

BAB II

LANDASAN TEORI

2.1 Pengertian Kos

Kos secara sederhana didefinisikan sebagai menempati satu ruang (kamar) rumah seseorang dengan perjanjian membayar dalam jumlah tertentu sebagai kompensasi sewa dan fasilitas lain didalamnya, seperti makan dan perabotan yang dipakai. Kos juga sering disebut rumah kedua untuk sebagian mahasiswa yang melanjutkan studinya ke perguruan tinggi yang lokasi perguruan tinggi tersebut berada diluar kota atau diluar pulau.

Menurut Wardhani & Nur (2017), Tempat kos merupakan suatu tempat tinggal yang disewakan kepada pihak lain dengan fasilitas-fasilitas tertentu dengan harga yang lebih terjangkau daripada di hotel/penginapan. Tempat kos lebih akrab digunakan sebagai domisili. Karena kebanyakan tempat kos disewa dalam jangka waktu yang cukup lama daripada hotel atau penginapan yang menggunakan hari. Dan juga istilah tempat kos sangatlah dekat dengan mahasiswa, Karena pada umumnya tempat kos disewakan untuk mahasiswa walaupun tidak jarang juga tempat kos yang disewakan untuk umum.

2.2 Sistem Pendukung Keputusan (SPK)

Menurut Wardhani & Nur (2017), Sistem Pendukung Keputusan (SPK) atau *Decision Support System* (DSS) adalah sebuah sistem yang mampu memberikan

kemampuan pemecahan masalah maupun kemampuan pengkomunikasian untuk masalah dengan kondisi semi terstruktur dan tak terstruktur. Sistem ini digunakan untuk membantu pengambilan keputusan dalam situasi semi terstruktur dan situasi yang tidak terstruktur, dimana tak seorangpun tahu secara pasti bagaimana keputusan seharusnya dibuat.

Menurut Elisabet Simbolon & Joslin Nababan (2018), Sistem Pendukung Keputusan (SPK) adalah sistem yang membantu untuk mengatasi masalah yang sepenuhnya terstruktur dan tidak terstruktur. Sistem pendukung keputusan (SPK) merupakan bagian dari sistem informasi berbasis komputer termasuk sistem berbasis pengetahuan atau manajemen pengetahuan yang dipakai untuk mendukung pengambilan keputusan dalam suatu organisasi atau perusahaan.

Menurut Nurhalimah, Tampubolon, Berutu, Simarmata, & mesran (2018), Sistem pendukung keputusan adalah bagian dari sistem informasi berbasis komputer (termasuk sistem pengetahuan) yang digunakan untuk memecahkan berbagai permasalahan keputusan dengan kriteria-kriteria yang berbeda. Metode ini merupakan pengambil keputusan yang menginginkan solusi yang mendekati ideal dan sesuai dengan kriteria yang ditetapkan. SPK merupakan penggabungan sumber- sumber kecerdasan individu dengan kemampuan komponen untuk memperbaiki kualitas keputusan.

2.3 Metode Fuzzy Tahani

2.3.1 Logika Fuzzy

Dikutip dari Astari & Komarudin (2018), Logika fuzzy adalah salah satu komponen pembentuk *soft-computing*, yang pertama kali diperkenalkan oleh Prof. Lotfi A. Zadeh pada tahun 1965. Dasar logika fuzzy adalah teori himpunan fuzzy yang terdapat peranan derajat keanggotaan untuk penentu keberadaan elemen dalam suatu himpunan yang sangat penting. Yang menjadi ciri utama dari logika fuzzy yaitu nilai keanggotaan (*membership function*)

Dikutip dari Syahroni & Rachmatullah (2018), Menurut teori logika fuzzy, kata fuzzy lebih dikenal sebagai sebuah *technical adjective*. Pemberian nama sistem fuzzy bukan berarti mengacu pada sistem yang kabur atau tidak jelas definisinya, deskripsinya dan cara kerjanya, namun yang dimaksud sistem fuzzy adalah sistem yang dibuat dengan definisi, deskripsi dan cara kerja yang jelas berdasarkan pada teori logika fuzzy. Yang ingin ditekankan disini adalah bahwa meskipun sebuah peristiwa yang akan dibuat menggunakan sistem fuzzy bersifat fuzzy namun sistem fuzzy yang dibuat tersebut harus tetap mempunyai definisi, deskripsi dan cara kerja yang jelas berdasarkan pada teori logika fuzzy.

