

BAB II

LANDASAN TEORI

II.1. Penelitian Terkait

Adapun penelitian terkait yang akan di gunakan sebagai sumber acuan yang relavan dan terkini yaitu :

Berdasarkan penelitian dari Rofiqoh Dewi (2017) dengan judul “Penerapan Metode Fuzzy SAW Untuk Penyeleksian Beasiswa Bidik Misi (Studi Kasus : Universitas Potensi Utama). Metode yang dipakai dalam pengambilan keputusan seleksi beasiswa bidikmisi adalah Fuzzy SAW, metode tersebut dipilih karena metode tersebut merupakan suatu bentuk model pendukung keputusan dimana inputdaerah balige utamanya menggunakan konsep dasar mencari penjumlahan terbobot. Penelitian dilakukan dengan mencari nilai bobot untuk setiap kriteria, kemudian dilakukan proses perangkingan yang akan menentukan alternatif yang optimal yaitu mahasiswa terbaik yang akan dipertimbangkan oleh pengambil keputusan untuk memperoleh beasiswa bidikmisi.

Berdasarkan penelitian dari (Yusnaeni & Marlina, 2020) dengan judul “MABAC Method Dalam Penentuan Kelayakan Penerima Bantuan SPP” Dalam pendataan pihak panitia mendata secara satu persatu serta dilakukan survei dan keputusan dari pihak panitia. Perlunya waktu dalam penginputan dan keputusan yang hanya dilakukan oleh pihak panitia tidak menjamin keputusan diambil secara tepat dan akurat. Oleh karena itu, perlunya sistem pendukung keputusan

yang bisa membantu menentukan hasil yang lebih objektif sesuai dengan kriteria yang telah ditentukan. Salah satu metode yang dibahas penulis disini MABAC Method. Adanya kestabilan dalam solusi dan rasional dibandingkan beberapa metode sistem pendukung keputusan lainnya. Penggunaan metode dilakukan terhadap 128 pendaftar siswa dengan data kriteria yang telah ditentukan. Dari data tersebut ditentukan bobot penilaian serta bobot alternative terhadap kriteria yang menghasilkan layak tidak layak nya siswa menerima beasiswa atau bantuan SPP

Berdasarkan penelitian dari (Yosafat et al., 2020) dengan judul “Perancangan Sistem Pendukung Keputusan Pembelian Laptop Dengan Metode Mabac (Studi Kasus: Sigma Komputer)” Hal ini membuat toko sigma komputer memerlukan sebuah sistem yang dapat membantu pemilik sekaligus pembeli untuk menentukan keputusan dalam pemilihan laptop yang akan dibeli secara cepat dan efisien. Sistem dibangun dengan menggunakan bahasa pemrograman C# dan database SQLite, serta menggunakan metode Mabac (Multi-Attributive Border Approximation Area Comparison) yang digunakan dalam perhitungan pada sistem pendukung keputusan. Data yang diperlukan berupa data laptop beserta spesifikasinya. Hasil dari perancangan sistem pendukung keputusan ini adalah dapat membantu baik dari sisi penjual yang dapat memberikan saran laptop yang cocok untuk pembeli secara efisien dan cepat, dan dari sisi pembeli yang dapat membantu mengambil keputusan memilih laptop sesuai dengan kriteria yang sesuai dengannya.

Berdasarkan penelitian dari (Ihwa et al., 2020) dengan judul “Sistem Pendukung Keputusan Pemilihan Jaksa Terbaik dengan Menerapkan Metode MABAC (Studi Kasus: Kejaksaan Negeri Medan)” Kriteria-kriteria yang diambil dalam penilaian ialah segi orientasi, segi pelayanan, segi integritas, segi komitmen, segi disiplin, dan segi kerja sama memerlukan sistem pendukung keputusan (SPK) untuk mempercepat dan mempermudah dalam membuat suatu keputusan. Sistem pendukung keputusan digunakan untuk membantu pengambilan keputusan dalam sebuah organisasi untuk mempermudah dalam pengambilan sebuah keputusan. Di mana tidak seorang pun mengetahui secara pasti bagaimana seharusnya keputusan di buat. Salah satu metode yang digunakan pada sistem pendukung keputusan ini adalah MABAC. Sistem pendukung keputusan untuk membantu pemerintah daerah untuk pemilihan jaksa terbaik yang sesuai dengan syarat dan kriteria-kriteria yang ditentukan. Metode MABAC merupakan salah satu dari sekumpulan metode sistem pendukung keputusan yang termasuk dalam kelompok metode multi criteria decision making (MABAC)

