# **BAB ll**

# **LANDASAN TEORI**

## **2.1 Kajian Penelitian Sejenis**

### 1. **Kajian Penelitian Ghulan Mahmudi Al Aziz, Imam Cholissodin, M. Tanzil Furqon Universitas Brawijaya**

Penelitian yang di lakukan oleh Ghulan Mahmudi Al Aziz, Imam Cholissodin, M. Tanzil Furqon Universitas Brawijaya mengambil obyek penelitian di Ponorogo. Dengan judul Sistem pendukung keputusan untuk rekomendasi wirausaha menggunakan metode AHP-TOPSIS. Sistem tersebut bertujuan untuk membantu para calon wirausaha dalam memilih wirausaha dengan tepat dan dapat memilih wirausahanya sesuai dengan minatnya atau bakatnya, maka akan ada semangat, usaha dan kerja keras yang lebih untuk mencapai keberhasilannya, Perbedaan penelitian yang sekarang akan di buat adalah meneliti tentang lokasi usaha, Sedangkan penelitian ini meneliti tentang usaha yang akan di lakukan

### **Kajian Penelitian Erna Ningsih, Dedih, Supriyadi STMIK Kharisma Karawang**

Penelitian yang dilaksanakan oleh Erna Ningsih, Dedih, Supriyadi dengan judul Sistem pendukung keputusan menentukan peluang usaha makanan yang tepat menggunakan metode Weighted Product (WP). Sistem tersebut bertujuan untuk membuat sistem penunjang keputusan yang dapat membantu pengguna untuk menentukan peluang usaha makanan yang tepat. Perbedaan penelitian yang sekarang akan di buat adalah meneliti tentang lokasi usaha, Sedangkan penelitian ini meneliti tentang usaha yang akan di lakukan

### **3. Kajian Penelitian Al Azis, G. M., Cholissodin, I., & Furqon, M. T. (2017). Sistem Pendukung Keputusan untuk Rekomendasi Wirausaha Menggunakan Metode AHP Dan TOPSIS (Studi Kasus Kab. Probolinggo).**

Sistem pendukung keputusan dapat digunakan untuk merekomendasikan suatu wirausaha bagi pengangguran ataupun semua orang. Metode AHP dan metode TOPSIS merupakan salah satu metode sistem pendukung keputusan yang dapat dikombinasikan dengan menghitung bobot kriteria menggunakan metode AHP,kemudian dilanjutkan dengan menghitung nilai preferensi untuk perangkingan dari alternatif wirausaha menggunakan metode TOPSIS. Metode AHP-TOPSIS akan merekomendasikan hasil berupa 3 wirausaha dengan nilai preferensi tertinggi. Sesuai dengan hasil pengujiannya,tujuan dari aplikasi ini dapat membantu untuk merekomendasikan wirausaha dalam membantu mendukung keputusan useruntuk memilih suatu wirausaha yang tepat

### **4. Kajian Penelitian Z., & Jamaaluddin, J. SISTEM PENUNJANG KEPUTUSAN PENENTUAN LOKASI PEMBANGUNAN MINIMARKET DENGAN METODE AHP**

Pembahasan utama dalam penelitian ini adalah perancangan pembuatan sistem penunjang keputusan penentuan lokasi strategis pembangunan Minimarket.Perencanaan ini menggunakan metode Analitycal Hierarchy Process (AHP),Masalah dalam pembangunan Minimarket sering di anggap rendah sehingga tidak sedikit minimarket yang tidak dapat mencapai target penjualan. Tujuan dari penelitian ini adalah menghasilkan sebuah Sistem Pendukung Keputusan (SPK) untuk mempermudah Investor dalam mengambil keputusan menentukan lokasi strategis.

## **2.2 Sistem Penunjang Keputusan**

### **2.2.1 Pengertian Sistem**

Sistem pada dasarnya adalah sekelompok unsur yang erat berhubungan satu dengan yang lainnya, yang berfungsi bersama-sama untuk mencapai tujuan. Adanya sebuah sistem didalam suatu perusahaan merupakan wujud dari rangkaian kegiatan yang dilakukan perusahaan dalam menjalankan operasionalnya, dengan adanya sistem ini diharapkan penyelenggaraan operasional perusahaan atau institusi dapat terjalin rapi dan terkoordinasi dengan baik, sehingga dapat mencapai hasil sesuai dengan yang menjadi harapan.

Rudy Tantra dalam bukunya Manajemen Proyek Sistem Informasi menerangkan bahwa “ Sistem *merupakan* satu kesatuan yang memiliki dua atau lebih komponen yang saling terhubung antara satu dengan lainnya untuk mencapai satu tujuan.” (Tantra, 2012).

### **2.2.2 Pengertian Sistem Penunjang Keputusan**

Pada dasarnya sistem pendukung keputusan merupakan pengembangan lebih lanjut dari sistem informasi manajemen terkomputerisasi yang dirancang sedemikian rupa sehingga bersifat interaktif dengan pemakainya. Sifat interaktif dimaksudkan untuk memudahkan integrasi antara berbagai Konsep Sistem Pendukung Keputusan (SPK)/Decision

Support Sistem (DSS) pertama kali diungkapkan pada awal tahun 1970-an oleh Michael S. Scott Morton dengan istilah Management Decision Sistem. Sistem tersebut adalah suatu sistem yang berbasis komputer yang ditujukan untuk membantu pengambil keputusan dengan memanfaatkan data dan model tertentu untuk memecahkan berbagai persoalan yang tidak terstruktur

## **2.4 Analitycal Hierarchy Process (AHP)**

### **2.4.1** ***Analitycal Hierarchy Process* (AHP)**

Menurut Taylor (2014), AHP adalah sebuah metode untuk memeringkat alternatif keputusan dan memilih yang terbaik dengan beberapa kriteria. AHP mengembangkan satu nilai numerik untuk memeringkat setiap alternatif keputusan, berdasarkan pada sejauh mana tiap-tiap alternatif memenuhi kriteria pengambil keputusan.

