# BAB II

# LANDASAN TEORI

## 1.1. Konsep Dasar Sistem

Pada umumnya setiap organisasi mempunyai sistem informasi dalam mengumpulkan, menyimpan, melihat, dan menyalurkan informasi dalam membuat perancangan sistem informasi.

Konsep dasar sistem merupakan sekelompok komponen berbasis komputer yang dibuat oleh manusia dalam mengelola data, menyimpan, menghimpun kerangka kerja serta mengkoordinasikan sumber daya manusia dan computer untuk mengubah sistem masukan menjadi sistem keluaran untuk mencapai tujuan dan sasaran yang telah ditetapkan sebelumnya

## A. Model Pembelajaran Berbasis Web

Pembelajaran berbasis web berakar pada potensi yang bersumber dari teknologi. Sebagian besar pembelajaran berbasis web dapat menghemat biaya dan terasa nyaman.(Wulandari, 2020)

Pembelajaran meliputi perencanaan, pelaksanaan, dan evaluasi kurikulum atau materi yang tercakup dalam ruang lingkup dan urutannya. Ruang lingkup kurikulum mencakup luas dan kedalaman dari apa yang diajarkan, dan urutannya adalah urutan pembelajarannya. Sebagaimana media pembelajaran lainnya pembelajaran dengan menggunakan web juga memiliki kelebihan dan kekurangan.

## B. Pengertian Sistem

Sistem berasal dari Bahasa latin (*systema*) dan Bahasa Yunani (*sustema*) adalah suatu kesatuan yang tediri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi untuk mencapai suatu tujuan. Dalam pengertian yang paling umum, sebuah sistem adalah sekumpulan benda yang memiliki hubungan di antara mereka.

Perancangan sistem adalah serangkaian kegiatan yang selalu beriringan dalam setiap pengembangan software, sebagai hubungan sebab dan akibat yang memunculkan siklus hidup sistem.(Muharam et al., 2018)

OPP (*Object Oriented Programming*) adalah "suatu metode pemrograman yang berorientasi kepada objek. Tujuan dari OOP diciptakan adalah untuk mempermudah pengembangan program dengan cara mengikuti model yang telah ada dikehidupan sehari-hari".

Ssistem yang baik sangat diperlukan bagi suatu organisasi. Untuk menunjang kinerja organisasi, maka dibutuhkan kerjasama diantara unsur atau elemen yang berkaitan dengan sistem tersebut.

suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu:

1) Komponen Sistem

   Komponen sistem terdiri dari sejumlah komponen yang saling berinteraksi yang artinya saling bekerjasama membentuk satu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu sub sistem atau bagian-bagian dari sistem. Setiap sistem tidak terdiri betapapun kecilnya, selalu mengandung komponen-komponen atau sub sistem-sub sistem. Setiap sub sistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

2) Batas Sistem

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini kemungkinan suatu dipandang sebagai suatu kesatuan. Batas suatu sistem menunjukan ruang lingkup (*scope*) dari sistem tersebut.

3) Lingkungan Luar Sistem

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga merugikan suatu sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus ditahan dan dikendalikan, kalau tidak maka akan mengganggu kelangsungan hidup dari sistem.

4) Penghubung Sistem

Penghubung sistem merupakan media penghubung antara suatu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari suatu subsistem ke subsistem yang lainnya. Keluaran (*output*) dari suatu sistem akan menjadi masukkan (*input*) untuk subsistem yang lainnya dengan melalui penghubung.

5) Masukkan Sistem

Masukkan adalah energi yang di masukan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan signal (*signal input*). *Maintenance input* adalah energi yang di masukan supaya sistem tersebut dapat beroperasi. *Signal input* adalah energi yang di proses untuk dapat keluaran.

6) Keluaran Sistem

Keluaran adalah hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lain atau kepada supra sistem.

7) Pengolah Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran. Suatu sistem produksi akan mengolah masukan berupa bahan baku dan bahan bahan yang lain menjadi keluaran berupa barang jadi.

8) Sasaran Sistem

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Kalau suatu sistem/tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Sasran dari sistem sangat menetukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.

Sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya:

1) Sistem diklasifikasikan sebagai sistem abstrak (*abstract system*) dan sistem fisik (*physical system*). Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Sistem fisik merupakan sistem yang ada secara fisik.
2) Sistem diklasifikasikan sebagai sistem alamiah (*natural system*) dan sistem buatan manusia (*human made system*). Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia. Sistem buatan manusia adalah sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan interaksi antara manusia dan mesin disebut dengan *human machine system* atau ada yang menyebut dengan *man machine system*.

