

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Tinjauan Studi**

Dalam penyusunan penelitian ini, terdapat beberapa penelitian terdahulu yang penulis jadikan sebagai referensi penelitian. Terdapat 6 (enam) jurnal yang penulis jadikan sebagai pegangan, pedoman, acuan, serta pembandingan dengan penelitian yang penulis lakukan.

Pada jurnal berjudul “Sistem Pendukung Keputusan Penerima *Store Of The Month* Pada Toko Indomaret Menggunakan Metode Analytical Hierarchy Process ( *AHP* ) Studi Kasus PT Indomarco Prismatama Medan” yang disusun oleh Tison Nopember Simanjuntak pada tahun 2016 mengangkat permasalahan yang ada proses penentuan toko indomaret yang berhak menerima *Store Of the month* di PT Indomarco Prismatama Cabang Medan. Permasalahan yang di jumpai adalah proses penentuan dan penilaian toko indomaret masih manual dan berdasarkan persepsi orang, belum ada kriteria yang pasti untuk dapat dijadikan patokan sebagai kunci untuk menentukan toko indomaret yang berhak menerima *Store Of the month*. Berdasarkan masalah tersebut dirancanglah sebuah sistem pendukung keputusan berbasis desktop. Sistem tersebut menjawab permasalahan yang dihadapi sebelumnya, sistem tersebut dapat mengolah data toko indomaret yang di prioritaskan sebagai penerima *Store Of the month* berdasarkan kriteria yang telah di tetapkan perusahaan sebagai dasar penilaian (Simanjuntak, 2016).

Jurnal kedua adalah jurnal yang berjudul “Penentuan Bonus Pada Karyawan Dengan Menggunakan Metode *ANP* (Studi Kasus: Pt.Asahimas Flat Glass,Tbk Jakarta)” yang disusun oleh Sandra Jamu Kuryanti dan Novita Indriyani. Jurnal tersebut diterbitkan pada tahun 2016. Jurnal ini mengangkat permasalahan mengenai tentang penentuan bonus karyawan berdasarkan alternatif yang ada, hasilnya berdasarkan penelitian yang dilakukan di jurnal ini penerapan sistem berdampak positif terhadap penyeleksian karyawan yang ditentukan melalui perangkungan nilai bobot yang ada sehingga penentuan karyawan yang berhak mendapatkan bonus lebih optimal dan efisien (Kuryanti dan Indriyani, 2016) .

Jurnal ketiga adalah jurnal yang berjudul “Sistem Pendukung Keputusan Pemilihan Laptop Dengan Metode *TOPSIS*” yang disusun oleh Kurniasih dan Desi Leha yang diterbitkan pada tahun 2013. Jurnal ini mengangkat permasalahan tentang proses pemilihan laptop untuk pengguna berdasarkan faktor keinginan dan anggaran. Sehubungan dengan hal itu maka dirancanglah sistem menggunakan metode *TOPSIS* karna metode ini dapat mencari alternatif terbaik dari beberapa alternatif yang ditawarkan, hasilnya sistem yang dibangun mampu membantu permasalahan dalam menentukan laptop yang diinginkan pengguna dari segi anggaran maupun biaya, metode ini juga dapat mengurutkan alternatif dari nilai yang terbesar hingga ke nilai yang terkecil (Kurniasih, 2013).

Jurnal selanjutnya adalah jurnal yang berjudul “Sistem Pendukung Keputusan Rekomendasi Tempat Wisata Di Jawa Barat Menggunakan Metode Technique For Order Preference By Similarity To Ideal Solution (*TOPSIS*)” yang disusun oleh Dewi Purnamasari, Gunawan Abdillah, Agus Komarudin, diterbitkan pada tahun 2017. Jurnal ini membahas tentang permasalahan wisatawan yang mencari tempat wisata di Jawa Barat dari segi jarak, anggaran dan fasilitas. Dari kriteria tersebut akan menjadi patokan mengingat banyaknya obyek wisata yang menarik di Jawa Barat. Hasilnya sistem yang dibangun menghasilkan tempat wisata yang terekomendasi dari perbandingan tempat wisata yang memiliki kriteria sesuai yang diinginkan (Dewi Purnamasari dkk, 2017).

Jurnal berikutnya yang penulis jadikan sebagai acuan adalah jurnal yang berjudul “Sistem Pendukung Keputusan Reward Bonus Karyawan Dengan Metode *TOPSIS*” jurnal ini disusun oleh Darsono Nababan dan Robbi Rahim yang terbit pada tahun 2018. Jurnal ini membahas tentang pemilihan karyawan atau pekerja yang berhak dalam penerimaan intensif atau bonus kedisiplinan. Sistem yang dibangun menghasilkan sistem yang mampu menentukan karyawan mana saja yang berhak dalam menerima bonus kedisiplinan berdasarkan kriteria absen, perilaku, prestasi dan kerjasama tim menggunakan metode *TOPSIS* (Nababan & Rahim, 2018).

Jurnal terakhir yang penulis jadikan sebagai acuan dalam penelitian ini adalah jurnal yang berjudul “Sistem Pendukung Keputusan Pemilihan Dosen

Berprestasi Dengan Metode Anp Dan Topsis” jurnal ini disusun oleh Rendra Gustriansyah pada tahun 2016. Jurnal ini membahas tentang pemilihan Dosen berprestasi dimana pemilihan tersebut harus sesuai kriteria yang ada, untuk membantu penentuan urutan Dosen yang berprestasi digunakan metode ANP dalam menentukan bobot setiap kriterianya lalu menggabungkan metode TOPSIS yang digunakan untuk menghitung perangnya, hasilnya sistem yang dibangun mampu memberikan urutan Dosen berprestasi berdasarkan bobot prioritas kerja (Gustriansyah, 2016).