2.3.2 Fuzzy Tahani

Dikutip dari Astari & Komarudin (2018), Fuzzy Tahani merupakan salah satu metode fuzzy yang menggunakan basis data standar. Pada basis data standar, data diklasifikasikan berdasarkan bagaimana data tersebut dipandang oleh *user* (pengguna). Oleh karena itu pada basis data standar

data yang ditampilkan akan keluar seperti data yang telah disimpan. Fuzzy *database* model Tahani masih menggunakan relasi standar, tetapi model Tahani ini menggunakan teori himpunan fuzzy pada suatu variabel untuk mendapatkan informasi pada *query*-nya. Sehingga pada pencarian data menggunakan rumus dari derajat keanggotaan pada suatu variabel himpunan fuzzy.

Berikut ini adalah tahapan logika fuzzy model Tahani :

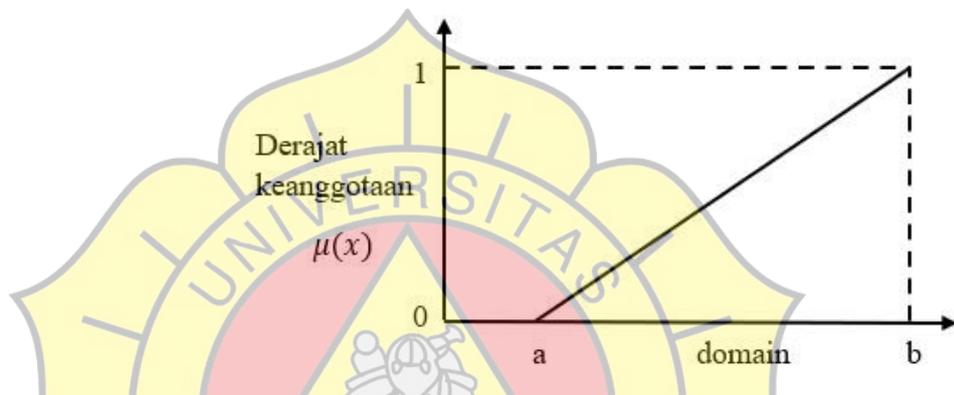
1. Menggambarkan fungsi keanggotaan (*membership function*) untuk setiap kriteria atau variabel fuzzy, ke dalam nilai keanggotaannya (derajat keanggotaan) yang memiliki interval antara 0 sampai 1, dengan pendekatan fungsi keanggotaan berbentuk kurva segitiga.

a. Fungsi Keanggotaan

Dikutip dari Susanti (2017), Fungsi keanggotaan atau sebagai *membership function* adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya yang memiliki interval antar 0–1.

Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan dengan melalui pendekatan fungsi. Macam-macam fungsi yang biasa digunakan dalam fuzzy yaitu representasi linier (linier naik & linier turun), representasi kurva segitiga, representasi kurva trapesium.

Representasi linier digambarkan sebagai suatu garis lurus. Bentuk ini paling sederhana tetapi mendekati suatu konsep yang tidak jelas. Ada dua keadaan himpunan fuzzy yang linier. Pertama, kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol bergerak ke kanan menuju nilai domain yang lebih tinggi.

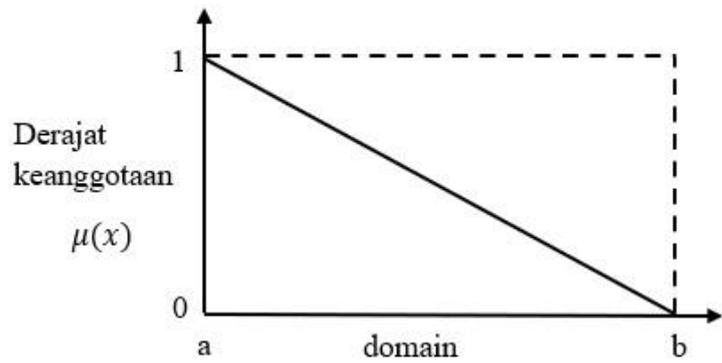


Gambar 2.1. Representasi Linier Naik

Fungsi untuk derajat keanggotaan dari linier naik dirumuskan sebagai berikut:

$$\mu(x) = \begin{cases} 0 & \text{jika } x \leq a \\ \frac{(x - a)}{(b - a)} & \text{jika } a \leq x \leq b \\ 1 & \text{jika } x \geq b \end{cases}$$

Kedua, merupakan kebalikan yang pertama yaitu garis lurus dimulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai yang lebih rendah.

