



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## BAB II

### LANDASAN TEORI

#### 2.1. Rumah

Rumah merupakan kebutuhan primer setiap orang. Biasanya, untuk dapat memenuhi kebutuhan rumah atau tempat tinggal, seseorang akan membeli kepada pihak *developer* atau pengembang perumahan. Banyak pengembang perumahan yang memberikan alternatif mengenai perumahan (Kurniawan & Kurniawan,, 2017). Rumah juga merupakan bangunan yang berfungsi sebagai tempat tinggal yang digunakan untuk dihuni, sarana keluarga, serta aset bagi pemiliknya. Rumah juga adalah tempat untuk beristirahat setelah melakukan berbagai aktifitas dan rumah harus dapat menampung kegiatan penghuninya. Sehingga, kebutuhan ruang setiap penghuni dapat terpenuhi dengan baik. Pada tahun 1946 di Inggris ada sebuah *Sub Committee on Standards of Fitness for Habitation* yang membuat rekomendasi terhadap rumah yang akan dihuni antara lain sebagai berikut (Perangin-Angin, 2016):

1. Dalam segala hal harus kering,
2. Dalam keadaan rumah diperbaiki.
3. Tiap kamar mempunyai lampu dan lubang ventilasi.
4. Mempunyai persediaan air yang cukup untuk segala keperluan rumah tangga.
5. Mempunyai kamar mandi.

6. Mempunyai tempat/kamar cuci, dengan pembuangan air limbah yang baik.
7. Mempunyai sistem drainase yang baik.
8. Mempunyai jamban yang memenuhi syarat kesehatan (di dalam atau di luar).
9. Cukup fasilitas untuk menyimpan, meracik, dan memasak makanan.
10. Tempat menyimpan makanan harus mempunyai ventilasi yang baik.
11. Jalan masuk ke rumah yang baik.
12. Mempunyai fasilitas alat pemanas/pendingin dikamar.

## 2.2. Sistem Pendukung Keputusan

Konsep sistem pendukung keputusan pertama kali diperkenalkan pada tahun 1970-an oleh Michael S.Scott Marton dengan istilah *management decision system*. Konsep sistem pendukung keputusan ditandai dengan sistem interaktif berbasis komputer yang membantu mengambil keputusan memanfaatkan data dan model untuk menyelesaikan masalah yang tidak terstruktur (Kusumadewi, 2010).

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 (Chamid, 2016).

Menurut (Kusumadewi, 2010) sistem pendukung keputusan adalah suatu sistem informasi spesifik yang ditujukan untuk membantu manajemen dalam mengambil keputusan yang berkaitan dengan persoalan yang bersifat:

1. Terstruktur, yaitu berhubungan dengan persoalan yang telah diketahui sebelumnya dengan penyelesaian standar aturan yang telah ditentukan.
2. Semi terstruktur, yaitu berhubungan dengan persoalan yang belum diketahui sebelumnya dengan parameter yang sudah ada.
3. Tidak terstruktur, yaitu berhubungan dengan persoalan baru yang cukup pelik karena banyaknya data yang belum diketahui.

### **2.2.1. Tahapan Proses Pengambilan Keputusan**

tahap-tahap dalam proses pengambilan keputusan yaitu (Kusumadewi, 2010):

#### 1. Penelusuran (*intelligence*)

Tahap pendefinisian masalah serta identifikasi informasi yang dibutuhkan dengan persoalan yang dihadapi serta keputusan yang diambil. Langkah ini sangat menentukan tingkat ketepatan keputusan yang akan diambil karena sebelum suatu tindakan diambil tentunya persoalan yang dihadapi harus dirumuskan terlebih dahulu secara jelas.

#### 2. Perancangan (*design*)

Tahap analisa dalam kaitan mencari atau merumuskan alternatif-alternatif pemecahan masalah. Setelah

permasalahan dirumuskan dengan baik, maka tahap berikutnya adalah merancang model pemecahan masalahnya dan menyusun berbagai alternatif pemecahan masalah

### 3. Pemilihan (*choice*)

Mengacu pada rumusan tujuan serta hasil yang diharapkan selanjutnya manajemen memilih alternatif solusi yang diperkirakan paling sesuai. Pemilihan alternatif ini akan mudah dilakukan jika hasil yang diharapkan terukur atau memiliki nilai kuantitas tertentu.