### **2.4.2 Karakteristik *Analitycal Hierarchy Process* (AHP)**

Analytical Hierarchy Process (AHP) adalah salah satu bentuk metode pengambilan keputusan yang pada dasarnya berusaha menutupi semua kekurangan dari metode sebelumnya. Peralatan utama dari metode AHP adalah sebuah hirarki fungsional dengan input utamanya adalah persepsi manusia. Dengan hirarki, suatu yang komplek tersebut diatur menjadi suatu bentuk hirarki(Permadi, 1992:5).

Perbedaan mencolok antara metode AHP dengan metode pengambilan keputusan lainnya terletak pada jenis inputnya. Metode yang sudah ada umumnya memakai input yang kuantitatif. Otomatis metode tersebut hanya dapat mengolah hal kuantitatif pula. Metode AHP menggunakan persepsi manusia yang dianggap ’expert’ sebagai input utamanya, kriteria ’expert’ disini bukan berarti bahwa orang tersebut harus jenius, pintar, bergelar doktor dan sebagainya, tapi lebih mengacu pada orang yang lebih mengerti benar permasalahan yang diajukan, merasakan akibat dari suatu masalah atau punya kepentingan terhadap masalah tersebut. Karena menggunakan input yang 11 kualitatif (persepsi manusia) maka AHP dapat mengolah hal kuantitatif disamping hal yang kualitatif.

### **2.4.3 Alasan Menggunakan *Analitycal Hierarchy Process* (AHP)**

Alasan menggunakan Metode AHP karena Metode AHP mempunyai beberapa kelebihandibanding metode lain, diantaranya sebagai berikut :

* + - * 1. Mampu memberikan dukungan pengambilan keputusan pada permasalahan yang multiobjective dan multicriteria.
        2. Memberikan dukungan pengambilan keputusan secara menyeluruh dengan memperhitungkan data kualitatif dan kuantitatif.
        3. Bersifat fleksibel yaitu menangkap beberapa tujuan dan kriteria sekaligus dalam sebuah model/hirarki.
        4. Inputan utamanya berupa data kuantitatif yaitu persepsi manusia dianggap sebagai expert.
        5. Memperhitungkan validitas sampai dengan batas toleransi inkonsistensi berbagai kriteria dan alternatif yang dipilih oleh para pengambil keputusan.
        6. Memperhitungkan daya tahan atau ketahanan output analisis sensitivitas pengambilan keputusan.

### **Langkah – Langkah Penggunaan AHP**

Mendefinisikan masalah dan menentukan solusi yang diinginkan.

Pada tahap ini kita harus menentukan sebuah permasalahan yang nantinya akan dipecah dengan detail dan juga mudah di pahami. Setelah itu menentukan sebuah solusi yang sesuai keinginan dari masalah tersebut, solusi dari permsalahan mungkin berjumlah lebih dari satu dan akan di kembangkan pada tahap selanjutnya.

* + - 1. Membuat struktur hirarki

Dengan tujuan utama membuat sebuah kriteria yang cocok untuk membuat sebuah penilaian terhadap alternatif yang telah di siapkan. Setiap kriteria memiliki sebuah intensitas yang berbeda antara yang satu dengan yang lainnya.

* + - 1. Membuat matrik perbandingan

Membuat matrik perbandingan berpasangan pendekatan dengan membandingkan sebuah kriteria perbandingan berpasangan. Dan untuk proses perhitungannya diambil mulai dari kriteria level paling atas kemudian level di bawahnya dan di bandingkan. Contoh S1-S2,S1-S3,S4-S4, S1=S5 dan kemudian lanjut dengan level di bawahnya S2-S3, S3-S4 dan seterusnya.

* + - 1. Mendefinisikan perbandingan berpasangan

Hasil dari perbandingan yang ada akan berupa sebuah angka atau nilai dari 1 sampai 9 yang menunjukan sebuah tingkat kepentingan masing masing alemen. Apabila suatu elemen dalam matriks dibandingkan dengan elemen itu sendiri maka hasil perbandingannya diberi nilai 1.Skala 9 telah terbukti bisa diterima dan dapat membedakan intensitas antar elemen.

* + - 1. Mengulangi langkah seluruh tingkat hirarki

Pada tahap ini harus melakukan pengulangan pada tahap 3 dan 4 untuk mendapatkan sebuah data yang akurat.

* + - 1. Menghitung vektor eigen

Menghitung vektor eigen dari masing masing matriks perbandingan berpasangan sedangkan pada tahap ini, dengan menggunakan sebuah vektor eigen adalah bobot yang digunakan setiap elemen untuk menentukan sebuah prioritas pada tingkatan hirarki terendah hingga mencapai tujuannya yaitu pada tingkat paling atas (satu elemen).

