

BAB II

TINJAUAN PUSTAKA

II.1 Penelitian Terkait

Dalam penelitian terdahulu, penulis mengambil beberapa referensi yang berkaitan dengan latar belakang masalah yang penulis ambil. Salah satunya yaitu penelitian yang dilakukan oleh Hendra Priatna, Iyan Mulyana, M.Kom, Aries Maesya, M.Kom, 2017 “Penerapan Perbandingan Metode *Profile Matching* Dan Metode *SAW (Simple Additive Weighting)*. Untuk Kenaikan Pangkat Pegawai” berkesimpulan bahwa dengan menggunakan sistem pendukung keputusan kenaikan pangkat pegawai dengan metode *profile matching* dan metode *Simple Additive Weigthing (SAW)* dapat mempermudah instansi dalam memutuskan kenaikan jabatan menjadi lebih mudah dan efisien dalam penanganan sistem kepegawaian.

Penelitian yang dilakukan oleh Sunarti, Jenie Sundari, 2018 “Perbandingan Metode *Simple Additive Weighting (SAW)* Dan Metode *Profile Matching* Pada Pemilihan Rumah Tinggal (Studi Kasus: Perumahan Depok)” berkesimpulan bahwa dengan menggunakan sistem pendukung keputusan pemilihan rumah tinggal dengan menggunakan metode *simple additive weigthing* dan metode *profile matching* dapat mempermudah dalam pemilihan rumah tinggal untuk membantu pengambilan keputusan.

Penelitian yang dilakukan Muhammad Isfan Fajar, Fahrul Agus, Haviluddin, 2018 “Analisis Penerapan Perbandingan Metode *Profile Matching* Dan *Simple Additive Weigthing (SAW)* Pada Kasus Pemilihan Bibit Unggul

Kelapa Sawit” berkesimpulan bahwa dengan menggunakan sistem pendukung keputusan pemilihan bibit kelapa sawit dengan menggunakan metode *profile matching* dan metode *simple additive weighing (SAW)* dapat mempermudah pemilihan bibit unggul kelapa sawit dalam pengambil keputusan.

II.2. Studi Literatur

II.2.1. Sistem Informasi

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk melakukan sasaran yang tertentu. Pendekatan Sistem yang merupakan jaringan kerja dari prosedur lebih menekankan urutan-urutan operasi di dalam sistem. (Abdul Rozak, et al; 2018). Informasi merupakan data yang diolah menjadi bentuk yang berguna untuk membuat keputusan karena menirukan ketidak pastian (atau meningkatkan pengetahuan). Informasi tersebut merupakan hasil pengolahan data atau fakta yang dikumpulkan dengan metode atau cara-cara tertentu (Abdul Rozak, et al; 2018). Menurut (Hutahean, 2014, p. 13) Sistem Informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang dibutuhkan. (Abdul Rozak, et al; 2018).

II.2.3. Sistem Pendukung Keputusan

Sistem yaitu kelompok yang menekankan aturan dari awal dan akhir untuk mencapai tujuan. pendukung keputusan yaitu pola pikir membantu menyelesaikan masalah. Sistem pendukung keputusan adalah aturan dari awal dan akhir dalam menyelesaikan masalah keputusan yang rinci dan jelas. Tahapan dasar tersebut adalah Informasi input disampaikan dalam matriks alternatif dan atribut, yang didasarkan pada matriks keputusan (Pristiwati Fitriani, 2020).

II.2.4. Tujuan Pendukung Keputusan

Menurut (Arief Budiman dkk, 2020) Sistem Pendukung Keputusan tidak dimaksudkan untuk mengotomatisasikan pengambilan keputusan, tetapi memberikan perangkat interaktif yang memungkinkan pengambilan keputusan untuk melakukan berbagai analisis menggunakan model-model yang tersedia.

Tujuan dari sistem pendukung keputusan adalah :

1. Membantu manager dalam pengambilan keputusan atas masalah semiterstruktur.
2. Memberikan dukungan atas pertimbangan manager dan bukannya dimaksudkan untuk menggantikan fungsi manager.
3. Meningkatkan efektifitas keputusan yang diambil manager lebih daripada perbaikan efesiensinya.
4. Komputer memungkinkan para pengambilan keputusan untuk melakukan banyak komputasi secara cepat dengan biaya yang rendah.
5. Peningkatkan Produktifitas.