Berdasarkan penelitian dari (Kristianto, 2018) dengan judul “MABAC: Pemilihan Penerima Bantuan Rastra Menggunakan Metode MultiAttributive Border Approximation Area Comparison” bahwa Mendukung keputusan pemilihan penerima bantuan rastra yang sifatnya multi alternative maka diperlukan metode pendukung keputusan. Metode MABAC adalah salah satu metode pendukung keputusan yang jenis keputusannya multi alternative, metode ini dapat membantu proses pengambilan keputusan pemilihan bantuan rastra

dengan optimal. MABAC merupakan metode perbandingan multikriteria. Metode ini dipilih karena, metode ini menyediakan stabil (konsisten) solusi dan handal untuk pengambilan keputusan rasional, dibandingkan dengan metode lain multi-kriteria pengambilan keputusan (SAW, COPRAS, Moora, TOPSIS dan VI-KOR).

Berdasarkan penelitian dari (Purba, 2020) dengan judul “Sistem Pendukung Keputusan Pemilihan Dokter Terbaik di Dinas Kesehatan Kab. Simalungun Menggunakan Metode MABAC” Maka dari itu untuk memperoleh informasi atau proses yang cepat, akurat, dan tidak adanya kecurangan dalam pemilihan dokter terbaik dengan prestasi Dokter yang benar dan memenuhi kriteria yang diharapkan, dibutuhkan suatu sistem pendukung keputusan untuk memecahkan situasi kompleks dan tidak terstruktur. Saya menggunakan metode MABAC (Multi-Attributive Border Approximation area Comparison) Metode ini digunakan untuk mendapatkan koefisien bobot kriteria, berdasarkan alternatif yang ada dievaluasi dan mendapatkan hasil yang benar nyata dan tidak adanya kecurangan lagi. Berdasarkan metode yang digunakan tersebut maka akan menghasilkan nilai berdasarkan data yang ada dapat dipastikan hasilnya akan lebih akurat dan tidak adanya lagi kecurangan dalam ajang pemilihan dokter terbaik tersebut

II.2. Sistem

Sistem dapat terdiri dari sistem-sistem bagian (*subsystem*). Sebagai misal, sistem komputer dapat terdiri dari subsistem perangkat keras dan subsistem

perangkat lunak. Masing-masing subsistem dapat terdiri dari subsistem-subsistem yang lebih kecil atau terdiri dari komponen-komponen. Subsistem perangkat keras (*hardware*) dapat terdiri dari alat masukan, alat pemroses, alat keluaran dan simpanan luar, dan kemudian subsistem-subsistem tersebut akan berinteraksi sedemikian rupa sehingga dapat mencapai satu kesatuan yang terpadu. Dalam buku *Analisa dan Design Sistem Informasi Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*, sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu (Han & goleman, daniel; boyatzis, Richard; Mckee, 2019)

II.3. Sistem Pendukung Keputusan

Sistem pendukung keputusan adalah sistem informasi berbasis komputer yang interaktif, dengan cara mengolah data dengan berbagai model untuk memecahkan masalah-masalah yang tidak terstruktur sehingga dapat memberikan informasi yang bisa digunakan oleh para pengambil keputusan dalam membuat sebuah keputusan. Dalam sebuah sistem pendukung keputusan, sumber daya intelektual yang dimiliki seseorang dipadukan dengan kemampuan computer untuk membantu meningkatkan kualitas dari keputusan yang diambil. Pengambilan keputusan merupakan sebuah proses memilih sebuah tindakan diantara beberapa alternatif yang ada, sehingga tujuan yang diinginkan dapat tercapai. (Alfina & Harahap, 2019)

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 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 terstruktur. 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 rangka mengidentifikasi 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. (Alfina & Harahap, 2019)

Konsep Sistem Pendukung Keputusan (SPK) pertama kali diungkapkan pada tahun 1971 oleh Michael Scoot Morton dengan istilah Management Decision System. Kemudian sejumlah perusahaan, lembaga penelitian dan perguruan tinggi mulai melakukan penelitian dan membangun Sistem Pendukung Keputusan, sehingga dari produksi yang dihasilkan dapat disimpulkan bahwa sistem ini merupakan suatu sistem berbasis komputer yang ditujukan untuk membantu pengambilan keputusan dalam memanfaatkan data dan model tertentu untuk memecahkan berbagai persoalan yang tidak terstruktur. (Rinaldy Manurung, 2020)