3) Sistem diklasifikasikan sebagai sistem tertentu (*deterministic system*) dan sistem tak tentu (*probabilistic system*). Sistem tertentu beroprasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi diantara bagian-bagiannya dapat dideteksi dengan pasti, sehingga keluaran dari sistem dapat diramalkan. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.
4) Sistem diklasifikasikan sebagai sistem tertutup (*closed system*) dan sistem terbuka (*open system*). Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya. Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk lingkungan luar atau subsistem yang lainnya.

## C. Pengertian Sistem Informasi

Sistem informasi terdiri dari beberapa komponen yang saling bekerja sama untuk mengolah data dan informasi. Komponen sistem informasi terdiri dari unsur-unsur seperti masukkan (*input*), pengolahan (*processing*), keluaran (*output*), serta umpan balik (*feedback*). (Fernando et al., 2018)

Definisi Sistem Informasi adalah Sistem informasi adalah kumpulan dari subsistem apapun baik phisik ataupun non phisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan yaitu mengolah data menjadi informasi yang berarti dan berguna.

## D. Website

Website merupakan sebuah media informasi yang ada di internet.Website tidak hanya dapat digunakan untuk penyebaran infomasi saja melainkan bisa digunakan untuk membuat toko online. Website adalah kumpulan dari halaman-halaman situs, yang biasanya terangkum dalam sebuah domain atau subdomain, yang tempatnya berada di dalam World Wide Web (WWW) di Internet. Sebuah halaman webadalah dokumen yang ditulis dalam format HTML (Hyper Text Markup Language), yang hampir selalu bisa diakses melalui HTTP, yaitu protokol yang menyampaikan informasi dari server website untuk ditampilkan kepada para pemakai melalui web browser. Semua publikasi dari website-website tersebut dapat membentuk sebuah jaringan informasi yang sangat besar.

Halaman-halaman dari website akan bisa diakses melalui sebuah URL yang biasa disebut Homepage. URL ini mengatur halaman-halaman situs untuk menjadi sebuah hirarki, meskipun, hyperlink-hyperlink yang ada di halaman tersebut mengatur para pembaca dan memberitahu mereka susunan keseluruhan dan bagaimana arus informasi ini berjalan. Beberapa website membutuhkan subskripsi (data masukan) agar para user bisa mengakses sebagian atau keseluruhan isi website tersebut.

Website adalah Kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gambar gerak, animasi, suara atau gabungan dari semuanya, baik yang bersifat dinamis yang membentuk satu rangkaian bangunan yang saling terkait yang masing-masing dihubungkan dengan jaringan-jaringan halaman.(Maimunah et al., 2020).

"*Website* terhubung dengan suatu jaringan internet yang akan membawa pengguna kesuatu tujuan yang diinginkan oleh pengguna dengan cara mengklik link yang berupa teks, gambar" (Endra & Aprilita, 2018).

"Situs Web (*Website*) awalnya merupakan suatu layanan sajian informasi yang menggunakan konsep hiperlink yang memudahkan surfer ( sebutan bagi pemakai komputer yang melakukan penyelusuran informasi di Internet) untuk mendapatkan informasi dengan cukup mengklik suatu link berupa teks atau gambar maka informasi dari teks atau gambar akan ditampilkan secara lebih terperinci (*detail*)" (Nurmalasari et al., 2019).

## E. Basis Data

Basis data terdiri dari kata basis dan data. Basis merupakan markas atau gudang, sednagkan data adalah catatan atas kumpulan fakta dan informasi yang mewakili objek, peristiwa, ataupun symbol.

Basis Data Merupakan data yang saling terhubung dan berkaitan dengan subjek tertentu pada tujuan tertentu pula. Hubungan antar data ini dapat dilihat oleh adanya *fiel* ataupun kolom.

## F. Pengertian PHP

Menurut (Firman et al., 2016) PHP atau kependekan dari Hypertext Preprocessor adalah salah satu bahasa pemrograman open source yang sangat cocok atau dikhususkan untuk pengembangan web dan dapat ditanamkan pada sebuah skripsi HTML. Bahasa PHP dapat dikatakan menggambarkan beberapa bahasa pemrograman seperti C, Java, dan Perl serta mudah untuk dipelajari.