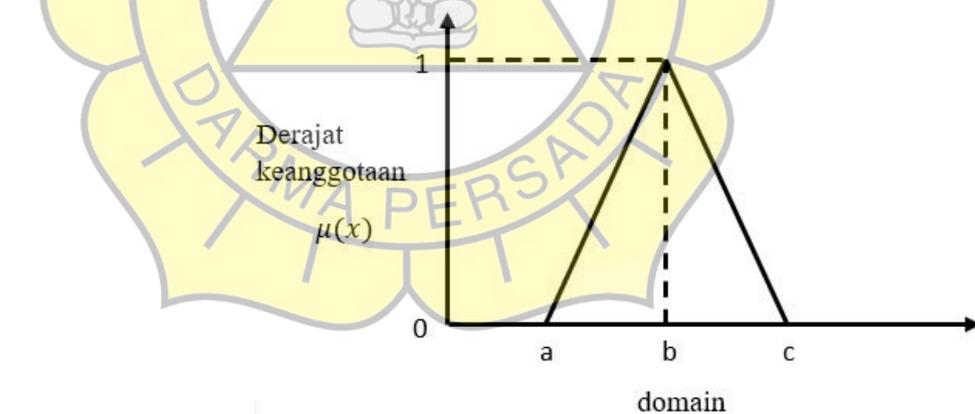


Gambar 2.2. Representasi Linier Turun

Fungsi untuk derajat keanggotaan dari linier turun dirumuskan sebagai berikut:

$$\mu(x) = \begin{cases} 1 & \text{jika } x \leq a \\ \frac{(b-x)}{(b-a)} & \text{jika } a \leq x \leq b \\ 0 & \text{jika } x \geq b \end{cases}$$

Representasi kurva segitiga merupakan gabungan dari representasi kurva linier naik dan kurva linier turun.

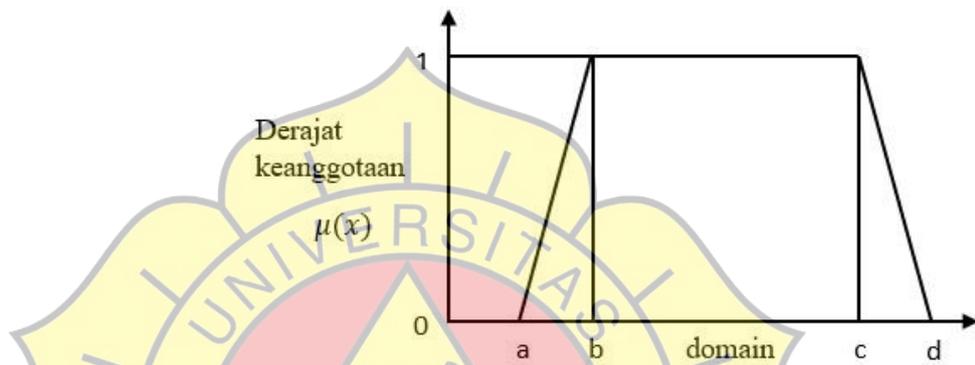


Gambar 2.3. Representasi Kurva Segitiga

Untuk representasi kurva segitiga fungsi keanggotaanya dirumuskan sebagai berikut:

$$\mu(x) = \begin{cases} 0 & \text{jika } x \leq a \text{ atau } x \geq c \\ \frac{(x-a)}{(b-a)} & \text{jika } a \leq x \leq b \\ \frac{(c-x)}{(c-b)} & \text{jika } b \leq x \leq c \end{cases}$$

Representasi kurva trapesium merupakan representasi kurva segitiga dengan beberapa titik (domain) memiliki derajat keanggotaan yang sama.



Gambar 2.4. Representasi Kurva Trapesium

Fungsi untuk derajat keanggotaan dari kurva trapesium dirumuskan berikut ini:

$$\mu(x) = \begin{cases} 0 & \text{jika } x \leq a \text{ atau } x \geq d \\ \frac{(x-a)}{(b-a)} & \text{jika } a \leq x \leq b \\ 1 & \text{jika } b \leq x \leq c \\ \frac{(d-x)}{(d-c)} & \text{jika } c \leq x \leq d \end{cases}$$

2. Fuzzifikasi, adalah fase pertama dari perhitungan fuzzy yaitu pengubahan nilai tegas ke nilai fuzzy. Dimana setiap variabel fuzzy dihitung nilai derajat keanggotaannya terhadap setiap himpunan fuzzy.
3. Fuzzifikasi Query, Diasumsikan sebuah query non konvensional (*nonfuzzy*) DBMS yang akan mencoba membuat dan

menerapkan sebuah sistem dasar logika fuzzy query atau disebut juga dengan pembentukan *query* dengan menggunakan relasi dasar.