II.4 Kriteria Sistem Pendukung Keputusan

Sistem pendukung keputusan dirancang secara khusus untuk mendukung seseorang yang harus mengambil keputusan-keputusan tertentu. Berikut ini beberapa kriteria sistem pendukung keputusan:

1. Interaktif

Sistem pendukung keputusan memiliki *user interface* yang komunikatif, sehingga pemakai dapat melakukan akses secara cepat ke data dan memperoleh informasi yang dibutuhkan.

2. Fleksibel

Sistem pendukung keputusan memiliki sebanyak mungkin variabel masukan, kemampuan untuk mengolah dan memberikan keluaran yang menyajikan alternatif-alternatif keputusan kepada pemakai.

3. Data Kualitas

Sistem pendukung keputusan memiliki kemampuan untuk menerima data kualitas yang dikuantitaskan yang sifatnya subyektif dari pemakainya, sebagai data masukan untuk pengolahan data.

4. Prosedur Pakar

Sistem pendukung keputusan mengandung suatu prosedur yang direncanakan berdasarkan rumusan formal atau juga berupa prosedur kepakaran seseorang atau kelompok dalam menyelesaikan suatu bidang masalah dengan fenomena tertentu (Dian Pratiwi, 2018)

II.5. Pemodelan Sistem Pendukung Keputusan

Pemodelan dalam pembangunan Sistem Pendukung Keputusan dilakukan langkah-langkah sebagai berikut:

1. Studi Kelayakan (*Intelligence*)

Pada langkah ini, sasaran ditentukan dan dilakukan pencarian prosedur, pengumpulan data, identifikasi masalah, identifikasi kepemilikan masalah, klasifikasi masalah, hingga akhirnya terbentuk sebuah pernyataan masalah. Kepemilikan masalah berkaitan dengan bagian apa yang akan dibangun oleh DSS dan apa tugas dari bagian tersebut sehingga model tersebut bisa relevan dengan kebutuhan si pemilik masalah.

2. Perancangan (*Design*)

Pada tahapan ini akan diformulasikan model yang akan digunakan dan kriteria-kriteria yang ditentukan. Setelah itu, dicari alternatif model yang bisa menyelesaikan permasalahan tersebut. Langkah selanjutnya adalah memprediksi keluaran yang mungkin. Kemudian, ditentukan variabel-variabel model.

3. Pemilihan (*Choice*)

Setelah pada tahap design ditentukan berbagai alternatif model beserta variabel-variabelnya, pada tahapan ini akan dilakukan pemilihan modelnya, termasuk solusi dari model tersebut. Selanjutnya, dilakukan analisis sensitivitasnya, yakni dengan mengganti beberapa variabel.

4. Membuat SPK

Setelah menentukan modelnya, berikutnya adalah mengimplementasikannya dalam aplikasi SPK.(B, Muhammad Auliya. S, Yan Watequlis .R, 2018)

II.6 Struktur Sistem Pendukung Keputusan

Keputusan yang diambil untuk menyelesaikan suatu masalah dilihat dari struktur masalahnya terbagi menjadi 3 (tiga) yaitu:

1. Keputusan Terstruktur (*Structured Decision*)

Keputusan yang dilakukan secara berulang-ulang dan bersifat rutin. Prosedur pengambilan keputusan sangatlah jelas. Keputusan tersebut terutama dilakukan pada manajemen tingkat bawah.

2. Keputusan Semi Terstruktur (*Semi Structured Decision*)

Keputusan yang memiliki dua sifat. Sebagian keputusan bisa ditangani oleh komputer dan yang lain tetap harus dilakukan oleh pengambilan keputusan. Prosedur dalam pengambilan keputusan tersebut secara garis besar sudah ada, tetapi ada beberapa hal yang masih memerlukan kebijakan dari pengambil keputusan. Biasanya keputusan semacam ini diambil oleh manajer level menengah dalam suatu organisasi.