* + - 1. Memeriksa konsistensi hirarki

Dengan memeriksa konsitensi yang dapat di ukur adalah rasio konsistensi dengan melihat sebuah index konsistensi.Sedangkan konsistensi yang di harapkan harus mendekati nilai sempurna agar memiliki keputusan yang baik.

### **Perancangan Sistem**

Urut proses pada sistem meliputi :

Menentukan tujuan, kriteria, dan alternatif.

Menyusun kriteria-kriteria ke dalam bentuk matriks berpasangan.

Menjumlahkan matriks kolom.

Menghitung nilai elemen kolom kriteria dengan rumus masing-masing elemen kolom dibagi dengan jumlah matriks kolom

Menghitung nilai prioritas kriteria dengan rumus menjumlah matriks baris hasil langkah ke-4 dan hasilnya dibagi dengan jumlah kriteria.

Menguji konsistensi setiap matriks berpasangan. Dengan langkah sebagai berikut:

- Menghitung Indeks Konsistensi (CI) dengan rumus:

**CI =** λ maks – n

**n – 1**

**CI =** indeks konsisten

**λ** = nilai eigen terbesar dari matrik berordo

**n** = jumlah kriteria

Nilai eigen terbesar didapat dengan menjumlahkan hasil perkalian jumlah kolom dengan vektor prioritas.

Menghitung Rasio Konsistensi (CR) dengan rumus :

CR = Ci

Cr

Dimana nilai Random Index (RI) berdasarkan perhitungan Saaty, dapat dilihat dari tabel 2.1

**Tabel 2 2** Random Index

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| N | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| RI | 0.00 | 0.00 | 0.58 | 0.90 | 1.12 | 1.24 | 1.32 | 1.41 | 1.45 | 1.49 |

* + - 1. Menyusun matriks baris antara alternatif versus kriteria yang isinya hasil perhitungan proses langkah 2 sampai dengan langkah 5.
      2. Hasil akhirnya berapa proritas global l sebagai nilai yang diambil untuk mengambil keputusan berdasarkan nilai tertinggi.

## **2.5 TOPSIS (*Technique for Order Preference by Similarity to Ideal Solution)***

### **2.5.1 TOPSIS *(Technique for Order Preference by Similarity to Ideal Solution)***

TOPSIS *(Technique for Order Preference by Similarity to Ideal Solution)* adalah salah satu metode pengambilan keputusan multikriteria yang pertama kali diperkenalkan oleh Kwangsun Yoon and Hwang Ching-Lai pada tahun 1981. Menurut Hwang dan Zeleny dikutip (Kusumadewi, 2006) pada dasarnya, proses pengambilan keputusan adalah memilih suatu alternatif. TOPSIS didasarkan pada konsep dimana alternatif terpilih yang terbaik tidak hanya memiliki jarak terpendek dari solusi ideal positif (A + ), namun juga memiliki jarak terpanjang dari solusi ideal negatif (A -) (Hwang, 1981) (Zeleny, 1982). Hal ini disebabkan konsepnya sederhana dan mudah dipahami, komputasinya efisien, dan memiliki kemampuan untuk mengukur kinerja relatif dari alternatif-alternatif keputusan dalam bentuk matematis yang sederhana. Mekanisme pengambilan keputusan dengan konsep fundamental dari metode ini adalah untuk penentuan jarak Euclide terpendek dari solusi ideal positif dan jarak Euclide terjauh dari solusi ideal negatif. TOPSIS banyak digunakan dengan alasan, konsepnya sederhana dan mudah dipahami, komputasinya efisien dan memiliki kemampuan untuk mengukur kinerja relatif II-14 dari alternatif-alternatif keputusan dalam bentuk matematis yang sederhana. Metode ini banyak digunakan untuk menyelesaikan pengambilan keputusan secara praktis

### **2.5.2 Tujuan TOPSIS**

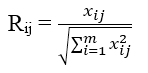
TOPSIS bertujuan untuk menentukan solusi ideal positif dan solusi ideal negatif. Solusi ideal positif memaksimalkan kriteria manfaat dan meminimalkan kriteria biaya, sedangkan solusi ideal negatif memaksimalkan kriteria biaya dan meminimalkan kriteria manfaat (Fan dan Cheng, 2009). Kriteria manfaat merupakan kriteria dimana ketika nilai kriteria tersebut semakin besar maka semakin layak pula untuk dipilih. Sedangkan kriteria biaya merupakan kebalikan dari kriteria manfaat, semakin kecil nilai dari kriteria tersebut maka akan semakin layak untuk dipilih. Dalam metode TOPSIS, alternatif yang optimal adalah yang paling dekat dengan solusi ideal positif dan paling jauh dari solusi ideal negatif.

### **2.5.3 Prinsip TOPSIS**

TOPSIS menggunakan prinsip bahwa alternatif yang terpilih harus mempunyai jarak terdekat dari solusi ideal positif dan terjauh dari solusi ideal negatif dari sudut pandang geometris dengan menggunakan jarak Euclidean untuk menentukan kedekatan relatif dari suatu alternatif dengan solusi optimal. Solusi ideal positif didefinisikan sebagai jumlah dari seluruh nilai terbaik yang dapat dicapai untuk setiap atribut, sedangkan solusi ideal negatif terdiri dari seluruh nilai terburuk yang dicapai untuk setiap atribut.TOPSIS mempertimbangkan keduanya, jarak terhadap solusi ideal positif dan jarak terhadap solusi ideal negatif dengan mengambil kedekatan relatif terhadap solusi ideal positif. Berdasarkan perbandingan terhadap jarak relatifnya, susunan prioritas alternatif bisa dicapai.

