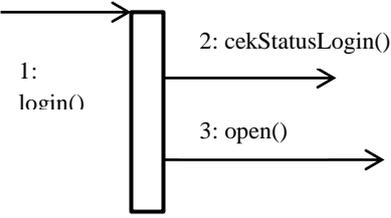
Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
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

4. *Sequence Diagram*

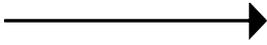
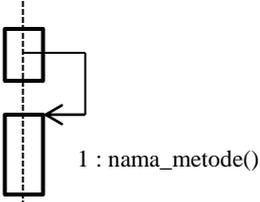
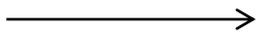
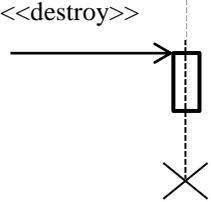
Menurut (Rosa & Shalahuddin, 2018: 165), Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu, membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

Berikut adalah simbol-simbol yang ada pada diagram sekuen:

Tabel 2.4 Simbol *Diagram Sequence*

Simbol	Deskripsi
<p>Aktor</p>  <p>nama aktor</p> <p>atau</p>  <p>nama aktor</p> <p>tanpa waktu aktif</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor</p>
<p>Garis hidup / <i>lifetime</i></p> 	<p>Menyatakan kehidupan suatu objek</p>
<p>Objek</p>  <p>nama objek : nama kelas</p>	<p>Menyatakan objek yang berinteraksi pesan</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p>  <p>Maka <code>cekStatusLogin()</code> dan <code>open()</code> dilakukan di dalam metode <code>login()</code> Aktor tidak memiliki waktu aktif</p>

Tabel 2.4 Lanjutan

<p>Pesan tipe <i>create</i></p> <p style="text-align: center;"><<create>></p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat</p>
<p>Pesan tipe <i>call</i></p> <p style="text-align: center;">1 :</p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,</p>  <p style="text-align: right;">1 : nama_metode()</p> <p>Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi</p>
<p>Pesan tipe <i>send</i></p> <p style="text-align: center;">1 :</p> 	<p>Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim</p>
<p>Pesan tipe <i>return</i></p> <p style="text-align: center;">1 :</p> 	<p>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>Pesan tipe <i>destroy</i></p> <p style="text-align: center;"><<destroy>></p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang di akhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i></p>

2.1.6. SDLC

Waterfall model sebagai salah satu teori dasar dan seakan wajib dipelajari dalam konteks siklus hidup perangkat lunak, merupakan sebuah siklus hidup yang terdiri dari mulai fase hidup perangkat lunak sebelum terjadi hingga pasca produksi. *Waterfall* model memiliki definisi sendiri bahwa sebuah hidup perangkat lunak memiliki sebuah proses yang linear dan sekuensial. Metode pengembangan sistem metode SDLC (*System Development Life Cycle*) atau sering disebut sebagai pendekatan air terjun (*waterfall*). Metode *waterfall* pertama kali diperkenalkan oleh Windows W. Royce pada tahun 1970. *Waterfall* merupakan model klasik yang sederhana dengan aliran sistem yang linier *Output* dari setiap tahap merupakan *input* bagi tahap berikutnya (Ferdika & Kuswara, 2017: 178).

Menurut (Rosa & Shalahuddin, 2018: 26), SDLC atau *Software Development Life Cycle* atau sering disebut juga *System Development Life Cycle* ada proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan *best practice* atau cara-cara yang sudah teruji baik)

Tahapan-tahapan yang ada pada SDLC secara *global* dalam buku (Rosa & Shalahuddin, 2018: 26) adalah sebagai berikut:

1. Inisiasi (*initiation*), tahap ini biasanya ditandai dengan pembuatan proposal proyek perangkat lunak.
2. Pengembangan konsep sistem (*system concept development*) mendefinisikan lingkup konsep termasuk dokumen lingkup sistem,

analisis manfaat biaya, manajemen rencana, dan pembelajaran kemudahan sistem.