### 4. Implementasi (*implemmentation*)

Tahap pelaksanaan dari keputusan yang telah diambil. Pada tahap ini perlu disusun serangkaian tindakan yang terencana sehingga hasil keputusan dapat dipantau dan disesuaikan apabila terdapat perbaikan-perbaikan yang dibutuhkan.

## 2.2.2. Karakteristik dan Nilai Guna Sistem Pendukung Keputusan

Sistem pendukung keputusan berbeda dengan sistem informasi. membedakan beberapa karakteristik yaitu (Kusumadewi, 2010):

1. Sistem pendukung keputusan dirancang untuk membantu mengambil keputusan dalam memecahkan masalah yang sifatnya semi terstruktur ataupun tidak terstruktur.
2. Dalam proses pengolahan sistem pendukung keputusan mengkombinasikan penggunaan model-model atau teknik-teknik

analisis dengan teknik pemasukan data konvensional serta fungsi-fungsi pencari atau integrasi informasi.

3. Sistem pendukung keputusan dirancang sedemikian rupa sehingga dapat digunakan dengan mudah oleh orang-orang yang tidak mempunyai dasar kemampuan pengoperasian komputer yang tinggi. Oleh karena itu pendekatan yang digunakan biasanya model interaktif.
4. Sistem pendukung keputusan dirancang dengan menekankan pada aspek fleksibilitas serta kemampuan adaptasi yang tinggi, sehingga mudah disesuaikan dengan berbagai perubahan lingkungan yang terjadi dan kebutuhan pemakai.

### 2.2.3. Komponen-Komponen Sistem Pendukung Keputusan

membagi sistem pendukung keputusan menjadi 3 komponen utama atau sub sistem yaitu (Kusumadewi, 2010):

1. *Subsystem data (database)* merupakan komponen sistem pendukung keputusan penyedia data bagi sistem. Data yang dimaksud disimpan dalam suatu pangkalan data (*database*) yang di organisasikan oleh suatu sistem yang disebut dengan sistem manajemen pangkalan data (*Data Base Management System*). Melalui DBMS inilah data dapat diambil dan dievaluasi dengan cepat. Pangkalan data dalam sistem pendukung keputusan berasal dari dua sumber yaitu sumber internal

dan sumber eksternal. Data eksternal ini sangat berguna bagi manajemen dalam mengambil keputusan.

2. *Subsystem model (model base)* keunikan sistem pendukung keputusan adalah kemampuannya dalam mengintegrasikan data dengan model-model keputusan. Model adalah suatu peniruan dari alam nyata. Kendala yang sering dihadapi dalam merancang suatu model adalah bahwa 21 model yang disusun ternyata tidak mampu mencerminkan seluruh variable alam nyata. Sehingga keputusan yang diambil didasarkan pada model tersebut menjadi tidak akurat dan tidak sesuai dengan kebutuhan. Oleh karena itu dalam menyimpan berbagai model pada sistem pangkalan model harus tetap dijaga fleksibilitasnya, artinya harus ada fasilitas yang mampu membantu pengguna untuk memodifikasi atau menyempurnakan model seiring dengan perkembangan pengetahuan. Hal lain yang perlu diperhatikan adalah pada setiap model yang disimpan hendaknya ditambahkan rincian keterangan dan penjelasan yang komprehensif mengenai model yang dibuat sehingga pengguna atau perancang:

- a. Mampu membuat model yang baru dengan mudah dan cepat.
- b. Mampu mengakses dan mengintegrasikan sub rutin model.
- c. Mampu menghubungkan model dengan model yang lain melalui pangkalan data.

U N  
M L  
N U S A N I A R A

- d. Mampu mengelola model base dengan fungsi manajemen yang analog dengan manajemen *database*.