3. Keputusan Tak Terstruktur (*Unstructured Decision*)

Keputusan yang penanganannya rumit karena tidak terjadi berulang-ulang atau tidak selalu terjadi. Menurut pengalaman dan berbagai sumber yang bersifat eksternal. Keputusan tersebut umumnya terjadi pada manajemen tingkat atas. (B, Muhammad Auliya. S, Yan Watequlis .R, 2018)

II.7. Karakteristik Sistem Pendukung Keputusan

Menurut Turban dalam Ade Dwi Setya (2018), SPK yang ideal memiliki beberapa karakteristik sebagai berikut:

1. Mendukung untuk pengambilan keputusan terutama pada situasi semi terstruktur dan terstruktur, dengan menyertakan penilaian manusia dan informasi terkomputerisasi.
2. Mendukung untuk keputusan yang independen atau sequential. Keputusan dapat dibuat satu kali, beberapa kali, atau berulang (dalam interval yang sama).
3. Mendukung di semua fase proses pengambilan keputusan, yaitu intelligence, design, choice, dan implementation.
4. SPK bersifat flexible, dapat menyesuaikan perubahan-perubahan kondisi yang terjadi secara tepat.
5. Kontrol penuh oleh pengambil keputusan terhadap semua langkah proses pengambilan keputusan dalam memecahkan suatu masalah. SPK secara khusus menekankan untuk mendukung pengambil keputusan, bukan menggantikan. (B, Muhammad Auliya. S, Yan Watequlis .R, 2018)

II.8. Subsistem dalam Sistem Pendukung Keputusan

Menurut Turban dalam Ade Dwi Setya (2018), suatu SPK memiliki beberapa subsistem yang menentukan kapabilitas dari teknis SPK. Subsistem tersebut antara lain:

1. Subsistem manajemen data. Termasuk database, yang mengandung data yang relevan untuk berbagai situasi dan dikelola oleh perangkat lunak yang disebut Database Management System (DBMS).

2. Subsistem Manajemen Model. Melibatkan model finansial, statistika, manajemen pengetahuan, atau berbagai model kuantitatif lainnya, sehingga dapat memberikan suatu kemampuan analitis, dan manajemen perangkat lunak yang diperlukan bagi sistem. Perangkat lunak ini sering disebut Model Base Management System (MBMS).
3. Subsistem Antarmuka Pengguna. Kemampuan dan pengetahuan pengguna atau pembuat keputusan dalam berinteraksi secara intensif SPK sehingga dapat melakukan analisis.
4. Subsistem Manajemen Berbasis Pengetahuan. Subsistem ini dapat mendukung subsistem lainnya. Subsistem ini dapat terinterkoneksi dengan repository pengetahuan perusahaan (bagian dari sistem manajemen pengetahuan), yang terkadang disebut sebagai basis pengetahuan organisasional. (B, Muhammad Auliya. S, Yan Watequlis .R, 2018)

II.9. Tujuan Sistem Pendukung Keputusan

Menurut Turban dalam Ade Dwi Setya (2018), sistem pendukung keputusan pada hakekatnya memiliki beberapa tujuan, yaitu :

1. Membantu manajer dalam pengambilan keputusan atas masalah semi terstruktur.
2. Memberikan dukungan atas pertimbangan manajer dan bukannya dimaksudkan untuk menggantikan fungsi manajer.

3. Meningkatkan efektifitas keputusan yang diambil manajer lebih dari perbaikan efisiensinya.
4. Kecepatan Komputasi, Komputer memungkinkan para pengambil keputusan untuk melakukan banyak komputerisasi secara cepat dengan biaya yang rendah.
5. Dukungan kualitas, Komputer bisa meningkatkan kualitas keputusan yang dibuat, seperti semakin banyak data yang di akses, makin banyak juga alternatif yang bisa dievaluasi.
6. Mengatasi keterbatasan kognitif dalam pemrosesan data penyimpanan. Menurut Simon dalam Ade Dwi Setya (2018), otak manusia memiliki kemampuan yang terbatas untuk memproses dan menyimpan informasi. Orang-orang kadang sulit mengingat dan menggunakan sebuah informasi dengan cara yang bebas dari kesalahan

II.12. Metode *Mabac*

Mabac (*Multi-Attributive Border Approximation area Comparison*) merupakan salah satu metode sistem pendukung keputusan yang bersifat multikriteria dan dianggap sebagai salah satu metode yang handal dalam pengambilan keputusan rasional. Langkah penyelesaian metode *Multi-Attributive Border Approximation Area Comparison* (MABAC) sebagai berikut:

1. Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan dan penentuan bobot.
2. Membentuk tabel alternatif beserta kriteria masing – masing alternatif.