## G. Pengertian XAMPP

XAMPP adalah perangkat lunak gratis yang mendukung banyak sistem operasi dan merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang terdiri sendiri (localhost), yang terdiri atas program Apache HTTP Server, MySQL Database, dan penterjemah bahasa yang ditulis dengan bahasa pemograman PHP dan Perl. (Sukma et al., 2020)

## H. Model Pengembangan Perangkat Lunak

*Rapid Application Development* (RAD) sebuah model pengembangan sistem dengan pendekatan prototyping yang dirancang untuk menghasilkan sistem yang berkualitas tinggi dengan waktu yang relatif cepat serta biaya yang cukup rendah (Sikumbang et al., 2020).

"Model air terjun (*Waterfall*) menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain pengodean, pengujian dan tahap pendu.(Rosa & Shalahuddin, 2018)

1. Analisis Kebutuhan Perangkat Lunak

   Analisis kebutuhan perangkat lunak Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dubutuhkan oleh user.

2. Desain

   Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program Perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosuder pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

3. Pembuatan Kode Program

   Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai desain yang telah dibuat pada tahap desain.

4. Penguji

Pengujian fokus pada perangkat lunak secara dari segi lojik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (error) dan memastikan keluaran yang Dihasilkan sesuai dengan yang diinginkan

5. Pendukung (*support*) atau pemeliharaan

Pendukung (*support*) Atau pemeliharaan (*maintenance*) Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke user. Perubahan bisa terjadi karena adanya kesalahn yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

## 2.2. Teori Pendukung

Teori-teori pendukung yang dimaksud untuk merancang model sistem yang baru pada penulisan laporan Tugas Akhir ini adalah:

### A. *Entity Relationship Diagram* (ERD)

*Entitas Relationship Diagram* adalah suatu alat yang dapat digunakan untuk mengorganisasikan data yang dikumpulkan dimana dalam diagram ini dapat diperlihatkan entitas-entitas berserta hubungan entitas tersebut. Komponen utama pembentukan ERD yaitu *Entity* (entitas) dan *Relation* (relasi) sehingga dalam hal ini ERD merupakan komponen-kompenen himpunan entitas dan himpunan relasi yang dideskripsikan lebih jauh melalui sejumlah atribut-atribut (*proprety*) yang menggambarkan seluruh pakta yang ditinjau.

*”Entity Relationship Diagram* (ERD) merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh System Analys dalam tahap analisis persyaratan proyek pengembangan system” (Muharam et al., 2018)

”ERD (*Entity Relationship Diagram*) merupakan suatu model data yang dikembangkan berdasarkan objek”.(Muharam et al., 2018)

Elemen-elemen pada ERD (*Entity Relationship Diagram*) adalah sebagai berikut:

1. Entitas (*Entity*).

   Pada ERD, *entity* digambarkan dengan sebuah bentuk persegi panjang. *Entity* adalah sesuatu yang nyata maupun abstrak dimana data tersimpan yang ada di dalam sistem.

2. Hubungan (*Relationship*).

   Pada ERD, *relationship* dapat digambarkan sebuah bentuk belah ketupat. *Relationship* adalah hubungan yang terjadi yaitu bentuk belah ketupat dihubungkan dengan dua bentuk persegi panjang.

3. Derajat Relasi (*Relationship Degree*).

   Derajat relasi adalah jumlah entitas yang berpartisipasi dalam suatu *relationship.*

4. Atribut

   Atribut yaitu menjelaskan apa yang dimaksud entitas maupun *relationship,* sehingga sering dikatakan bahwa atribu adalah elemen dari setiap entitas.

5. Kardinalitas

   Kardinalitas menjelaskan jumlah maksimum tupel yang dapat berelasi dengan entitas pada entitias yang lain, dari sejumlah kemungkinan

banyaknya hubungan antar entitas, kardinalitas relasi merujuk pada hubungan maksimum yang terjadi dari entitas ke antitas lain dan sebaliknya. Terdapat tiga kardinalitas relasi, yaitu:

1. *One to one*

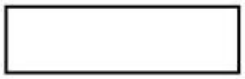
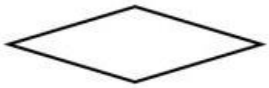
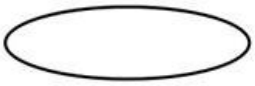
   Tingkat hubungan satu ke satu, yaitu dinyatakan satu kejadian pada entitas pertama, dan hanya mempunyai satu hubungan dengan satu kejadian pada entitas kedua dan sebaliknya.