3. *Subsystem dialog (user system interface)* keunikan lain dari sistem pendukung keputusan adalah adanya fasilitas yang mampu mengintegrasikan sistem yang terpasang dengan pengguna secara interaktif. Fasilitas atau subsystem ini dikenal sebagai subsystem dialog, inilah sistem diartikulasikan dan diimplementasikan sehingga pengguna atau pemakai dapat berkomunikasi dengan sistem yang dirancang. Fasilitas yang dimiliki oleh subsystem ini adalah:

- a. Bahasa aksi (*action language*). Perangkat lunak yang dapat digunakan pengguna untuk berkomunikasi dengan sistem. Komunikasi ini dilakukan melalui berbagai media seperti *keyboard*.
- b. Bahasa tampilan. Perangkat lunak yang berfungsi sebagai sarana untuk menampilkan sesuatu. Peralatan yang digunakan merealisasi tampilan diantaranya adalah printer, grafik monitor dan lain-lain.
- c. Bahasa pengetahuan. Bagian yang mutlak diketahui oleh pengguna sehingga sistem ini dirancang dapat berfungsi secara efektif.

U N  
M L  
N U S A N T A R A



### 2.3. TOPSIS

Pada dasarnya, proses pengambilan keputusan adalah memilih suatu alternatif. Menurut Hwang dan Zeleny dalam TOPSIS (*Technique for Order Preference by Similarity to Ideal Solution*) didasarkan pada konsep dimana alternatif terpilih yang terbaik tidak hanya memiliki jarak terpendek dari solusi ideal positif, namun juga memiliki jarak terpanjang dari solusi ideal negatif (Anggraini & Jasmir, 2016). TOPSIS telah digunakan dalam banyak aplikasi termasuk keputusan investasi keuangan, perbandingan performa dari perusahaan, perbandingan dalam suatu industri khusus, pemilihan sistem operasi, evaluasi pelanggan, dan perancangan robot (Muzakkir, 2017). konsep ini banyak digunakan pada beberapa model MADM (*Multi Attribute Decision Making*) untuk menyelesaikan masalah keputusan secara praktis. Hal ini disebabkan konsepnya sederhana, mudah dipahami, komputasinya efisien, dan memiliki kemampuan untuk mengukur kinerja relatif dari alternatif-alternatif keputusan dalam bentuk matematis yang sederhana (Anggraini & Jasmir, 2016).

Secara umum, prosedur TOPSIS mengikut langkah – langkah sebagai berikut (Ma'ruf, 2016):

1. Menghitung matriks keputusan yang ternormalisasi
2. Menghitung matriks keputusan ternormalisasi yang terbobot
3. Menghitung matriks solusi ideal positif dan matriks solusi ideal negatif
4. Menghitung jarak antara nilai setiap alternatif dengan matriks solusi ideal positif dan matriks solusi ideal negatif
5. Menghitung nilai preferensi untuk setiap alternatif

TOPSIS membutuhkan rating kinerja setiap alternatif  $A_i$  pada setiap kriteria  $C_j$  yang ternormalisasi yaitu (Ma'ruf, 2016):

$$r_{ij} = \frac{X_{ij}}{\sqrt{\sum_{i=1}^m X_{ij}^2}} ; \text{ dengan } i = 1, 2, \dots, m ; \text{ dan } j = 1, 2, \dots, n$$

### Rumus 2.1. Normalisasi

Sumber: (Ma'ruf, 2016)

Solusi ideal positif  $A^+$  dan solusi ideal negatif  $A^-$  dapat ditentukan berdasarkan rating bobot ternormalisasi ( $y_{ij}$ ) sebagai:

$$y_{ij} = w_j r_{ij} \text{ dengan } i = 1, 2, \dots, m ; j = 1, 2, \dots, n.$$

$$A^+ = (y_1^+, y_2^+, \dots, y_n^+);$$

$$A^- = (y_1^-, y_2^-, \dots, y_n^-);$$

dengan

$$y_j^+ \begin{cases} \max_i y_{ij} ; \text{ jika } j \text{ adalah atribut keuntungan} \\ \min_i y_{ij} ; \text{ jika } j \text{ adalah atribut biaya} \end{cases}$$

$$y_j^- \begin{cases} \min_i y_{ij} ; \text{ jika } j \text{ adalah atribut keuntungan} \\ \max_i y_{ij} ; \text{ jika } j \text{ adalah atribut biaya} \end{cases}$$

$$j = 1, 2, \dots, m.$$

### Rumus 2.2. Bobot Ternormalisasi

Sumber: (Ma'ruf, 2016)