### **2.5.4 Langkah-langkah Metode TOPSIS Secara umum**

**1. Membangun normalized decision matrix**  
Elemen Rij hasil dari normalisasi decision matrix R dengan metode Euclidean length of a vector adalah:



dengan i=1,2,3, …m; dan j=1,2,3 … n

**2.  Membangun weighted normalized decision matrix**Solusi ideal positif A+dan solusi ideal negatif A-dapat ditentukan berdasarkan rating bobot ternormalisasi (Yij) sebagai:

http://anindyadev.com/source/Tutorial%20Lain-Lain/TOPSIS/T2.png

dengan i=1,2,3, … m dan j=1,2,3, … n

**3.  Menentukan matriks solusi ideal dan matriks solusi ideal negatif**Solusi ideal positif (A+) dihitung berdasarkan:

http://anindyadev.com/source/Tutorial%20Lain-Lain/TOPSIS/T3.png

Solusi ideal negatif (A-) dihitung berdasarkan:

http://anindyadev.com/source/Tutorial%20Lain-Lain/TOPSIS/T4.png

**4.  Menentukan jarak antara nilai setiap alternatif dengan matriks solusi ideal positif dan matrik ideal negatif.**Jarak antara alternatif Aidengan solusi ideal positif dirumuskan sebagai:

http://anindyadev.com/source/Tutorial%20Lain-Lain/TOPSIS/T5.png

Jarak antara alternatif Aidengan solusi ideal negatif dirumuskan sebagai:

http://anindyadev.com/source/Tutorial%20Lain-Lain/TOPSIS/T6.png

**5.  Menentukan nilai preferensi untuk setiap alternatif**Kedekatan setiap alternatif terhadap solusi ideal dihitung berdasarkan rumus:

http://anindyadev.com/source/Tutorial%20Lain-Lain/TOPSIS/T7.png

Nilai preferensi untuk setiap alternatif merupakan hasil akhir dari perhitungan metode TOPSIS, semakin tinggi nilai nya maka alternatif tersebut merukan alternatif yang diinginkan.

## **2.6 Unified Modelling Language (UML)**

UML (Unified Modeling Language) adalah tujuan umum, perkembangan, bahasa pemodelan dibidang rekayasa perangkat lunak, yang dimaksudkan untuk menyediakan cara standar untuk memvisualisasikan desain sistem..

Menurut Nugroho (2009), UML (Unified Modeling Language) adalah Metodologi kolaborasi antara metoda-metoda Booch, OMT (Object Modeling Technique), serta OOSE (Object Oriented Software Enggineering) dan beberapa metoda lainnya, merupakan metodologi yang paling sering digunakan saat ini untuk analisa dan perancangan sistem dengan metodologi berorientasi objek mengadaptasi maraknya penggunaan bahasa “pemrograman berorientasi objek” (OOP).

### **2.6.1 Use Case Diagram**

Use Case Diargam merupakan pemodelan untuk melakukan (behavior) sistem informai yang akan dibuat. Use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Berikut adalah simbol-simbol yang ada pada diagram use case (Rosa dan Shalahuddin, 2014:156):

**Tabel 2.3** Simbol Use Case Diagram

|  |  |
| --- | --- |
|  | *ACTOR*  Orang proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari actor adalah gambar orang, biasanya dinyatakan menggunakan kata benda di awal frase nama *actor*. |
|  | *USE CASE*  Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesar antar unit atau actor biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama use case. |
|  | ASOSIASI/*ASSOCIATION*  Komunikasi antara *actor* dan use case yang berpartisipasi pada use case atau use case memiliki interraksi dengan *actor*. |
|  | EKSTENSI/*EXTEND*  Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan memiliki nama depan yang sama dengan use case yang di tambahkan. |
|  | GENERALISASI/*GENERALIZATION*  Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya. |
|  | MENGGUNAKAN/*INCLUDE*  Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsional atau sebagai syarat dijalankan use case ini. |

### **2.6.2 Activity Diagram**

**Sukamto dan Shalahuddin (2013), “Diagram aktivitas atau activity diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”.**

Activity Diagram menggambarkan aliran kerja atau aktivitas dari sebuah sistem yang ada pada perangkat lunak. Diagram ini memodelkan event yang terjadi dalam suatu use case. Dari pertama sampai akhir, diagram ini menunjukkan langkah apa saja yang terjadi dalam proses kerja sistem yang dibuat. Sehingga dapat memudahkan dalam proses pembuatan system.

Simbol-simbol activity diagram sebagai berikut :

**Tabel 2.4** Simbol Activity Diagram

|  |  |
| --- | --- |
| **Simbol** | **Keterangan** |
|  | **Start Point**  Menggambarkan awal dari aktivitas. |
|  | **End Point**  Menggambarkan akhir dari aktivitas. |
|  | **Activity**  Menggambarkan suatu proses atau kegiatan bisnis. |
|  | **Fork**  Menggambarkan aktivitas yang dimulai dengan sebuah aktivitas dan diikuti oleh dua atau lebih aktivitas yang harus dikerjakan. |
|  | **Decision**  Menggambarkan keputusan atau pilihan. |
|  | **State Transition**  Menggambarkan aliran perpindahan kontrol antara state. |
|  | **Swimlane**  Menggambarkan pemisahan atau pengelompokkan aktivitas berdasarkan actor. |

### **2.6.3.Sequence Diagram**

**Sukamto dan Shalahuddin (2013), “Diagram aktivitas atau activity diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”.**

Activity Diagram menggambarkan aliran kerja atau aktivitas dari sebuah sistem yang ada pada perangkat lunak. Diagram ini memodelkan event yang terjadi dalam suatu use case. Dari pertama sampai akhir, diagram ini menunjukkan langkah apa saja yang terjadi dalam proses kerja sistem yang dibuat. Sehingga dapat memudahkan dalam proses pembuatan system.