Jurnal – jurnal yang telah dipaparkan di atas merupakan dasar dari penelitian yang penulis akan lakukan. Dibandingkan dengan jurnal-jurnal tersebut penulis akan melakukan sebuah penelitian yang serupa namun memiliki pembeda. Penulis akan membangun sebuah sistem pendukung keputusan yang akan membantu proses penentuan toko Indomaret mana saja yang berhak mendapatkan *Store of the month* menggunakan metode TOPSIS dan ANP. Penelitian tersebut merupakan kombinasi dari penelitian sebelumnya yang membahas tentang pembangunan sistem pendukung keputusan di bidang yang lain yang terbukti memberi dampak positif, yang diimplementasikan pada perusahaan dalam menentukan suatu keputusan berdasarkan kriteria yang ada.

## **2.2 Tinjauan Pustaka**

### **2.2.1 Sistem Pendukung Keputusan (*Decision Support System Atau DSS*)**

Sistem Pendukung Keputusan merupakan suatu sistem interaktif yang dapat membantu pekerjaan atau tugas top manajer dalam mengambil sebuah keputusan melalui penggunaan model serta data keputusan untuk mencari solusi dan memecahkan masalah-masalah yang sifatnya abstrak atau semi terstruktur (Abadi & Latifah, 2016).

Tahapan-tahapan yang harus dilakukan dalam pengambilan keputusan adalah sebagai berikut (Abadi & Latifah, 2016):

- a. Tahap Pemahaman (*Intelligence Phase*)

Tahap ini adalah proses pencarian dan pendeteksian dari lingkup masalah yang ada serta proses pengenalan problem yang tersedia. Data inputan atau masukan didapatkan, diproses serta di olah lalu diuji dalam rangka pengidentifikasian masalah.

b. Tahap Perancangan (Desigen Phace)

Tahap ini adalah proses pencarian dan pengembangan alternatif solusi/tindakan yang dapat diambil. Tahap perencanaan tersebut merupakan konsep atau ide kejadian nyata yang diperkecil, sehingga dilakukan proses vertifikasi dan validasi yang bertujuan untuk mengetahui keakuratan model dalam meneliti kasus atau permasalahan yang ada.

c. Tahap Pemilihan (Choice Phace)

Tahap ini dilakukan proses seleksi atau pemilihan terhadap berbagai alternatif atau solusi yang muncul ketika tahap perencanaan agar dapat ditentukan mana kriteria-kriteria yang sesuai berdasarkan tujuan yang ingin dicapai.

d. Tahap Implementasi (Implementation Phace)

Tahap ini diimplementasikan kedalam perancangan sistem yang sudah dibuat di tahap perencanaan serta memilih alternatif tindakan yang telah dipilih ketika tahap pemilihan berlangsung.

### **2.2.2 TOPSIS (*Technique For Order Preference by Similiarity to Ideal Solution*)**

*TOPSIS* merupakan salah satu metode pengambilan keputusan multi kriteria. Prinsip *TOPSIS* adalah alternatif yang dipilih harus mendapatkan jarak terjauh dari solusi ideal negatif dan terdekat dari solusi ideal positif dengan menggunakan jarak Euclidean / distance Euclidean untuk menentukan suatu alternatif paling dekat dengan solusi optimal (Renaldo dkk, 2019). Solusi ideal positif diartikan bahwa keseluruhan dari jumlah nilai terbaik yang dapat dicapai untuk setiap atribut, sedangkan solusi ideal negatif adalah kebalikannya yaitu terdiri dari keseluruhan dari jumlah nilai terburuk yang dicapai untuk setiap atribut. Susunan prioritas alternatif bisa tercapai berdasarkan perbandingan terhadap jarak relatifnya yang didapatkan dari kedua solusi tersebut lalu dipertimbangkan, jarak Terhadap

solusi ideal negatif dan jarak terhadap solusi ideal positif dengan jalan mengambil kedekatan relatif dari solusi ideal positif. Metode *TOPSIS* sering digunakan dalam menyelesaikan pengambilan keputusan. Hal ini dikarenakan konsepnya lumayan simple dan sederhana, efisien, mudah dipahami, komputasinya efektif dan memiliki kemampuan mengukur kinerja relatif dari alternatif- alternatif keputusan yang ada (Kurniasih, 2013).

Kelebihan metode *TOPSIS* :

- A. Konsepnya sederhana dan mudah dipahami, kesederhanaan ini dilihat dari alur proses metode *TOPSIS* yang tidak begitu rumit. Karena menggunakan indikator kriteria dan variabel alternatif sebagai pembantu untuk menentukan keputusan
- B. Komputasinya efisien, perhitungan komputasinya lebih efisien dan dan cepat,
- C. Mampu dijadikan sebagai pengukur kinerja alternatif dan juga alternatif keputusan dalam sebuah bentuk output komputasi yang sederhana
- D. Dapat digunakan sebagai metode pengambilan keputusan yang lebih cepat

Kekurangan metode *TOPSIS* :

- A. Belum adanya penentuan bobot prioritas yang menjadi prioritas hitungan terhadap kriteria, yang berguna untuk meningkatkan validitas nilai bobot perhitungan kriteria. Maka dengan alasan ini, metode ini dapat di kombinasikan misalnya dengan metode *AHP* agar menghasilkan output atau keputusan yang lebih maksimal
- B. Belum adanya bentuk linguistik untuk penilaian alternatif terhadap kriteria, biasanya bentuk linguistik ini di interpretasikan dalam sebuah bilangan fuzzy

- C. Belum adanya mediator seperti hirarki jika di proses secara mandiri maka dalam ketepatan pengambilan keputusan cenderung belum menghasilkan keputusan yang sempurna
- D. Metode *TOPSIS* ini dapat digunakan dalam menentukan perangsangan alternatif dengan memperhitungkan solusi ideal dari suatu masalah dan penentuan bobot setiap kriteria. Namun, kurang baik jika digunakan dalam mendapatkan bobot yang memperhitungkan hubungan antara kriteria. Walaupun dapat dilakukan dengan pairwise comparison, tetapi membutuhkan matriks dan perhitungan yang lebih rumit. Oleh karena itu, dilakukan penggabungan dengan metode lain seperti ANP Analytic Network Process dalam mengatasi masalah pembobotan tersebut.