Jarak antara alternatif  $A_i$  dengan solusi ideal positif dirumuskan sebagai:

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

$$D_i^+ = \sqrt{\sum_{j=i}^n (y_i^+ - y_{ij})^2}$$

$$I = 1, 2, \dots, m.$$

**Rumus 2.3. Jarak Antara Alternatif dengan solusi ideal positif**

**Sumber: (Ma'ruf, 2016)**

Jarak antara alternatif  $A_i$  dengan solusi ideal negatif dirumuskan sebagai:

$$D_i^- = \sqrt{\sum_{j=i}^n (y_{ij} - y_i^-)^2}$$

$$I = 1, 2, \dots, m$$

**Rumus 2.4. Jarak Antara Alternatif dengan solusi ideal negatif**

**Sumber: (Ma'ruf, 2016)**

Nilai preferensi untuk setiap alternatif ( $V_i$ ) diberikan sebagai:

$$V_i = \frac{D_i^-}{D_i^- + D_i^+}$$

$$I = 1, 2, \dots, m$$

**Rumus 2.5. Nilai Preferensi untuk setiap alternatif**

**Sumber: (Ma'ruf, 2016)**

#### 2.4. Metode *Waterfall*

Metode *Waterfall* adalah metode yang melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan sistem lalu menuju ke tahap *analysis*, *design*, *coding*, *testing / verification*, dan *maintenance*. Disebut dengan *Waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya

dan berjalan berurutan. Sebagai contoh tahap desain harus menunggu selesainya tahap sebelumnya yaitu tahap requirement (Hutahaean & Purba, 2016). Berikut penjelasan tahapan model *Waterfall* (Radhitya, Hakim, & Solechan, 2016).

#### 1. Requirement Definition

Pengembangan sistem dimulai dengan mengadakan penelitian terhadap elemen-elemen kebutuhan sistem bersangkutan dan mendefinisikan kebutuhan-kebutuhan tersebut dan menjabarkannya ke dalam panduan bagi pengembangan sistem di tahap berikutnya. Tahap ini merupakan tahap penting dalam mendapatkan gambaran utuh sistem untuk pengembangan sistem ke dalam bentuk penerapan sistem basis data.

#### 2. System and Software Design

Pada tahap ini menyiapkan dan menyusun sistem baru, kemudian mengembangkan secara tertulis dan mendefinisikan spesifikasi, kemudian diperiksa oleh tim SQA. Selanjutnya jika disetujui oleh klien, maka dokumen tersebut merupakan kontrak kerja antara klien dan pengembang software. Selanjutnya merencanakan jadwal pengembangan software. Jika disetujui tim SQA, tahap desain baru dilakukan Pada tahap ini menyiapkan dan menyusun sistem baru, kemudian mengembangkan secara tertulis

kemudian membagi kebutuhan-kebutuhan menjadi sistem perangkat lunak atau perangkat keras. Proses tersebut menghasilkan sebuah arsitektur sistem keseluruhan. Desain perangkat lunak termasuk menghasilkan fungsi sistem perangkat lunak dalam bentuk yang mungkin ditransformasi ke dalam satu atau lebih program yang dapat dijalankan. Tahapan ini telah menentukan

alur software hingga pada tahap algoritma yang detil. Diakhir tahap ini, kembali diperiksa oleh tim SQA.

### 3. Implementation and Unit Testing

Selama tahap ini desain perangkat lunak disadari sebagai sebuah program lengkap atau unit program. Desain yang telah disetujui, diubah dalam bentuk kode-kode program. Tahap ini, kode-kode program yang dihasilkan masih pada tahap modul-modul. Di akhir tahap ini, tiap modul di uji coba tanpa diintegrasikan.

### 4. Integration and System Testing

Unit program diintegrasikan dan diuji menjadi sistem yang lengkap untuk meyakinkan bahwa persyaratan perangkat lunak telah dipenuhi. Setelah uji coba, sistem disampaikan ke konsumen.