**Tabel 2.5** Simbol Sequence Diagram

|  |  |
| --- | --- |
| **Simbol** | **Keterangan** |
|  | **Actor**  Menggambarkan orang yang sedang berinteraksi dengan sistem. |
|  | **Boundary**  Menghubungkan antara userdengan sistem. |
|  | **Entity**  Menggambarkan informasi yang harus disimpan oleh sistem(struktur data dari sebuah sistem). |
|  | **Control**  Menggambarkan “perilaku mengatur”, mengkoordinasikan perilaku sistem dan dinamika dari suatu sistem, menangani tugas utama dan mengontrol alur kerja suatu sistem. |
|  | **Object Message**  Menggambarkan pengiriman pesan. |
|  | **Message to Self**  Sebuah objek yang mempunyai sebuah pesan kepada dirinya sendiri. |
|  | **Lifeline**  garis titik-titik yang terhubung dengan objek, sepanjang lifeline terdapat |
|  | **Activation**  mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi |

### **2.6.4 Class Diagram**

Class diagaram adalah struktur dan deskripsi class dan obyek beserta hubungan antara class. Sukamto dan Shalahuddin (2013), “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 atau operasi.

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.

2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

1. Kelas main Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

2. Kelas yang menangani tampilan sistem (view) Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

3. Kelas yang diambil dari pendefinisian use case (controller) Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian use case, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak.

4. Kelas yang diambil dari pendefinisian data (model) Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Berikut ini simbol-simbol dari class diagram yang sudah saya sediakan ditabel ini

**Tabel 2.6** Simbol Class Diagram

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Keterangan** |
| https://1.bp.blogspot.com/-iUPBEAw34Yc/XoKfvcJLtkI/AAAAAAAABD8/_Tm_7nsNBlYY_95Ens2ET9gc3e15Xz8eACLcBGAsYHQ/s1600/34.JPG | Kelas | Kelas pada struktur  sistem. |
| <https://1.bp.blogspot.com/-ssEe2Lm9BX4/Xn2DSSOZFiI/AAAAAAAAA6E/29go2I2rm4wxvEjCe3tQG0xQawNoetzyACPcBGAYYCw/s1600/8.JPG> | Interface | Sama   dengan    konsep interface  dalam pemrograman berorientasi objek. |
| <https://1.bp.blogspot.com/-tiFv0q8EpH8/XoAiqyNM_BI/AAAAAAAABBE/_p3_Pia32BQs7RobDPDlm1MEYnqzveLRgCPcBGAYYCw/s1600/31.JPG> | Association | Relasi antarclass  dengan arti umum,  asosiasi biasanya juga disertai dengan  Multiplicity. |
| https://1.bp.blogspot.com/-5r_Ex4vd44Y/Xn7d3fB3HhI/AAAAAAAAA-Q/xHDeQm03vB0lH5Ge1nxR08_pzS4ggDO3gCPcBGAYYCw/s1600/18.JPG | Directed Association | Relasi antarkelas  dengan makna kelas  yang atau  digunakan  oleh kelas  yang lain, asosiasi   biasanya juga disertai dengan  multiplicity. |
| https://1.bp.blogspot.com/-Hp7DJUrhqw0/Xn_tkcGCeYI/AAAAAAAAA_s/IScX-kQCxFQ40LodNWtt20J35_9xqd2bwCPcBGAYYCw/s1600/26.JPG | Generalisasi | Relasi antarkelas  dengan makna  generalisasi-spesialisasi (umum khusus). |
| <https://1.bp.blogspot.com/-WBnKMoenB6s/Xn7eOnouGwI/AAAAAAAAA-U/pxv9gmG_PgUWoGmQgGRt2FKaQONM080VACPcBGAYYCw/s1600/19.JPG> | Dependency | Relasi antarkelas  dengan makna kebergantungan  antarkelas’ |
| https://1.bp.blogspot.com/-eVC098piIYk/XoKhdMEQQAI/AAAAAAAABEI/6AO7_qyK4Sc6jD8NcUk4n4Yd3HhE5zSGwCLcBGAsYHQ/s200/35.JPG | Aggregation | Relasi antarkelas dengan  makna   semua-bagian  (whole-part) |

Class Diagram mampu memberikan  penjelasan implementasi-independen dari suatu jenis program yang digunakan, kemudian dilewatkan diantara berbagai komponennya.(Ansori,2019)