Langkah-langkah metode *TOPSIS* :

- A. Mendefinisikan permasalahan yang akan diselesaikan dengan metode *TOPSIS*.
- B. Membuat matriks keputusan sesuai dengan permasalahan yang akan dipecahkan, kemudian lakukan normalisasi matriks dengan persamaan berikut ini :

$$rij = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}} \quad (2.2.1)$$

Dimana  $rij$  merupakan matriks hasil normalisasi dari matriks dasar permasalahannya, dengan  $i = 1,2,3,\dots,m$ , dan  $j = 1,2,3 \dots N$ . Sedangkan  $x_{ij}$  merupakan matriks dasar yang akan dinormalisasikan. Untuk setiap  $i$  menunjukkan baris dari matriks, dan untuk setiap  $j$  menunjukkan kolom dari setiap matriks.

- C. Lakukan normalisasi matrik  $rij$  menggunakan rating bobot sehingga diperoleh matrik rating bobot ternormalisasi, rumus yang digunakan adalah  $Y_{ij} = w_i.rij$ .

Dimana  $y_{ij}$  adalah matriks rating terbobot,  $w_i$  adalah bobot rating ke  $i$ , dan  $r_{ij}$  adalah matriks hasil normalisasi pada langkah ke dua. Untuk  $i = 1, 2, \dots, m$ , dan  $j = 1, 2, \dots, n$ . Dalam hal ini, bobot rating harus ditentukan berdasarkan jumlah variabel keputusan yang sedang diselesaikan.

- D. Tentukan solusi ideal positif ( $A^+$ ) dan solusi ideal negatif ( $A^-$ ) berdasarkan nilai matriks rating terbobot pada langkah ke-3. Berikut persamaan yang digunakan untuk mencari nilai solusi ideal positif dan nilai solusi ideal negatif dengan rumus  $a^+ = \max(y_{1+}, y_{2+}, \dots, y_{n+})$  dan  $a^- = \max(y_{1-}, y_{2-}, \dots, y_{n-})$ .
- E. Tentukan jarak antara nilai terbobot setiap alternatif terhadap solusi ideal positif dan solusi ideal negatifnya. Untuk menentukan jarak antara nilai terbobot setiap alternatif terhadap solusi ideal positif, digunakan rumus Sedang untuk menghitung jarak antara nilai terbobot setiap alternatif terhadap solusi ideal negatif, digunakan persamaan berikut :

$$D_i^+ = \sqrt{\sum_{j=1}^n (y_j^+ - y_{ij})^2} \quad (2.2.2)$$

- F. Langkah terakhir adalah menghitung nilai preferensi untuk setiap alternatif dengan persamaan berikut :

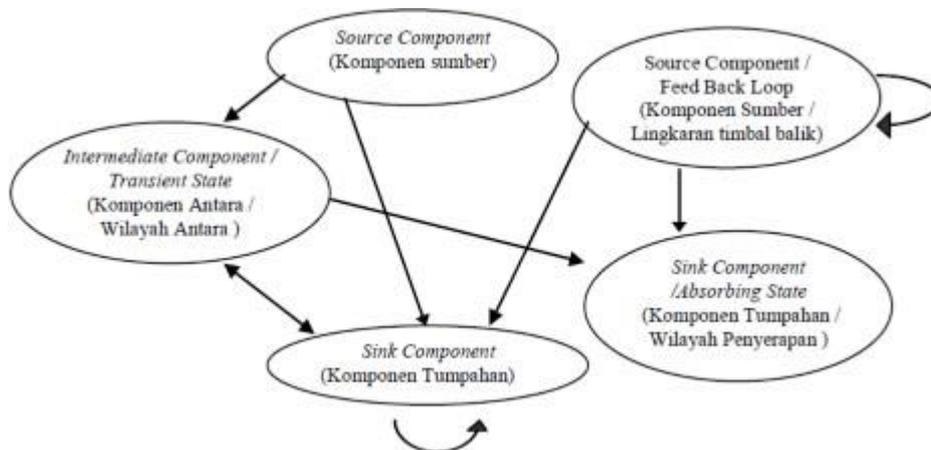
$$V_i = \frac{D_i^-}{D_i^- + D_i^+} \quad (2.2.3)$$

### 2.2.3 ANP (*Analytic Network Process*)

Metode ANP merupakan pengembangan metode AHP. Metode ANP mampu memperbaiki kelemahan AHP berupa kemampuan mengakomodasi keterkaitan antar alternatif atau kriteria. Model dari metode ANP yaitu berupa jaringan sehingga dapat diketahui saling keterkaitan antara setiap elemen yang ada pada satu kriteria yang sama, ataupun terhadap elemen-elemen yang berbeda kriteria (Rahmanda, 2017).

Model ini merupakan pengembangan dari *AHP* sehingga lebih memiliki kompleksitas dibanding metode *AHP*. Metode *ANP* mampu memperbaiki kelemahan *AHP* berupa kemampuan mengakomodasi keterkaitan antar kriteria atau alternatif. Keterkaitan pada metode *ANP* ada 2 jenis yaitu keterkaitan dalam satu set dalam elemen (*inner dependence*) dan keterkaitan antara elemen yang berbeda (*outer dependence*). Adanya keterkaitan antar elemen tersebut menyebabkan metode *ANP* lebih kompleks dibanding metode *AHP*.