Operator yang digunakan untuk relasi dasar dalam pembentukan *query* pada himpunan fuzzy yaitu sebagai berikut :

- a. Interseksi, operator ini berhubungan dengan operasi interseksi pada himpunan. α -predikat sebagai hasil operasi dengan operator AND diperoleh dengan mengambil nilai keanggotaan terkecil antar elemen pada himpunan-himpunan yang bersangkutan dengan persamaan berikut:

$$\mu_{A \cap B} = \min(\mu_{A(X)}, \mu_{B(Y)}) \dots$$

- b. Union, operator ini berhubungan dengan operasi union pada himpunan. A-predikat sebagai hasil operasi dengan operator OR diperoleh dengan mengambil nilai keanggotaan terbesar antar elemen pada himpunan-himpunan yang bersangkutan dengan persamaan berikut:

$$\mu_{A \cup B} = \max(\mu_{A(X)}, \mu_{B(Y)}) \dots$$

- c. Komplemen, operator ini berhubungan dengan operasi komplemen pada himpunan. A-predikat sebagai hasil operasi dengan operator NOT diperoleh dengan mengurangi nilai keanggotaan elemen pada himpunan yang bersangkutan dari 1 dengan persamaan berikut:

$$\mu_{\bar{A}} = 1 - \mu_{A(X)} \dots$$

4. Setelah diperoleh hasil operasi relasi dari pembentukan *query*, maka data hasil rekomendasi baik operator AND atau OR adalah nilai rekomendasi > 0 .

2.4 Metode VIKOR (*VIšekriterijumsko KOmpromisno Rangiranje*)

Dikutip dari Trisnani (2018), Metode VIKOR merupakan metode sederhana yang dapat dilakukan dan membantu pengambilan keputusan yang terbaik dari beberapa alternatif. Metode VIKOR diperkenalkan pertama kali oleh Serafim Opricovic pada tahun 1998.

Dikutip dari Nurhalimah, Tampubolon, Berutu, Simarmata, & mesran (2018), Metode VIKOR adalah salah satu metode dari *Multi-Criteria Decision Making* (MCDM) yang penyelesaiannya sederhana dan menghasilkan suatu keputusan yang lebih efektif. Pada metode VIKOR yang menjadi nilai terbaik adalah nilai yang paling rendah.

Dikutip dari Sutrikanti, Situmorang, Fachrurrazi, Nurdianto, & Mesran (2018), Metode VIKOR memiliki kelebihan dalam mengkompromi alternatif yang ada, serta dapat menyelesaikan pengambilan keputusan bersifat diskret pada kriteria yang bertentangan dan *non commensurable*, yaitu perbedaan unit antar kriteria.

Dikutip dari Nurhalimah, Tampubolon, Berutu, Simarmata, & mesran (2018), Langkah - langkah perhitungan dengan metode VIKOR, sebagai berikut:

1. Melakukan normalisasi menggunakan rumus sebagai berikut:

$$N_{ij} = \frac{(f_j^+) - (f_{ij})}{(f_j^+) - (f_j^-)}$$

Keterangan:

- a. f_j^+ = nilai terbaik dalam satu kriteria j
- b. f_j^- = nilai terjelek dalam satu kriteria j
- c. f_{ij} = fungsi respon alternatif i pada kriteria j
- d. $i = 1, 2, 3, \dots, m$ nomor urutan alternatif
- e. $j = 1, 2, 3, \dots, n$ nomor urutan kriteria

2. Menghitung nilai (*utility*)S dan (*regret*)R menggunakan rumus:

$$S_j = \sum_{j=1}^n \left[W_j \left(\frac{(f_j^+) - (f_{ij})}{(f_j^+) - (f_j^-)} \right) \right]$$

$$R_j = \text{MAX}_j \left[W_j \left(\frac{(f_j^+) - (f_{ij})}{(f_j^+) - (f_j^-)} \right) \right]$$

Keterangan:

- a. S_i (*maximum group utility*)
- b. R_i (*minimum individual regret of the opponent*)
- c. W_j = bobot kriteria

3. Menentukan nilai indeks

$$Q_i = v \left[\frac{S_i - S^-}{S^+ - S^-} \right] + (1 - v) \left[\frac{R_i - R^-}{R^+ - R^-} \right]$$

Keterangan:

- a. Q_i = nilai indeks VIKOR
- b. $i = 1, \dots, m$
- c. v = bobot dari *majority of criteria*, biasanya diberi nilai 0,5 namun bisa lebih besar atau lebih kecil tergantung pengambil keputusan.

- d. S^+ = nilai maksimal dari S_i .
- e. S^- = nilai minimal dari S_i .
- f. R^+ = nilai maksimal dari R_i .
- g. R^- = nilai minimal dari R_i .