2. *One to Many* atau *Many to One*

   Tingkat hubungan satu ke banyak yaitu sama dengan banyak ke satu. Tergantung dari arah hubungan tersebut dilihat. Untuk satu kejadian pada entitas yang pertama mempunyai banyak hubungan dengan kejadian pada entitas yang kedua.

3. *Many to Many*

   Tingkat hubungan banyak ke banyak terjadi jika tiap kejadian pada sebuah entitas mempunyai banyak hubungan dengan entitas lainnya, baik dilihat dari sisi entitas yang pertama maupun dari sisi yang kedua.

| Simbol | Keterangan |
|---|---|
|  | Entitas, yaitu kumpulan dari objek yang dapat diidentifikasikan secara unik |
|  | Relasi, yaitu hubungan yang terjadi antara salah satu lebih entitas. Jenis hubungan antara lain. one to one, One to many, dan many to many. |
|  | Atribut, yaitu karakteristik dari entitas atau relasi yang merupakan penjelasan detail tentang entitas. |
|  | Hubungan antara entitas dengan atributnya dan himpuan entitas dengan himpunan relasinya. |

**Gambar II. 1**
**Simbol Entity Relationship Diagram**

## B. *Logical Record Structure* (LRS)

LRS adalah hasil dari permodelan Entity Relationship Diagram (ERD) beserta atributnya sehingga bisa melihat hubungan-hubungan antara entitas. *Logical record structur* dibentuk menggunakan nomor dari tipe *record*. *Logical record structure* mencakup link-link diantara tipe *record*. *Link* ini mengarah dari satu tipe *record* ke tipe r*ecord* lainnya.

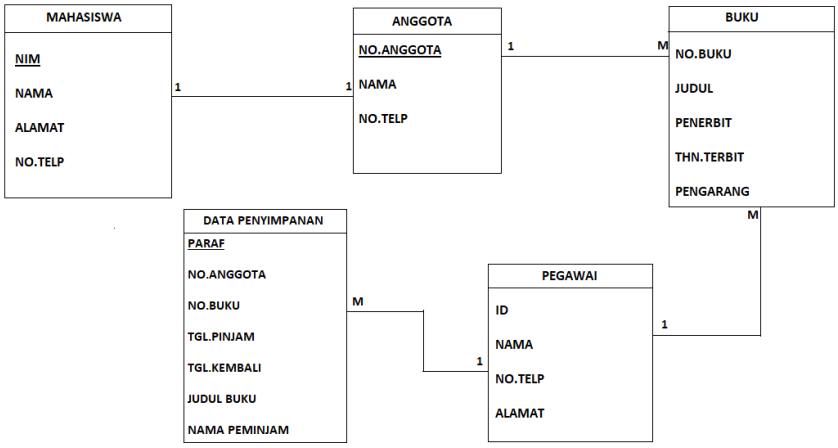
Logical Record Structure (LRS) digambarkan kotak persegi panjang dan dengan nama yang unik. File record pada LRS ditempatkan dalam kotak. LRS terdiri dari link diantara tipe record lainnya, banyaknya link dari LRS yang diberi nama oleh filed-filed yang kelihatan pada kedua link tipe record(Sukmaindrayana & Sidik, 2017).

Menurut (Taufik & Ermawati, 2017)“sebelum tabel dibentuk dari field atau atribut entitas secara fisik atau level internal, maka harus dibuatkan suatu bentuk

relational model yang dibuat secara logic atau level external dan konsep, dari pernyataan tersebut dibutuhkan yang disebut dengan Logical Record Structure".

Dalam pembuatan Logical Record Structure (LRS) terdapat tiga hal yang dapat mempengaruhi, yaitu :

1. Jika tingkat hubungan (*cardinality*) satu pada satu (*one-to-one*), maka digabungkan dengan entitas yang lebih kuat (*strong entity*), atau digabungkan dengan entitas yang memiliki atribut yang lebih sedikit.
2. Jika tingkat hubungan (*cardinality*) satu pada banyak (*one-to-many*), maka hubungan relasi atau digabungkan dengan entitas yang tingkat hubungannya banyak.
3. Jika tingkat hubungan (*cardinality*) banyak pada banyak (*many-to-many*), maka hubungan relasi tidak akan digabungkan dengan entitas manapun, melainkan menjadi sebuah LRS




**Gambar II. 2**
**Logical Record Structure**

## C. *Unified Modelling Language* (UML)

"*Unified Modelling Language* (UML) adalah Bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi objek" (Nugroho Anggun, 2018).