3. Perencanaan (*planning*), mengembangkan rencana manajemen proyek dan dokumen perencanaan lainnya.
4. Analisis kebutuhan (*requirements analysis*), menganalisis kebutuhan pemakai sistem perangkat lunak (*user*) dan mengembangkan kebutuhan *user*, membuat dokumen kebutuhan fungsional.
5. Desain (*design*), mentransformasikan kebutuhan detail menjadi kebutuhan yang sudah lengkap, dokumen desain sistem fokus pada bagaimana dapat memenuhi fungsi-fungsi yang dibutuhkan.
6. Pengembangan (*development*), mengkonversi desain ke sistem informasi yang lengkap termasuk bagaimana memperoleh dan melakukan instalasi lingkungan sistem yang dibutuhkan.
7. Integrasi dan pengujian (*integration and test*), mendemonstrasikan sistem perangkat lunak bahwa telah memenuhi kebutuhan yang dispesifikasikan pada dokumen kebutuhan fungsional.
8. Implementasi (*implementation*), termasuk pada persiapan implementasi, implementasi perangkat lunak pada lingkungan produksi (lingkungan pada user) dan menjalankan resolusi dari permasalahan yang teridentifikasi dari fase integrasi dan pengujian.
9. Operasi dan pemeliharaan (*operations and maintenance*), mendeskripsikan pekerjaan untuk mengoperasikan dan memelihara

sistem informasi pada lingkungan produksi, termasuk implementasi akhir dan masuk pada proses peninjauan.

10. Disposisi (*disposition*), mendeskripsikan aktifitas akhir dari pengembangan sistem dan membangun data yang sebenarnya sesuai dengan aktivitas *user*.

2.2. Teori Khusus

2.2.1 Penjualan

Menurut (Ferdika & Kuswara, 2017: 176), Penjualan merupakan kegiatan yang dilakukan oleh penjual dalam menjual barang atau jasa dengan harapan akan memperoleh laba dari adanya transaksi-transaksi tersebut dan penjualan dapat diartikan sebagai pengalihan atau pemindahan hak kepemilikan atas barang atau jasa dari pihak penjual ke pembeli.

Menurut (Anthony et al., 2017: 138), yang dimaksud dengan penjualan adalah menjual barang dagang yang menjadi usaha pokok suatu perusahaan dengan dilakukan secara terus menerus dan tepat. Penjualan juga adalah salah satu langkah pemasaran dari suatu perusahaan, sehingga perusahaan dapat memperoleh keuntungan yang akibatnya adalah kegiatan operasional perusahaan dapat tetap dijalankan. Sebuah perusahaan mempunyai tiga tujuan dalam penjualan yaitu:

- a. Mencapat batas tertentu dari penjualan
- b. Mendapat keuntungan
- c. Mendukung perkembangan perusahaan.

Penjualan bila diidentifikasi berdasarkan perusahaannya maka dapat dikategorikan menjadi beberapa jenis, yaitu :

- a. Penjualan langsung, dimana penjualan ini adalah dengan cara mengambil barang dari *supplier* kemudian secara langsung dikirim ke *customer*
- b. Penjualan stok gudang, dimana penjualan ini adalah dengan cara menjual barang dari stok yang ada di gudang.
- c. Penjualan kombinasi, adalah penjualan dimana dengan mengambil sebagian barang dari *supplier* serta sebagian dari stok yang ada di gudang.

Dalam jurnal (Anthony et al., 2017: 138), terdapat beberapa bentuk-bentuk dari penjualan, antara lain:

- a. Penjualan tunai/*cash*, adalah penjualan bersifat *cash and carry* yang mana penjualan dilakukan setelah terdapat kesepakatan harga antara penjual dengan pembeli, lalu pembeli dapat membayar secara langsung dan barang dapat langsung dimiliki.
- b. Penjualan kredit/*non-cash*, yaitu penjualan *non-cash*, dengan memberikan tenggat waktu tertentu, biasanya diatas satu bulan.
- c. Penjualan tender, adalah penjualan yang dilaksanakan melalui prosedur tender yang digunakan untuk memenuhi permintaan pihak pembeli yang membuka tender.

- d. Penjualan ekspor, yaitu penjualan yang dilaksanakan dengan pihak pembeli dimana mengimpor barang dari luar negeri yang biasanya dengan menggunakan *letter of credit*.
- e. Penjualan konsinyasi, yaitu penjualan barang dengan cara menitipkan kepada pembeli yang juga berperan sebagai penjual dimana jika barang yang dititipkan tersebut tidak terjual maka barang akan dikembalikan lagi kepada penjual.
- f. Penjualan grosir, yaitu penjualan yang dilakukan tidak langsung kepada pembeli, tetapi melalui perantara pedagang.

2.2.2 Internet

Interconnected Network atau yang lebih dikenal dengan sebutan internet adalah suatu jaringan komputer raksasa dengan daya jangkauan yang mendunia. Pengertian lain internet (*inter-network*) merupakan jaringan yang menggabungkan beberapa komputer yang terhubung dalam sebuah *internet protocol* (IP) yang mencakup secara luas ke seluruh dunia (Prasetyo & Rahel, 2016: 2).