4. Perangkingan alternatif

Hasil perangkingan merupakan hasil pengurutan dari S_i , R_i , dan Q_i . Solusi alternatif peringkat terbaik berdasarkan dengan nilai Q_i minimum menjadi peringkat terbaik dengan syarat:

- a. Kondisi 1 :

$$Q(A^{(2)}) - Q(A^{(1)}) \geq DQ \text{ dengan } DQ = \frac{1}{(n-1)}$$

Dimana $A^{(2)}$ = alternatif dengan urutan kedua pada perangkingan Q_i dan $A^{(1)}$ = alternatif dengan urutan terbaik pada perangkingan Q_i sedangkan n merupakan jumlah alternatif. Jika kondisi tidak memenuhi syarat, maka solusi kompromi dapat diajukan sebagai berikut:

Memilih alternatif A_1, A_2, \dots, A_n jika kondisi pertama tidak memuaskan. A_n merupakan alternatif yang ditentukan dengan menggunakan persamaan:

$$Q(A^{(n)}) - Q(A^{(1)}) \geq DQ \text{ dengan } DQ = \frac{1}{(n-1)}$$

- b. Kondisi 2 :

Diterima dengan melihat stabilitas perankingan alternatif. Stabilitas alternatif perankingan dinilai ketika nilai $v > 0,5$, atau $v \approx 0,5$, atau $v < 0,5$. Jika kondisi tidak memenuhi syarat, maka solusi kompromi dapat diajukan

sebagai berikut: Memilih alternatif A_1 dan A_2 jika hanya kondisi 2 tidak memuaskan.

2.5 Bahasa Pemrograman

2.5.1 Definisi Bahasa Pemrograman

Menurut Budiman (2017), Bahasa pemrograman atau bahasa komputer merupakan teknik komunikasi standar untuk memerintah komputer. Bahasa pemrograman adalah suatu himpunan aturan sintaks dan semantik yang dipakai untuk mendefinisikan program komputer. Suatu bahasa memungkinkan seorang *programmer* secara persis menentukan data yang mana akan diolah oleh komputer, bagaimana data ini akan disimpan/diteruskan, dan langkah apa secara tepat yang akan diambil dalam berbagai situasi.

2.5.2 Bahasa Pemrograman Dalam Perancangan Website

Ada beberapa unsur-unsur bahasa pemrograman yang digunakan penulis dalam perancangan *website* terdiri dari PHP, HTML, dan *Javascript*.

2.5.2.1 PHP

Menurut Wibawa & Rajab (2017), PHP (*Hypertext Preprocessor*) adalah Bahasa *server-side –scripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis. Karena

PHP merupakan *server-side-scripting* maka sintaks dan perintah-perintah PHP akan dieksekusi di server kemudian hasilnya akan dikirimkan ke *browser* dengan format HTML.

Menurut Rubiati (2018), PHP merupakan bahasa berbentuk *script* yang ditempatkan di dalam server baru kemudian diproses. Kemudian hasil pemrosesan dikirim kepada web browser klien. Bahasa pemrograman ini dirancang khusus untuk membentuk web dinamis.

Tabel 2.1. Tabel penjelasan PHP

<i>Include</i>	Memanggil file
<?php	Awal dari PHP
?>	Penutup dari PHP

2.5.2.2 HTML

Menurut Dhika, Lukman, Fitriansyah, (2016), HTML mempunyai kepanjangan dari *Hyper Text Markup Language*, yaitu suatu bahasa pemrograman *hyper text*. HTML memiliki fungsi untuk membangun kerangka ataupun format *web* berbasis HTML. Dokumen HTML adalah *text file* murni yang dapat dibuat dengan *editor teks* sembarang. Dokumen ini dikenal sebagai *web page*. *File-file* HTML ini berisi instruksi-instruksi yang kemudian diterjemahkan oleh *browser* yang ada di komputer *client (user)*

sehingga isi informasinya dapat di tampilkan secara visual di komputer pengguna.

Menurut Mariko (2019), *Hypertext Markup Language* (HTML) adalah bahasa standar yang digunakan untuk menampilkan konten pada halaman website. Fungsi-fungsi yang dapat dilakukan dengan bahasa programan HTML adalah mengatur serta mendesain tampilan isi halaman website, membuat tabel pada halaman website, mempublikasikan halaman website secara *online*, membuat form yang dapat menjadi input serta menangani *registrasi* dan transaksi via website, menampilkan area gambar pada browser.

Tabel 2.2. Tabel penjelasan HTML

<html> dan </html>	Tanda awal mulai dan akhir HTML
<head> dan </head>	Letak informasi <i>page header</i>
<body> dan </body>	Letak atribut halaman seperti warna latar belakang, isi konten, berbagai <i>link</i> , dll.

2.5.2.3 JavaScript

Secara umum *JavaScript* adalah bahasa pemrograman tingkat tinggi yang digunakan dalam pengembangan website agar lebih dinamis dan interaktif.