Dalam implementasi pemecahan masalah, *ANP* bergantung pada alternatif-alternatif dan kriteria yang ada. Pada jaringan *AHP* terdapat level tujuan, kriteria, dan alternatif, yang masing-masing level memiliki elemen. Sedangkan pada jaringan *ANP*, level dalam *AHP* disebut cluster yang dapat memiliki kriteria dan alternatif di dalamnya. Elemen dalam suatu komponen atau cluster dapat 23 mempengaruhi elemen lain dalam komponen atau cluster yang sama (*inner dependence*), dan dapat pula mempengaruhi elemen cluster yang lain (*outer dependence*) dengan memperhatikan setiap kriteria. Yang diinginkan dalam *ANP* adalah mengetahui keseluruhan pengaruh dari semua elemen. Oleh karena itu, semua kriteria harus diatur dan dibuat prioritas dalam suatu kerangka kerja hirarki atau jaringan, melakukan perbandingan dan sintesis untuk memperoleh urutan prioritas dari sekumpulan kriteria yang ada. Kemudian diturunkan pengaruh dari elemen dalam sistem umpan balik (*feedback*) dengan memperhatikan masing-masing kriteria (Gustriansyah, 2016). Struktur jaringan ditunjukkan pada gambar 2.2.11.



Gambar 2.2. 1 Struktur jaringan metode ANP

#### 2.2.4 Store Of The Month

Store of the month (toko terbaik bulan ini) adalah suatu insentif atau kompensasi (pemberian uang diluar gaji) penghargaan yang diberikan oleh pihak pimpinan perusahaan kepada karyawannya yang sudah berkontribusi dalam memenuhi target perusahaan, agar mereka bekerja dengan motivasi yang tinggi, berprestasi dalam mencapai tujuan-tujuan perusahaan atau sebagai pengakuan terhadap prestasi hasil kerja dan kontribusi karyawan kepada perusahaan (Simanjuntak, 2016).

#### 2.2.5 Basis Data

Basis data atau database adalah sekumpulan data atau gudang data yang saling berhubungan atau berelasi dan disusun sedemikian rupa dengan aturan tertentu yang logis untuk menghasilkan informasi yang nantinya dapat digunakan oleh sistem maupun user kembali untuk di manfaatkan dan di olah lagi dengan cepat dan mudah. Dengan basis data seseorang dapat menyimpan sebuah informasi ke dalam media penyimpanan, baik penyimpanan secara hardware maupun penyimpanan cloud untuk nantinya dapat digunakan kembali (Febriani & Permadi, 2017).

### 2.2.6 Java

Java adalah bahasa pemrograman yang mampu berjalan di berbagai sistem saat ini, di antaranya adalah sistem komputer dan handphone. Fungsi dari java ini diantaranya adalah multi platform dimana program yang di buat dapat di jalankan di berbagai OS berbasis dekstop seperti windows dan linux. Selain itu java juga dapat di jalankan di dalam mobile application android yang di bantu dengan IDE android studio (Natsir, 2016).

### 2.2.7 NetBeans

Netbeans adalah salah satu IDE (Integrated Development Environment) yang dikembangkan dengan bahasa pemrograman java. Netbeans mempunyai lingkup pemrograman yang terintegrasi dalam satu paket software atau perangkat lunak yang didalamnya menyediakan pemrograman untuk membangun suatu aplikasi dengan interpreter, GUI (antarmuka grafis), text editor (pengeditan dalam bentuk teks), dan compiler (penerjemah). Netbeans adalah perangkat lunak yang dapat digunakan secara gratis, entah itu untuk komersial maupun non komersial yang secara user dapat diginakan open source dan didukung oleh Sun Microsystem. (Febriani & Permadi, 2017).

Berikut salah satu kelebihan dan kekurangan NetBeans :

a. Kelebihan Netbeans

Diantara IDE lainnya NetBeans mempunyai kelebihan yaitu tidak berbayar atau gratis, sehingga dapat digunakan semua kalangan developer. Selain itu netbeans sangat kompatibel dengan swing karena langsung dikembangkan oleh microsystem yang tidak lain adalah pengembang swing.

b. Kekurangan Netbeans

Salah satu yang menjadi kekurangan Netbeans yaitu Hanya mensupport satu pengembangan java yaitu swing, padahal java sendiri dikembangkan juga oleh eclipse yang terkenal dengan nama Jface dan SWT yang lumayan cukup populer. Netbeans mempermanenkan source untuk java yang sedang dikerjakan dalam sebuah code generated, sehingga para programmer tidak dapat mengeditnya secara manual.



Gambar 2.2. 2 Software NetbeansIDE

### 2.2.8 IDE (Integrated Development Environment)

Integrated Development Environment atau sering disingkat IDE adalah program untuk komputer sebagai alat untuk mengembangkan sebuah program atau aplikasi komputer yang memiliki macam-macam fasilitas yang dibutuhkan dalam pembangunan software atau perangkat lunak. Tujuan dari IDE adalah untuk menyediakan dan memfasilitasi semua kebutuhan yang diperlukan dalam membangun sebuah perangkat lunak (Wihidayat & Maryono, 2017). IDE sangat diperlukan seorang programmer karna harus mengetahui berbagai bahasa pemrograman untuk membangun perangkat lunak. Contoh IDE yang di gunakan para developer adalah Netbeans IDE, Intelijs IDE, Reactjs, Nodejs, dan masih banyak lagi. Salah satu fasilitas IDE adalah sebagai berikut :

- a. Editor, fasilitas ini digunakan untuk membuat kode atau fungsi-fungsi yang di inputkan oleh developer dapat diterjemahkan oleh otak/mesin komputer sebagai kode programming.
- b. Compiler, fasilitas ini digunakan untuk menjalankan bentuk kode programing yang tadi dituliskan di Editor lalu diubah ke dalam bentuk binary yang kemudian ditampilkan sesuai perintah kode mesin.
- c. Debugger, fasilitas ini digunakan untuk mengecek dan mencoba alur jalannya kode program untuk mencari kesalahan yang terjadi dari program yang di bangun atau sering disebut bug.

- d. Views/Frame, fasilitas ini digunakan untuk menampilkan dan menentukan hasil perintah dari kode program yang di jalankan oleh Compiler dari Editor.

### 2.2.9 ERD (Entity Relationship Diagram)

Menurut (Fatimah & Samsudin, 2019), Entity Relationship Diagram (ERD) adalah teknik atau suatu model untuk mempresentasikan hubungan antar data berdasarkan data yang mempunyai relasi. ERD digunakan untuk memodelkan kebutuhan data dari suatu perusahaan atau organisai, ada tahap analisis yang dijalankan oleh system analys seolah-olah ada bahan dasara sebagai desain database yang dirancang sedemikian rupa sebagai dasar dari sistem yang dikembangkan (Fatimah & Samsudin, 2019).

Atau pengertian sempitnya ERD merupakan sebuah konsep yang menjelaskan hubungan antara data yang disimpan maupun penyimpanan data (database) dan dilandasi dari kenyataan yang ada yang terdiri dari sekumpulan objek yaitu disebut sebagai entitas dan relasi atau hubungan antar objek- objek tersebut (Fatimah & Samsudin, 2019).

Terdapat 3 komponen dalam pembentukan ERD yaitu :

a. Entitas

Entitas adalah objek atau individu yang mempunyai ekstensi menarik dan berbeda dari yang lain di bidang organisasi yang dimodelkan.

Contoh : Pengendara, Kartu Tanda Penduduk (KTP), dan SIM (Surat Izin Mengemudi).

b. Hubungan (relasi/relationship)

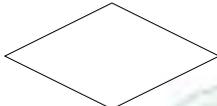
Hubungan (relasi/relationship) adalah hubungan antar dua entitas maupun lebih yang berfungsi untuk mengatur operasi suatu database.

Contoh : Pengendara mendaftar dengan KTP untuk mendapatkan SIM), relasinya adalah mendaftar.

c. Atribut

Atribut adalah karakter yang membedakan entitas yang satu dengan entitas lainnya, untuk lebih rincinya dapat dilihat pada tabel 2.2.1.

Tabel 2.2. 1 Simbol ERD

No	Simbol	Nama	Keterangan
1		Entitas	Entitas dapat berupa elemen besar dalam sebuah organisasi atau lingkungan.
2		Hubungan atau Relasi	Menunjukkan nama relasi antar satu entitas dengan entitas lainnya.
3		Atribut	Atribut adalah karakteristik dari sebuah entitas.
4		Garis Relasi	Menunjukkan hubungan (keterkaitan) antar entitas.
5		Entitas Lemah	Entitas yang kemunculannya tergantung dari entitas lain yang lebih kuat.

Jenis-jenis atribut :

a. Atribut Kunci

Atribut kunci merupakan atribut yang unik karena pada atribut yang dijadikan sebagai kunci tidak boleh ada baris data yang mempunyai nilai sama. Atribut kunci adalah satu atau gabungan beberapa atribut yang mampu dengan sendirinya membedakan semua baris dalam tabel secara unik.

Contoh : Nomor Induk Karyawan (NIK), NIS dan nomor inti lainnya

b. Atribut sederhana

atribut yang bernilai otomatis dan tidak bisa dipecah lagi. Contoh pada tabel customer adalah nomor\_identitas dan jaminan dimana atribut ini tidak dapat dipecah lagi.

c. Atribut Multivalued (Bernilai Banyak)

Atribut multivalued adalah atribut yang dapat diisi lebih dari satu nilai tetapi masih pada jenis yang sama.

Contoh : atribut hobi pada mahasiswa yang punya banyak hobi, ada yang cuma satu hobi dan ada juga yang tidak mempunyai hobi sama sekali.

d. Atribut Composite (Komposit)

Atribut composite adalah atribut yang masih dapat diuraikan lagi menjadi sub-sub atribut yang masing-masing memiliki makna.

Contoh : pada tabel customer adalah atribut alamat, dimana dapat diuraikan lagi menjadi alamat, kota, kode\_pos dan no\_fax.

Kardinalitas atau derajat relasi merujuk kepada hubungan maksimum yang terjadi dari himpunan entitas yang satu ke himpunan entitas yang lain dan begitu juga sebaliknya.

a. One to One (1:1).

artinya setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B dan begitupun juga sebaliknya.

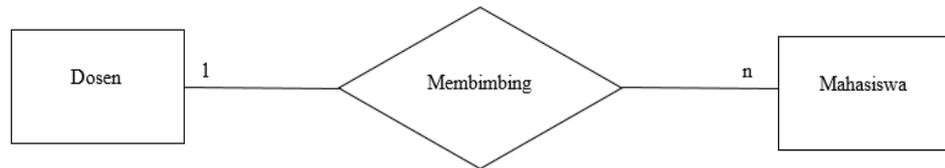


Gambar 2.2. 3 Contoh kardinalitas one to one

b. One to many (1:N / Many)

artinya setiap anggota entitas pada himpunan A dapat saling berhubungan dengan lebih dari satu anggota entitas pada himpunan

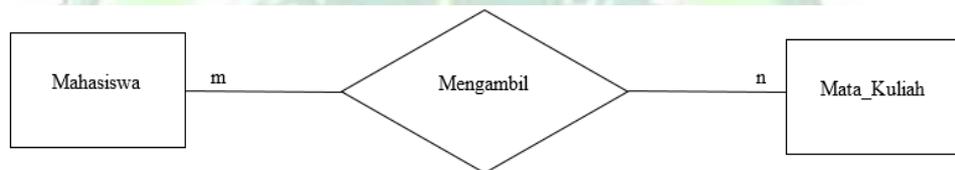
B tetapi tidak sebaliknya.



Gambar 2.2. 4 Contoh kardinalitas one to many

c. Many to Many (M:N)

artinya setiap entitas pada himpunan A dapat berhubungan dengan banyak entitas pada himpunan entitas B dan demikian juga sebaliknya.



Gambar 2.2. 5 Contoh kardinalitas many to many

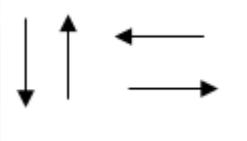
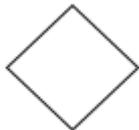
### 2.2.10 Flowchart

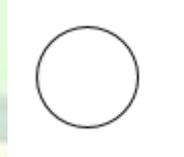
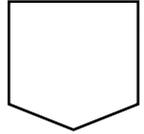
Flowchart adalah bagan atau gambar yang menunjukkan hubungan dan urutan sebuah proses sekaligus apa saja instruksinya. flowchart ini lambangkan dengan simbol. Setiap simbol melambangkan atau menggambarkan proses sesuai fungsinya masing-masing, sedangkan proses antar simbol dan hubungan ditunjukkan atau digambarkan dengan garis penghubung. Flowchart ini digunakan ketika seorang developer membuat sebuah program pada tahap awal. Flowchart sangat membantu dalam proses kegiatan pembuatan sebuah program karna alurnya menjadi menjadi lebih runutu dan jelas. Setelah flowchart selesai disusun, selanjutnya pemrogram (programmer) menerjemahkannya ke bentuk program dengan bahasa pemrograman (Abdillah dkk., 2019).

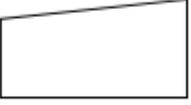
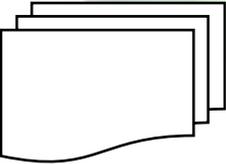
Flowchart adalah sajian grafik yang terdiri atas sekumpulan simbol, dimana setiap simbol menampilkan kegiatan tertentu. Simbol ini digunakan untuk

merepresentasikan atau bisa juga sebagai tool bantu dalam menggambarkan proses suatu program. Simbol-simbol yang digunakan adalah sebagai berikut :

Tabel 2.2. 2 Simbol Flowchart

No	Simbol	Nama	Keterangan
1		Terminal	Digunakan untuk memulai atau mengakhiri suatu proses pada program
2		<i>Flow Direction</i>	Digunakan untuk menunjukkan alir atau garis sebuah proses, simbol ini juga digunakan untuk menghubungkan satu simbol ke simbol lainnya.
3		<i>Processing</i> atau Proses	Simbol ini digunakan dalam kegiatan atau proses yang sedang dilakukan oleh komputer.
4		<i>Decision</i> atau keputusan	Digunakan untuk menentukan suatu keputusan berdasarkan kondisi yang ada di dalam program.

5		<i>Input-Output</i>	Menunjukkan proses masukan dan keluaran yang terjadi tetapi tidak terikat oleh jenis peralatannya.
6		<i>Predefined Proses</i> atau <i>Proses Terdefinisi</i>	Digunakan untuk menunjukkan bagian sub-prosedur atau prosedur pokok. Dengan kata lain, prosedur yang terinformasi di sini belum detail dan akan dirinci di tempat lain.
7		<i>Connector (On-page)</i>	Fungsinya ketika ada sebuah simbol yang saling berjauhan atau tata letaknya rumit jika dihubungkan, sehingga simbol ini digunakan untuk menyederhanakan.
8		<i>Connector (Off-page)</i>	Sama seperti connector on-page, tetapi bedanya simbol ini digunakan untuk menghubungkan simbol ketika halamannya berbeda. simbol ini bisa menggunakan angka ataupun huruf.

9		<i>Preparation</i> atau Persiapan	Digunakan untuk menyiapkan wadah simpan atau tempat penyimpanan di dalam sebuah media penyimpanan
10		<i>Manual Input</i>	Digunakan untuk memasukkan data secara manual dengan menggunakan keyboard
11		Manual <i>Operation</i> atau Kegiatan Manual	Digunakan untuk menunjukkan proses/kegiatan yang dilakukan secara manual atau tanpa menggunakan system
12		Document	Artinya dokumen ini berbentuk kertas atau output yang di cetak di kertas
13		Multiple Documents	Hampir sama dengan simbol document, bedanya hanya dalam jumlahnya lebih dari satu.
14		Display	Digunakan untuk menunjukkan penggunaan peralatan output seperti printer, monitor, dll

Secara garis besar dalam pembuatan flowchart, setiap pengolahan selalu terdiri dari tiga bagian utama, yaitu;

- a. Input berupa bahan mentah
- b. Proses pengolahan
- c. Output berupa bahan jadi.

Dalam proses pengolahan data dengan komputer, dapat dirangkum urutan dasar problem solving (pemecahan suatu masalah) diantaranya yaitu :

- a. Start: berisi instruksi-instruksi untuk mempersiapkan peralatan-peralatan yang dibutuhkan sebelum menangani permasalahan untuk dipecahkan.
- b. Read: berisi instruksi-instruksi untuk dapat membaca data inputan yang di hasilkan oleh peralatan input.
- c. Process: berisi kegiatan untuk menangani persoalan/pemecahan masalah sesuai dengan data yang telah dibaca oleh sistem.
- d. Write: berisi instruksi-instruksi untuk merekam lalu menerjemahkan hasil kegiatan ke dalam peralatan output.
- e. End: mengakhiri kegiatan pengolahan.