3. Membentuk matriks keputusan awal (X) berdasarkan rating
4. Normalisasi elemen matriks awal (X)

$$t_{ij} = \frac{x_{ij} - x_{i+}}{x_i - x_{i+}}, \text{ jika jenis kriteria merupakan Biaya/Cost}$$

$$t_{ij} = \frac{x_{ij} - x_{i1}}{x_i - x_{i+}} \text{ jika jenis kriteria merupakan keuntungan/Benefit}$$

5. Perhitungan matriks tertimbang (V)

$$V_{ij} = (w_i * t_{ij}) + w_i,$$

6. Penentuan matriks area perkiraan perbatasan (G)

$$G_i = v_{ij} \cdot 1 = m$$

7. Perhitungan elemen matriks jarak alternatif dari daerah perkiraan perbatasan (Q)

$$Q_{ij} = V_{ij} - G_i$$

8. Perankingan alternatif. Perankingan alternatif dilakukan dengan cara menambahkan setiap elemen kriteria dari masing – masing alternatif berdasarkan matriks jarak alternatif dari daerah perkirann perbatasan (Q).
(Hengky Yosafat : 2020)

II.1. PHP (*Hypertext Preprocessor*)

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. PHP merupakan bahasa *scripting*

server – side, dimana pemrosesan datanya dilakukan pada sisi *server*. Sederhananya, *server* yang akan menterjemahkan skrip program, baru kemudian hasilnya akan dikirim kepada *client* yang melakukan permintaan. Adapun pengertian lain PHP adalah *akronim* dari *Hypertext Preprocessor*, yaitu suatu bahasa pemrograman berbasis kode – kode (*script*) yang digunakan untuk mengolah suatu data dan mengirimkannya kembali ke *web browser* menjadi kode HTML”. (Muhammad Faisal Widad : 2017).

PHP (singkatan rekursif: *PHP Hypertext Preprocessor*) merupakan script yang dapat disisipkan ke dalam HTML. PHP dipakai untuk membuat sistem berbasis web yang dinamis. PHP bahasa pemrograman yang *server-side* sehingga program tersebut akan dijalankan atau diproses oleh server . Hingga saat ini PHP sudah rilis versi 7.4.1 pada 18 desember 2019. (Roberto kaban,. Dkk : 2019)

PHP adalah bahasa pelengkap HTML yang memungkinkan dibuatnya aplikasi dinamis yang memungkinkan adanya pengolahan data dan pemrosesan data. Semua syntax yang diberikan akan sepenuhnya dijalankan pada server sedangkan yang dikirimkan ke browser hanya hasilnya saja. Kemudian merupakan bahasa berbentuk script yang ditempatkan dalam server dan diproses di server. Hasilnya akan dikirimkan ke client, tempat pemakai menggunakan browser. PHP dikenal sebagai sebuah bahasa scripting, yang menyatu dengan tag-tag HTML, dieksekusi di server, dan digunakan untuk membuat halaman web yang dinamis seperti halnya *Active Server Pages (ASP)* atau *Java Server*

Pages (JSP). PHP merupakan sebuah software Open Source (Reza Hermiati : 2021).

II.14. MySQL (*My Structure Query Language*)

MySQL (*My Structure Query Language*) merupakan sebuah program pembuat *database* yang bersifat *Open Source*, artinya semua orang dapat menggunakannya dan dapat dijalankan pada semua *platform* baik *Windows* maupun *linux*. MySQL juga merupakan sebuah perangkat lunak sistem manajemen basis data *SQL* yang bersifat jaringan sehingga dapat digunakan untuk aplikasi multi *user*. MySQL juga sering dikenal dengan nama sistem manajemen *database* relasional. Suatu *database* relasional menyimpan data dalam tabel yang terpisah. Tabel -tabel tersebut terhubung oleh suatu relasi terdefinisi yang memungkinkan memperoleh kombinasi data dari beberapa tabel dalam suatu permintaan. Untuk administrasi *database*, seperti pembuatan *database*, pembuatan tabel, dan sebagainya dapat digunakan aplikasi berbasis web seperti *PHP MyAdmin* dengan aplikasi *XAMPP*. (Muhammad Faisal Widad : 2017)