6. Komputer bisa meningkatkan kualitas keputusan yang dibuat.
7. Berdaya saing.
8. Mengatasi keterbatasan kognitif dalam pemrosesan dan penyimpanan.

II.3. Definisi pegawai

Pegawai adalah orang pribadi yang bekerja pada pemberi kerja, baik sebagai pegawai tetap atau pegawai tidak tetap atau tenaga kerja lepas berdasarkan perjanjian atau kesepakatan kerja baik secara tertulis maupun tidak tertulis. Para pegawai bertugas untuk melaksanakan suatu pekerjaan dalam jabatan atau kegiatan tertentu dengan memperoleh imbalan yang dibayarkan berdasarkan periode tertentu, penyelesaian pekerjaan, atau ketentuan lain yang ditetapkan pemberi kerja (Anita B. Wandanaya dalam Jurnal Pradita, 2017 : 2).

II.4. Metode *Profile Matching*

Metode *Profile Matching* merupakan proses membandingkan antara kemampuan individu ke dalam kemampuan posisi sehingga dapat diketahui perbedaan kemampuannya (*GAP*) sehingga semakin kecil *gap* yang dihasilkan maka semakin besar peluang mendapatkan posisi yang diinginkan (kusrini, 2007).

Langkah-langkah metode *profile matching* adalah:

1. Menentukan *variabel* data-data yang dibutuhkan.
2. Menentukan aspek-aspek yang digunakan untuk penilaian.
3. Pemetaan *Gap* profil. $Gap = \text{Profil Minimal} - \text{Profil data tes}$

4. Setelah diperoleh nilai *Gap* selanjutnya diberikan bobot untuk masing-masing nilai *Gap*.

5. Perhitungan dan pengelompokan *Core Factor* dan *Secondary Factor*. Setelah menentukan bobot nilai *gap*, kemudian dikelompokkan menjadi 2 kelompok yaitu:

1. *Core Factor* (Faktor Utama), yaitu merupakan kriteria (kompetensi) yang paling penting atau menonjol atau paling dibutuhkan oleh suatu penilaian yang diharapkan dapat memperoleh hasil yang optimal. $NFC = ENC / EIC$ Keterangan:

NFC : Nilai rata-rata *core factor*

NC : Jumlah total nilai *core factor*

IC : Jumlah item *core factor*

2. *Secondary Factor* (faktor pendukung), yaitu merupakan item-item selain yang ada pada *core factor*. Atau dengan kata lain merupakan faktor pendukung yang kurang dibutuhkan oleh suatu penilaian. $NFS = ENS / EIS$ Keterangan:

NFS : Nilai rata-rata *secondary factor*.

NS : Jumlah total nilai *secondary factor*

IS : Jumlah item *secondary factor*.

6. Perhitungan Nilai Total. Nilai Total diperoleh dari prosentase *core factor* dan *secondary factor* yang diperkirakan berpengaruh terhadap hasil tiap-tiap profil. $N = (x) \% NCF + (x) \% NSF$ Keterangan:

N : Nilai Total dari kriteria

NFS : Nilai rata-rata *secondary factor*

NFC : Nilai rata-rata *core factor*

(x) % : Nilai persen yang diinputkan

7. Perhitungan penentuan ranking. Hasil Akhir dari proses *profile matching* adalah ranking. Penentuan ranking mengacu pada hasil perhitungan tertentu. $\text{Ranking} = (x) \% \text{ NMA} + (x) \% \text{ NSA}$ Keterangan :

NMA : Nilai total kriteria Aspek Utama

NSA : Nilai total kriteria Aspek Pendukung

(x) % : Nilai persen yang diinputkan

II.5. Metode (SAW) *Simple Additive Weighting*

Metode (SAW) *Simple Additive Weighting* adalah penjumlahan terbobot. Konsep dasar dari metode *simple additive weighting* adalah mencari pejumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut. Metode (saw) *simple additive weighting* membutuhkan proses normalisasi matrik keputusan (X) ke skala yang dapat diperbandingkan dengan semua rating alternatif yang ada (Kusumadewi).

