

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Penelitian terkait yang dilakukan dengan analisa metode SMART (*Simple Multi Attribute Rating Technique*), diantaranya adalah :

Penelitian yang dilakukan Bambang TJ Hutagalung (2021) yang mengangkat judul “Penerapan Metode SMART dalam Seleksi Penerima Bantuan Sosial Warga Masyarakat Terdampak COVID-19”. Hasil dari penelitian ini adalah menunjukkan bahwa penentuan bobot yang tepat pada setiap kriteria sangat mempengaruhi hasil perhitungan terhadap rekomendasi pemberian bantuan dana sosial bagi masyarakat terdampak Covid-19. Kemudian agar dapat memperoleh hasil yang lebih akurat maka perlu dilakukan uji validitas terhadap kriteria untuk memperoleh kriteria yang lebih tepat sesuai dengan kebutuhan seleksi kelayakan penerima bantuan dana sosial bagi warga masyarakat terdampak Covid-19 dari PW Tabita.

Penelitian yang dilakukan Gumilar Ramadhan Pangaribuan (2019) yang mengangkat judul “Pemilihan Jenis Sapi bagi Peternak Sapi Potong dengan Metode SMART”. Hasil dari penelitian ini adalah merekomendasikan jenis sapi terbaik dengan menggunakan metode SMART. Sumber data diperoleh dengan cara wawancara dan pemberian angket kepada 25 peternak sapi potong secara random di desa parbalogan, Kecamatan Tanah Jawa, Kabupaten Simalungun. Adapun enam jenis sapi (alternatif) yang digunakan seperti Lemosin (A1),

Simental (A2), Bali (A3), Perah (A4), Brahma (A5) dan Madras (A6). Sementara kriteria penilaian yang digunakan, yaitu: Asal (C1), Harga (C2), Usia (C3), Bobot (C4), dan Ukuran (C5). Hasil dari penelitian menyebutkan bahwa jenis Sapi Lemosin (A1) menjadi rekomendasi pertama dengan nilai akhir 1 dan jenis sapi bali (A3) sebagai rekomendasi kedua dengan nilai akhir 0,702543.

Penelitian yang dilakukan Hidayatus Sibyan (2019) yang mengangkat judul “Implementasi Metode Smart Pada Sistem Pendukung Keputusan Penerima Beasiswa Sekolah”. Hasil dari penelitian ini adalah untuk membuat sistem yang bisa mempermudah proses seleksi beasiswa agar penerima beasiswa sesuai dengan urutan prioritas yang tepat. Dalam membangun Sistem Pendukung Keputusan (SPK) ini digunakan metode SMART yang merupakan singkatan dari *Simple Multi Attribute Rating Technique*, yaitu cara pengambilan keputusan dengan menghitung bobot kriteria pada masing-masing alternatif. Hasil penelitian telah menghasilkan sebuah SPK dengan metode SMART yang dapat membantu pimpinan SMK Gema Nusantara dalam menentukan siswa penerima beasiswa.

Penelitian yang dilakukan Mohammad Guntur (2019) yang mengangkat judul “Penerapan Metode SMART untuk Seleksi Kelayakan Penerima Bantuan Pengembangan Usaha Pangan Masyarakat”. Hasil dari penelitian ini adalah bertujuan untuk membantu menyelesaikan masalah seleksi pemberian bantuan dana pengembangan usaha pangan masyarakat agar tepat sasaran, adapun metode yang digunakan adalah metode SMART dimana metode ini mampu melakukan proses perangkingan terhadap alternatif dengan kriteria yang telah dipilih sebagai hasil penilaian berdasarkan kebutuhan seleksi yaitu Legalitas, pengalaman

kegiatan dagang, memiliki AD/ART, mesin penggiling dan luas gudang penyimpanan. Hasil dari analisa menggunakan metode SMART dengan memperhatikan kriteria yang dipilih diketahui bahwa kelompok tani yang sangat layak untuk menerima bantuan pengembangan usaha pangan adalah kelompok tani rukun tani dengan bobot nilai 94, sedangkan rekomendasi layak dan dipertimbangkan adalah kelompok tani karya bersama dan kelompok tani usaha mandiri dengan bobot nilai masing masing yaitu 82,75 dan 74. Sehingga dari hasil analisa menggunakan metode SMART dapat membantu pihak pengambil keputusan dalam memilih kelompok tani yang layak menerima bantuan pengembangan usaha pangan masyarakat dengan tepat dan objektif.