Menurut Firdaus & Rachmanto (2019), *Javascript* adalah bahasa pemrograman tingkat tinggi dan dinamis. *JavaScript* populer di internet dan dapat bekerja di sebagian besar penjelajah web populer seperti Google Chrome, Internet Explorer, Mozilla Firefox, Netscape dan Opera. Kode *JavaScript* dapat disisipkan dalam halaman web menggunakan tag *SCRIPT*, yang dulu masih menggunakan PHP (Hypertext Preprocessor) sebagai bahasa mesinnya, PHP pun tidak begitu jadal akan tetapi kalo di bandingkan dengan bahasa mesin *Javascript*, lebih efektif dalam segi kecepatan sistemnya maka *Javascript* lebih unggul.

Menurut Variastuti & Subkhi (2019), *Javascript* termasuk kedalam bahasa pemrograman tingkat tinggi atau sering disebut dengan *high level language* yang artinya bahasanya mendekati bahasa manusia sehingga memudahkan pengguna untuk mengerti bahasa pemrograman *javascript*. *Javascript* juga memiliki kelebihan karena pengguna tidak perlu membeli lisensi, serial number dan *software* karena dapat diakses dengan mudah di web browser. Untuk pembuatan kode program *javascript* tidak memerlukan alat pengembangan yang mahal hanya dengan memiliki *editor text* seperti notepad sudah bisa untuk membuat kode *javascript*.

2.6 Konsep Basis Data (*Database*)

Basis data pada dasarnya dapat dianggap sebagai tempat atau lokasi untuk sekumpulan berkas data yang sudah terkomputerisasi dengan tujuan untuk memelihara informasi, dan juga memuat informasi tersebut, terutama apabila informasi tersebut sedang dibutuhkan

2.6.1 Definisi Basis Data (*Database*)

Menurut Rubiati (2018), *Database* sering didefinisikan sebagai kumpulan data yang terkait. Secara teknis, yang berada dalam sebuah *database* adalah sekumpulan tabel atau objek lain (*indeks, view, dan lain – lain*). Tujuan utama pembuatan *database* adalah untuk memudahkan dalam mengakses data. Data dapat ditambahkan, diubah, atau dibaca dengan relatif mudah dan cepat.

Menurut Fitri, Rukun, & Dwiyani (2016), dengan penggunaan *database* dalam penyimpanan data memberikan kerapian dan keamanan data. Sebab dengan menggunakan *database* data yang tersimpan terhindar dari redudansi dan kehilangan data. Karena data dapat disimpan dalam waktu yang lama dan mudah untuk dicari jika dibutuhkan dibandingkan dengan penyimpanan data menggunakan buku. Penggunaan *database* berbentuk *online* juga memberikan kelebihan yang lebih baik dibandingkan dengan menggunakan penyimpanan data berupa *hardisk*. Hal ini dikarenakan resiko kehilangan data atau data rusak akan lebih kecil, serta data tersebut dapat diakses kapanpun dan dimanapun.

2.6.2 phpMyAdmin

Pada umumnya phpMyAdmin berfungsi sebagai salah satu aplikasi untuk mengelola *database*. phpMyAdmin digunakan untuk membantu administrasi *database* MySQL yang dibuat oleh phpMyAdmin.net.

Menurut Erma Standsyah, Rahmawati, & Intannia Sari Restu (2017), PhpMyAdmin adalah sebuah aplikasi/perangkat lunak bebas (*opensource*) yang ditulis dalam bahasa pemrograman PHP yang digunakan untuk menangani administrasi *database* MySQL melalui jaringan lokal maupun internet. phpMyAdmin mendukung berbagai operasi MySQL, diantaranya (mengelola basis data, tabel-tabel, bidang (*fields*), relasi (*relations*), indeks, pengguna (*users*), perijinan (*permissions*), dan lain-lain). Perbedaan phpMyAdmin dengan MySQL terletak pada fungsi. PhpMyAdmin merupakan alat untuk memudahkan dalam mengoperasikan *database* MySQL, sedangkan MySQL adalah *database* tempat penyimpanan data. Phpmyadmin sendiri digunakan sebagai alat untuk mengolah/ mengatur data pada MySQL

2.7 Software Pendukung

2.7.1 MySQL

Menurut Sihotang (2019), MySQL (*My Structure Query Language*) adalah sebuah *software database*, yang merupakan tipe data relasional yang artinya MySQL penyimpanan datanya dalam bentuk tabel-tabel yang saling berhubungan.