## **2.7 Entity Relationship Diagram (ERD)**

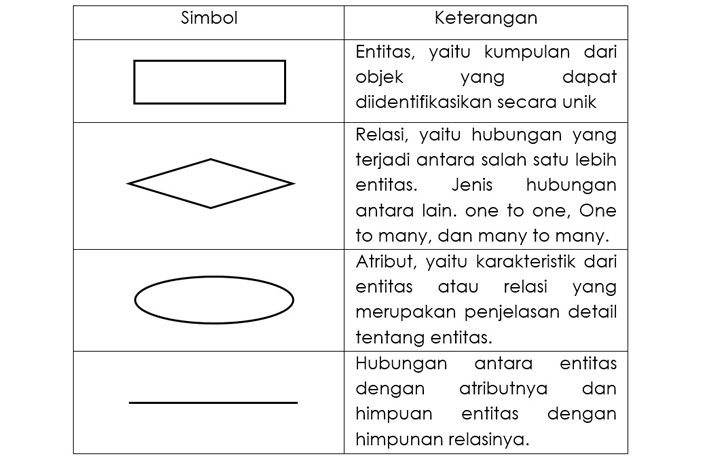
Menurut Sutanta (2011:91) mengatakan ERD mempunyai dua macam model data, keduanya merupakan cara mendeskripsikan perancangan *database* pada tingkatan logika. Dua model tersebut adalah :

1. Model ERD atau *Conceptual Data Model (*CDM) : model yang dibuat berdasarkan anggapan bahwa dunia nyata terdiri dari koleksi obyek-obyek dasar yang dinamakan entitas (*entity)* serta hubungan (*relationship*) antara entitas-entitas itu.

2. Model Relasional atau *Physical Data Model* (PDM) : model yang menggunakan sejumlah tabel untuk menggambarkan data serta hubungan antara data-data tersebut. Setiap tabel mempunyai sejumlah kolom dimana setiap kolom memiliki nama yang unik.

#### **2.7.1 ENTITAS (entity)**

Entitas menggambarkan sekumpulan obyek atau benda di dunia nyata yang setiap itemnya (*intance).* Entitas biasanya berupa orang, obyek, lokasi, kejadian, dll. (contoh : pegawai, mahasiswa, hobi, orang tua, anggota, suku, dll).



**Gambar2.1** ENTITAS (entity)

#### **2. 7.2 ATRIBUT (attribute)**

Atribut menggambarkan properti dari entitas. Setiap entitas memiliki sejumlah atribut yang relevan bagi sistem yang dimodelkannya (contoh: mahasiswa memiliki atribut nama, nim, alamat, dll yang berguna bagi sistem informasi perusahaan dia kuliah). Atribut diklasifikasikan sebagai *entity key* dan *entity descriptor. Entity key* digunakan untuk mengidentifikasi suatu *instance* dari entitas secara unik. Atribut yang memiliki nilai yang unik disebut *Candidate Key,* yang salah satunya akan dipilih untuk menjadi *primary key.* Sebagai konsekuensi logis hubungan antar entitas, *primary key* dari entitas dasar akan muncul lagi sebagai *foreign key* di entitas subordinat. *Foreign Key* adalah *primary key* dari sebuah entitas yang digugnakan oleh entitas lain untuk mengidentifikasikan sebuah *instance* dari *relationship.*

#### **2.7.3. RELASI (relation)**

Relasi atau hubungan adalah kejadian atau transaksi yang terjadi di antara dua entitas yang keterangannya perlu disimpan dalam database.

Aturan penggambaran relasi antar entitas adalah:

1. Relasi dinyatakan dengan simbol belah ketupat.

2. Nama relasi dituliskan di dalam simbol belah ketupat.

3. Relasi menghubungkan dua entitas.

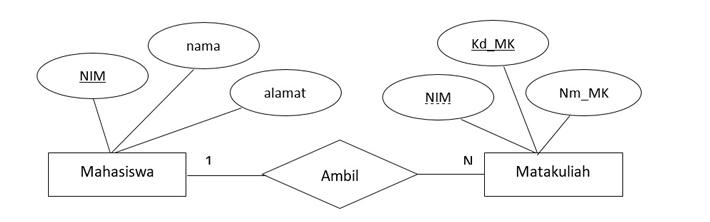
4. Nama relasi menggunakan kata kerja aktif (diawali awalan me) tunggal.

5. Nama relasi sebisa mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.

**Relasi ada 3 variasi yaitu :**

**Relasi Binery**

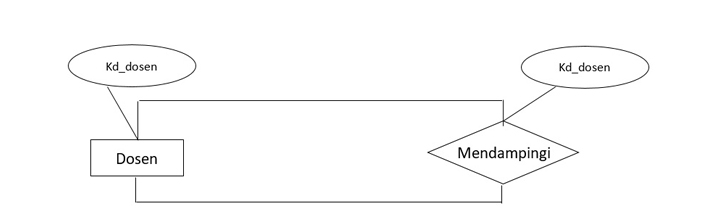
Relasi binary merupakan relasi yang terjadi antara 2 himpunan entitas yang berbeda. Relasi ini merupakan relasi yang umum digunakan. Relasi antara mahasiswa mengambil matakuliah yang menunjukkan binery relation.



**Gambar2.2 Relasi Binery**

**Relasi Unary**

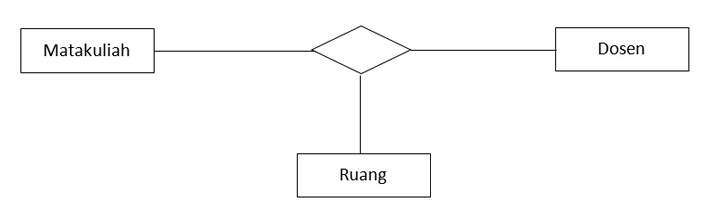
Relasi Unary merupakan variasi relasi yang terjadi dari sebuah himpunan entitas ke himpunan entitas yang sama, dan unary sering disebut dengan relasi tunggal. Relasi antara dosen dan mendampingi yang menunjukkan unary relation.