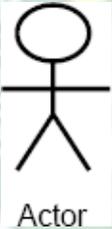
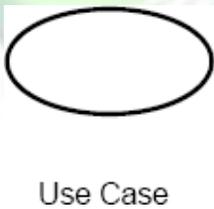
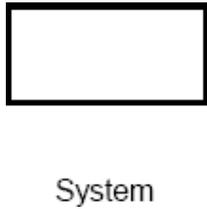
### **2.2.11 UML (Unified Modeling Language)**

UML (Unified Modeling Language) merupakan nodel perancangan yang dibuat sebagai pengganti dari metode analisis dan design berorientasi object (OOAD&D/object oriented analysis and design) yang perkenalkan pertama kali sekitar tahun 80-an sampai tahun 90-an. UML merupakan gabungan dari metode Booch, Rumbaugh (OMT) dan Jacobson. Tetapi UML ruang lingkupnya lebih luas daripada OOAD. Pada pertengahan saat pengembangan UML, dilakukan standarisasi proses dengan OMG(Object Management Group) dengan harapan UML akan menjadi bahasa standar pemodelan internasional pada masa depan (yang sekarang mulai banyak dipakai oleh sebagian besar developer dan kalangan lainnya) (Abdillah dkk., 2019).

UML (Unified Modelling Language) Merupakan salah satu bahasa yang banyak di gunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML juga menyediakan diagram-diagram dalam bentuk perangkat lunak diantaranya sebagai berikut:

- a. Use Case atau Diagram merupakan pemodelan untuk kegiatan didalam sistem informasi yang ingin dibuat. Use Case mendeskripsikan sebuah interaksi antar aktor dengan sistem informasi yang ingin dibuat. Secara kasar, Use Case digunakan untuk mengetahui fungsi apa saja yang berada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu (Hendini, 2016).

Tabel 2.2. 3 Simbol Flowchart

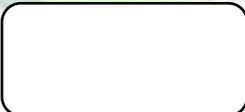
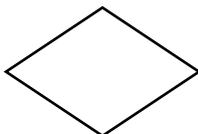
No	Simbol	Nama	Keterangan
1		<i>Actor</i>	Mewakili peran orang atau sistem yang lain untuk berkomunikasi dan berinteraksi dengan system.
2		<i>Use Case</i>	Proses penggambaran abstraksi suatu urutan interaksi antara satu atau lebih Aktor dengan usecase.
3		<i>System</i>	Bagian dari Sekumpulan Objek yang berelasi dan berinteraksi serta berhubungan antar objek. biasa juga disebut sebagai

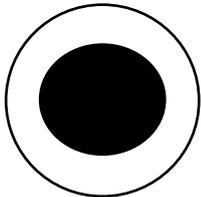
			satu kesatuan yang dirancang untuk mencapai tujuan tertentu
4	 Association	<i>Association</i>	Relasi antar benda yang Terhubung diantara obyek. Terjadi hubungan khusus untuk menggambarkan obyek struktural secara menyeluruh atau sebagian
5	 Generalization	<i>Generalization</i>	Menggambarkan hubungan khusus atau spesialis antara dua buah usecase dimana salah satu fungsi usecase sebagai fungsi umum dari lainnya.
6	 Dependency	<i>Dependency</i>	Hubungan antara dua buah benda dimana jika satu benda berubah maka akan mengakibatkan benda satunya juga akan berubah.
7	 Aggregation	<i>Aggregation</i>	Simbol ini maksudnya adalah satu elemen mempunyai elemen lainnya

8		<i>Realization</i>	<i>Realization</i> merupakan hubungan antara pengelompokan yang berelasi, hubungan ini dapat dibuat diantara <i>interface</i> dan <i>elements</i> , atau antara <i>use cases</i> dan <i>collaborations</i> .
---	---	--------------------	--

- b. Diagram Aktifitas atau Activity Diagram menggambarkan alur atau aliran kerja yang di sebut workflow atau aktifitas dari sebuah proses bisnis atau sistem menu yang ada pada perangkat lunak (Hendini, 2016).

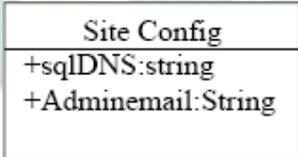
Tabel 2.2. 4 Simbol activity diagram

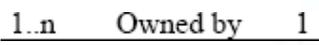
No	Simbol	Nama	Keterangan
1		<i>Status awal</i>	Digunakan untuk mengawali sebuah perancangan sistem.
2		Aktivitas	Aktivitas yang dilakukan system aktivitas biasanya untuk menggambarkan eksekusi dari suatu aksi.
3		<i>Decision</i>	Asosiasi atau percabangan digunakan ketika ada pilihan aktivitas lebih dari satu.

4		Status Akhir	Digunakan untuk mengakhiri sebuah perancangan sistem.
5		<i>Join</i>	Digunakan untuk menggabungkan beberapa aktivitas menjadi satu aktivitas.

- c. Diagram Kelas atau Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode operasi (Hendini, 2016).

Tabel 2.2. 5 Simbol class diagram

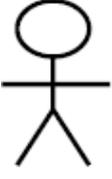
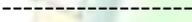
No	Simbol	Nama	Keterangan
1		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama, bentuknya kotak dan dibagi menjadi 3 bagian, yaitu nama kelas dibagian atas, nama properti/atribut dibagian tengah, lalu di bagian akhir adalah method dari sebuah class.  Bagian atas adalah bagian nama dari <i>class</i> .

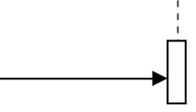
2		<i>Assosiation</i>	Sebuah asosiasi merupakan relasi antar kelas dengan makna umum dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 <i>class</i> , Asosiasi biasanya juga disertai dengan multiplicity untuk menampilkan sebuah relationship.
3		<i>Dependency</i>	Relasi antar kelas dengan arti kebergantungan antar kelas, maksudnya sebuah <i>class</i> bergantung untuk menggunakan <i>class</i> yang lain. Hal ini disebut <i>dependency</i> yang digunakan untuk menunjukkan operasi pada suatu <i>class</i> .
4		<i>Aggregation</i>	Sebuah <i>Aggregation</i> dilambangkan sebagai sebuah panah bertitik-titik. relasi antar class dengan makna semua bagian/whole part, <i>Aggregation</i> Adalah keseluruhan

			bagian <i>relationship</i> yang biasanya disebut relasi. <i>aggregation</i> digambarkan sebagai sebuah garis dan sebuah jajaran genjang yang tidak solid.
5		<i>Generalization</i>	Sebuah relasi dengan makna generalisasi-spesialis (umum-khusus). <i>Generalization</i> dilambangkan dengan sebuah panah yang tidak solid mengarah ke kelas.

- d. Diagram Sekuen digunakan untuk menggambarkan kegiatan objek pada Use Case dengan menandai waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Oleh sebab itu untuk mengambar diagram sekuen harus diketahui terlebih dahulu objek-objek yang terlibat dalam sebuah Use Case beserta metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada Use Case (Hendini, 2016).

Tabel 2.2. 6 Simbol sequence diagram

No	Simbol	Nama	Keterangan
1	 <p>Actor</p>	<i>Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan sistem.
2		<i>Garis hidup / lifeline</i>	Menggambarkan sebuah kehidupan suatu objek.
3	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"> <p>nama objek : nama kelas</p> </div>	Objek	Menyatakan objek yang saling bertukar pesan.
4		Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan.
5	<p>&lt;&lt;create&gt;&gt;</p> 	Pesan tipe create	Menyatakan suatu objek membangun objek lainnya, arah panah mengarah pada objek yang dibangun.

6	<p>1 : nama_method ()</p> 	Pesan tipe call	Menyatakan suatu objek untuk memanggil operasi yang ada pada objek lain atau objeknya sendiri.
7	<p>1 : masukan</p> 	Pesan tipe send	Menyatakan bahwa suatu objek mengirimkan inputan data atau informasi ke objek lainnya, anak atau arah panah diarahkan kepada objek yang dikirim.
8	<p>1 : keluaran</p> 	Pesan tipe return	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.
9		Pesan tipe destroy	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy.

- e. Star UML merupakan perangkat lunak atau software dengan permodelan program yang mendukung UML (Unified Modeling Language). Berdasarkan pada UML version 3.0.2 dilengkapi 11 macam diagram yang berbeda, didukung dengan pendekatan MDA (Model Driver Architecture) dari dukungan konsep UML. StarUML dapat memaksimalkan produktifitas dan kualitas dari suatu software project.

### **2.2.12 Black Box**

Pengujian perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program untuk mengetahui apakah fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Metode BlackboxTesting merupakan salah satu metode yang mudah digunakan karena hanya memerlukan batas bawah dan batas atas dari data yang di harapkan,Estimasi banyaknya data uji dapat dihitung melalui banyaknya field data entri yang akan diuji, aturan entri yang harus dipenuhi serta kasus batas atas dan batas bawah yang memenuhi. Dan dengan metode ini dapat diketahui jika fungsionalitas masih dapat menerima masukan data yang tidak diharapkan maka menyebabkan data yang disimpan kurang valid (Nababan & Rahim, 2018).

### **2.2.13 Skala Likert Dan Akurasi**

Skala Likert adalah skala yang digunakan untuk mengukur persepsi, sikap atau pendapat seseorang atau kelompok mengenai sebuah peristiwa atau fenomena sosial, berdasarkan definisi operasional yang telah ditetapkan oleh peneliti. Skala ini merupakan suatu skala psikometrik yang biasa diaplikasikan dalam angket dan paling sering digunakan untuk riset yang berupa survei, termasuk dalam penelitian survei deskriptif.

Dalam mengevaluasi performance algoritma dari Machine Learning (ML) (khususnya supervised learning), kita menggunakan acuan Confusion Matrix. Confusion Matrix merepresentasikan prediksi dan kondisi sebenarnya(aktual) dari data yang dihasilkan oleh algoritma ML. Berdasarkan Confusion Matrix, kita bisa

menentukan Accuracy, Precision, Recall dan Specificity (Riandari dkk, 2017).

Akurasi mengukur ketepatan dan kemiripan hasil pada waktu yang sama dengan membandingkannya terhadap nilai absolut. Oleh karena itu, semakin mendekati ukurannya, semakin tinggi level akurasi. Hal itu tergantung secara utama pada caranya; data dikumpulkan, Akurasi mengarah pada ketepatan dan kemiripan hasil pada waktu yang sama dengan membandingkannya terhadap nilai absolut. Akurasi menjangkau pengukuran aktual mendekati ukuran standar, yaitu tepat sasaran. Maka, semakin mendekati ukurannya, semakin tinggi level akurasi. Akurasi dapat dinyatakan sebagai persen perolehan kembali (% recovery) dari nilai yang diukur terhadap nilai sebenarnya atau nilai target. Jika proses pengukuran menghasilkan hasil nilai rata-rata mempunyai selisih dari nilai sebenarnya atau nilai target, proses tersebut dikatakan bias (Riandari dkk, 2017).

