

BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem Informasi

2.1.1 Pengertian Sistem

Whitten mendefinisikan sistem sebagai sekelompok komponen yang saling terhubung dan berfungsi secara bersamaan untuk mencapai hasil yang diinginkan (whitten:2009:6).

Menurut gaol, sistem adalah hubungan satu unit dengan unit-unit lainnya yang saling berhubungan satu sama lainnya dan yang tidak dapat dipisahkan serta menuju suatu kesatuan dalam rangka mencapai tujuan yang telah ditetapkan (gaol:2008:9).

“Suatu sistem biasa dianggap merupakan suatu himpunan bagian yang saling berkaitan yang membentuk satu keseluruhan yang rumit atau kompleks tetapi merupakan satu kesatuan”. Contoh wujud (“entitas”) atau benda yang sesuai dengan definisi itu banyak sekali, misalnya saja mobil, jam, lembaga pemerintah, dan banyak lagi” (Tatang M. Amirin. 2008 : 4).

2.1.2 Pengertian Informasi

Menurut laudon, informasi adalah data yang telah diolah menjadi sebuah bentuk yang berguna bagi penerima dan sangat berarti dalam tindakan mengambil sebuah keputusan (laudon:2012:15).

Didalam bukunya gupta menjelaskan informasi adalah data yang telah diambil, diolah atau digunakan untuk tujuan informasi semata yang mana bisa diatur dan di presentasikan sehingga pembuat keputusan dapat mengambil tindakan secara langsung (gupta:2011:26)

2.1.3 Pengertian Sistem Informasi

Laudon mendefinisikan sistem informasi merupakan perangkat komponen yang saling terkait yang mana mengumpulkan, memproses, menyimpan, dan mendistribusikan informasi untuk mendukung pengambilan keputusan dan kontrol dalam sebuah organisasi. Disamping mendukung pengambilan keputusan, koordinasi, dan kontrol, sistem informasi juga dapat membantu manajer dan pekerja menganalisis masalah, memvisualisasikan subyek yang kompleks, menciptakan produk baru (laudon:2012:15)

2.2 Sistem Pendukung Keputusan

Sistem Pendukung Keputusan definisi SPK secara sederhana adalah sebuah sistem yang digunakan sebagai alat bantu menyelesaikan masalah untuk membantu pengambil keputusan (manajer) dalam menentukan keputusan tetapi tidak untuk menggantikan kapasitas manajer hanya memberikan pertimbangan. SPK ditujukan untuk keputusan-keputusan yang memerlukan penilaian atau pada keputusan-keputusan yang sama sekali tidak dapat didukung oleh algoritma (Turban, 2011).

Sistem pendukung keputusan merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan dan manipulasi data. Sistem itu digunakan untuk membantu pengambilan keputusan dalam situasi yang semi terstruktur dan

situasi tidak terstruktur, dimana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat.(Kusrini, 2007)

2.2.1 Tujuan Sistem Pendukung Keputusan

Tujuan dari sistem pendukung keputusan adalah (Turban, Efraim dan Jay Aronson. 2005):

1. Membantu manajer dalam pengambilan keputusan atas masalah semi-terstruktur.
2. Memberikan dukungan atas pertimbangan manajemen dan bukannya dimaksudkan untuk menggantikan fungsi manajer.
3. Meningkatkan efektivitas keputusan yang diambil manajer lebih dari pada perbaikan efisiensinya.
4. Kecepatan komputasi. Komputer memungkinkan para pengambil keputusan untuk melakukan banyak komputasi secara cepat dengan biaya yang rendah.
5. Peningkatan produktivitas. Membangun satu kelompok pengambil keputusan, terutama para pakar, bisa sangat mahal.
6. Dukungan kualitas. Komputer bisa meningkatkan kualitas keputusan yang dibuat.
7. Berdaya saing. Manajemen dan pemberdayaan sumber daya perusahaan.
8. Mengatasi keterbatasan kognitif dalam pemrosesan dan penyimpanan.

2.2.2 Komponen dalam Sistem Pendukung Keputusan

Menurut Kusrini [2007:25] sistem pendukung pengambilan keputusan terdiri dari

beberapa komponen atau subsistem yaitu :

1. Subsistem Manajemen Data

Subsistem manajemen data memasukan satu database yang berisi data yang relevan untuk situasi yang dikelola oleh perangkat lunak yang disebut sistem manajemen database.

2. Subsistem Manajemen Model

Subsistem manajemen model, yaitu sebuah paket perangkat lunak yang berisi model-model finansial, statistik, management science, atau model kuantitatif yang lain yang menyediakan kemampuan analisis sistem dan management software yang terkait.

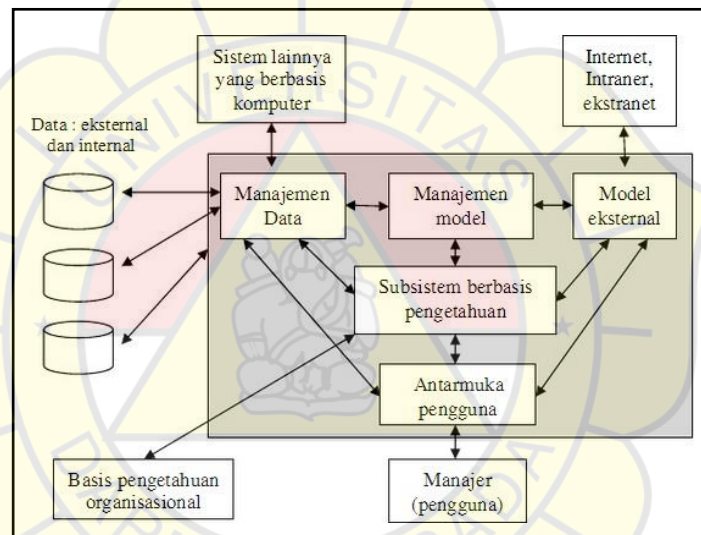
3. Subsistem Antarmuka Pengguna

Pengguna berkomunikasi dengan memerintahkan sistem pendukung keputusan melalui subsistem tersebut, pengguna adalah bagian yang dipertimbangkan dari sistem. Subsistem ini tempat komunikasi antara pengguna dan sistem pendukung keputusan serta tempat pengguna memberikan perintah kepada sistem pendukung keputusan.

4. Subsistem Manajemen Berbasis Pengetahuan

Subsistem manajemen pengetahuan (knowledge) yaitu subsistem yang mampu mendukung subsistem yang lain atau berlaku sebagai sebuah komponen yang berdiri sendiri (independen) .

Arsitektur sistem pendukung keputusan dan komponen yang ditunjukkan pada gambar 2.1 memberikan pemahaman mendasar mengenai struktur umum suatu sistem pendukung keputusan atau DSS.



Gambar 2.1 Arsitektur DSS

Sumber : Buku Decision Support Systems and Intelligent Systems oleh Turban

Berdasarkan tingkat dukungannya, sistem pendukung keputusan dibagi menjadi 6, yaitu:

1. *Retrieve Information Elements*

Inilah dukungan terendah yang bisa diberikan oleh sistem pendukung keputusan, yakni berupa akses selektif terhadap informasi.

2. *Analyze Entire File*

Dalam tahapan ini, para manajer diberi akses untuk melihat dan menganalisis file secara lengkap.

3. *Prepare Reports from Multiple Files*

Dukungan seperti ini cenderung dibutuhkan mengingat manajer berhubungan dengan banyak aktivitas dalam suatu momen tertentu.

4. *Estimate Decision Consequences*

Dalam tahapan ini, manajer dimungkinkan untuk melihat dampak dari setiap keputusan yang diambil.

5. *Propose Decision*

Dukungan di tahapan ini sedikit lebih maju lagi. Suatu alternative keputusan bisa disodorkan ke hadapan manajer untuk dipertimbangkan.

6. *Make Decision*

Ini adalah jenis dukungan yang sangat diharapkan dari sistem pendukung keputusan. Tahapan ini akan memberikan sebuah keputusan yang tinggal menunggu legitimasi dari manajer untuk dijalankan.

2.3 Metode *Benefit Cost Ratio* (BCR)

Metode *Benefit Cost Ratio* (BCR) merupakan metode yang digunakan untuk mengetahui besaran keuntungan/ kerugian serta kelayakan suatu proyek. Metode BCR ini memberikan penekanan terhadap nilai perbandingan antara aspek manfaat (*benefit*) yang akan diperoleh dengan aspek biaya dan kerugian yang akan ditanggung (*cost*). (Giatman, 2006)

Secara umum, metode analisis BCR dapat dijelaskan sebagai berikut :

$$\text{BCR} = \frac{\sum \text{Benefit}}{\sum \text{Cost}}$$

Kriteria untuk mengetahui apakah suatu rencana proyek layak atau tidak secara ekonomis dengan metode BCR adalah :

- Jika $\text{BCR} \geq 1$ Proyek layak (feasible)
- Jika $\text{BCR} < 1$ Proyek tidak layak (unfeasible)

2.3.1 Cara Kerja Metode *Benefit Cost Ratio* (BCR)

Benefit and Cost Ratio adalah salah satu konsep yang bisa digunakan untuk menentukan kelayakan dari sebuah proyek. Pada umumnya *Benefit and Cost Ratio* dimanfaatkan di dalam menentukan kelayakan dari sebuah proyek yang berkaitan dengan kepentingan masyarakat umum.

B/C Ratio (*Benefit Cost Ratio*) merupakan suatu ukuran perbandingan antara pendapatan dengan Total Biaya produksi ($\text{Cost} = C$). B yaitu Benefit, kemudian untuk C berarti cost. Dalam batasan besaran nilai *Benefit Cost Ratio* dipakai sebagai

alat di dalam mengetahui apakah suatu usaha tersebut menguntungkan atau suatu usaha tersebut tidak menguntungkan. Berikut ini rumus b/c ratio yang bisa anda gunakan untuk menghitung keuntungan dari suatu usaha.

2.4 Pengertian Profit Margin

Pengertian Profit Margin, dapat disimpulkan bahwa net profit margin merupakan perbandingan antara laba bersih dengan penjualan. Net profit margin dapat disebut juga sebagai ukuran keuntungan dengan membandingkan antara laba setelah bunga dan pajak dibandingkan dengan penjualan. Sehingga menurut Sartono (2008:123) rasio ini dapat dirumuskan sebagai berikut:

$$\text{Profit Margin} = (\text{Laba Usaha} / \text{Penjualan Netto}) \times 100\%$$

2.5 Data Mining

Data mining merupakan suatu proses penemuan pengetahuan yang bermanfaat dan menarik di dalam kumpulan data yang besar, tujuan utama data mining yaitu prediksi (*prediction*) dan uraian (*description*). Data mining juga mempunyai beberapa fungsi utama yaitu *classification* (klasifikasi), *regression* (regresi), *clustering* (pengelompokan), *summarization* (ringkasan), *dependency modeling* (pemodelan ketergantungan), *changeand deviation detection* (pendeteksian perubahan dan deviasi) (Nurani, Susanto, & Proboyekti, 2007).

Data mining juga dapat diartikan sebagai rangkaian kegiatan untuk menemukan pola yang menarik dari data dalam jumlah besar, data – data tersebut dapat disimpan dalam database, data *warehouse* atau penyimpanan informasi lainnya. Ilmu – ilmu yang berkaitan dengan data mining diantaranya adalah

database system, data warehouse, statistik, machine learning, information retrieval dan koputasi tingkat tinggi. Data mining juga didukung oleh ilmu lain seperti *spatial data analysis, signal processing, neural network* dan pengenalan pola (Meilani & Susanti, 2014).

Menurut Han dan Kamber alasan utama mengapa data mining diperlukan adalah karena adanya sejumlah besar data yang dapat digunakan untuk menghasilkan informasi dan *knowledge* yang berguna (Junanto, 2013).

2.6 Klasifikasi

Klasifikasi merupakan proses untuk menemukan fungsi dan model yang dapat membedakan atau menjelaskan konsep atau kelas data dengan tujuan memperkirakan kelas yang tidak diketahui dari suatu objek. Dalam proses pengklasifikasian biasa terdapat dua proses yang harus dilakukan, yaitu (Nugroho & Subanar, 2013) :

1. Proses *Training*

Pada proses ini akan digunakan data *training set* atau data sampel yang telah diketahui label – label atau atribut dari data sampel tersebut untuk membangun model.

2. Proses *Testing*

Pada proses *testing* ini dilakukan untuk mengetahui keakuratan model yang telah dibuat pada proses training maka dibangun data yang disebut dengan data testing untuk mengklasifikasi label – labelnya.

Klasifikasi merupakan penempatan objek – objek kesalah satu dari beberapa kategori yang telah ditetapkan sebelumnya. Klasifikasi sekarang ini telah banyak digunakan dalam berbagai aplikasi, sebagai contoh pendeteksian pesan email, spam berdasarkan header dan isi atau mengklasifikasikan galaksi berdasarkan bentuk – bentuknya. Pada proses klasifikasi data yang diinputkan adalah data record atau data sampel. Pada setiap *record* dikenal sebagai *instance* atau contoh yang ditentukan oleh sebuah tuple (x,y) . Dimana x adalah himpunan atribut dan y adalah atribut tertentu yang menyatakan sebagai label class (Nugroho & Subanar, 2013).

2.7 *Naïve Bayes*

Naive Bayes merupakan sebuah pengklasifikasian probalistik sederhana yang menghitung sekumpulan probabilitas dengan menjumlahkan frekuensi dan kombinasi nilai dari dataset yang diberikan. Algoritma menggunakan teorema bayes dan mengansumsikan semua atribut independen atau tidak saling ketergantungan yang diberikan oleh nilai pada variabel kelas. *Naive Bayes* juga didefinisikan sebagai pengklasifikasian dengan metode probabilitas dan statistik yang dikemukakan oleh ilmuwan inggis Thomas Bayes, yaitu memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya (Saleh, 2015).

Naive Bayes didasarkan pada asumsi penyederhanaan bahwa nilai atribut secara kondisional saling bebas jika diberikan nilai output. Dengan kata lain, diberikan nilai output, probabilitas mengamati secara bersama adalah produk dari probabilitas individu.

Keuntungan penggunaan *Naive Bayes* adalah bahwa metode ini hanya membutuhkan jumlah data pelatihan (*Training Data*) yang kecil untuk menentukan estimasi parameter yang diperlukan dalam proses pengklasifikasian. *Naive Bayes* sering bekerja jauh lebih baik dalam kebanyakan situasi dunia nyata yang kompleks dari pada yang diharapkan (Saleh, 2015).

Persamaan dari teorema Bayes dapat dilihat di bawah ini :

$$P(H|X) = \frac{P(X|H) \cdot P(H)}{P(X)}$$

X : data dengan class yang belum diketahui

H : hipotesis data menggunakan suatu class spesifik

$P(H|X)$: probabilitas hipotesis H berdasar kondisi X (posteriori probabilitas)

$P(H)$: probabilitas hipotesis H (prior probabilitas)

$P(X|H)$: probabilitas X berdasarkan kondisi pada hipotesis H

$P(X)$: probabilitas H

Untuk menjelaskan metode *Naive Bayes*, perlu diketahui bahwa proses klasifikasi memerlukan sejumlah petunjuk untuk menentukan kelas apa yang cocok bagi sampel yang di analisis tersebut. Karena itu, metode *Naive Bayes* di atas disesuaikan sebagai berikut (Saleh, 2015) :

$$P(C|F_1 \dots F_n) = \frac{P(C)P(F_1 \dots F_n|C)}{P(F_1 \dots F_n)}$$

Di mana Variabel C mempresentasikan kelas, sementara variabel F1...Fn mempresentasikan karakteristik petunjuk yang dibutuhkan untuk menentukan klasifikasi. Maka rumus tersebut menjelaskan bahwa peluang masuknya sampel karakteristik tertentu dalam kelas C (Posterior) adalah peluang munculnya kelas C (sebelum masuknya sampel tersebut, seringkali disebut prior), dikali dengan peluang kemunculan karakteristik – karakteristik sampel pada kelas C (disebut *like lihood*), dibagi dengan peluang kemunculan karakteristik – karakteristik secara global (disebut juga *evidence*). Karena itu, rumus di atas dapat pula ditulis secara sederhana sebagai berikut (Saleh, 2015):

$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}$$

Nilai *evidence* selalu tetap untuk setiap kelas pada satu sampel. Nilai dari Posterior tersebut nantinya akan dibandingkan dengan nilai – nilai posterior kelas lainnya untuk menentukan ke kelas apa suatu sampel akan diklasifikasikan.

2.8 Teori Mengenai Pemograman dan Database

2.8.1 Xampp

Xampp adalah sebuah paket kumpulan software yang terdiri dari apache, mysql, phpmyadmin, php, perl, Freetype2, dll. Xampp berfungsi untuk memudahkan instalasi lingkungan PHP, dimana biasanya lingkungan

pengembangan web memerlukan php, apache, mysql, dan phpmyadmin serta software-software yang terkait dengan pengembangan web. Dengan menggunakan xampp, kita tidak perlu menginstall aplikasi-aplikasi tersebut.

2.8.2 PHP

Menurut (Budi Raharjo, 2012) dalam Buku “Modul Pemrograman Web (HTML, PHP, & MYSQL)”. PHP adalah salah satu pemrograman skrip yang dirancang untuk membangun aplikasi web. Ketika dipanggil dari web browser, program yang ditulis dengan PHP akan di-parsing didalamweb server oleh *interpreter* PHP dan diterjemahkan dalam dokumen HTML, yang selanjutnya akan ditampilkan kembali ke web browser. Karena pemrosesan program PHP dilakukan dilingkungan web server, PHP dikatakan sebagai bahasa sisi server (*server-side*) oleh sebab itu, seperti yang telah dikemukakan sebelumnya, kode PHP tidak akan terlihat pada saat user memilih perintah “view source” pada web browser yang mereka gunakan. Selain menggunakan PHP, aplikasi web juga dapat dibangun dengan Java (JSP- *JavaServer Pages* dan Servlet), Perl, maupun ASP (*Active Server Pages*).

Syntax Program PHP adalah bahasa yang dirancang untuk mudah diletakkan dalam kode HTML. Banyak dijumpai kode PHP yang menyatu dengan kode HTML. Kode PHP diawali dengan tag `<?php` dan diakhiri dengan tag `?>`. apabila kita melakukan konfigurasi terhadap file PHP ini untuk mengizinkan pengguna tag pendek (short tag) dengan mengubah nilai `short_open_tag` menjadi on, maka tag tersebut dapat digantikan dengan `<? dan ?>`. dalam PHP, nilai default dari `short_open_tag` adalah off. Selain itu, PHP kita juga dapat menggunakan tag gaya ASP `<% dan %>`, dengan mengubah nilai `asp_tags` dalam file PHP ini menjadi

on. Berikut ini contoh kode PHP :

```
<? Php  
  
echo "Hello World";  
  
?>
```

Perintah echo di dalam PHP berguna untuk mencetak nilai, baik teks maupun numerik ke layar web browser.

2.8.3 MYSQL

Menurut (Betha Sidik, 2005) dalam buku “Mysql untuk Pengguna, Administrator, dan Pengembangan Web”, MySQL merupakan software sistem manajemen database (*Database Management System – DBMS*) yang sangat populer di kalangan pemrograman *web*, terutama dilingkungan *Linux* dengan menggunakan *script PHP* dan *Perl*. *Software database* ini kini telah tersedia juga pada platform sistem operasi *Windows (98/ME* atau pun *NT/2000/XP*).

MySQL merupakan *database* yang paling populer digunakan untuk membangun aplikasi *web* yang menggunakan *database* sebagai sumber dan pengelola datanya.

Kepopuleran *MySQL* dimungkinkan karena kemudahan untuk digunakan, cepat secara kerja *query*, dan mencukupi untuk kebutuhan *database* perusahaan-perusahaan skala menengah kecil. *MySQL* merupakan *database* yang digunakan oleh situs-situs terkemuka di *internet* untuk menyimpan datanya.

Software database MySQL kini dilepas sebagai *software* manajemen *database* yang *open source*, sebelumnya merupakan *software database* yang

shareware. *Shareware* adalah suatu *software* yang dapat didistribusikan secara komersial maka pemakai harus mempunyai lisensi dari pembuatnya. *Software open source* menjadikan *software* dapat didistribusikan secara bebas dan dapat dipergunakan untuk keperluan pribadi ataupun pun komersial, termasuk di dalamnya *source code* dari *software* tersebut.

Data *MySQL* tersedia secara bebas cuma-cuma dan boleh digunakan oleh setiap orang, dengan lisensi open source *GNU General Public License (GPL)* ataupun lisensi komersial non *GPL*. Saat ini diperkirakan lebih dari 3 juta pemakai di seluruh dunia, dengan lebih dari setengah juta *server* yang memasangnya, termasuk dalam *Yahoo!*, *MP3.com*, *Motorola*, *NASA*, *Silicon Graphics*, *HP*, *Cisco*, dan *Texas Instruments*.

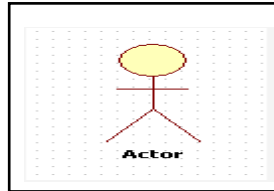
Database *MySQL*, merupakan *database* yang menjajikan sebagai alternatif pilihan *database* yang digunakan untuk sistem *database* personal atau organisasi kita. *Oracle* sebagai *database* besar telah membuat *kit* (modul) untuk memudahkan proses migrasi dari *MySQL* ke dalam *Oracle*, hal ini dapat menunjukkan bahwa *Oracle* telah memperhitungkan *database MySQL* sebagai *database* alternatif masa depan. Demikian juga dengan *database MySQL*, menunjukkan banyaknya pengguna

2.8.4 Diagram Use Case

Menurut (Jeffrey L Whitten , 2004) dalam Buku “ *Systems Analysis and Design Methods*”. Use case adalah Diagram yang menggambarkan interaksi antara sistem dengan sistem eksternal dengan pengguna. Dengan kata lain, secara grafis menggambarkan siapa yang akan menggunakan sistem dan dengan cara apa pengguna mengharapkan untuk berinteraksi dengan sistem.

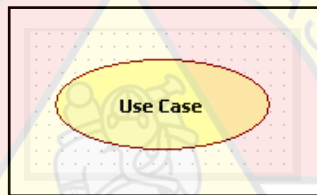
Komponen-komponen pembentuk diagram *use case* sebagai berikut :

1. Aktor : Merupakan pelaku eksternal yang berinteraksi dengan sistem untuk pertukaran informasi. Yang disimbolkan sebagai berikut :



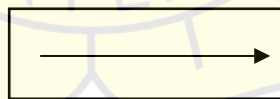
Gambar 2.2 Simbol Aktor

2. Use Case : Merupakan urutan langkah-langkah yang secara tindakan saling terkait (skenario), baik terotomatisasi maupun secara manual, untuk tujuan melengkapi satu tugas bisnis tunggal.



Gambar 2.3 Simbol *Use Case*

3. Relasi : Merupakan garis atau simbol yang menghubungkan antara *element use case* dan aktor.



Gambar 2.4 Simbol Relasi

Relasi memiliki tipe relasi yang mungkin terjadi pada diagram *use case* :

1. <<include>> yaitu kelakuan yang harus terpenuhi agar sebuah *event* dapat terjadi, dimana pada kondisi ini sebuah *use case* adalah bagian dari *use case* lainnya.
2. <<extends>> yaitu kelakuan yang hanya berjalan di bawah kondisi tertentu

seperti menggerakkan alarm.

2.8.5 Diagram *Sequence*


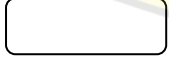
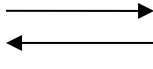

Sequence diagram secara grafis menggambarkan bagaimana *object* berinteraksi dengan satu sama lain melalui pesan pada eksekusi sebuah usecase atau operasi. Diagram ini mengilustrasikan bagaimana pesan terkirim dan diterima di antara *object* dan *sequence* (ruang waktu) (Whitten,2008). Adapun simbol simbol dari *sequence* diagram terdapat pada daftar simbol.

2.8.6 Diagram *Activity*

Menurut (Jeffry L Whitten , 2004) dalam Buku “ *Systems Analysis and Design Methods*”. Diagram *activity* merupakan sebuah diagram yang dapat digunakan untuk menggambarkan secara grafis aliran proses bisnis, langkah-langkah sebuah *use case* atau logika *behaviour* (metode) *object*.

Komponen-komponen pembentuk diagram *activity* sebagai berikut ;

Tabel 2.1 Pembentuk Diagram *Activity* (Jeffry L Whitten , 2004)

NO	SIMBOL	KETERANGAN
1		Menggambarkan awal sebuah proses
2		Menggambarkan sebuah kegiatan atau tugas yang perlu dilakukan
3		Menggambarkan sasaran yang mengawali kegiatan
4		Sebuah bar sinkronisasi. Simbol ini memperbolehkan anda untuk meggambarkan

		kegiatan yang dapat muncul secara paralel
5	[]	Menggambarkan sebuah sasaran yang merupakan sebuah hasil dari kegiatan keputusan
6	◇	Menggambarkan sebuah kegiatan keputusan / tindakan yang harus diambil pada kondisi tertentu
7	●	Menggambarkan akhir dari sebuah proses

2.8.7 Database (Basis Data)

Menurut junindar, basis data merupakan kumpulan data yang saling berhubungan satu dengan lainnya yang tersimpan di perangkat keras komputer dan diperlukan suatu perangkat lunak memanipulasi basis data tersebut (junindar:2008:9). Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. (rossa:2014:43).

2.9 Software Engineering Paradigm dan Waterfall Model

Software Engineering Paradigma seringkali disebut sebagai proses model, dimana proses model untuk *software engineering* dipilih berdasarkan sifat alamiah dari proyek dan aplikasi, metode, dan *tool* yang digunakan, serta kontrol dan *deliverable* yang diperlukan. *Software process model* merupakan strategi yang mencakup *process*, *method*, dan *tool* yang digunakan untuk memecahkan permasalahan aktual, yang biasa dilakukan oleh seorang *software engineer* atau *team engineer*.

(Pressman, 2001, p26) Seorang pelanggan biasanya mengalami suatu masalah dan dia menginginkan agar masalah tersebut bisa diselesaikan dengan bantuan suatu aplikasi. Oleh sebab itu, si pelanggan terutama yang mengerti teknik informatika tentunya akan memberikan sejumlah *requirement* yang mendasar, termasuk *input*, proses dan *output* yang diharapkan dari aplikasi tersebut. Dalam situasi ini, waterfall model adalah pilihan terbaik. (Pressman, 2001, p30) Metode perancangan sistem yang digunakan dalam pembuatan sistem pendukung keputusan penialian proyek ini adalah *The Classic Life Cycle* atau yang sering disebut dengan *waterfall* model (Pressman, 1997, pp31-32). Tahapan-tahapan yang dilakukan *waterfall* model ini adalah :

1. Analisis dan Perancangan Sistem (*Sistem Engineering and Analysis*) Perangkat lunak merupakan bagian dari suatu sistem yang lebih besar sehingga langkah pertama yang harus dilakukan adalah menetapkan kebutuhan untuk semua elemen sistem. Hal ini penting karena perangkat lunak harus berinteraksi dengan elemen-elemen lainnya seperti perangkat keras yang digunakan, pengguna, dan *database*.
2. Analisis Kebutuhan Perangkat Lunak (*Software Requirement Analysis*) Proses pengumpulan kebutuhan dilakukan secara intensif dan terfokus, khususnya pada perangkat lunaknya. Untuk mengerti sistem yang akan dibangun, seorang pembuat sistem harus memahami informasi yang dibutuhkan oleh perangkat lunak itu nantinya, fungsi-fungsi, *performace*, dan *interface* yang akan digunakan.
3. Perancangan (*Design*) Perancangan dipusatkan pada empat atribut program yaitu struktur data, arsitektur perangkat lunak, perincian prosedur, dan

karakteristik user *interfacenya*. Proses perancangan ini menterjemahkan kebutuhan sistem menjadi representasi perangkat lunak yang dapat diperkirakan kualitasnya sebelum proses pengkodean dimulai. Hasil perancangan kemudian didokumentasikan dan menjadi bagian dari konfigurasi perangkat lunak.

4. Pengkodean (*Coding*) Pada tahap ini rancangan sistem akan diimplementasikan ke dalam bahasa mesin yang dapat dijalankan oleh komputer.
5. Pengujian (*Testing*) Setelah proses pengkodean selesai, dilakukan pengujian sampai semua perintah selesai diuji. Pengujian ini bertujuan untuk menemukan kesalahan dan memastikan keluaran yang dihasilkan sesuai dengan apa yang diharapkan.
6. Pemeliharaan (*Maintenace*) Pada tahap ini pemeliharaan dengan mengadakan modifikasi atau perubahan perangkat lunak setelah perangkat lunak tersebut diserahkan kepada pemakai. Perubahan ini akan terjadi bila ditemukan adanya kesalahan di dalam sistem, atau dikarenakan perubahan lingkungan perangkat lunaknya, seperti perubahan perangkat keras atau sistem operasi yang digunakan, atau untuk meningkatkan fungsi dan kinerja dari perangkat lunak itu sendiri.