Internet adalah alat penghubung antara organisasi dan pelanggannya, sehingga tercipta sebuah organisasi baru secara *virtual*. Menurut DeFleur & Dennis dalam Zin, Muda, & Nordin, *internet* adalah sebuah sistem komputasi di seluruh dunia yang menggunakan sarana umum untuk menghubungkan perangkat keras dan mentransmisikan informasi digital, komunitas orang dengan menggunakan sebuah teknologi komunikasi yang umum dan mendistribusikan sistem informasi secara *global* (Pibriana & Ricoida, 2017: 105).

2.2.3 Website

Menurut (Ferdika & Kuswara, 2017: 176), *Website* adalah kumpulan halaman *web* yang saling terhubung dan *file-filenya* saling terkait. *Web* terdiri dari *page* atau halaman, dan kumpulan halaman yang dinamakan *homepage*. *Homepage* berada pada posisi teratas, dengan halaman- halaman terkait berada di bawahnya. Biasanya setiap halaman di bawah *homepage* disebut *child page*, yang berisi *hyperlink* ke halaman lain dalam *web*.

World Wide Web (biasa disingkat WWW) atau *web* merupakan salah satu aplikasi internet yang paling populer. *Web* adalah sebuah sistem dimana informasi dalam bentuk teks, gambar, suara dan lainnya yang tersimpan dalam sebuah *internet webserver* ditampilkan dalam bentuk HTML (*hypertext Markup language*). Pengertian lain *web* atau *www* adalah dokumen atau informasi yang saling berhubungan yang dihubungkan melalui *hyperlink* atau URL (*Uniform Resource Locator*) (Prasetyo & Rahel, 2016: 2)

2.2.4 HTML5

“HTML (*Hyper Text Markup Language*) adalah bahasa yang digunakan pada dokumen *web* sebagai bahasa untuk pertukaran dokumen *web*”. HTML (*Hyper Text Markup Language*) adalah bahasa mark-up internet (*web*) berupa kode dan simbol yang dimasukkan kedalam sebuah *file* yang ditujukan untuk ditampilkan didalam sebuah *website*. *Website* yang dibuat dengan HTML ini, dapat dilihat oleh semua orang yang terkoneksi dengan *internet*. Tentunya dengan menggunakan aplikasi penjelajah internet (*browser*) seperti Internet Explorer, Mozilla Firefox dan Google Chrome (Ferdika & Kuswara, 2017: 177).

Dalam jurnal (Yatini, 2014: 2), HTML 5 merupakan teknologi inti dari *Internet* adalah bahasa *markup* untuk penataan dan penyajian konten *world wide web*. Tujuan utama HTML 5 adalah meningkatkan bahasa dengan dukungan multimedia yang tetap mudah dibaca dan dimengerti. Fitur baru HTML 5 antara lain:

- a. Unsur `<canvas>` untuk menggambar 2D.
- b. Unsur `<video>` dan `<audio>` untuk media pemutar.
- c. Dukungan untuk penyimpanan lokal.
- d. Konten baru dengan elemen spesifik seperti `<article>`, `<footer>`, `<header>`, `<nav>`, `<section>`.

2.2.5 CSS

Menurut (Maudi et al., 2014: 103), CSS merupakan singkatan dari *Cascading Style Sheet*. CSS biasa digunakan dalam dokumen HTML untuk menciptakan suatu *style* yang dipakai untuk mengatur penampilan elemen HTML. Dengan menggunakan *style*, suatu elemen dapat diformat dengan fitur yang jauh lebih kaya daripada yang disediakan oleh elemen HTML itu sendiri. Sebagai contoh, pengaturan seperti warna tulisan bisa ditangani melalui *style* tanpa melibatkan *tag* HTML yang berfungsi untuk mengatur warna.

2.2.6 PHP

Menurut (Ferdika & Kuswara, 2017: 177), PHP atau *Hypertext Preprocessor* merupakan bahasa pemrograman berbasis *web* yang memiliki kemampuan untuk memproses data dinamis. Artinya, dapat membentuk suatu tampilan berdasarkan permintaan terkini. PHP merupakan teknologi *open source* yang bebas diperoleh

dan diatur untuk memenuhi kebutuhan perusahaan, pendidikan, atau professional yang didukung oleh komunitas pengguna dan pengembangnya. Untuk menjalankan PHP dibutuhkan suatu *web server*.