**Gambar2.3** Relasi Unary

**3 . Relasi N-ary**

Relasi N-ary merupakan relasi dari 3 entitas atau lebih. Relasi ini untuk menghubungkan dari tiga entitas yang dimasukan ke relasi multi entitas.N-ary relation menunjukkan secara lebih jelas bahwa bahwa beberapa entitas berpartisipasi dalam sebuah relasi tunggal. Bentuk relasi semacam ini sebisa mungkin dihindari karena akan mengaburkan derajat relasi yang ada dan akan menyebabkan perencanaan database semakin kompleks.



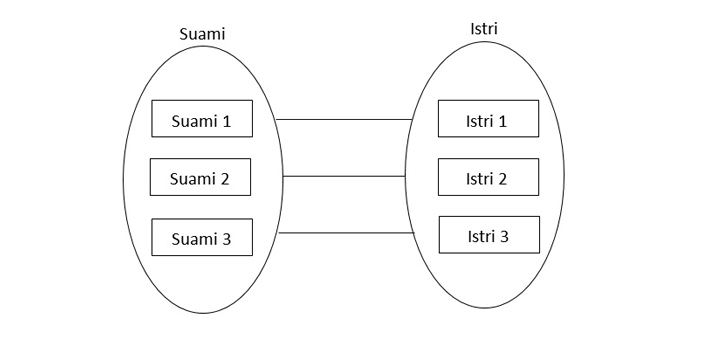
**Gambar 2.4** Relasi N-ary

**KARDINALITAS (Derajat Relasi)**

Model relasi ini berdasarkan persepsi dunia nyata diantaranya himpunan objek dasar dan relasi antara entitas.Entitas dapat diartikan sebagai objek dan diidentifikasikan secara unik, dan objeknya dapat berbentuk orang, barang, dan sebagainya.Kardinalitas relasi menunjukkan maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas lain.Kadinalitas relasi yang terjadi di antara dua himpunan entitas dapat berupa satu ke satu (one to one), satu ke banyak (one to many), dan banyak ke banyak (many to many).

**Satu ke satu (one to one)**

Hubungan satu ke satu (one to one) berarti setiap himpunan entitas hanya boleh berhubungan dengan satu himpunan entitas lainnya. Sebagai contoh himpunan suami dan istri berikut:

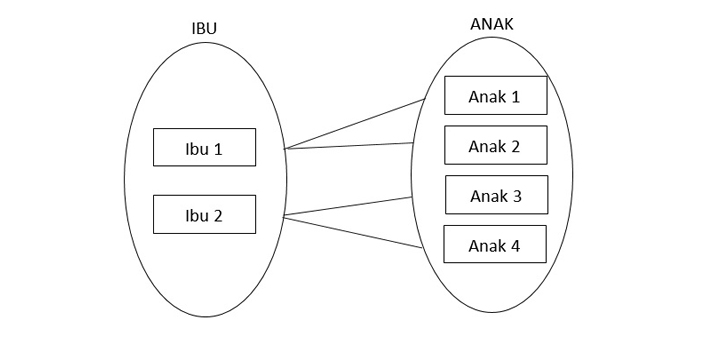


**Gambar 2.5** Relasi Satu ke satu (one to one)

Pada gambar diatas dapat dilihat bahwa satu himpunan entitas suami hanya berhubungan tepat dengan satu himpunan entitas istri. Dalam arti kata suami atau istri tidak boleh selingkuh.

**Satu ke Banyak (one to many)**

Hubungan satu ke banyak (one to many) berarti  satu dari setiap himpunan entitas boleh berhubungan dengan banyak himpunan entitas lainnya.Sebagai contoh himpunan ibu dan anak berikut:

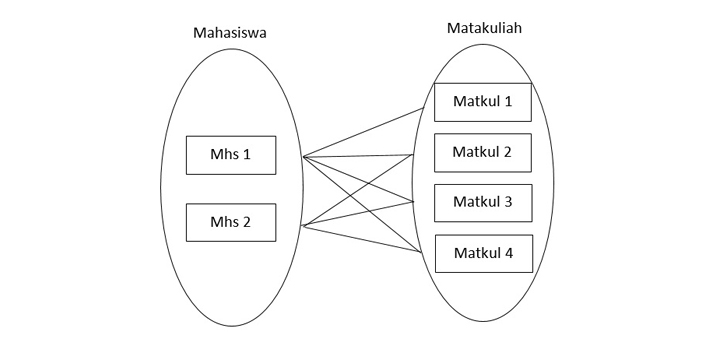


**Gambar2.6** Relasi Satu ke Banyak (one to many)

Pada gambar diatas dapat dilihat bahwa satu himpunan ibu memiliki banyak hubungan ke himpunan entitas anak.Dalam arti kata satu ibu bisa memiliki banyak anak dan satu anak hanya dimiliki oleh satu ibu.