Penelitian yang dilakukan Sundari Retno Andani (2019) yang mengangkat judul “Penerapan Metode SMART Dalam Pengambilan Keputusan Penerima Beasiswa Yayasan AMIK Tunas Bangsa”. Hasil dari penelitian ini adalah menentukan penerima beasiswa yayasan. Metode penelitian yang diterapkan dalam menentukan penerima beasiswa yayasan AMIK Tunas Bangsa adalah metode SMART (*Simple Multi Attribute rating Technique*), karena metode ini mampu menyelesaikan masalah pengambilan keputusan multikriteria berdasarkan teori bahwa setiap alternatif terdiri dari sejumlah kriteria yang memiliki nilai-nilai dan setiap kriteria memiliki bobot yang menggambarkan seberapa penting kriteria tersebut dengan kriteria lain. Kriteria yang digunakan dalam menentukan penerima beasiswa yayasan pada penelitian ini adalah IPK, penghasilan orang tua dan jumlah tanggungan. Hasil dari penelitian ini adalah

rekomendasi penerima beasiswa. Dengan menggunakan metode SMART, maka rekomendasi yang dihasilkan lebih akurat.

Sedangkan penelitian yang dilakukan oleh penulis yang mengangkat judul “Sistem Pendukung Keputusan Penerimaan Bantuan UMKM Pada Kecamatan Medan Timur Menggunakan Metode SMART”. Perbedaan penelitian yang penulis lakukan dengan penelitian sebelumnya adalah penulis mengangkat masalah dalam penentuan penerimaan bantuan UMKM sedangkan pada penelitian sebelumnya membahas mengenai masalah seleksi pemberian bantuan pada pengembang usaha. Sehingga hasil penelitian yang dilakukan oleh penulis merancang sistem pendukung keputusan penerimaan bantuan UMKM berbasis web yang menggunakan bahasa pemrograman *PHP* dan *Mysql*. Dalam perancangan ini dengan penelitian sebelumnya sama-sama menerapkan suatu metode yaitu metode SMART (*Simple Multi Attribute Rating Technique*). Metode SMART lebih sering digunakan karena kesederhanaannya dalam merespon kebutuhan pembuatan keputusan dan menganalisis. Tujuan dari penelitian ini adalah dapat menjadi solusi dalam memecahkan permasalahan yang dihadapi pihak instansi. Harapan hasil dari penelitian dapat menjadi *output* atau keluaran yang bermanfaat bagi pihak terkait dalam jangka yang panjang.

II.2. Landasan Teori

Untuk mendukung keberhasilan penelitian ini, penyusun melakukan pendekatan teoritis melalui beberapa literatur yang berhubungan dengan penelitian yang dilakukan. Beberapa landasan teori pada penelitian ini yaitu:

II. 2.1 Sistem Pendukung Keputusan (SPK)

Sistem pendukung keputusan (SPK) merupakan sistem inFormasi interaktif yang menyediakan informasi, pemodelan, dan pemanipulasian data. Sistem ini digunakan untuk membantu pengambilan keputusan dalam situasi yang semiterstruktur dan situasi yang tidak terstruktur. Tujuan dari SPK adalah untuk membantu pengambil keputusan memilih berbagai alternatif keputusan yang merupakan pengolahan informasi-informasi yang diperoleh atau tersedia dengan menggunakan model pengambilan keputusan. Ciri utama sekaligus keunggulan dari sistem pendukung keputusan tersebut adalah kemampuannya untuk menyelesaikan masalah-masalah yang tidak erstruktur. Pengambilan keputusan merupakan proses pemilihan alternatif tindakan untuk mencapai tujuan atau sasaran tertentu. Pengambilan keputusan dilakukan dengan pendekatan sistematis terhadap permasalahan melalui proses pengumpulan data menjadi informasi serta ditambah dengan faktor-faktor yang perlu dipertimbangkan dalam pengambilan keputusan. Tahap-tahap yang harus dilalui dalam proses pengambilan keputusan sebagai berikut :

- a. Tahap Pemahaman (*Intelligence Phase*) Tahap ini merupakan proses penelusuran dan pendeteksian dari lingkup problematika serta proses pengenalan masalah. Data masukan diperoleh, diproses, dan diuji dalam rangkamengidentifikasi masalah.
- b. Tahap Perancangan (*Design Phase*) Tahap ini merupakan proses pengembangan dan pencarian alternatif tindakan atau solusi yang dapat diambil. Tersebut merupakan representasi kejadian nyata yang

disederhanakan, sehingga diperlukan proses validasi dan verifikasi untuk mengetahui keakuratan model dalam meneliti masalah yang ada.

- c. Tahap Pemilihan (*Choice Phase*) Pada tahap ini dilakukan pemilihan terhadap berbagai alternatif solusi yang dimunculkan pada tahap perencanaan agar ditentukan atau dengan memperhatikan kriteria-kriteria berdasarkan tujuan yang akan dicapai. Tahap Implementasi (*Implementation Phase*) Pada tahap ini dilakukan penerapan terhadap rancangan sistem yang telah dibuat pada tahap perancangan serta pelaksanaan alternatif tindakan yang telah dipilih pada tahap pemilihan, (Dyna Marisa Khairina; 2016 : 17).

II.2.2 UMKM (Usaha Mikro Kecil dan Menengah)

UMKM adalah unit usaha produktif yang berdiri sendiri, yang dilakukan oleh orang perorangan atau Badan Usaha disemua sektor ekonomi (Tambunan, 2012). Pada prinsipnya, pembedaan antara Usaha Mikro (UMi), Usaha Kecil (UK), Usaha Menengah (UM) dan Usaha Besar (UB) umumnya didasarkan pada nilai asset awal (tidak termasuk tanah dan bangunan), omset rata-rata pertahun atau jumlah pekerja tetap. Namun definisi UMKM berdasarkan ketiga alat ukur ini berbeda disetiap Negara. Di Indonesia, definisi UMKM diatur berdasarkan Undang-Undang Republik Indonesia Nomor 20 Tahun 2008 tentang Usaha Mikro, Kecil, dan Menengah. Definisi menurut UU No. 20 Tahun 2008 tersebut adalah :

- a. Usaha Mikro adalah usaha produktif milik orang perorangan dan/atau badan usaha perorangan yang memenuhi kriteria Usaha Mikro sebagaimana diatur dalam Undang-Undang ini.
- b. Usaha Kecil adalah usaha ekonomi produktif yang berdiri sendiri yang dilakukan oleh orang perorangan atau badan usaha yang bukan merupakan anak perusahaan atau bukan cabang perusahaan yang dimiliki, dikuasai atau menjadi bagian baik langsung maupun tidak langsung dari usaha menengah atau usaha besar yang memenuhi kriteria Usaha Kecil sebagaimana dimaksud dalam Undang-undang.
- c. Usaha Menengah adalah usaha ekonomi produktif yang berdiri sendiri yang dilakukan oleh orang perseorangan atau badan usaha yang bukan merupakan anak perusahaan atau cabang perusahaan yang dimiliki, dikuasai, atau menjadi bagian baik langsung maupun tidak langsung dengan Usaha Kecil atau usaha besar dengan jumlah kekayaan bersih atau hasil penjualan tahunan sebagaimana diatur dalam Undang-undang. (Nirfandi Gonibala: 2019).

II.2.3 Metode *Simple Multi Attribute Rating Technique* (SMART)

Simple Multi-Attribute Rating Technique (SMART) merupakan metode pengambilan keputusan multikriteria yang dikembangkan oleh Edward pada tahun 1977. SMART merupakan teknik pengambilan keputusan multikriteria yang didasarkan pada teori bahwa setiap alternatif terdiri dari sejumlah kriteria yang memiliki nilai-nilai dan setiap kriteria memiliki bobot yang menggambarkan

seberapa penting ia dibandingkan dengan kriteria lain. Pembobotan ini digunakan untuk menilai setiap alternatif agar diperoleh alternatif terbaik.

SMART menggunakan *linear additive model* untuk meramal nilai setiap alternatif. SMART merupakan metode pengambilan keputusan yang fleksibel. SMART lebih banyak digunakan karena kesederhanaannya dalam merespon kebutuhan pembuat keputusan dan caranya menganalisa respon. Analisa yang terlibat adalah transparan sehingga metode ini memberikan pemahaman masalah yang tinggi dan dapat diterima oleh pembuat keputusan. Pembobotan pada SMART menggunakan skala antara 0 sampai 1, sehingga mempermudah perhitungan dan perbandingan nilai pada masing-masing alternatif.

Model fungsi *Utility linear* yang digunakan oleh SMART seperti pada Persamaan

$$SMART = \sum_{j=1}^k w_j u_{ij} \dots\dots\dots (1)$$

Di mana :

w_j adalah nilai pembobotan kriteria ke-j dari k kriteria,

u_{ij} adalah nilai *Utility* alternatif i pada kriteria j.

Pemilihan keputusan adalah mengidentifikasi mana dari n alternatif yang mempunyai nilai fungsi terbesar. Nilai fungsi ini juga dapat digunakan untuk meranking alternatif.

Langkah-langkah dalam melakukan metode ini adalah:

1. w_j adalah nilai pembobotan kriteria ke -j dari k kriteria,
2. U_{ji} adalah nilai *Utility* alternatif i pada kriteria j,

3. Pemilihan keputusan adalah mengidentifikasi mana dari n alternatif yang mempunyai nilai fungsi terbesar.
4. Nilai fungsi ini dapat digunakan untuk meranking n alternatif

Menghitung nilai normalisasi bobot, seperti pada Persamaan :

$$Nw_j = \frac{w_j}{\sum_{j=1}^k w_j} \dots\dots\dots (2)$$

Di mana:

Nw_j adalah normalisasi bobot kriteria ke $-j$

W_j adalah nilai bobot kriteria ke $-j$

K adalah jumlah kriteria

n adalah bobot kriteria ke $-n$

Menghitung nilai *Utility*, seperti pada Persamaan

$$u_{ij} = (v_{ij}) \dots\dots\dots (3)$$

Di mana:

U_{ij} adalah *Utility* kriteria ke $-j$ untuk alternatif i

V_{ij} adalah nilai kriteria ke $-j$ untuk alternatif i

$F(v_{ij})$ adalah fungsi kriteria ke $-j$ untuk alternatif i , (M. Safii; 2018).

II.2.3.1 Studi Kasus Metode SMART

Penelitian yang dilakukan oleh Gumilar Ramadhan Pangaribuan (2019) yang mengangkat judul “Pemilihan Jenis Sapi Bagi Peternak Sapi Potong Dengan Metode SMART”. Dalam penerapan metode SMART untuk penentuan jenis sapi terbaik ada beberapa hal yang perlu dilakukan sebelum melakukan perhitungan nilai, salah satunya yaitu :

1. Identifikasi Kriteria

Dalam sistem penilaian ada 5 kriteria yang didapatkan melalui hasil wawancara dengan peternak sapi yaitu asal, harga, usia, bobot, dan ukuran.

2. Pemberian Bobot Kriteria

Pemberian bobot kriteria didapat melalui hasil wawancara penilaian dan berdasarkan pada pemberian bobot terbesar hingga bobot terkecil dengan interval 1-100 dan dijadikan nilai default pada sistem. Kemudian semua nilai tersebut dijumlahkan.

Tabel II.1. Bobot Kriteria

No	Kriteria	Bobot
1	Asal	10
2	Harga	10
3	Usia	20
4	Bobot	50
5	Ukuran	10
Total		100

(Sumber : Gumilar Ramadhan Pangaribuan; 2019)

Setelah di dapatkan nilai untuk setiap kriteria, kemudian di lakukan normalisasi, yaitu dengan membagi antara antara nilai bobot kriteria dengan jumlah nilai menggunakan persamaan (1).

Tabel II.2. Normalisasi Bobot Kriteria

No	Kriteria	Bobot Kriteria	Normalisasi
1	Asal	10/100	0,1
2	Harga	10/100	0,1
3	Usia	20/100	0,2
4	Bobot	50/100	0,5

5	Ukuran	10/100	0,1
---	--------	--------	-----

(Sumber : Gumilar Ramadhan Pangaribuan; 2019)

3. Memberikan Nilai Kriteria Pada Semua Alternatif

Nilai tersebut bisa dilihat sebagai berikut :

Tabel II.3. Pemberian Nilai Kriteria

No	Alternatif	Kriteria				
		Asal	Harga	Usia	Bobot	Ukuran
1	Lemosin	0,9	45	37	1100	1,5
2	Simental	0,8	17	24	460	1
3	Bali	0,7	25	35	900	1,3
4	Perah	0,6	10	36	250	1
5	Brahma	0,9	15	12	310	1,1
6	Madras	0,7	18	37	500	1,1

(Sumber : Gumilar Ramadhan Pangaribuan; 2019)

Nilai-nilai kriteria tersebut kemudian dikonversikan menjadi sebuah nilai kriteria data baku untuk menentukan nilai utiliti yang didapat dari persamaan(2). Bisa dilihat pada tabel 4 Asal, Harga, Usia, Bobot dan Ukuran untuk masing-masing usaha dihitung berdasarkan persamaan (2).

Nilai Asal dihitung dengan persamaan 2

$$\text{Maks(Asal)} = \text{maks}\{0.9, 0.8, 0.7, 0.6, 0.9, 0.7\} = 0.9$$

$$\text{Min (Asal)} = \text{min}\{0.9, 0.8, 0.7, 0.6, 0.9, 0.7\} = 0.6$$

Sehingga :

$$\text{Lemosin (Asal)} = \left(\frac{0.9-0.6}{0.9-0.6} \right) * 100\% = 1$$

$$\text{Simental (Asal)} = \left(\frac{0.8-0.6}{0.9-0.6} \right) * 100\% = 0,67$$

$$\text{Bali (Asal)} = \left(\frac{0.7-0.6}{0.9-0.6} \right) * 100\% = 0,33$$

$$\text{Perah (Asal)} = \left(\frac{0.6-0.6}{0.9-0.6} \right) * 100\% = 0$$

$$\text{Brahma (Asal)} = \left(\frac{0.9-0.6}{0.9-0.6} \right) * 100\% = 1$$

$$\text{Madras (Asal)} = \left(\frac{0.7-0.6}{0.9-0.6} \right) * 100\% = 0,33$$

Sehingga diperoleh perhitungan secara lengkap Nilai *Utility* Alternatif seperti yang ditunjukkan pada tabel berikut:

Tabel II.4. Matriks Perhitungan Nilai *Utility* Alternatif

No	Alternatif	Kriteria				
		Asal	Harga	Usia	Bobot	Ukuran
1	Lemosin	1	1	1	1	1
2	Simental	0,67	0,2	0,48	0,25	0
3	Bali	0,33	0,42	0,92	0,76	0,6
4	Perah	0	0	0,96	0	0
5	Brahma	1	0,14	0	0,07	0,2
6	Madras	0,33	0,22	1	0,29	0,2

(Sumber : Gumilar Ramadhan Pangaribuan; 2019)

4. Menghitung Masing-Masing Nilai Alternatif

Menghitung nilai alternatif menggunakan rumus persamaan (3) dengan mengkoverensikan antara nilai utiliti dengan nilai normalisasi bobot kriteria sehingga diperoleh nilai terakhir. Perhitungannya adalah sebagai berikut :

$$\text{Lemosin (Asal)} = 1 * 0.1 = 0$$

$$\text{Lemosin (Harga)} = 1 * 0.1 = 0.1$$

$$\text{Lemosin (Usia)} = 1 * 0.2 = 0.2$$

$$\text{Lemosin (Bobot)} = 1 * 0.5 = 0.5$$

$$\text{Lemosin (Ukuran)} = 1.5 * 0.1 = 0.1$$

Tabel II.5. Menentukan Nilai Akhir Menggunakan Persamaan (4)

No	Alternatif	Kriteria					Nilai Akhir
		Asal	Harga	Usia	Bobot	Ukuran	
1	Lemosin	0,1	0,1	0,2	0,5	0,1	1
2	Simental	0,07	0,02	0,096	0,123	0	0,306
3	Bali	0,03	0,04	0,184	0,382	0,06	0,702

4	Perah	0	0	0,192	0	0	0,192
5	Brahma	0,1	0,01	0	0,035	0,02	0,169
6	Madras	0,03	0,02	0,2	0,147	0,02	0,423

(Sumber : Gumilar Ramadhan Pangaribuan; 2019)

Dengan demikian alternatif terbaik adalah alternatif yang mendominasi alternatif lainnya. Hasil dari perhitungan dengan metode SMART maka diperoleh peringkat yang paling tinggi A1 yaitu jenis Sapi Lemosin dengan nilai akhir 1 dan A3 sebagai peringkat kedua dengan nilai akhir 0,702543.

5. Kesimpulan

Berdasarkan hasil penelitian dapat diterangkan bahwa metode SMART dapat diterapkan untuk merekomendasikan jenis sapi terbaik untuk peternakan sapi potong dengan mempertimbangkan beberapa kriteria, yaitu: Asal, Harga, Usia, Bobot dan Ukuran. Hasil dari penelitian menyebutkan menyebutkan bahwa jenis Sapi Lemosin (A1) menjadi rekomendasi pertama dengan nilai akhir 1 dan jenis sapi bali (A3) sebagai rekomendasi kedua dengan nilai akhir 0,702543. (Gumilar Ramadhan Pangaribuan: 2019).

II.2.4 Database

Basis data atau *database* adalah kumpulan data terstruktur. Agar dapat menambahkan, mengakses, dan memproses data yang tersimpan dalam *database* komputer, dibutuhkan sistem manajemen basis data (*database management system*). Dalam pengembangan perangkat lunak tradisional yang memanfaatkan pemrosesan *file*, setiap kelompok pengguna menyimpan *file-file*-nya sendiri untuk menangani aplikasi pengolahan datanya masing-masing. Hal ini mengakibatkan

adanya kerangkapan data atau disebut dengan *redundancy*. Redundansi dalam proses penyimpanan data yang terjadi berkali-kali dapat mengakibatkan beberapa masalah. Pertama, ada kebutuhan untuk melakukan pembaruan logis tunggal, misalnya seperti memasukkan data pada siswa baru beberapa kali: satu kali untuk setiap *file* tempat data siswa direkam. Hal ini menyebabkan duplikasi data. Kedua, ruang penyimpanan terbuang ketika data yang sama disimpan berulang kali, dan masalah ini mungkin serius untuk *database* yang besar. Ketiga, file yang mewakili data yang sama mungkin menjadi tidak konsisten. Hal ini bisa terjadi karena update diaplikasikan pada beberapa file tapi tidak untuk file yang lain, (Cosmas Eko Suharyanto; 2017).

II.2.5 Website

Website merupakan kumpulan dari halaman-halaman web yang berhubungan dengan *file file* lain yang saling terkait. Dalam sebuah *Website* terdapat suatu halaman yang dikenal dengan sebutan *Home page*. *Home page* adalah sebuah halaman yang pertama kali ketika seseorang mengunjungi sebuah *Website*. dari *Home page*, pengunjung dapat mengklik *hyperlink* untuk pindah ke halaman lain yang terdapat dalam *Website* tersebut. Sebuah *Home page* biasanya merupakan sebuah *file* dengan nama *index.htm* atau *index.html*. (Muhammad Saed Novendr; 2019).

II.2.6 *Hypertext Markup Language (HTML)*

Hypertext Markup Language (HTML) adalah sebuah bahasa *markup* yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi di dalam sebuah penjelajah web internet dan *Formating hypertext* sederhana yang ditulis kedalam *Format ASCII* agar dapat menghasilkan tampilan wujud yang terintergrasi. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata dan disimpan kedalam *Format ASCII* normal sehingga menjadi *Home page* dengan perintah-perintah HTML. Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan SGML (*Standart Generalized Markup Language*), HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman web. HTML saat ini merupakan standar internet yang didefenisikan dan dikendalikan penggunaannya oleh *World Wide Web Consortium (W3c)*, HTML dibuat oleh kolaborasi *Caillau TIM* dengan *Berners lee robert* ketika mereka bekerja di CERN pada tahun 1989 CERN adalah lembaga penelitian fisika energi tinggi di jenewa. (Harison; 2016).

II.2.7 PHP

PHP adalah singkatan dari "PHP: *Hypertext Preprocessor* ", yang merupakan sebuah bahasa scripting yang terpasang pada *HyperText Markup Language (HTML)*. Sebagian besar sintaks mirip dengan bahasa C, Java dan Perl, ditambah beberapa fungsi PHP yang spesifik. Tujuan utama penggunaan

bahasa ini adalah untuk memungkinkan perancang web menulis halaman web dinamik dengan cepat. (Nursahid; 2015).

II.2.8 My SQL

Salah satu contoh *database management system* adalah *MySQL*. *MySQL* adalah *database open source* terpopuler di dunia. Dengan kinerja, kehandalan dan kemudahan penggunaan yang terbukti, *MySQL* telah menjadi pilihan *database* terdepan untuk aplikasi berbasis web, yang digunakan oleh properti web profil tinggi termasuk *Facebook*, *Twitter*, *YouTube*, *Yahoo!* dan banyak lagi. Kata “SQL” dari “*MySQL*” adalah singkatan dari “*Structured Query Language*”. SQL adalah bahasa standar yang paling umum digunakan untuk mengakses *database*. Bergantung pada lingkungan pemrograman yang kita pakai, kita bisa memasukkan SQL secara langsung (misalnya, untuk *men-generate* laporan), memasukkan pernyataan SQL ke dalam kode yang ditulis dalam bahasa lain, atau menggunakan API khusus yang dapat menyembunyikan sintaks SQL, (Cosmas Eko Suharyanto; 2017).