Langkah-langkah metode *simple additive weighting* adalah:

1. Menentukan kriteria yang akan dijadikan acuan dalam pengambilan keputusan, yaitu C_i .
2. Memberikan nilai bobot untuk masing-masing kriteria sebagai W .
3. Memberikan nilai rating kecocokan setiap alternatif pada setiap kriteria.
4. Membuat matriks keputusan berdasarkan kriteria (C_i), kemudian melakukan normalisasi matriks berdasarkan persamaan yang disesuaikan

degan jenis atribut (atribut keuntungan ataupun atribut biaya) sehingga diperoleh matriks ternormalisasi R.

$$R_{ij} = \begin{cases} \frac{x_{ij}}{\max_i x_{ij}} & \text{JIKA J ADALAH ATRIBUT KEUNTUNGAN (BENEFIT)} \\ \frac{\min_i x_{ij}}{x_{ij}} & \text{JIKA J ADALAH ATRIBUT BIAYA (COST)} \end{cases}$$

Keterangan Setiap Kriteria :

R_{ij} : Nilai rating Kinerja ternormalisasi

X_{ij} : Nilai atribut yang dimiliki dari

$\max X_{ij}$: Nilai terbesar dari setiap kriteria

$\min X_{ij}$: Nilai terkecil dari setiap kriteria

Benefit : jika nilai terbesar adalah terbaik

Cost : jika nilai terkecil adalah terbaik

5. Hasil akhir diperoleh dari proses perangkingan yaitu penjumlahan dan perkalian matriks ternormalisasi R dengan vector bobot sehingga diperoleh nilai terbesar yang dipilih sebagai alternatif yang terbaik (A_i) sebagai solusi.

$$V_i = \sum_{j=1}^n W_j R_{ij}$$

Keterangan :

V_i : Ranking untuk setiap alternatif

W_j : Nilai Bobot Dari Setiap Kriteria

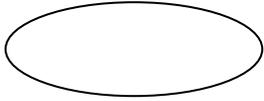
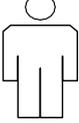
II.6. UML (*Unified Modeling Language*)

Unified Modelling Language (UML) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi objek. Abstraksi konsep dasar *UML* terdiri dari *structural classification*, *dynamic behavior*, dan model *management* dapat kita pahami main *concepts* sebagai term yang akan muncul pada saat membuat *diagram* dan *view* adalah kategori dari *diagram* tersebut. *UML* mendefinisikan *diagram-diagram* sebagai *Use case diagram*, *Class diagram*, *Statechart diagram*, *Activity diagram*, *Sequence diagram*, *Collaboration diagram*, *Component diagram*, dan *Deployment diagram*. (Suendri, 2018 : 2)

A. *Use Case Diagram*

Use Case Diagram merupakan pemodelan untuk kelakuan sistem informasi yang akan dibangun. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibangun. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada pada sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. (Julianto Simatupang, 2019).

Tabel II 1. Simbol *Use Case Diagram*

Simbol	Diskripsi
<i>Use Case</i> 	<i>Fungsionalitas</i> yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor: biasanya dinyatakan dengan menggunakan kata kerja diawal di awal <i>frase</i> nama <i>use case</i> .
Aktor/actor 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.
Asosiasi /association 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan actor.
Ekstensi / extend <<extends>> 	<i>Relasi 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 Arah panah mengarah pada <i>use case</i> yang ditambahkan.
Generalisasi / generalization 	Hubungan <i>generalisasi</i> dan spesialisasi (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
Menggunakan / <i>include</i> 	<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

B. *Class Diagram*

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. (Julianto Simatupang, 2019).