Menurut Dhika, Lukman, Fitriansyah, (2016), MySQL merupakan *databases* yang sangat populer. Beberapa keuntungan yang dimiliki MySQL yaitu: bersifat open source, menggunakan bahasa SQL (structure query language), *super performance* dan *reliable*, Mudah dipelajari, mampu bekerja dilintas *platform* dan *multi user*.

2.7.2 XAMPP

Menurut Sihotang (2019), XAMPP adalah salah satu paket instalasi apache, PHP, dan MySQL secara instant yang dapat digunakan untuk membantu proses instalasi ketiga produk tersebut.

Menurut Trimarsiah & Arafat (2017), XAMPP adalah sebuah *software* yang berfungsi untuk menjalankan website berbasis PHP dan menggunakan pengolah data MYSQL di komputer lokal. XAMPP adalah proyek non profit yang di kembangkan oleh apache friend yang didirikan oleh kai oswalad seilder dank ayvogelgesang pada tahun 2002 proyek ini bertujuan mempromosikan penggunaan apache web server Paket amp (apache, mysql, php), salah satunya adalah XAMPP yang sudah terintegasi amp di dalamnya dan menghemat *resource* computer daripada menginstal amp satu persatu.

Penulis menyimpulkan bahwa XAMPP merupakan *software* sebagai tool pembantu untuk pengembangan paket perangkat lunak berbasis *open source* yang menggabungkan Apache web server, MySQL, PHP dan beberapa modul lainnya di dalam satu paket aplikasi.



Gambar 2.5. XAMPP

2.7.3 Google Chrome

Google Chrome merupakan salah satu dari sekian banyak program aplikasi web browser yang tersedia pada saat ini dan biasanya digunakan untuk menjelajahi konten atau isi yang ada pada media *World Wide Web* (WWW) dengan memanfaatkan jaringan internet.

Dikutip dari Mustika, Mardian, & Rinawati (2018), menjelaskan bahwa web browser merupakan aplikasi untuk dapat menjelajah internet, yang berfungsi menampilkan dan melakukan interaksi dengan server. Berkembangnya internet dan semakin banyaknya website yang ada, menjadikan internet sebagai sumber informasi yang utama. Web browser juga merupakan perangkat lunak yang fungsinya menerima dan menyajikan informasi di internet. Sumber informasi bisa dilihat menggunakan web browser, baik yang berupa halaman web, gambar, video, atau jenis konten lainnya.

Dikutip dari Mustika, Mardian, & Rinawati (2018), Google Chrome dibentuk dan dikembangkan oleh sang raksasa mesin pencari Google. Browser ini dirilis pada 2 September 2008, dan pada saat itu hanya digunakan untuk Microsoft Windows karena masih dalam status beta. Google chrome menjadi browser terpopuler karena banyaknya fitur yang tersedia didalamnya dan kecepatan respon yang dimilikinya.

2.8 UML (*Unified Modelling Language*)

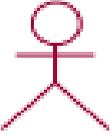
Menurut Sihotang (2019), UML (*Unified Modeling Language*) memiliki sintaks dan semantic. Ketika kita membuat model menggunakan konsep UML (*Unified Modeling Language*) ada aturan-aturan yang harus diikuti. Bagaimana elemen pada model-model yang kita buat berhubungan satu dengan lainnya harus mengikuti standar yang ada.

2.8.1 Use Case Diagram

Menurut Fadila (2021), Use Case Diagram menggambarkan interaksi secara usecase dan actor. Use case mempresentasikan fungsionalitas sistem dan kebutuhan sistem dari sudut pandang pengguna.

Menurut Febriandirza (2020), Diagram ini bersifat statis. Diagram ini memperlihatkan komponen Use Case dan aktor-aktor. Diagram ini sangat penting untuk mengorganisasikan dan memodelkan perilaku dari sistem yang dibutuhkan dan diharapkan pengguna

Tabel 2.3. Tabel komponen use case diagram

Komponen <i>Use Case</i>	Penjelasan
	<i>Actor</i> , Merupakan sebuah komponen yang menggambarkan seseorang atau sesuatu (seperti perangkat atau sistem lainnya) yang berinteraksi dengan sistem
	<i>Use case</i> adalah gambaran fungsional dari suatu sistem, sehingga pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun.

Tabel 2.4. Tabel relasi use case diagram

Relasi <i>Use Case</i>	Penjelasan
 <p>Relasi <i>Association</i></p>	<i>Association</i> , menghubungkan <i>link</i> antara elemen.
 <p>Relasi <i>Generalization</i></p>	<i>Generalization</i> disebut juga <i>inheritance</i> (pewarisan), sebuah elemen dapat merupakan spesialisasi dari elemen lainnya.
 <p>Relasi <i>Dependency</i></p>	<i>Dependency</i> , sebuah elemen bergantung dalam beberapa cara ke elemen lainnya.

Tabel 2.5. Tabel *stereotype* yang mungkin terjadi pada use case diagram

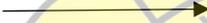
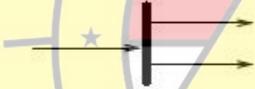
Relasi / <i>Stereotype</i>	Penjelasan
<< <i>include</i> >>	Kelakuan yang harus terpenuhi agar sebuah <i>event</i> dapat terjadi, dimana pada kondisi ini sebuah <i>use case</i> adalah bagian <i>use case</i> lainnya.
<< <i>extends</i> >>	Kekuan yang hanya berjalan dibawah kondisi tertentu seperti menggerakkan <i>alarm</i> .
<< <i>Communicates</i> >>	Ditambahkan untuk asosiasi yang mungkin menunjukkn asosiasinya adalah <i>communication association</i> . Ini merupakan pilihan selama asosiasi hanya tipe <i>relationship</i> yang dibolehkan antara <i>actor</i> dan <i>use case</i> .

2.8.2 Activity Diagram

Menurut Fitri, Rukun, & Dwiyani (2016), Activity diagram menggambarkan proses dan urutan aktivitas dalam suatu proses sistem informasi. Activity diagram ini dapat membantu dalam memahami proses secara keseluruhan.

Menurut Febriandirza (2020), Diagram ini bersifat dinamis. Diagram ini memperlihatkan aliran dari suatu aktifitas ke aktifitas lainnya dari suatu sistem. Diagram ini penting dalam pemodelan fungsi-fungsi dalam suatu sistem dan memberi tekanan pada aliran kendali antar objek.

Tabel 2.6. Tabel komponen activity diagram

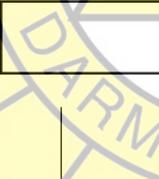
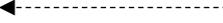
<i>Activity Diagram</i>	Penjelasan
 <i>Start State</i>	<i>Start State</i> , sebagai tanda awal proses dari <i>activity diagram</i> .
 <i>State</i>	<i>State</i> , berfungsi menampung event dalam <i>activity diagram</i>
 <i>Activity</i>	<i>Activity</i> , memiliki fungsi yang sama dengan state. Menampung <i>event</i> atau aktivitas pada proses sistem.
 <i>State Transition</i>	<i>State Transition</i> , berfungsi untuk menunjukkan aliran atau urutan dari <i>event</i> atau aktivitas pada diagram.
 <i>Fork</i>	<i>Fork</i> , Menggambarkan aktivitas yang dimulai dengan sebuah aktivitas dan diikuti oleh dua atau lebih aktivitas yang harus dikerjakan.
 <i>Decision</i>	<i>Decision</i> , digunakan ketika terjadi pemilihan 2 kondisi <i>event</i> pada diagram.
 <i>End State</i>	<i>End State</i> , sebagai tanda akhir dari <i>activity diagram</i> .

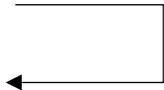
2.8.3 Sequence Diagram

Menurut Fitri, Rukun, & Dwiyani (2016), Sequence diagram menjelaskan interaksi objek yang disusun dalam suatu urutan waktu, yaitu berupa urutan kejadian yang dilakukan oleh seorang *user* yang menjalankan sistem. Sequence diagram menunjukkan operasi secara detail.

Menurut Febriandirza (2020), Sequence diagram adalah salah satu jenis diagram pada UML menjelaskan interaksi objek yang berdasarkan urutan waktu, sequence diagram menggambarkan tahapan atau urutan yang harus dilakukan agar menghasilkan sesuatu seperti pada use case diagram.

Tabel 2.7. Tabel komponen sequence diagram

<i>Sequence Diagram</i>	Penjelasan
	<i>Actor</i> , menggambarkan seseorang atau sesuatu (seperti perangkat atau sistem lain) yang berinteraksi dengan sistem.
	<i>General/Object</i> , sebuah kontak yang tampil pada posisi paling atas diagram, yang mewakili <i>object</i> , <i>use case</i> , <i>class</i> , dan <i>actor</i> .
	<i>Message (call)</i> , menggambarkan pesan atau hubungan antara obyek yang menunjukkan urutan kejadian yang terjadi.
	<i>Lifeline</i> , eksekusi objek selama sequence (<i>message</i> dikirim atau diterima dan aktifitasnya).

	<p><i>Message Return</i>, menggambarkan pesan atau hubungan obyek itu sendiri yang menunjukkan urutan kejadian yang terjadi.</p>
	<p><i>Object Message</i>, menggambarkan pesan atau hubungan antar obyek yang menunjukkan urutan kejadian yang terjadi.</p>