MySQL merupakan suatu jenis *database server* yang sangat terkenal. MySQL termasuk jenis RDBMS (*Relational Database Manajement System*). MySQL mendukung bahasa pemrograman PH, bahasa permintaan yang terstruktur, karena pada penggunaannya SQL memiliki beberapa aturan yang telah distandarkan oleh asosiasi yang bernama ANSI. MySQL merupakan RDBMS (*Relational Database Management System*) server. RDBMS adalah program yang memungkinkan pengguna database untuk membuat, mengelola,

dan menggunakan data pada suatu model relational. Dengan demikian, tabel-tabel yang ada pada database memiliki relasi antara satu tabel dengan tabel lainnya. Beberapa keunggulan dari MySQL yaitu :

- a. Cepat, handal dan mudah dalam penggunaannya. MySQL lebih cepat tiga sampai empat kali dari pada database server komersial yang beredar saat ini, mudah diatur dan tidak memerlukan seseorang yang ahli untuk mengatur administrasi pemasangan MySQL.
- b. Didukung oleh berbagai bahasa Database Server MySQL dapat memberikan pesan Error dalam berbagai bahasa seperti Belanda, Portugis, Spanyol, Inggris, Perancis, Jerman, dan Italia.
- c. Mampu membuat tabel berukuran sangat besar. Ukuran maksimal dari setiap tabel yang dapat dibuat dengan MySQL adalah 4 GB sampai dengan ukuran file yang dapat ditangani oleh sistem operasi yang dipakai.
- d. Lebih murah MySQL bersifat open source dan didistribusikan dengan gratis tanpa biaya untuk UNIX platform, OS/2 dan Windows Platform. Melekatnya integrasi PHP dengan MySQL. Keterikatan antara PHP dengan MySQL yang sama-sama Software Open-Source sangat kuat, sehingga koneksi yang terjadi lebih cepat jika dibandingkan dengan menggunakan database server lainnya. Modul MySQL di PHP telah dibuat Built-in sehingga tidak memerlukan konfigurasi tambahan pada File konfigurasi Php ini (Reza Hermiati : 2021).

II.15. Database

Database adalah basis dari data, dengan kata lain database merupakan kumpulan data, dasar yang digunakan untuk menampilkan data atau informasi, sekumpulan data atau informasi teratur berdasarkan criteria tertentu yang saling berhubungan. Selain itu juga database dapat di defenisikan sebagai susunan *record* data operasional lengkap dari suatu organisasi perusahaan, yang terorganisir dan disimpan secara terintegrasi. (Herlina Trisnawati, 2018)

Data adalah penggambaran suatu fakta atau keadaan. Informasi merupakan sekumpulan data yang telah diolah menjadi suatu bentuk sehingga berguna bagi penerima dan pemakai. Dalam dunia komputer *database* bisa dikategorikan bisa sangat spesial karena selalu menjadi hal utama dalam perancangan sistem suatu perusahaan, tentunya ada alasan tertentu mengapa *database* menjadi prioritas sendiri dalam kinerja manajemen perusahaan. (Herlina Trisnawati, 2018)

II.16. Normalisasi

Normalisasi adalah proses pembentukan struktur basis data sehingga sebagian besar ambiguity bisa dihilangkan. Normalisasi merupakan sebuah teknik dalam logical desain sebuah basis data relasional yang mengelompokkan atribut dari suatu tabel sehingga membentuk struktur tabel yang normal. Adapun kriteria tabel dikatakan normal adalah ketika tidak ada kerangkapan data (redudansi data). (Puspitasari et al., 2018)

Tujuan dari normalisasi adalah untuk :

1. Untuk menghilangkan kerangkapan data sehingga meminimumkan pemakaian *storage* yang dipakai oleh *base relations (file)*.
2. Untuk mengurangi kompleksitas.
3. Untuk mempermudah pemodifikasian data.