Tabel II 2. Simbol *Class Diagram*

Simbol	Diskripsi
--------	-----------

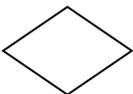
<p>Kelas</p> <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <p>Nama_ <i>Class</i></p> <hr/> <p>-atribut</p> <hr/> <p>-operasi()</p> </div>	<p>Kelas pada struktur sistem.</p>
<p>Antar muka/interface</p> <div style="text-align: center; margin: 5px 0;">  </div> <p>nama_ <i>interface</i></p>	<p>Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.</p>
<p>Asosiasi / association</p> <div style="text-align: center; margin: 5px 0;">  </div>	<p><i>Relasi</i> antar kelas dengan makna umum, <i>asosiasi</i> biasanya juga disertai dengan <i>multiplicity</i>.</p>
<p>Asosiasi berarah / <i>directed association</i></p> <div style="text-align: center; margin: 5px 0;">  </div>	<p><i>Relasi</i> antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, <i>asosiasi</i> biasanya juga disertai dengan <i>multiplicity</i>.</p>
<p><i>Generalisasi</i></p> <div style="text-align: center; margin: 5px 0;">  </div>	<p><i>Relasi</i> antar kelas dengan makna <i>generalisasi- generalisasi- spesialisasi</i> (umum khusus).</p>
<p>Kebergantungan / <i>dependency</i></p> <div style="text-align: center; margin: 5px 0;">  </div>	<p><i>Relasi</i> antar kelas dengan makna kebergantungan antar kelas</p>
<p>Agregasi /aggregation\</p> <div style="text-align: center; margin: 5px 0;">  </div>	<p><i>Relasi</i> antar kelas dengan makna semua-bagian (<i>whole- part</i>).</p>

C. Activity Diagram

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Penekanan pada diagram aktivitas adalah menggambarkan aktivitas sistem atau

aktivitas yang dapat dilakukan oleh sistem, bukan apa yang dilakukan aktor (Julianto Simatupang, 2019).

Tabel II 3. Simbol *Activity Diagram*

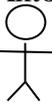
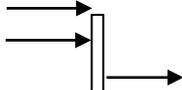
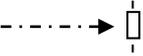
Simbol	Diskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan <i>aktivitas</i> lebih dari Satu
Penggabungan / Join	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu..
Status akhir 	Status akhir yang dilakukan sistem, sebuah <i>diagram aktivitas</i> memiliki sebuah status akhir.

D. *Sequence Diagram*

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar *diagram sequence* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-

metode yang dimiliki kelas yang *diinstansiasi* menjadi objek itu. (Julianto Simatupang, 2019).

Tabel II 4. *Sequence Diagram*

Simbol	Diskripsi
Aktor 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.
Garis hidup / lifeline 	Menyatakan kehidupan suatu objek.
Objek 	Menyatakan objek yang berinteraksi pesan.
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya
Pesan tipe create << create >> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat. 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.
Pesan tipe send 1 : masukan 	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
Pesan tipe return 1 : keluaran 	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.
Pesan destroy 	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah yang mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy.

II.7. Website

Menurut (Raniya dkk, 2020) *Website* merupakan sekumpulan halaman yang berhubungan yang dapat diakses public melalui *world wide web*. Suatu *website* biasanya mengandung teks, gambar, suara bahkan video interaktif dan

mempunyai kelebihan untuk menghubungkan (*link*) satu dokumen dengan dokumen lainnya (*hypertext*) yang dapat diakses melalui sebuah *browser*. *website* adalah kumpulan halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara dan/atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan – jaringan halaman. Maka dapat disimpulkan bahwa *website* adalah sekumpulan informasi yang terdiri dari teks, gambar diam atau gerak, animasi, suara yang membentuk satu rangkaian bangunan yang terhubung ke internet.

II.8. HTML (*Hypertext Markup Language*)

HTML (Hyper Text Mark Up Language) bahasa pemrograman standar yang digunakan untuk membuat sebuah halaman web, yang kemudian dapat diakses untuk menampilkan berbagai informasi di dalam sebuah penjelajah web Internet HTML dapat juga digunakan sebagai link link antara file-file dalam situs atau dalam komputer dengan menggunakan *localhost*, atau *link* yang menghubungkan antar situs dalam dunia internet. (*Sumber : Pengertian HTML dan Fungsinya, Sejarah HTML, Cara Kerja HTML (temukanpengertian.com)*).

II.8.1. CSS (*Cascading Style Sheet*)

CSS (Cascading Style Sheet) adalah salah satu kode pemrograman yang bertujuan untuk menghias dan mengatur gaya tampilan atau layout halaman web supaya lebih elegan dan menarik (Didik Setiawan, 2017 : 16).

II.8.2. PHP

Menurut (Khairunnisa Fauzia, 2020), *PHP Hypertext Preprocessor* adalah suatu bahasa *scripting* khususnya digunakan untuk *web server*. *PHP* memiliki sifat *server side scripting* sehingga untuk menjalankan *PHP* harus menggunakan *web server*.