### 5. Operation and Maintenance

Umumnya, ini adalah tahap yang terpanjang. Sistem dipasang dan digunakan. Pemeliharaan termasuk pembetulan kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaikan implementasi unit sistem dan peningkatan jasa sistem sebagai kebutuhan baru ditemukan

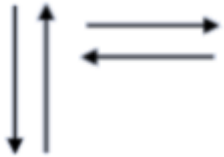
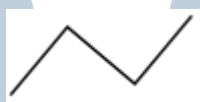
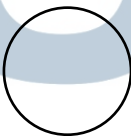
## 2.5. *Flowchart*


*Flowchart* adalah sebuah metode yang digunakan untuk menggambarkan tahap-tahap dalam menyelesaikan masalah (prosedur). Dalam kehidupan sehari-hari, *Flowchart* banyak digunakan di pusat-pusat layanan seperti kantor pemerintahan, bank, rumah sakit, organisasi masyarakat dan perusahaan.

*Flowchart* dapat diartikan sebagai suatu alat atau saran yang menunjukkan langkah-langkah yang harus dilaksanakan dalam menyelesaikan suatu permasalahan untuk komputasi dengan cara mengekspresikannya ke dalam serangkaian simbol-simbol grafis khusus (Nuraini, 2015). Berikut akan dijelaskan mengenai simbol-simbol *Flowchart* yang dibagi ke dalam 3 kategori diantaranya (Solikin & Putra, 2018):

1. Simbol Arus (*Flow Direction Symbols*) pada tabel 2.1 biasanya simbol yang termasuk ke dalam kategori ini digunakan sebagai simbol penghubung.

**Tabel 2.1. *Flowchart* Simbol Penghubung**

Simbol	Nama	Fungsi
	Flow Direction Symbol / Connecting Line	Berfungsi untuk menghubungkan simbol yang satu dengan yang lainnya, menyatakan arus suatu proses.
	Communication Link	Berfungsi untuk transmisi data dari suatu lokasi ke lokasi lain.
	Connector	Digunakan untuk menyatakan sambungan dari proses yang satu ke proses berikutnya di halaman yang sama.



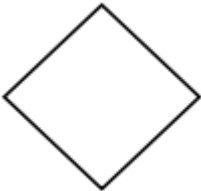
	<p>Offline Connector</p>	<p>Digunakan untuk menyatakan sambungan dari proses yang satu ke proses berikutnya di halaman yang berbeda.</p>
---	--------------------------	---

Sumber : (Solikin & Putra, 2018)



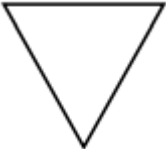

2. Simbol Proses (*Processing Symbols*)

Sesuai dengan namanya, simbol proses pada Tabel 2.2 digunakan untuk menyatakan simbol yang berkaitan dengan serangkaian proses yang dilakukan.

**Tabel 2.2. Flowchart Simbol Process**

Simbol	Nama	Fungsi
	<p>Processing</p>	<p>Digunakan untuk menunjukkan pengolahan yang akan dilakukan dalam komputer.</p>
	<p>Manual Operation</p>	<p>Digunakan untuk menunjukkan pengolahan yang tidak dilakukan oleh komputer</p>
	<p>Decision</p>	<p>Digunakan untuk memilih proses yang akan dilakukan berdasarkan kondisi tertentu.</p>



Simbol	Nama	Fungsi
	Predefined Process	Digunakan untuk mempersiapkan penyimpanan yang sedang/akan digunakan dengan memberikan harga awal.
	Terminal	Digunakan untuk memulai atau mengakhiri program.
	Offline Storage	Berfungsi untuk menunjukkan bahwa data akan disimpan ke media tertentu.
	Manual Process Symbol	Digunakan untuk mengProcesskan data secara manual dengan keyboard.

Sumber : (Solikin & Putra, 2018)





### 3. Simbol I/O (*Process-Output*)

Simbol pada Tabel 2.3 termasuk ke dalam bagian *Process-output* berkaitan dengan masukan dan keluaran.


UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



**Tabel 2.3. Flowchart Simbol Process – Output**

Simbol	Nama	Fungsi
	<p><i>Process/Output</i></p>	<p>Digunakan untuk menyatakan <i>Process</i> dan output tanpa melihat jenisnya.</p>
	<p>Punched Card</p>	<p>Digunakan untuk menyatakan masukan dan keluaran yang berasal dari card.</p>
	<p>Disk Storage</p>	<p>Digunakan untuk menyatakan masukan dan keluaran yang berasal dari pita magnetis.</p>
	<p>Magnetic Tape</p>	<p>Digunakan untuk menyatakan masukan dan keluaran yang berasal dari dokumen.</p>

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

Simbol	Nama	Fungsi
	Document	Digunakan untuk menyatakan keluaran layar monitor.

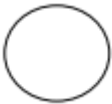


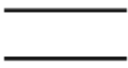
Sumber : (Solikin & Putra, 2018)

## 2.6. *Data Flow Diagram*

*Data Flow Diagram* (DFD) adalah diagram yang digunakan untuk menggambarkan proses-proses yang terjadi pada sistem yang akan dikembangkan, dengan model ini, data-data yang terlibat pada masing-masing proses dapat diidentifikasi (Elistri, Wahyudi, & Supardi, 2014). DFD merupakan representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang di aplikasikan sebagai data yang mengalir dari masukan (*Process*) dan keluaran (output) (Ermatita, 2016). DFD ini sering disebut juga dengan nama *Bubble chart*, *Bubble diagram*, model proses, diagram alur kerja, atau model fungsi. DFD ini adalah salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang dimanipulasi oleh sistem (Haryanto & Nasihin, 2018). Di bawah ini terdapat beberapa level yang lebih detail untuk merepresentasikan informasi atau fungsi yang lebih detail pada tabel 2.4 (Kusuma & Prasetya, 2017).

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

**Tabel 2.4. Simbol DFD**

Simbol	Nama	Keterangan
	Proses Transformasi	Proses yang mengubah data dari <i>Process</i> menjadi output
	Sumber & Tujuan Data	Karyawan & organisasi yang mengirim data ke dan menerima data dari sistem.
	Arus Data	Arus data yang masuk ke dalam dan keluar dari sebuah proses.
	Penyimpanan Data	Penyimpanan Data

Sumber : (Kusuma & Prasetya, 2017)

## 2.7. Entity Relationship Diagram

*Entity Relationship Diagram* (ERD) adalah pemodelan data utama yang membantu mengorganisasikan data dalam suatu proyek ke dalam entitas – entitas dan menentukan hubungan antar entitas (Fridayanthie & Mahdiati, 2016). ERD merupakan notasi grafis dalam pemodelan data konseptual yang mendeskripsikan hubungan antara penyimpanan. ERD digunakan untuk memodelkan struktur data dan hubungan antar data ERD berbeda dengan DFD yang merupakan suatu model jaringan fungsi yang akan di lakukan oleh sistem, sedangkan ERD merupakan model jaringan data yang menekankan pada struktur-struktur dan relationship data (Elistri, Wahyudi, & Supardi,

2014). Konsep paling dasar di model ER adalah entitas (*entity*), relasi (*relationship*), dan atribut (*atributes/properties*). Notasi-notasi simbolik di dalam diagram *Entity Relationship* yang dapat digunakan adalah (Wibowo, Kanedi, & Jumadi, 2015):

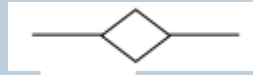
1. Pada ERD, entitas digambarkan dengan sebuah bentuk persegi panjang. Entitas adalah sesuatu apa saja yang ada di dalam *system*, nyata maupun abstrak dimana data tersimpan atau dimana terdapat data. Entitas diberi nama dengan kata benda dan dapat di kelompokkan dalam empat jenis nama, yaitu orang, benda, lokasi, kejadian (terdapat unsur waktu didalamnya). Symbol entitas dapat dilihat pada gambar 2.1.



**Gambar 2.1. Simbol Entitas**

**Sumber : (Wicaksono, Hakim, & Utomo , 2016)**

2. Pada E-R diagram, *relationship* dapat digambarkan dengan sebuah bentuk belah ketupat. *Relationship* adalah hubungan alamiah yang terjadi antara entitas. Pada umumnya penghubung (*relationship*) diberi nama dengan kata kerja dasar, sehingga memudahkan untuk melakukan pembacaan relasinya (bisa dengan kalimat aktif atau pasif). Penggambaran hubungan yang terjadi adalah sebuah bentuk belah ketupat dihubungkan dengan dua bentuk empat persegi panjang. Symbol relasi dapat dilihat pada gambar 2.2.