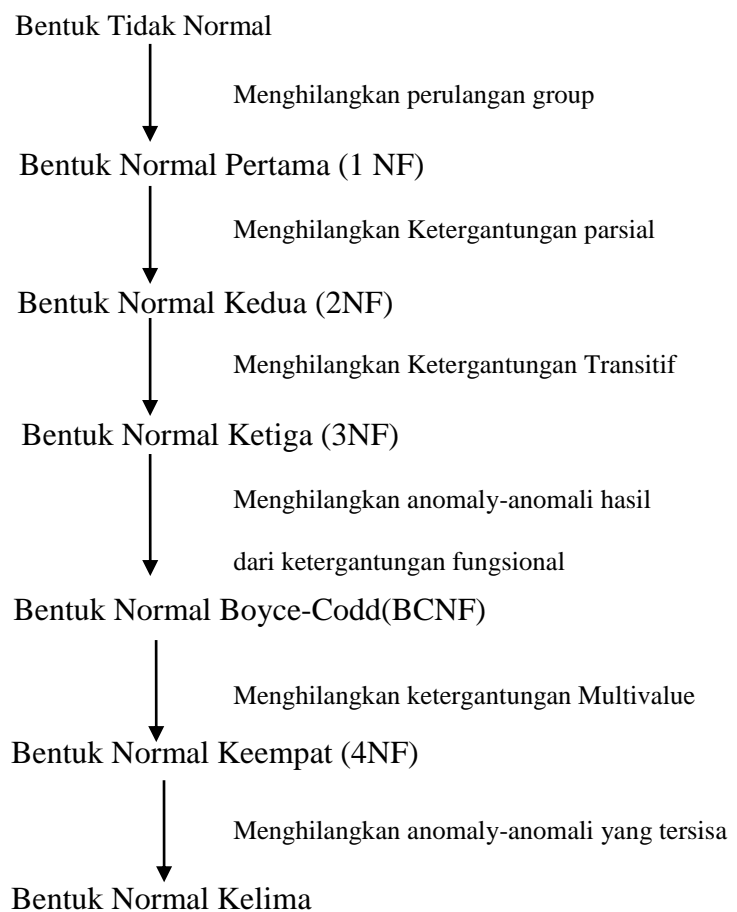
II.16.1. Proses Normalisasi

Gambaran proses normalisasi yaitu:

1. Data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu ke beberapa tingkat, kemudian.
2. Apabila tabel yang diuji belum memenuhi persyaratan tertentu, maka tabel tersebut perlu dipecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi bentuk yang optimal. Untuk melakukan proses tersebut dibutuhkan beberapa tahapan.

Tahapan dalam normalisasi dimulai dari tahap paling ringan (*1NF*) hingga paling ketat (*5NF*). Biasanya hanya sampai pada tingkat *3NF* atau *BCNF* karena sudah cukup memadai untuk menghasilkan tabel-tabel yang berkualitas baik.

Urutan tahapan normalisasi tampak seperti gambar II.1.



Gambar II.1. Tahapan pada Normalisasi
(Sumber : Dwi Puspitasari, dkk ; 2018)

Adapun aturan dalam normalisasi adalah suatu tabel dikatakan baik (efisien) atau normal jika memenuhi 3 kriteria sbb:

1. Jika ada dekomposisi (penguraian) tabel, maka dekomposisinya harus dijamin aman (*Lossless-Join Decomposition*). Artinya, setelah tabel tersebut diuraikan / didekomposisi menjadi tabel-tabel baru, tabel-tabel baru tersebut bisa menghasilkan tabel semula dengan sama persis.
2. Terpeliharanya ketergantungan fungsional pada saat perubahan data (*Dependency Preservation*).
3. Tidak melanggar *Boyce-Codd Normal Form (BCNF)*.

Jika kriteria ketiga (*BCNF*) tidak dapat terpenuhi, maka paling tidak tabel tersebut tidak melanggar Bentuk Normal tahap ketiga (*3rd Normal Form / 3NF*). Pada penelitian ini formula yang dibuat sampai memenuhi bentuk normal ke 3 yaitu *3NF*.

Proses Normalisasi Berdasarkan tahapan normalisasi, terdapat enam bentuk normal yaitu Elmasri & Navathe (2013):

1. Bentuk Normal Tahap Pertama (*1st Normal Form / 1NF*).
2. Bentuk Normal Tahap Kedua (*2nd Normal Form / 2NF*).
3. Bentuk Normal Tahap (*3rd Normal Form / 3NF*).
4. *Boyce-Code Normal Form (BCNF)*.
5. Bentuk Normal Tahap (*4th Normal Form / 4NF*) 6. Bentuk Normal Tahap (*5th Normal Form / 5NF*)

Bentuk Normal Pertama / *1NF*, memiliki aturan:

- a. Tidak adanya atribut *multi-value*, atribut komposit atau kombinasinya.
- b. Mendefinisikan atribut kunci.
- c. Setiap atribut dalam tabel tersebut harus bernilai atomic (tidak dapat dibagi-bagi lagi).

Bentuk Normal Kedua / *2NF*, memiliki aturan :

- a. Sudah memenuhi dalam bentuk normal kesatu (*1NF*).
- b. Semua atribut bukan kunci hanya boleh tergantung (*functional dependency*) pada atribut kunci. Jika ada ketergantungan parsial maka atribut tersebut harus dipisah pada tabel yang lain.

- c. Perlu ada tabel penghubung ataupun kehadiran foreign key bagi atribut-atribut yang telah dipisah tadi.

Bentuk Normal Ketiga / *3NF*, memiliki aturan:

- a. Sudah memenuhi dalam bentuk normal kedua (*2NF*)
- b. Tidak ada ketergantungan transitif (dimana atribut bukan kunci tergantung pada atribut bukan kunci lainnya). (Puspitasari et al., 2018)

II.17. UML (*Unified Modelling Language*)

UML (Unified Modelling Language) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek. Hal ini disebabkan karena *UML* menyediakan Bahasa pemodelan visual yang memungkinkan bagi pengembangan sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*Sharing*) dan mengkomunikasikan rancangan dengan baik. (Helmi Kurniawan : 2015)

Unified Modelling Language (UML) merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek. UML merupakan salah satu alat bantu yang sangat handal dalam bidang pengembangan sistem berorientasi objek karena UML menyediakan Bahasa pemodelan visual yang memungkinkan pengembang sistem membuat *blue print* atas visinya dalam bentuk yang baku. UML berfungsi sebagai jembatan dalam mengkomunikasikan beberapa aspek dalam sistem melalui jumlah elemen grafis yang bisa dikombinasikan menjadi.

Unified Modeling Language (UML) biasa digunakan untuk :

- a. Menggambarkan batasan sistem dan fungsi - fungsi sistem secara umum, dibuat dengan *use case* dan *actor*.
- b. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan *interaction diagrams*.
- c. Menggambarkan representasi struktur *static* sebuah sistem dalam bentuk *class diagrams*.
- d. Membuat model behavior “yang menggambarkan kebiasaan atau sifat sebuah sistem” dengan *state transition diagrams*.
- e. Menyatakan arsitektur implementasi fisik menggunakan *component and development*.
- f. Menyampaikan atau memperluas *functionality* dengan *stereotypes*.

(Alfina & Harahap, 2019)


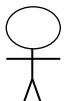

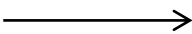
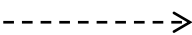
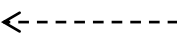
UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat dan pengembangan sistem. (Urva et al., 2015)

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah Empat alat bantu sebagai berikut: *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram* dan *Class Diagram*.

1. Use Case Diagram

Use Case Diagram merupakan pemodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat. Use Case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan Use Case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Symbol-simbol yang digunakan dalam Use Case Diagram Yaitu

Tabel II.3. Simbol Use Case



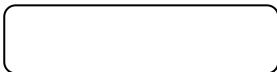
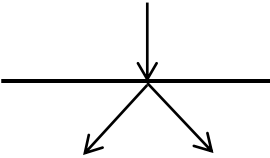
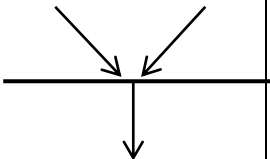
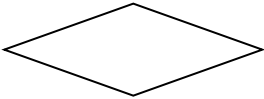

Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang 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 : Gellysa Urva, dkk ; 2015)

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.4. 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 rake, 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.

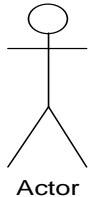
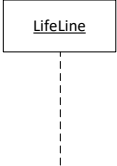

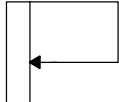


(Sumber : Gellysa Urva,dkk ; 2015)

3. *Sequence Diagram*

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan

diterima antara objek. Simbol-simbol yang digunakan dalam *Sequence Diagram* yaitu:

Tabel II.5. *Sequence Diagram*

Gambar	Keterangan
	Orang,Proses atau sistem lain yang berinteraksi dengan sistem informasi.
	Merepresentasikan entitas tunggal dalam sequence diagram, digambarkan dengan kotak. Entitas ini memiliki nama, <i>Stereotype</i> atau berupa <i>instance</i> .
	<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> , <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 : Gellysa Urva, dkk ; 2015)

4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antara kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan

tanggung jawab entitas yang merupakan 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 kardinaliti.

Tabel II.6. 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 : Gellysa Urva, dkk ; 2015)