UML adalah salah satu tool/model untuk merancang pengembangan software yang berbasis object-oriented. UML sendiri juga memberikan standar penulisan sebuah sistem blueprint, yang meliputi konsep proses bisnis, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen yang diperlukan dalam sistem software (Sonata, 2019).

"Tehnik pengembangan sistem yang menggunakan Bahasa grafis sebagai alat untuk mendokumentasi dan melakukan spesifikasi pada sistem" (Mulyani, 2016).

UML diaplikasikan untuk maksud tertentu sebagai berikut :

a) Merancang perangkat lunak.
b) Sarana komunikasi antara perangkat lunak dengan proses bisnis.
c) Menjabarkan sistem secara rinci untuk Analisa dan mencari apa yang diperlukan sistem.
d) Mendokumentasikan sistem yang ada proses-proses dan organisasinya.

### 1. *Activity Diagram*

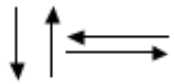
*Activity* Diagram adalah diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem.Diagram ini penting dalam

pemodelan fungsi-fungsi suatu sistem dan memberi tekanan pada aliran kendali antar objek.

Activity Diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem yang ada pada perangkat lunak

*Activity* diagram atau diagram aktivitas menggambarkan proses yang terjadi di dalam sebuah sistem dengan bentuk mengalir atau urut dati satu proses ke proses selanjutnya. Diagram ini khususnya menggambarkan aliran *use case* dimana interaksi antara pengguna normal dan alternatif terjadi

Alur kerja, logika, dan hubungan antara aktor dengan alur kerja dalam *use case* digambarkan dalam diagram yang disebut dengan aktivitas atau *activity diagram*

| NO | GAMBAR | NAMA | KETERANGAN |
|---|---|---|---|
| 1 | | *Activity* | Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain |
| 2 | | *Action* | State dari sistem yang mencerminkan eksekusi dari suatu aksi |
| 3 | | *Initial Node* | Bagaimana objek dibentuk atau diawali. |
| 4 | | *Activity Final Node* | Bagaimana objek dibentuk dan diakhiri |
| 5 | | *Decision* | Diguanakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu |
| 6 |  | *Line Connector* | Digunakan untuk menghubungkan satu simbol dengan simbol lainnya |

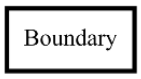
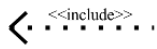
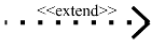
**Gambar II. 3**
**Simbol Activity Diagram**

**2.** ***Use Case*** Diagram

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 aktor dengan sistem informasi yang akan dibuat. Secara kasar, use case digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Diagram use case menyajikan interaksi antara use case dan aktor. Dimana aktor dapat berupa orang, peralatan atau sistem lain yang berinteraksi dengan sistem yang sedang dibangun. Use case menggambarkan fungsionalitas sistem atau persyaratan yang harus dipenuhi sistem dari pandangan"

| No | Gambar | Nama | Keterangan |
|---|---|---|---|
| 1. | | Actor | - Orang atau sistem lain yang berinteraksi dengan sistem saat ini.<br>- Sebuah peran, bukan pengguna spesifik.<br>- Memberikan input, menerima output, ataupun keduanya. |
| 2. | | Use Case | - Bagian utama dari fungsionalitas sistem.<br>- Bisa extend (memperluas) use case lainnya.<br>- Ditempatkan didalam boundary sistem (batasan sistem).<br>- Dilabeli dengan kata kerja-frase kata benda. |
| 3. |  | System Boundary | - Berisi nama dari sistem yang diletakkan didalam atau diluar bagian boundary.<br>- Mewakili ruang lingkup sistem.<br>- Actor berada diluar lingkup sistem. |
| 4. | | Association Relationship | - Menghubungkan actor dengan use case.<br>- Menunjukkan komunikasi dua arah.<br>- Tanda (*) untuk keragaman dari asosiasi (Multiplicity of the association).<br>- Namun umumnya hanya digambarkan garis. |
| 5. |  | Include Relationship | - Memasukkan satu use case kedalam use case lainnya.<br>- Perilaku (behavior) yang harus terpenuhi agar sebuah event dapat terjadi, dimana kondisi ini sebuah use case adalah bagian dari use case lainnya.<br>- Memberikan input, menerima output, ataupun keduanya.<br>- Tanda panah mengarah dari base use case menuju use case yang di include. |
| 6. |  | Extend Relationship | - Memperluas use case untuk memasukkan perilaku opsional.<br>- Tanda panah mengarah dari use case tambahan ke base use case. |