PHP merupakan singkatan dari *Hypertext Preprocessor*. Ia merupakan bahasa berbentuk skrip yang ditempatkan dalam *server* dan diproses di *server*. Hasilnya yang dikirim ke klien, tempat pemakai menggunakan *browser*. Kelahiran PHP bermula saat Rasmus Lerdorf membuat sejumlah skrip Perl yang dapat mengamati siapa saja yang melihat-lihat daftar riwayat hidupnya, yakni pada tahun 1994. Skrip-skrip ini selanjutnya dikemas menjadi *tool* yang disebut “*Personal Home Page*”. Paket inilah yang menjadi cikal bakal PHP. Pada tahun 1995, Rasmus menciptakan PHP/FI versi 2. Pada versi inilah pemrograman dapat menempelkan kode terstruktur di dalam *tag* HTML. Yang menarik, kode PHP juga bisa berkomunikasi dengan *database* dan melakukan perhitungan-perhitungan yang kompleks sambil jalan (Prasetyo & Rahel, 2016: 2)

2.2.7 JavaScript

Dalam jurnal (Yatini, 2014: 2), JavaScript adalah bahasa *scripting* kecil, ringan, berorientasi objek yang ditempelkan pada kode HTML dan di proses di sisi *client*. JavaScript digunakan dalam pembuatan *website* agar lebih interaktif dengan memberikan kemampuan tambahan terhadap HTML melalui eksekusi perintah di sisi *browser*. JavaScript dapat merespon perintah *user* dengan cepat dan menjadikan halaman *web* menjadi responsif. JavaScript memiliki struktur sederhana, kodenya dapat disisipkan pada dokumen HTML atau berdiri sebagai satu kesatuan aplikasi.

Menurut (Prayitno & Safitri, 2015: 2) JavaScript adalah bahasa pemrograman yang digunakan untuk membuat program yang digunakan agar dokumen HTML yang ditampilkan dalam *browser* menjadi lebih interaktif, tidak sekedar indah saja

2.2.8 XAMPP

Menurut (Ferdika & Kuswara, 2017: 177), XAMPP adalah sebuah *software web server apache* yang di dalamnya sudah tersedia *database server* MySQL dan dapat mendukung pemograman PHP”. Berdasarkan definisi di atas, maka dapat disimpulkan XAMPP adalah sebuah *software web server apache* yang didalamnya sudah tersedia *database server* MySQL dan *support* PHP *programming*.

Sedangkan menurut (Mulayana & Gustina, 2016: 164), XAMPP merupakan perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri atas program Apache HTTP Server, MySQL *database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Program ini tersedia dalam GNU *General Public License* dan Bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan *web* yang dinamis.

2.2.9 Dreamweaver CS6

Menurut (Maudi et al., 2014: 102), Adobe Dreamweaver merupakan program penyunting halaman *web* keluaran Adobe Systems yang dulu dikenal sebagai Macromedia Dreamweaver keluaran Macromedia. Program ini banyak

digunakan oleh pengembang *web* karena fitur-fiturnya yang menarik dan kemudahan penggunaannya. Versi terakhir Macromedia Dreamweaver sebelum Macromedia dibeli oleh Adobe Systems yaitu versi 8.

2.2.10 Basis Data

Menurut (Prayitno & Safitri, 2015: 2), *Database* atau biasa disebut basis data merupakan kumpulan data yang saling berhubungan. Data tersebut biasanya terdapat dalam tabel-tabel yang saling berhubungan satu sama lain, dengan menggunakan *field*/kolom pada tiap tabel yang ada.

Menurut (Rosa & Shalahuddin, 2018: 43), sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat, kebutuhan basis data dalam sistem informasi yaitu memasukkan, menyimpan, dan mengambil data, serta membuat laporan berdasarkan data yang telah disimpan.

2.2.11 SQL

Menurut (Rosa & Shalahuddin, 2018: 46), SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengelola data RDBMS. SQL awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus. Meskipun SQL diadopsi dan diacu sebagai bahasa standar oleh hampir sebagian besar RDBMS yang beredar saat ini, tetapi tidak semua standar yang tercantum dalam SQL diimplementasikan oleh seluruh DBMS tersebut. Sehingga kadang-

kadang ada perbedaan perilaku oleh DBMS yang berbeda padahal *query* yang dimasukkan sama.