II.8.3. JavaScript

JavaScript adalah bahasa pemrograman untuk *web*. Ini didukung oleh sebagian besar *Web Browser* termasuk *Chrome, Firefox, Safari, internet Explorer, Edge, Opera*, dan lainnya. Sebagian besar *browser ponsel* untuk ponsel pintar juga mendukung *JavaScript*. Ini terutama digunakan untuk meningkatkan halaman *web* untuk memberikan pengalaman yang lebih ramah pengguna. Ini termasuk memperbarui halaman *web* secara dinamis, peningkatan antarmuka pengguna seperti *menu* dan kotak dialog, *animasi*, grafik 2D dan 3D, peta *interaktif*, pemutar video, dan banyak lagi. Mode penggunaan *JavaScript* ini di *browser web* juga disebut sebagai *javascript sisi klien* atau *Client-Side Javascript*. (Sumber : *Pengertian JavaScript: Sejarah, Cara Kerja JavaScript dan Manfaatnya!* (rifqimulyawan.com)).

II.8.4. Bootstrap

Menurut (Fahrul Khoirul Khadziq dkk, 2020). Pengertian *bootstrap* adalah sebuah *framework CSS* dari *Twitter* yang menyediakan komponen-komponen antarmuka siap pakai dan telah dirancang sedemikian rupa untuk keperluan desain

halaman *website* artistik. Selain komponen-komponen dasar untuk membangun antarmuka, *bootstrap* juga menyediakan *grid* 12 kolom yang dapat menunjang kinerja anda dalam mendesain *layout* yang rapi. Sebagai *CSS framework*, *Bootstrap* tergolong paket lengkap. Teknologi *HTML*, *CSS*, dan *Javascript* yang ada pada *Bootstrap*, anda tidak hanya dapat *styling* dengan *CSS* saja, akan tetapi dapat juga menggunakan komponen-komponen seperti ikon, tombol dan *navigasi* dengan desain unik khas *Bootstrap*.¹² Selain itu, tersedia banyak *plugin JQuery* yang dapat membantu untuk membuat tampilan *website* yang *atraktif*. *Framework Bootstrap* dibangun menggunakan *Less*, yaitu sebuah *preprocessor* dengan fungsi tambahan seperti *variable*, *mixin* dan fungsi untuk *mengompilasi CSS*. Fitur-fitur yang ditawarkan pada *Bootstrap* dapat membuat sang *developer Bootstrap* kaya jika berorientasi dengan uang. *Framework* ini seharusnya layak untuk menjadi *Framework* berbayar, mengingat fitur-fitur yang ditawarkan sangat lengkap. Kenyataannya, *framework* ini justru tersedia gratis, yaitu dengan menggunakan *lisensi open source*, dimana semua kalangan bebas menggunakan *framework* ini secara gratis tanpa takut terkena masalah legalitas.

II.8.5. *JQuery*

Menurut (Omar Pahlevi dkk, 2018). *JQuery* adalah *open source add-on* pustaka yang menekankan pada interaksi antara *JavaScript* dan *HTML*. *JQuery JavaScript* merupakan kode *JavaScript* yang telah ditulis dan tinggal menambahkan satu atau dua baris kode untuk memanggil *JQuery*.

II.8.6 *Sublime Text*

Menurut (Vivi Afifah dkk, 2021), Sublime Text adalah aplikasi editor untuk kode dan teks yang dapat berjalan diberbagai platform sistem operasi menggunakan teknologi Phyton API.

II.9. *Database*

Menurut (Novri Ultariani dkk, 2020), *Database* adalah suatu aplikasi yang menyimpan sekumpulan data. Setiap *database* mempunyai *API* tertentu untuk membuat, mengakses, mengatur, mencari, menyalin data yang ada di dalamnya. Atau juga *Database* yaitu kumpulan file-file yang berhubungan satu dengan yang lainnya, diatur sedemikian rupa sehingga dapat digunakan oleh beberapa program aplikasi *database*.

II.9.1. *MySQL*

Menurut (Aulia Ikhsan dkk, 2020). *MySQL* adalah *Relational Database Management System (RDMS)* yang cepat dan akurat. Sebuah basis data dapat membuat pengguna untuk menyimpan, mencari, mengurutkan, dan mendapatkan data dengan sangat efisien. *Server MySQL* mengendalikan akses dalam data untuk memastikan bahwa para pengguna dapat bekerja dalam waktu yang bersamaan.