**Gambar II. 4**
**Simbol Use Case Diagram**

### 3. Class Diagram

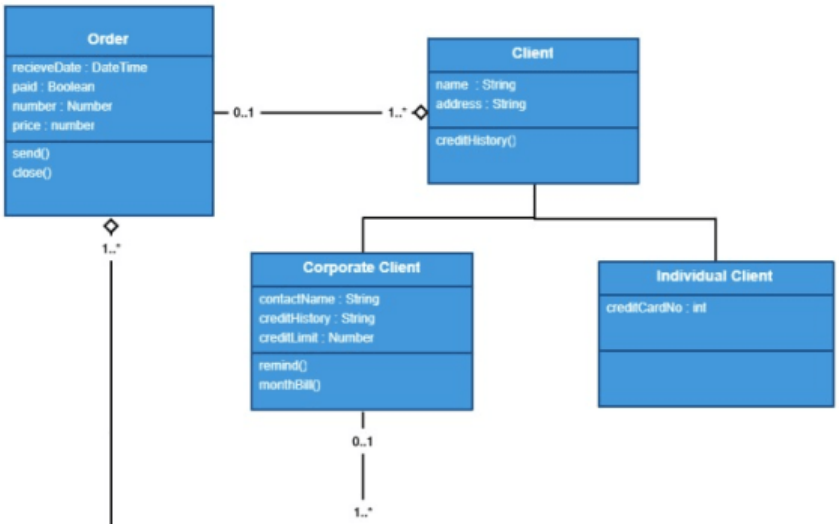
*Class Diagram* merupakan model statis yang menggambarkan kelas dan relasi antar kelas yang konstan di dalam sistem dari waktu ke waktu. Class diagram menggambaran kelas, yang mencakup perilaku dan kedudukan dengan hubungan antar kelas

*Class diagram* berfungsi untuk menjelaskan tipe dari objek sistem dan hubungannya dengan objek lain. Objek adalah nilai tertentu dari setiap atribut *class entity*. Adapun komponen *class diagram* adalah :

a) *Object* merupakan *instance* dari sebuah *class* dan dituliskan tersusun secara *horizontal*. Digambarkan sebagai sebuah class (kotak) dengan nama object di dalamnya yang diawali dengan sebuah titik koma.
b) *Class* merupakan blok-blok pembangun pada pemrograman berorientasi objek. Sebuah *class* digambarkan sebagai sebuah kotak yang terbagi dalam atas tiga bagian. Bagian atas adalah bagian nama dari *class*. Bagian tengah mendefinisikan properti atau *atribut class*. Bagian akhir mendefinisikan fungsi atau *method-method* dari sebuah *class*.
c) *Association* sebuah asosiasi merupakan sebuah *relationship* paling umum antara dua class dan dilambangkan oleh sebuah garis yang menghubungkan antara dua class. Garis ini bisa melambangkan tipe-tipe *relationship* dan juga dapat menampilkan hukum-hukum multiplisitas pada sebuah relationship. (Contoh: *One-to-one, one-to-many, many-to-many*).
d) *Dependency* kadangkala sebuah *class* menggunakan *class* yang lain. Hal ini disebut dependency. Umumnya penggunaan *dependency* digunakan untuk

menunjukkan operasi pada suatu *class* yang menggunakan *class* yang lain. Sebuah *dependency* dilambangkan sebagai sebuah panah bertitik-titik.

e) *Aggregation* mengindikasikan keseluruhan bagian *relationship* dan biasanya disebut sebagai relasi.



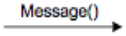

**Gambar II. 5**
**Class Diagram**

## 4. *Sequence* Diagram

> "Diagram sekuen menggambarkan kelakuan objek pada *user case* dengan mendeskripsikan waktu hidup objek dan *message* 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"* (Sukamto & Salahuddin, 2016)

"*Sequence* diagram menggambarkan bagaimana sistem merespon kegiatan *user*. *Sequence* diagram yang dibuat yaitu yang berhubungan langsung dengan kegiatan utama dari sistem informasi anggaran pendapatan dan belanja berbasis objek" (Irmayani & Susyatih, 2017).

Sequence diagram menggambarkan interaksi antara objek didalam dan sekitar sistem (termasuk pengguna dan display). Berupa messageyang digambarkan terhadap waktu. Squence diagram terdiri antar dimensi vertical (waktu) dan dimensi horizontal (obyek-obyek yang terkait).

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

**Gambar II. 6**
**Simbol Sequence Diagram**