**Gambar 2.2. Simbol Relasi**

**Sumber : (Wicaksono, Hakim, & Utomo , 2016)**

3. Atribut adalah deksripsi data yang dapat mengidentifikasi suatu entitas. Symbol atribut dapat dilihat pada gambar 2.3.



**Gambar 2.3. Simbol Atribut**

**Sumber : (Wicaksono, Hakim, & Utomo , 2016)**

## 2.8. PHP

PHP atau kependekan dari *Hypertext PreProcessor* adalah salah satu bahasa pemrograman *open source* yang sangat cocok atau dikhususkan untuk pengembangan *Website* dan dapat di tanamkan pada sebuah skrip HTML. Bahasa PHP dapat dikatakan menggambarkan beberapa bahasa pemrograman seperti C, Java, dan Perl serta mudah untuk dipelajari.

PHP diciptakan untuk mempermudah pengembang *website* dalam menulis halaman *website* dinamis dengan cepat, bahkan lebih dari itu kita dapat mengeksplorasi hal-hal yang luar biasa dengan PHP. Sehingga dengan demikian PHP sangat cocok untuk para pemula, menengah maupun expert sekalipun (Djaelangara, Sengkey, & Lantang, 2015).

## 2.9. MySQL

MySQL merupakan turunan dari salah satu konsep utama dalam basis data sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian basis data terutama untuk proses seleksi, pemasukan, perubahan dan penghapusan data yang dimungkinkan dapat dikerjakan dengan mudah dan otomatis (Fahrival, Pohan, & Nasution, 2018). Keunggulan dari *database* Mysql adalah sebagai berikut (Wongso, 2015):

- a. *Portability*, dapat berjalan stabil pada berbagai sistem operasi.
- b. *Multi user*, dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan,
- c. *Performance tuning*, mempunyai kecepatan yang tinggi dalam menangani query,
- d. *Column typers*, memiliki tipe data yang sangat kompleks,
- e. *Comand and function*, memiliki operator dan fungsi penuh yang mendukung *select* dan *where* dalam query,
- f. *Security*, memiliki beberapa lapisan sekuritas seperti tingkat subnet *mask*, *hostname*, *privilege*, *user* dengan sistem perizinan yang mendetail serta *Password* yang terenkripsi
- g. *Scalability dan limits*, mampu menangani basis data dalam jumlah besar, dengan jumlah field lebih dari 40 juta, 60ribu tabel dan 5 milyar record.  
Batas indeks mencapai 32 buah pertabel
- h. *Localization*, dapat mendeteksi pesan kesalahan (error code) pada klien dengan menggunakan lebih dari 20bahasa.

- i. *Connectivity*, dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix Socket, Named Pipes.
- j. *Interface*, memiliki antarmuka terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API,
- k. *Client and tools*, dilengkapi dengan berbagai *tools* yang dapat digunakan untuk *Administrasi* basis data sekaligus dokumen petunjuk online
- l. *Structure table*, memiliki struktur tabel yang lebih fleksibel dalam menangani alter table dibandingkan dengan PostgreSQL dan Oracle

## 2.10. Google Maps API

Sebelum membahas *Google Maps API*, API sendiri adalah kependekan dari *Application Programming Interface*. API sendiri adalah fungsi pemrograman yang disediakan oleh aplikasi atau layanan agar layanan tersebut bisa diintegrasikan dengan aplikasi yang kita buat. Google Maps adalah layanan gratis yang diberikan oleh Google dan sangat populer. Google Maps adalah suatu peta dunia yang dapat kita gunakan untuk melihat suatu daerah. Dengan kata lain, Google Maps merupakan suatu peta yang dapat dilihat dengan menggunakan suatu browser. Kita dapat menambahkan fitur Google Maps dalam *website* yang telah kita buat atau pada blog kita yang berbayar maupun gratis sekalipun dengan Google Maps API. Google Maps API adalah suatu library yang berbentuk JavaScript (Ariyanti, Khairil, & Kanedi, 2015)