* + - 1. **Banyak ke Banyak (**Many to Many**)**

Hubungan banyak ke banyak (many to many) berarti setiap himpunan entitas boleh berhubungan dengan banyak himpunan entitas lainnya dan sebaikanya. Sebagai contoh himpunan matakuliah dan mahasiswa berikut:



**Gambar2.7** Relasi Banyak ke Banyak (Many to Many)

Pada gambar diatas dapat dilihat bahwa satu himpunan mahasiswa memiliki banyak hubungan ke himpunan entitas matakuliah dan satu dari himpunan matakuliah memiliki banyak hubungan ke himpunan entitas mahasiswa.Dalam arti kata satu mahasiswa bisa memiliki banyak matakuliah dan satu matakuliah bisa dimiliki oleh banyak mahasiswa.

## **2.8 Data Base**

Menurut (Hesananda et al., 2017), Database ialah suatu wadah untuk menampung sebuah data yang ada pada sebuah sistem. Database juga bias diartikan sebagai kumpulan data. Database juga biasa dikenal formal dan tegas. Database juga bias diartikan dengan kumpulan data yang terintegrasi yang dapat dimanipulasi, diambil dan dicari secara cepat.

## **2.9 DBMS (Database Management System)**

DBMS (Database Management System)adalah suatu sistem atau software yang dirancang khusus untuk mengelola suatu database dan menjalankan operasi terhadap data yang diminta oleh banyak pengguna. merupakan perangkat lunak yang dipakai untuk membaangun data yang berbasis komputerisasi.

Ada 2 macam interuksi pada DBMS yaitu :

1. ***DDL (Data Definition Language)***

Yang pertama adalah bahasa DDL atau kepanjangannya *Data Definition Language*, yaitu dipakai untuk menggambarkan desain dari basis data secara menyeluruh. DDL (*Data Definition Language*) dapat dipakai untuk membuat tabel baru, memuat indeks, maupun mengubah tabel. Hasil dari kompilasi DDL akan disimpan di kamus data. Itulah definisi dari DDL.

1. D***ML(Data Manipulation Language)***Dan yang kedua adalah DML atau kepanjangannya D*ata Manipulation Language*, yaitu dipakai untuk memanipulasi dan pengambilan data pada suatu basis data

## **2.10 Framework**

Menurut Betha Sidik (2012) Framework adalah kumpulan intruksiintruksi yang dikumpulkan dalam class dan function-function dengan fungsi masing-masing untuk memudahkan developer dalam memanggilnya tanpa harus menuliskan syntax program yang sama berulang-ulang serta dapat menghemat waktu

## **2.11** **PHP**(***Hypertext Preprocessor***)

PHP Menurut (Das & Saikia, 2016) PHP merupakan sebuah bahasa pemrograman **server side scripting** yang bersifat **open source.** Bahasa pemrograman ini banyak digunakan untuk pengembangan website. PHP menjalankan instruksi pemrograman saat proses runtime. Hasil dari instruksi tentu akan berbeda tergantung data yang diproses. PHP merupakan bahasa pemrograman server-side, maka script dari PHP nantinya akan diproses di server.Selain itu, PHP juga merupakan bahasa pemrograman yang bersifat open source. Pengguna bebas memodifikasi dan mengembangkan sesuai dengan kebutuhan mereka.

## **2.12 Codeigniter**

Menurut Betha Sidik (2012) CodeIgniter adalah Sebuah framework php yang bersifat open source dan menggunakan metode MVC (Model, View, Controller) untuk memudahkan developer atau programmer dalam membangun sebuah aplikasi berbasis web tanpa harus membuatnya dari awal. Framework Codeigniter dikembangkan oleh Rick Ellis, CEO Ellislab, Inc. kelebihan dari framework codeigniter jika dibandingkan dengan frameworklain adalah sebagai berikut :

* 1. Gratis (Open-Source) Kerangka kerja Codeigniter memiliki lisensi dibawah Apache/BSDopen-source sehingga bersifat bebas atau gratis.
  2. Berukuran kecil Ukuran yang kecil merupakan keunggulan tersendiri jika dibandingkan framework lain yang berukuran besar dan29 membutuhkan resource yang besar dan juga dalam eksekusi maupunpenyimpanannya.
  3. Model Kode merupakan program (berupa OOP class) yang digunakan untuk berhubungan dengan database MySQL sekaligus untuk memanipulasinya (input-edit-delete).
  4. Model Kode merupakan program (berupa OOP class) yang digunakan untuk berhubungan dengan database MySQL sekaligus untuk memanipulasinya (input-edit-delete).
  5. View Merupakan kode program berupa template atau PHP untuk menampilkan data pada browser.
  6. Controller merupakan Kode program (berupa OOP class ) yangdigunakan untuk mengontrol aliran atau dengan kata lain sebagai pengontrol model dan view

## **2.13 HTML (HyperText Markup Language)**

Menurut (Endra & Aprilita, 2018), HTML yaitu bahasa pemrograman standar yang digunakan untuk membuat sebuah halaman web, yang kemudian dapat diakses untuk menampilkan berbagai informasi di dalam sebuah penjelajah web Internet (Browser). 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.Supaya dapat menghasilkan tampilan wujud yang terintegerasi Pemformatan hiperteks sederhana ditulis dalam berkas format ASCII sehingga menjadi halaman web dengan perintah-perintah HTML.

## **2.14 CSS**

Menurut (Wahyudi, 2017), CSS adalah suatu Bahasa pemrograman web yang digunakan untuk mengendalikan dan membangun berbagai komponen dalam web sehingga tampilan web akan lebih rapi, terstruktur, dan seragam