Berikut ini adalah contoh pengaksesan data pada DBMS dengan SQL menurut (Rosa & Shalahuddin, 2018: 46) yang secara umum terdiri dari 4 hal sebagai berikut:

1. Memasukkan data (*insert*)
2. Mengubah data (*update*)
3. Menghapus data (*delete*)
4. Menampilkan data (*select*)

2.2.12 MySQL

Menurut (Ferdika & Kuswara, 2017: 177), MySQL merupakan salah satu perangkat lunak sistem manajemen basis data (*database management system*) atau DBMS yang menggunakan perintah standar SQL (*structured Query Language*). Dimana MySQL mampu untuk melakukan banyak eksekusi perintah *query* dalam satu permintaan, baik itu menerima dan mengirimkan data. MySQL juga *multi-user* dalam arti dapat dipergunakan oleh banyak pengguna dalam waktu bersamaan. MySQL tersedia dalam perangkat lunak gratis dibawah lisensi GNU *General Public Lisenca* (GPL) dan juga menjual dalam lisensi komersial untuk keperluan jika pengguna tidak cocok menggunakan lisensi *General Public Lisenca* (GPL). Pengguna MySQL yang merupakan sebuah *database server* sekaligus dapat sebagai *client*, dan dapat berjalan di *multi-OS* (*Operating System*) memiliki keunggulan lainnya seperti *Open Source* sehingga penggunanya tidak perlu membayar lisensi kepada pembuatnya. Bahasa pemrograman yang dapat

digunakan untuk mengakses MySQL diantaranya adalah dengan C, C++, Java, Perl, PHP, Python, dan APIs.

MySQL adalah *database* yang cepat dan tangguh, sangat cocok jika digabungkan dengan PHP, dengan *database* kita bisa menyimpan, mencari dan mengklasifikasikan data dengan lebih akurat dan profesional. MySQL menggunakan SQL language (*Structur Query Language*) artinya MySQL menggunakan *query* atau bahasa pemrograman yang sudah standar di dalam dunia *database* (Ipnuwati, 2010: 15).

2.3. Penelitian Terdahulu

Penelitian terdahulu dari penelitian ini adalah sebagai berikut:

Tabel 2.5 Penelitian Terdahulu

No	Peneliti	Judul	Hasil
1	Dwi Hartini, Sarjono (2016)	Analisis Dan Perancangan Sistem Informasi Penjualan Pada Pt. Jaya Mandiri Strategic	Hasil penelitian yang dilakukan berupa prototype dari sistem informasi penjualan untuk menghemat waktu dan biaya operasional bentuk usaha dagang.
2	Andri Prasetyo, Rahel Susanti (2016)	Sistem Informasi Penjualan Berbasis <i>Web</i> Pada PT. Cahaya Sejahtera Sentosa Blitar	Hasil penelitian yang dilakukan berupa aplikasi sistem informasi pemesanan produk ini dapat memudahkan pelanggan dalam pencarian informasi tiwul instan dengan pemesanan produk yang cukup mudah dengan bergabung menjadi pelanggan atau member di dalam sistem informasi ini dan pembayaran dilakukkan melalui <i>transfer bank</i> .
3	Nurul Azwanti (2017)	Sistem Informasi Penjualan Tas Berbasis <i>Web</i> Dengan Pemodelan Uml	Hasil penelitian yang dilakukan berupa Sebuah aplikasi yang dapat dijadikan sebagai media penjualan pada Mendi Shopping. Membantu pihak Mendi Shopping dalam menyimpan data-data mereka dengan cara membuat suatu <i>database</i> dan membantu mengurangi kesalahan, kehilangan data dalam pengolahan data penjualan.
4	Ujang Mulayana, Dian Gustina (2016)	Perancangan Sistem Informasi Penjualan Handphone Berbasis <i>Web</i> Pada Toko Ilham Cellular Jakarta	Hasil penelitian yang dilakukan berupa Sebuah Sistem Informasi Penjualan <i>Handphone</i> Berbasis <i>Web</i> Pada Toko Ilham Cellular Jakarta
5	Nandang Iriadi, Nia Rosdiana (2017)	Perancangan Sistem Informasi Penjualan Minuman Kemasan Berbasis <i>Web</i> Pada Toko Bambu Sejahtera Bekasi	Hasil penelitian yang dilakukan berupa Sebuah Sistem Informasi Penjualan Minuman Kemasan Berbasis <i>Web</i> Pada Toko Bambu Sejahtera Bekasi

Tabel 2.5 Lanjutan

6	Queiroz & Oliveira (2014)	Benefits Of The Marketing Information System In The Clothing Retail Business	Hasil dari penelitian ini menunjukkan bahwa informasi dan proses sistematisnya mempunyai kedudukan yang penting dalam pengembangan operasi bisnis dan untuk menciptakan kesempatan bisnis yang baru.
---	---------------------------	--	--