## 2.11. Penelitian Terdahulu

**Tabel 2.5. Tabel Penelitian Sebelumnya**

1	Penulis	Fatima Anggraini, Jasmir
	Judul	Analisis dan perancangan sistem pendukung keputusan rekomendasi pemilihan perumahan dengan metode TOPSIS
	Nama jurnal	Jurnal Manajemen Sistem Informasi Vol. 1, No. 5, Desember 2016
	Kesimpulan	Berdasarkan hasil penelitian, penentuan kriteria adalah tahapan awal dalam penerapan metode TOPSIS, metode TOPSIS digunakan untuk perhitungan perangkingan dan solusi yang di rekomendasikan
2	Penulis	Alpaniam, Dwi Marisa Midyanti
	Judul	Aplikasi pemilihan rumah di kota pontianak menggunakan metode TOPSIS berbasis <i>website</i>
	Nama jurnal	Jurnal Coding, Sistem Komputer Untan, Vol 6, No. 3 (2018)
	Kesimpulan	Berdasarkan hasil pengujian dapat disimpulkan bahwa spk yang dibuat berbasis <i>website</i> dengan menerapkan TOPSIS untuk mengurutkan preferensi akhir dari tertinggi sampai terendah untuk memberikan rekomendasi rumah terbaik dari sistem ke pengguna.
3	Penulis	Tomy, Zainal, Dyna
	Judul	Sistem pendukung keputusan pemilihan rumah tinggal di perumahan menggunakan metode <i>Simple Additive Weighting</i> (SAW) (studi kasus : Kota Samarinda)
	Nama jurnal	Prosiding Seminar Ilmu Komputer dan Teknologi Informasi Vol. 2, No. 1, Maret 2017



	Kesimpulan	Penerapan metode SAW mampu memberikan alternatif terbaik dalam penentuan perumahan dan sistem dapat membantu memberikan rekomendasi kepada calon pembeli rumah berdasarkan sistem sesuai dengan data yang dimasukkan oleh <i>Admin</i> .
4	Penulis	Meta Amalya Dewi
	Judul	Penggunaan <i>Simple Additive Weighting</i> Dalam Pengembangan Sistem Penunjang Keputusan Penentuan Bonus Karyawan
	Nama Jurnal	ULTIMA InfoSys, Vol. IX, No.1, Juni 2018
	Kesimpulan	Dibentuknya sistem dengan metode SAW dapat meningkatkan kinerja Manajer Personalia dalam proses perhitungan bonus tahunan karyawan dengan hasil perhitungan yang akurat dan dapat dipertanggungjawabkan dengan baik.

Sistem yang dibentuk oleh Fatima merupakan sistem pendukung keputusan yang menggunakan metode TOPSIS dengan tujuan untuk mencari perumahan terbaik untuk dijadikan tempat tinggal (Anggraini & Jasmir, 2016). Sistem yang dibentuk oleh alpaniam merupakan sistem pendukung keputusan menggunakan metode TOPSIS dengan tujuan untuk menentukan pemilihan rumah di kota Pontianak (Alpaniam & Midyanti, 2018). Sistem yang dibentuk oleh Tomy merupakan sistem pendukung keputusan yang menggunakan metode SAW dengan tujuan untuk mendapatkan rekomendasi sebuah rumah di kota Samarinda (Adianto, Arifin, & Khairina, 2017). Penelitian terdahulu yang terakhir merupakan dari penelitian mahasiswa UMN yang telah membentuk sistem dengan metode SAW dapat meningkatkan kinerja Manajer Personalia dalam proses perhitungan

bonus tahunan karyawan dengan hasil perhitungan yang akurat dan dapat dipertanggungjawabkan dengan baik (Dewi, 2018).

Perbedaan yang diteliti dalam rancangan aplikasi berbasis website ini terdapat pada sistem yang merupakan pendukung keputusan dengan menggunakan metode TOPSIS dan juga sistem yang bertujuan untuk membantu agent property menjual rumah kepada para calon pembelinya dengan menggunakan kriteria.

A large, light blue circular watermark logo is centered on the page. It features a stylized grid of white squares of varying sizes arranged in a pattern that resembles a staircase or a grid structure.

# UMN

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA