BAB II

TINJAUAN PUSTAKA

II.1. Sistem Pendukung Keputusan

Sistem Pendukung Keputusan adalah sistem informasi berbasis komputer yang interaktif, dengan cara mengolah data dengan berbagai model untuk memecahkan masalah-masalah yang tidak terstruktur sehingga dapat memberikan informasi yang bisa digunakan oleh para pengambil keputusan dalam membuat sebuah keputusan. Perusahaan(Jhon Fransdesker , Sri Primaini, Nazori Suhandi, 2015). Dalam sebuah Sistem Pendukung Keputusan, sumber daya intelektual yang dimiliki seseorang dipadukan dengan kemampuan komputer untuk membantu meningkatkan kualitas dari keputusan yang diambil. Pengambilan keputusan merupakan sebuah proses memilih sebuah tindakan diantara beberapa alternatif yang ada, sehingga tujuan yang diinginkan dapat tercapai.

Sistem Pendukung Keputusan (SPK) bagian dari sistem informasi berbasis komputer termasuk sistem berbasis pengetahuan atau manajemen pengetahuan yang dipakai untuk mendukung pengambilan keputusan dalam suatu organisasi atau perusahaan(Jhon Fransdesker , Sri Primaini, Nazori Suhandi, 2015).

Sistem Pendukung Keputusan (SPK) biasanya dibangun untuk mendukung solusi atas suatu masalah atau untuk suatu peluang, Aplikasi Sistem Pendukung Keputusan (SPK) digunakan dalam pengambilan keputusan. Aplikasi Sistem Pendukung Keputusan (SPK) menggunakan CBIS (*Computer Based Information System*) yang fleksibel, interaktif, dan dapat diadaptasi, yang dikembangkan untuk

mendukung solusi atas masalah manajemen spesifik yang tidak terstruktur.Sistem Pendukung Keputusan sebagai sistem berbasis komputer yang terdiri dari tiga komponen yang saling berinteraksi, sistem bahasa (mekanisme untuk memberikan komunikasi antara pengguna dan komponen sistem pendukung keputusan lain), sistem pengetahuan (respositori pengetahuan domain masalah yang ada pada Sistem Pendukung Keputusan atau sebagai data atau sebagai prosedur) dan sistem pemrosesan atau lebih kapabilitas manipulasi masalah umum yang diperlukan untuk pengambilan keputusan (DickyNofriansyah, 2014).

II.1.1. Karakteristik Sistem Pendukung Keputusan

Menurut (Dicky Nofriansyah, 2014), karakteristik dari sistem pendukung keputusan sebagai berikut:

1. Mendukung proses pengambilan keputusan suatu organisasi atau perusahaan.
2. Adanya *interface*manusia/mesin dimana manusia (*user*) tetap memegang kontrol proses pengambilan keputusan.
3. Mendukung pengambilan keputusan untuk membahas masalah terstruktur, semi terstruktur serta mendukung beberapa keputusan yang saling berinteraksi.
4. Memiliki kapasitas dialog untuk memperoleh informasi sesuai dengan kebutuhan.
5. Memiliki subsistem yang terintegrasi sedemikian rupa sehingga dapat berfungsi sebagai kesatuan sistem.
6. Memiliki dua komponen yaitu data dan model.

II.1.2. Ciri –Ciri Sistem Pendukung Keputusan

Menurut (Dicky Nofriansyah, 2014), ciri-ciri Sistem Pendukung Keputusan sebagai berikut :

1. Banyak pilihan / alternatif
2. Ada kendala
3. Mengikuti suatu pola atau model tingkah laku, baik yang terstruktur maupun tidak terstruktur
4. Banyak input/variabel
5. Ada faktor resiko, dibutuhkan kecepatan, ketepatan dan keakuratan.

II.1.3. Tahapan Sistem Pengambilan Keputusan

Ada 4 tahap yang harus dilalui dalam proses pengambilan keputusan yaitu:

a. Penelusuran(*Inteligensi*)

   Tahap ini merupakan tahap pendefinisian masalah serta identifikasi informasi yang di butuhkan yang berkaitan dengan persoalan yang di hadapi serta keputusan yang akan di ambil.

b. Perancangan (*Desain*)

   Tahap ini merupan tahap analisa dalam kaitan mencari atau merumuskan alternatif-alternatif pemecahan masalah .

c. Pemilihan (*Choice*)

   Yaitu memilih alternatif solusi yang di perkirakan paling sesuai.

d. Implementasi (*Implementation*)

Tahap ini merupakan tahap pelaksanaan dari keputusan yang telah diambil (Diana Fatmawati, Sulton, Sadikin , 2016).

II.1.4. Komponen Sistem Pendukung Keputusan

Sistem pendukung keputusan dibangun oleh tiga komponen yaitu subsistem manajemen data (*Database*), subsistem model (*Modelbase*), dan subsistem dialog (*User System Interface*).

1. Subsistem Manajemen Data (*Database*)

   Subsistem data merupakan komponen Sistem Pendukung Keputusan yang berguna sebagai penyedia data bagi sistem. Data tersebut di simpan untuk diorganisasikan dalam sebuah basis data yang di organisasikan oleh suatu sistem manajemen basis data (*Database Management System*)

2. Subsistem Model (*Model Base*)

   Model adalah tiruan dari alam nyata, kendala yang sering di hadapi dalam merancang model adalah bahwa model yang di rancang tidak mampu mencerminkan seluruh variable alam nyata, sehingga keputusan yang di ambil tidak sesuai dengan kebutuhan oleh karena itu, dalan penyimpanan berbagai model harus di perhatikan dan harus di jaga fleksibilitasnya. Hal lain yang harus di perhatikan adalah pada setiap model yang di simpan hendaknya di tambahkan rincian keterangan dan penjelasan yang komprehensif mengenai model yang di buat.

3. Subsistem Dialog(*User System Interface*)

   Subsistem dialog adalah fasilitas yang mampu mengintegrasikan sistem yang terpasang dengan pengguna secara interaktif , yang dikenal dengan subsistem dialog, sistem di implementasikan sehingga pengguna dapat berkomunikasi dengan sistem yang di buat. (Dicky Nofriansyah, 2014).

II.1.5. Tujuan Sistem Pendukung Keputusan

Tujuan dari Sistem Pendukung Keputusan adalah :

1. Meningkatkan efektivitas keputusan yang diambil manajer lebih dari pada perbaikan efesiensinya.
2. Membantu manajer dalam pengambilan keputusan atas masalah semi terstruktur .
3. Kecepatan komputasi. Komputer memungkinkan para pengambil keputusan untuk melakukan banyak komputasi secara cepat dengan biaya yang rendah.
4. Dukungan kualitas. Komputer bisa meningkatkan kualitas keputusan yang dibuat sebagai contoh, semakin banyak data yang diakses, makin banyak juga alternatif yang bisa dievaluasi.

Peningkatan produktivitas. Membangun satu kelompok pengambil keputusan, terutama para pakar, bisa sangat mahal. Pendukung terkomputerisasi bisa mengurangi ukuran kelompok dan memungkinkan para anggotanya untuk berada di berbagai lokasi yang berbeda-beda (menghemat biaya perjalanan). (Elena Monica, Dadang Sudrajad, Nana Suarna , 2015).

II.2. Metode TOPSIS

TOPSIS adalah metode multi kriteria yang digunakan untuk mengidentifikasi solusi dari himpunan alternatifberdasarkan minimalisasi simultan dari jarak titik ideal dan memaksimalkan jarak dari titik terendah.Metode TOPSIS merupakan metode penilaian yang ditafsirkandapat memberikan setiap objek untuk dievaluasi nilainya secara spesifik.(Ahmad abdul chamid,2016).

Konsep utama TOPSIS adalah mencari alternatif preferensi terbaik tidak hanya dari nilai alternatif jarak terpendek dari solusi ideal positif, tetapi juga dari nilai alternatif jarak terjauh dari solusi ideal negatif. Langkah-langkah perhitungan metode TOPSIS :

1. Menghitung Normalisasi

$$R_{ij} = \frac{X_{ij}}{\sum_{i=1}^{m} x_{ij}^{2}}. \qquad (1)$$

Dimana :

Rij = nilai normalisasi dari tiap alternatif (i) terhadap kriteria (j) dengan i = 1,2,…n; dan j = 1,2,…,n.

Xij = nilai dari suatu alternatif (i) terhadap kriteria (j) dengan i = 1,2,…n; dan j = 1,2,…,n.

2 Menghitung nilai normalisasi terbobot

$$Y_{ij} = W_i . R_{ij} \qquad (2)$$

Dimana :

Yij = nilai normalisasi terbobot.

Wi =adalah bobot masing-masing kriteria.

Rij = nilai normalisasi masing-masing alternatif dimana i=1,2,...,m;

dan j = 1,2,...,n.

3. Mengidentifikasi solusi ideal positif dan solusi ideal negatif

Solusi ideal positif

$$A^{+} = \left(y_1^{+}, y_2^{+}, \cdots, y_n^{+}\right); \quad (3)$$

Dimana :

A^{+}=hasil nilai maksimal dari nilai normalisasi terbobot (y_{ij}) jika atributnya adalah atribut keuntungan atau hasil nilai minimal dari nilai normalisasi terbobot (y_{ij}) jika atributnya adalah atribut biaya.

Solusi ideal negatif

$$A^{-} = \left(y_1^{-}, y_2^{-}, \cdots, y_n^{-}\right); \quad (4)$$

Dimana :

A^{-} = hasil nilai minimal dari nilai normalisasi terbobot (y_{ij}) jika atributnya adalah atribut keuntungan atauhasil nilai maksimal dari nilai normalisasi terbobot (y_{ij}) jika atributnya adalah atribut biaya.

4. Menghitung jarak setiap alternatif dengan solusi ideal positif dan negatif

Jarak Alternatif solusi ideal positif

$$D_i^{+} = \sqrt{\sum_{j=1}^{n}\left(y_i^{+} - y_{ij}\right)^2}; \quad (5)$$

Dimana :

Di^+ = nilai akar dari jumlah nilai tiap alternatif yang diperoleh dengan nilai normalisasi terbobot untuk setiap alternatif (y_{ij}) di kurangi solusi ideal positif (yi^+) kemudian di pangkat dua.

Jarak Alternatif solusi ideal negatif

$$D_i^- = \sqrt{\sum_{j=1}^{n}\left(y_{ij} - y_i^-\right)^2}; \quad (6)$$

Dimana :

Di^- = nilai akar dari jumlah nilai tiap alternatif yang diperoleh dengan nilai ternormalisasi terbobot untuk setiap alternatif (y_{ij}) di kurangi solusi ideal negatif (yj^-) kemudian di pangkat dua.

ideal negatif (yj^-) kemudian di pangkat dua.

5. Menentukan nilai kedekatan setiap alternatif dengan solusi ideal *(preferensi)*

$$V_i = \frac{D_i^-}{D_i^- + D_i^+}; \quad (7)$$

Dimana :

V_i = nilai jarak antara alternatif A_i dengan solusi ideal negatif (Di^-) dibagi dengan jumlah nilai jarak antara alternatif A_i dengan solusi ideal negatif (Di^-) ditambah jumlah nilai jarak antara alternatif A_i dengan solusi ideal negatif (Di^+).

Dari hasil Nilai $V_i$yang didapat dari tiap alternatif lalu alternatif akan diurutkan berdasarkan dari nilai terbesar hingga terkecil untuk mendapatkan hasil ranking alternatif terbaik.(Ahmad abdul chamid,2016)

II.3. UML (*Unified Modeling Language*)

Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah stadarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrorgaman berorientasi objek, yaitu Unified Modelling Language (UML).(Jhon Fransdesker , Sri Primaini, Nazori Suhandi , 2015).UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks.(Jhon Fransdesker , Sri Primaini, Nazori Suhandi, 2015)

UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. *UML* saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.(Gellysa Urva dan Helmi Fauzi Siregar, 2015).

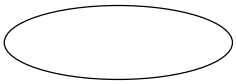
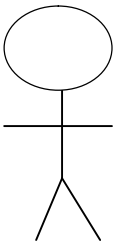
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML adalah sebagai berikut:

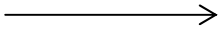
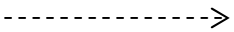
1. *Use case* Diagram

Use case diagram merupakan pemodelan untuk kelakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem

informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel II.1 dibawah ini:

Tabel II.1. Simbol *Use Case*

Gambar	Keterangan
	*Use case* menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama *use case*.
	Aktor adalah *abstraction* dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasikan aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan *use case*, tetapi tidak memiliki control terhadap *use case*.
	Asosiasi antara aktor dan *use case*, digambarkan dengan garis tanpa panah yang mengindikasikan

	siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.
	Asosiasi antara aktor dan *use case* yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	*Include*, merupakan di dalam *use case* lain (*required*) atau pemanggilan *use case* oleh *use case* lain, contohnya adalah pemanggilan sebuah fungsi program.
	Extend, merupakan perluasan dari *use case* lain jika kondisi atau syarat terpenuhi.

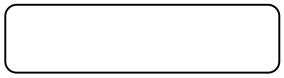
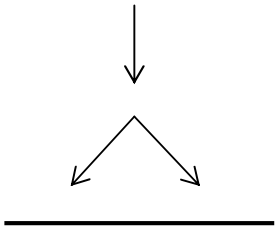
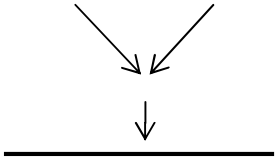
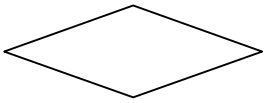
(Sumber: Gellysa Urva dan Helmi Fauzi Siregar, 2015)

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.2 dibawah ini:

Tabel II.2. Simbol *Activity Diagram*

Gambar	Keterangan
	*Start point*, diletakkan pada pojok kiri atas dan

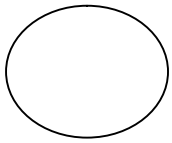
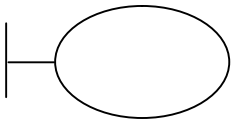
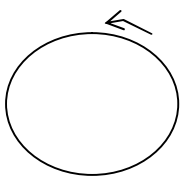
	merupakan awal aktifitas.
	*End point*, akhir aktifitas.
	*Activites*, menggambarkan suatu proses/kegiatan bisnis.
	*Fork* (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	*Join* (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	*Decision Points*, menggambarkan pilihan untuk pengambilan keputusan, *true, false.*
	*Swimlane*, pembagian *activity* diagram untuk menunjukkan siapa melakukan apa.

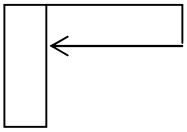
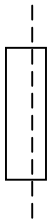
(Sumber : Gellysa Urva dan Helmi Fauzi Siregar, 2015)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.3 dibawah ini :

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Keterangan
	*EntityClass*, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	*Boundary Class*, berisi kumpulan kelas yang menjadi *interface* atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan *form* cetak.
	*Control class*, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.

	*Message,* simbol mengirim pesan antar *class*.
	*Recursive,* menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	*Activation, activation* mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	Lifeline, garis titik-titik yang terhubung dengan objek, sepanjang *lifeline* terdapat *activation*.

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar, 2015)

4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggng jawab entitas yang menentukan perilaku sistem.

Class diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization*dan *Aggregation,* Atribut (*Attributes*), Operasi

(Operation)*Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut.

Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti yang dapat dilihat pada tabel II.4 dibawah ini:

Tabel II.4. *Multiplicity Class Diagram*

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar, 2015)

II.4. Pengertian Basis Data (*Database*)

Basis data atau Database adalah kumpulan informasi yang disusun dan merupakan suatu kesatuan yang utuh yang disimpan di dalam perangkat keras (komputer) secara sistematis sehingga dapat diolah menggunakan perangkat lunak.(Ganda Yoga Swara,Yunes Pebriadi, 2016).Dengan sistem tersebut data yang terhimpun dalam suatu database dapat menghasilkan informasi yang

berguna.ditarik kesimpulan bahwa *database* adalah sekelompok data yang mempunyai ciri-ciri khusus dan dapat dikelola sedemikian rupa sehingga bisa menghasilkan sebuah format data yang baru.(Ganda Yoga Swara,Yunes Pebriadi, 2016).

II.4.1. Operasi Dasar Basis Data

Ada beberapa operasi basis data diantaranya :

1. Pembuatan basis data baru (*create database*), yang identik dengan pembuatan lemari arsip yang baru.
2. Penghapusan basis data (*drop database*), yang identik dengan perusakan lemari arsip (sekaligus beserta isinya jika ada).
3. Pembuatan file/tabel baru ke suatu basis data (*create table*), yang identik dengan penambahan map arsip baru ke sebuah lemari arsip yang telah ada.
4. Penghapusan file/tabel dari suatu basis data (*drop table*), yang identik dengan perusakan map arsip lama yang ada di sebuah lemari arsip
5. Penambahan/pengisian data baru ke sebuah file/tabel di sebuah basis data (*insert*), yang identik dengan penambahan lembaran arsip ke sebuah map arsip(Ganda Yoga Swara,Yunes Pebriadi, 2016).

II.4.2. Komponen Sistem Basis Data

Berikut berapa komponen operasi basis data diantaranya

a. Perangkat Keras (Hardware)

Perangkat keras yang biasanya terdapat dalam sebuah sistem basis data adalah :

1. Komputer (satu untuk sistem yang standalone atau lebih dari satu untuk sistem jaringan).
2. Memori sekunder yang on-line (Harddisk).
3. Memori sekunder yang off-line (tape atau Removable Disk) untuk keperluan backup data.
4. Media / perangkat komunkasi (untuk sistem jaringan).

b. Sistem Operasi (*Operating System*)

Secara sederhana, sistem opersi merupakan program yang mengaktifkan / memfungsikan sistem komputer, mengendalikan seluruh sumber daya (resource) dalam komputer dan melakukan operasioperasi dasar dalam computer.

c. Basis Data (*Database*)

Sebuah sistem basis data dapat memiliki beberapa basis data.Setipa basis data dapat berisi.Memiliki sejumlah objek basis data (seperti file / tabel, indeks dan lain-lain).Selain menyimpan data, setiap basis data juga mengandung / menyimpan definisi struktur (baik untuk basis data maupun objek-objeknya secara detail).

d. Sistem Pengelola Basis Data (*Database Management System/DBMS*)

Pengelola basis data secara fisik tidak dilakukan oleh pemakai secara langsung, tetapi ditangani oleh sebuah Perangkat Lunak (sistem) yang khusus / spesifik. Perangkat lunak inilah (DBMS) yang akan menentukan bagaiman data diorganisasi, disimpan, diubah dan diambil kembali. Ia juga menerapkan mekanisme pengamanan data pemakaian data secara bersama,

pemaksaan keakuratan/konsistensi data dan sebagainya. (Ganda Yoga Swara,Yunes Pebriadi, 2016).

II.4.3 MySQL

MySQL adalah sebuah perangkat lunak system manajemen basis data SQL (DBMS) yang multithread, dan multi-user.MySQL adalah implementasi dari system manajemen basisdata relasional (RDBMS). MySQL dibuah oleh TcX dan telah dipercaya mengelola system dengan 40 buah database berisi 10.000 tabel dan 500 di antaranya memiliki 7 juta baris. MySQL AB merupakan perusahaan komersial Swedia yang mensponsori dan yang memiliki MySQL.MySQL sebenarnya turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structured Query Language)*. SQL adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis Pendiri MySQL AB adalah dua orang Swedia yang bernama David Axmark, Allan Larsson dan satu orang Finlandia bernama Michael “Monty”. Setiap pengguna MySQL dapat menggunakannya secara bebas yang didistribusikan gratis dibawah lisensi GPL(General Public License) namun tidak boleh menjadikan produk turunan yang bersifat komersial. (Jhon Fransdesker , Sri Primaini, Nazori Suhandi , 2015)

II.4.4. Normalisasi

Normalisasi adalah suatu teknik untuk mengorganisasikan data kedalam tabel-tabel untuk memenuhi kebutuhan pemakai didalam suatu organisasi. Normalisasi digunakan sebagai pemandu dalam merancang basisdata relasional.

Pada dasarnya, normalisasi adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari tabel rasional.(Ganda Yoga Swara,Yunes Pebriadi, 2016).

Beberapa tujuan dari normalisasi adalah :

1. Untuk menghilangkan kerangkapan data
2. Untuk mengurangi kompleksitas.
3. Untuk mempermudah pemodifikasian data.

II.4.5. Bentuk-bentuk dari normalisasi

1. Bentuk tidak normal *(unformalized form)*

   Bentuk ini merupakan bentuk data yang direkam, tidak ada keharusan untuk mengikuti suatu format tertentu, dapat saja data tidak lengkap atau terduplikasi.

2. Bentuk normal pertama (1NF atau *first normal form*)

   Bentuk normal pertama mempunyai ciri-ciri yaitu setiap data dibentuk dalam *flat file* (file dasar) dan data dibentuk dalam satu record demi satu record. Tidak ada set atribut yang berulang-ulang atau atribut yang bernilai ganda.

3. Bentuk normal kedua (2NF atau *second normal form*)

   Bentuk normal kedua mempunyai syarat yaitu bentuk data telah memenuhi kriteria bentuk normal pertama, atribut bukan kunci haruslah bergantung secara fungsi pada kunci utama, atau *primery key*, sehingga untuk bentuk

normal kedua haruslah sudah ditentukan kunci-kunci *field*. Kunci *field* harus unik dan dapat mewakili atribut lain yang menjadi anggotanya.

4. Bentuk normal ketiga (3NF atau *three normal form*)

   Untuk menjadi bentuk normal ketiga maka relasi haruslah dalam bentuk normal kedua dan sama atribut bukan primer tidak punya hubungan yang transi, dengan kata lain setiap atribut bukab kunci haruslah bergantun pada *primery key* secara menyeluruh. (Ganda Yoga Swara, Yunes Pebriadi, 2016).

II.5. Aplikasi Berbasisi Web

Aplikasi web merupakan sebuah jenis aplikasi yang menggunakan tehnologi browser untuk menjalankan sebuah aplikasi tersebut dengan di akses melalu jaringan portable.(Angga Prayuda,2012) sedangkan pengertian aplikasi web yang lainya memiliki pengertian aplikasi web merupakan sebuah program yang tersimpan pada sebuah server kemudian dikirim melalui internet dan diakses dengan melalui tampilan muka browser.Dari pengertian-pengertian di atas jadi dapat di simpulkan bahwa aplikasi web merupakan sebuah aplikasi yang diakses melakui web browser dengan menggunakan jaringa internet dan intranet.aplikasi web ini juga merupakan sebuah perangkat lunak atau software yang di kodekan dengan bahasa pemogramman seperti html, java script, css, ruby, python, php, dan bahasa pemogramman lainya.Lain hal nya dengan menggunakan aplikasi dekstop yang harus terinstall secara permanen pada komputer user. Keduanya sama-sama ada keuntungan dan kelemahannya. Berikut adalah keuntungan dan kekurangan yang membandingkan aplikasi berbasis web dan desktop.

II.5.1 Keuntungan dan Kekurangan Aplikasi Web

a. Keuntungan Aplikasi Web :

1. Dapat diakses dimanapun tanpa harus terinstall di komputer user
2. Kompatibel dengan semua jenis komputer dan operating system, meski terkadang untuk beberapa browser user perlu dikonfigurasi
3. Tidak perlu ada lisensi tambahan pada komputer user
4. Bagi progammer, mudah jika harus update atau bug fix karena hanya perlu update di komputer server
5. Spesifikasi komputer user tidak terlalu tinggi

b. Kekurangan Aplikasi Web :

1. Sangat tergantung dengan kecepatan akses terhadap server
2. Dibutuhkan spesifikasi komputer server yang lumayan tinggi jika pemakainya semakin banyak
3. Terkadang kompatibiltas browser pada user berpengaruh pada layout aplikasi.

II.5.2 Keuntungan dan Kekurangan Aplikasi Desktop

a . Keuntungan Aplikasi Desktop:

1. Bisa berjalan independen tanpa tergantung adanya browser.
2. Jika aplikasi berjalan secara stand alone, maka koneksi jaringan tida dibutuhkan.
3. Mudah memodifikasi settingannya.

4. Umumnya aplikasi berjalan lebih cepat karena seluruh resource dan proses terjadi di komputer user sendiri.

b. Kekurangan Aplikasi Desktop :

1. Harus terinstall pada komputer masing-masing user.
2. Sangat tergantung kompatiblitas terhadap sistem operasi pada komputer user.
3. Umumnya, satu instalasi pada komputer user harus terdapat satu lisensi.
4. Biasanya membutuhkan spesifikasi hardware yang cukup tinggi.(Angga Prayuda,2012).

II.6. PHP

PHP adalah sebuah bahasa pemograman yangberjalan dalam sebuah *web-server* (*server side*). PHP diciptakan oleh programmer unix dan Perl yang bernamaRasmus Lerdoft pada bulan Agustus-September 1994. Pada awalnya, Rasmus mencoba menciptakan sebuah script dalam wesite pribadinya dengan tujuan untukmemonitor siapa saja yang pernah mengunjungi websitenya. Script PHP adalah bahasa program yang berjalanpada sebuah webserver, atau sering disebut *server-side*. Oleh karena itu,PHP dapat melakukan apa saja yang bisadilakukan program CGI lain, yaitu mengolah data dengan tipe apapun, menciptakan halaman web yang dinamis, serta menerima dan menciptakan cookies, dan bahkan PHP bisa melakukan lebih dari itu. Arti script server-side adalah, agar dapat menjalankan script ini dibutuhkan tiga program utama, yaitu web-server (dapat berupa IIS dari windows atau apache), modul PHP dan juga web browser. PHP dapat berjalan pada semua jenis system operasi, antara lain

pada Linux dan varian Unix (HP-UX, Solaris dan OpenBSD), pada Ms Windows, Mac dan masih banyak lag, selain itu PHP juga dapat berjalan pada beberapa jenis web-server antara lain Apache, Microsoft IIS, personal webserver, Netscape dan Iplanet Server, Caudium, Xitami, Omnihttpd dan masih banyak lagi.(Jhon Fransdesker , Sri Primaini, Nazori Suhandi , 2015).

II.6.1 Kelebihan PHP

Kelebihan yang dimiliki PHP dibandingkan dengan bahasa pemrograman yang lain, Diantaranya :

1. Bisa membuat Web menjadi Dinamis.
2. PHP bersifat Open Source yang berarti dapat digunakan oleh siapa saja secara gratis.
3. Program yang dibuat dengan PHP bisa dijalankan oleh Semua Sistem Operasi karena PHP berjalan secara Web Base yag artinya semua Sistem Operasi bahkan HP yang mempunyai Web Browser dapat menggunakan program PHP.
4. Aplikasi PHP lebih cepat dibandingkan dengan ASP maupun Java.
5. Mendukung banyak paket Database seperti MySQL, Oracle, PostgrSQL, dan lain-lain.
6. Bahasa pemrograman PHP tidak memerlukan Kompilasi / Compile dalam penggunaannya.
7. Banyak Web Server yang mendukung PHP seperti Apache, Lighttpd, IIS dan lain-lain.
8. Pengembangan Aplikasi PHP mudah karena banyak Dokumentasi,

Refrensi & Developer yang membantu dalam pengembangannya.. (Nurul Imam, 2013)

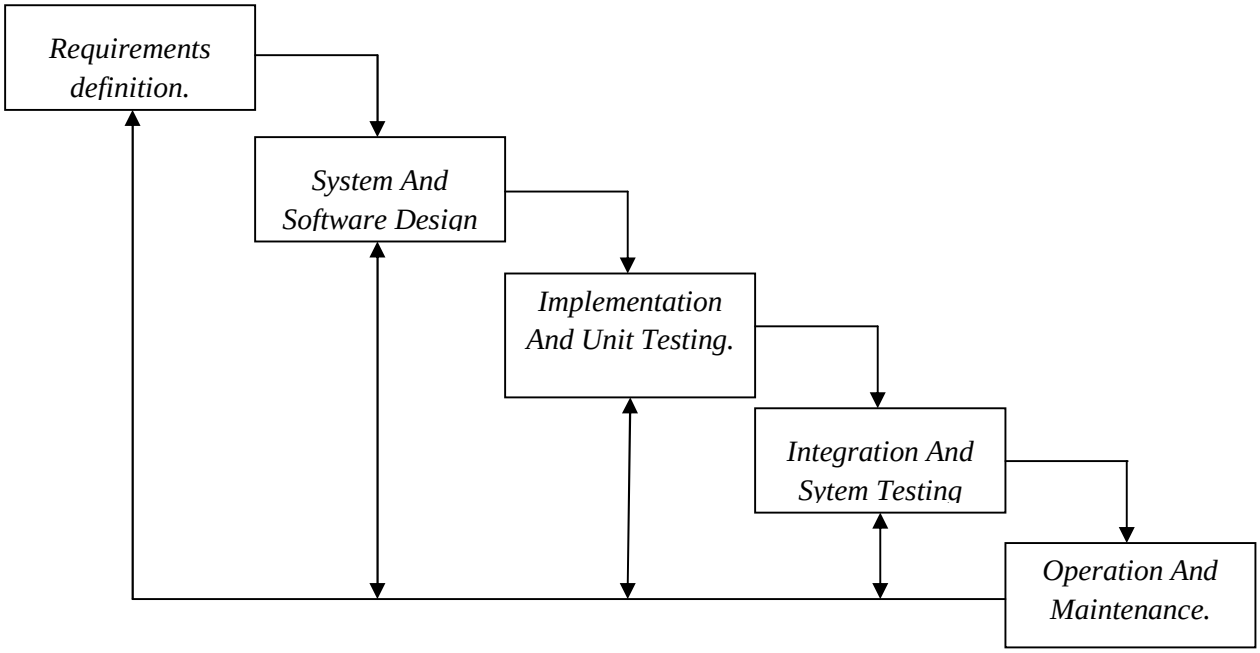
II.6.2. Kekurangan PHP

Selain kelebihan PHP, PHP juga mempunyai kekurangan. Namun masalah kekurangannya sangat sedikit. Diantaranya :

1. PHP Tidak mengenal Package.
2. Tidak ideal untuk pengembangan skala besar.
3. Jika tidak di encoding, maka kode PHP dabat dibaca semua orang & untuk meng encodingnya dibutuhkan tool dari Zend yang mahal sekali biayanya.
4. PHP memiliki kelemahan keamanan. Jadi Programmer harus jeli & berhati-hati dalam melakukan pemrograman & Konfigurasi PHP. (Nurul Imam, 2013)

II.7.Metode *Waterfall*

Metode*waterfall* melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan sistem lalu menuju ke tahap analisis, desain, coding, testing / verification, dan maintenance. (Rizki Alfiasca, Antok,2014). 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.Secara umum tahapan pada model waterfall dapatdilihat pada gambar 1 berikut.



Gambar II.1 Siklus Pengembangan Dengan Metode *Waterfall*

(Sumber : Rizki Alfiasca, Antok,2014)

Gambar di atas adalah tahapan umum dari model proses ini, memecah model ini menjadi 6 tahapan meskipun secara garis besar sama dengan tahapan-tahapan model waterfall pada umumnya. Berikut adalah penjelasan dari tahaptahap yang dilakukan di dalam model ini.

1. *Requirements definition*.

   Proses pencarian kebutuhan diintensifkan dan difokuskan pada software. Untuk mengetahui sifat dari program yang akan dibuat, maka para software engineer harus mengerti tentang domain informasi dari software, misalnya fungsi yang dibutuhkan, user interface. Dari 2 aktivitas tersebut (pencarian

kebutuhan sistem dan software) harus didokumentasikan dan ditunjukkan kepada pelanggan.

2. *System And Software Design.*

   Proses ini digunakan untuk mengubah kebutuhankebutuhan diatas menjadi representasi ke dalam bentuk “blueprint” software sebelum coding dimulai. Desain harus dapat mengimplementasikan kebutuhan yang telah disebutkan pada tahap sebelumnya. Seperti 2 aktivitas sebelumnya, maka proses ini juga harus didokumentasikan sebagai konfigurasi dari software.

3. *Implementation And Unit Testing*

   Untuk dapat dimengerti oleh mesin, dalam hal ini adalah komputer, maka desain tadi harus diubah bentuknya menjadi bentuk yang dapat dimengerti oleh mesin, yaitu ke dalam bahasa pemrograman melalui proses coding. Tahap ini merupakan implementasi dari tahap design yang secara teknis nantinya dikerjakan oleh programmer.

4. *Integration And Sytem Testing*

   Sesuatu yang dibuat haruslah diujicobakan.Demikian juga dengan software. Semua fungsi-fungsi software harus diujicobakan, agar software bebas dari error, danhasilnya harus benar- benar sesuai dengan kebutuhan yang sudah didefinisikansebelumnya.

5. *Operation And Maintenance*

   Pemeliharaan suatu software diperlukan, termasuk di dalamnya adalah pengembangan, karena software yang dibuat tidak selamanya hanya seperti itu.Ketika dijalankan mungkin saja masih ada error kecil yang tidak

ditemukansebelumnya, atau ada penambahan fitur-fitur yang belum ada pada software tersebut. (Rizki Alfiasca, Antok,2014).