II.2.9 XAMPP

Xampp merupakan singkatan dari X (empat sistem operasi apapun), XAMPP merupakan perangkat lunak bebas, yang mendukung banyak sistem operasi, dan merupakan kompilasi dari beberapa program. Seperti Apache, *MYSQL*, *PHPP*, dan *Perl*. Xampp adalah tool yang menyediakan paket perangkat lunak dalam satu buah paket. Dalam paket Xampp sudah terdapat

Apache(Web Server), *Mysql* (Database), *PHP* (server side scripting), *Perl*, *FTP* server, *PhpMyAdmin*, dan berbagai pustaka bantu lainnya. Kepanjangan dari Xampp yaitu :

1. (X) : Program ini dapat dijalankan dibanyak sistem operasi, seperti *Windows*, *Linux*, *Mac OS* dan juga *Solaris*.
2. (A) : *Apache* merupakan aplikasi web server. Tugas utama dari *Apache* adalah menghasilkan halaman web yang benar kepada user berdasarkan kode *PHP* yang dituliskan oleh pembuat web atau user.
3. (M) : *MySQL*, merupakan aplikasi data server. Perkembangannya disebut juga *Sql* yang merupakan kepanjangan dari *Structured Query Language*. *Sql* merupakan bahasa terstruktur yang digunakan untuk mengolah *database*. P : *PHP*, merupakan bahasa pemrograman web, dimana user dapat menggunakan bahasa pemrograman ini untuk membuat web yang bersifat server-side scripting.
4. (P) : *Perl*, yaitu merupakan bahasa pemrograman untuk segala keperluan, dan dikembangkan pertama kali oleh *Larry Wall* di mesin *Unix*, (Muhammad Saed Novendr; 2019).

II.2.10 UML (*Unified Modeling Language*)

Unified Modeling Language (UML). UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam

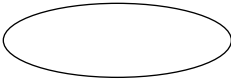
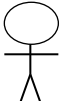
mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem, (Ade Hendini; 2016).

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

1. *Use Case Diagram*

Use Case Diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem inFormasi yang akan dibuat. *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sistem inFormasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *Use Case Diagram* yaitu :

Tabel II.6. Tabel Simbol *Use Case Diagram*

Gambar	Keterangan
	<i>Use Case</i> mengGambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki control terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , diGambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.
	Asosiasi antara aktor dan <i>Use Case</i> yang menggunakan panah terbuka untuk

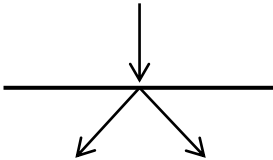
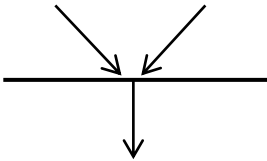
	mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
----->	<i>Include</i> , merupakan di dalam <i>Use Case</i> lain (<i>required</i>) atau pemanggilan <i>Use Case</i> oleh <i>Use Case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
<-----	<i>Extend</i> , merupakan perluasan dari <i>Use Case</i> lain jika kondisi atau syarat terpenuhi.

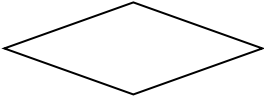
(Sumber : Ade Hendini; 2016)

2. Diagram Aktivitas (Activity Diagram)

Activity Diagram mengGambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *Activity Diagram*, yaitu :

Tabel II.7. Tabel Simbol Activity Diagram

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , mengGambarkan suatu proses/ kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi.

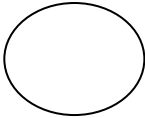
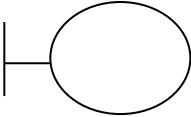
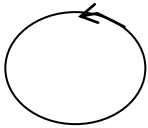
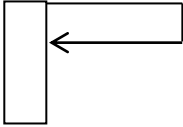
	<i>Decision Points</i> , mengGambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>Activity Diagram</i> untuk menunjukkan siapa melakukan apa.

(Sumber : Ade Hendini; 2016)

3. Diagram Urutan (*Sequence Diagram*)

Sequence Diagram mengGambarkan kelakuan objek pada *Use Case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *Sequence Diagram*, yaitu :

Tabel II.8. Tabel Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk Gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>Formentry</i> dan <i>Form</i> cetak
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , mengGambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding

	lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Ade Hendini; 2016)

4. Class Diagram (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class Diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class Diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau *cardinality*.

Tabel II.9. Tabel Multiplicity Class Diagram

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Ade Hendini; 2016)