

BAB II

TINJAUAN PUSTAKA

II.1 Teori Sistem

Menurut kusrini dan Andri koniyo (2007:5) kata sistem mempunyai beberapa pengertian, tergantung dari sudut pandang mana kata tersebut didefinisikan. Secara garis besar ada dua kelompok pendekatan, yaitu :

1. Pendekatan sistem yang lebih menekankan pada elemen-elemen atau kelompoknya, yang dalam hal ini system itu didefinisikan sebagai “ suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu aturan tertentu”.
2. Pendekatan sistem sebagai jaringan kerja dari prosedur yang lebih menekankan urutan proses di dalam system. Prosedur (procedure) didefinisikan oleh Richard F. Neushl sebagai “urutan operasi kerja (tulismenulis), yang biasanya melibatkan beberapa orang di dalam satu atau lebih departemen, yang diterapkan untuk menjamin penanganan yang seragam dari transaksi bisnis yang terjadi”.

Pendekatan sistem yang lebih menekankan pada elemen-elemen atau komponennya mendefinisikan sistem sebagai “sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan”. Dengandemikian di dalam suatu system, komponen-komponen ini tidak dapat

berdirisendiri, tetapi sebaliknya, saling berhubungan hingga membentuk satu kesatuan sehingga tujuan system itu dapat tercapa

II.1.1 Karakteristik Sistem

Menurut kusrini dan Andri koniyo (2007 : 6) Sistem mempunyai beberapa karakteristik atau sifat-sifat tertentu, antara lain :

1. sistem (*Component*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang saling bekerja sama membentuk suatu komponen sistem atau bagian-bagian dari sistem.

2. Batasan system (*Boundary*)

Merupakan daerah yang membatasi suatu sistem dengan sistem yang lain atau dengan lingkungan kerjanya.

3. Subsistem

Bagian-bagian dari sistem yang beraktivitas dan berinteraksi satu sama lain untuk mencapai tujuan dengan sasarannya masing-masing.

4. Lingkungan Luar sistem (*Evironment*)

Suatu sistem yang ada di luar dari batas sistem yang dipengaruhi oleh operasi sistem.

5. Penghubung Sistem (*Interface*)

Media penghubung antara suatu subsistem dengan subsistem lain. Adanya penghubung ini memungkinkan berbagai sumber daya mengalir dari suatu subsistem ke subsistem lainnya.

6. Masukan sistem (*Input*)

Energi yang masuk ke dalam sistem, berupa perawatan dan sinyal. Masukan perawatan adalah energi yang dimasukkan supaya sistem tersebut dapat berinteraksi

7. Keluaran sistem (*Output*)

Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

8. Pengolahan Sistem (*Process*)

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan mengubah masukan menjadi keluaran.

9. Sasaran Sistem (*Object*)

Tujuan yang ingin dicapai oleh sistem, akan dikatakan berhasil apabila mengenai sasaran atau tujuan.

II.1.2 Klasifikasi Sitem

Menurut kusrini dan Andri koniyo (2009:7) suatu sistem dapat diklasifikasikan menjadi seperti berikut :

1. Sistem abstrak dan sistem fisik

Sistem abstrak adalah suatu system yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, sedangkan system fisik adalah sistem yang ada secara fisik.

2. Sistem alamiah dan sistem buatan manusia

Sistem alamiah adalah sistem yang terjadi melalui proses alam sedangkan sistem buatan manusia adalah sistem yang dirancang oleh manusia.

3. Sistem tertentu dan sistem tak tentu

Sistem tertentu adalah suatu sistem yang operasinya dapat diprediksi secara tepat sedangkan sistem tak tertentu adalah sistem dengan perilaku ke depan yang tidak dapat diprediksi.

4. Sistem tertutup dan sistem terbuka

Sistem tertutup adalah sistem yang tidak terpengaruh oleh lingkungan luar otomatis, sedangkan sistem terbuka adalah sistem yang berhubungan dan terpengaruh oleh lingkungan luar.

II.2 Sistem Pendukung Keputusan

Sistem pendukung keputusan (*decision support*) adalah salah satu cara mengorganisir informasi (melibatkan penggunaan basis data) yang dimaksudkan untuk digunakan dalam membuat keputusan. SPK dirancang untuk pendekatan menyelesaikan masalah para pembuat keputusan dan kebutuhan-kebutuhan aplikasi, tetapi tidak untuk menggantikan keputusan maupun membuat suatu keputusan untuk pengguna.

II.2.1 Pengertian Sistem Pendukung Keputusan

Beberapa definisi keputusan yang dikemukakan para ahli dijelaskan sebagai berikut :

1. Menurut Ralph C. Davis

Keputusan adalah hasil pemecahan masalah yang dihadapinya dengan tegas. Suatu keputusan merupakan jawaban yang pasti terhadap suatu pertanyaan. Keputusan harus dapat menjawab pertanyaan tentang apa yang dibicarakan dalam hubungannya dengan perencanaan. Keputusan dapat pula berupa tindakan terhadap pelaksanaan yang sangat menyimpang dari rencana semula.

2. Menurut Keen dan Scoot Morton

Sistem Pendukung Keputusan merupakan penggabungan sumber-sumber kecerdasan individu dengan kemampuan komponen untuk memperbaiki kualitas keputusan. Sistem Pendukung Keputusan juga merupakan sistem informasi berbasis komputer untuk manajemen pengambilan keputusan yang menangani masalah-masalah semi struktur

Dengan pengertian diatas dapat dijelaskan bahwa SPK bukan merupakan alat pengambilan keputusan, melainkan merupakan sistem yang membantu pengambil keputusan dengan melengkapi mereka dengan informasi dari data yang telah diolah dengan relevan dan diperlukan untuk membuat keputusan tentang suatu masalah dengan lebih cepat dan akurat. Sehingga sistem ini tidak dimaksudkan untuk menggantikan pengambilan keputusan dalam proses pembuatan keputusan.

II.2.2 Karakteristik Sistem Pendukung Keputusan

Menurut (Sprague dan Carlson, 1993), karakteristik sistem pendukung keputusan yaitu :

- 1. Sistem yang berbasis komputer**
- 2. Dipergunakan untuk membantu para pengambil keputusan**
- 3. Untuk memecahkan masalah-masalah rumit yang “mustahil” dilakukan dengan kalkulasi manual**
- 4. Simulasi yang interaktif**
- 5. Data dan model analisis sebagai komponen utama**

II.2.3 Ciri – Ciri Sistem Pendukung Keputusan

Menurut Dicky Nofriansyah (2014 : 2), 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.2.4. Fase Dalam Sistem Pendukung Keputusan

Menurut Dicky Nofriansyah (2014: 2), Tiga fase dalam pengambil keputusan yaitu:

1. *Intelligence*

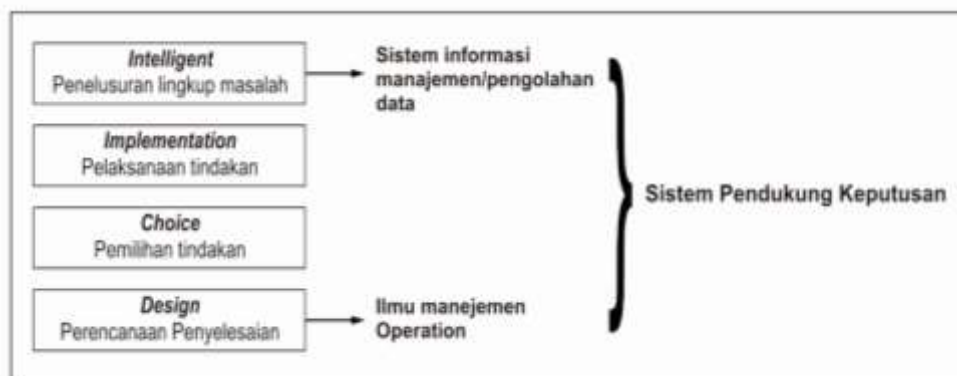
Tahap ini merupakan proses penelusuran dan pendekatan dari ruang lingkup problematika secara proses pengenalan masalah. Data masukan diperoleh dan diuji dalam rangkamengidentifikasi masalah

2. *Design*

Tahap ini merupakan proses menemukan,mengembangkan dan menganalisis alternatifindakan yang bisa dilakukan .tahapmelakukan kelayakan solusi

3. *Choice*

Pada tahap ini dilakukan proses pemilihan diantara berbagai alternatif tindakan yang akan dijalankan. Hasil pemilihan tersebut kemudian diimplementasikan kedalam proses pengambilan keputusan



Gambar II.1 Fase Proses Pengambil Keputusan
Sumber (Dicky Nofriansyah, S.Kom : 2014)

II.2.5 Komponen Sistem Pendukung Keputusan

Menurut Surbakti (2010), komponen-komponen dari SPK adalah sebagai berikut:

1. *Data Management*

Termasuk *database*, yang mengandung data yang relevan untuk berbagai situasi dan diatur oleh *software* yang disebut *Database Management System* (DBMS).

2. *Model Management*

Melibatkan model finansial, statistik, *management science*, atau berbagai model kualitatif lainnya, sehingga dapat memberikan ke sistem suatu kemampuan analitis, dan manajemen software yang dibutuhkan.

3. *Communication*

User dapat berkomunikasi dan memberikan perintah pada DSS melalui subsistem ini. Ini berarti menyediakan antarmuka.

4. *Knowledge Management*

Subsistem optional ini dapat mendukung subsistem lain atau bertindak atau bertindak sebagai komponen yang berdiri sendiri.

II.2.6 Tujuan Sistem Pendukung Keputusan

Scott Morton telah mendefenisikan tiga tujuan yang harus dicapai oleh sistem pendukung keputusan, yaitu:

1. Sistem harus dapat membantu manajer dalam membuat keputusan guna memecahkan masalah semi terstruktur.
2. Sistem harus dapat mendukung manajer, bukan mencoba menggantikannya.
3. Sistem harus dapat meningkatkan efektivitas pengambilan keputusan manajer

Tujuan-tujuan tersebut mengacu pada tiga prinsip dasar sistem pendukung keputusan yaitu :

1. Struktur masalah : untuk masalah yang terstruktur, penyelesaian dapat dilakukan dengan menggunakan rumus-rumus yang sesuai, sedangkan untuk masalah terstruktur tidak dapat dikomputerisasi. Sementara itu, sistem pendukung keputusan dikembangkan khususnya untuk menyelesaikan masalah yang semi-terstruktur.
2. Dukungan keputusan : sistem pendukung keputusan tidak dimaksudkan untuk menggantikan manajer, karena komputer berada di bagian terstruktur, sementara manajer berada dibagian tak terstruktur untuk memberikan penilaian dan melakukan analisis. Manajer dan komputer bekerja sama sebagai sebuah tim pemecah masalah semi terstruktur.
3. Efektivitas keputusan : tujuan utama dari sistem pendukung keputusan bukanlah mempersingkat waktu pengambilan keputusan, tetapi agar keputusan yang dihasilakn dapat lebih baik.

II.3. Prestasi Belajar

Menurut Sumadi Suryabrata (2007: 297), prestasi dapat pula didefinisikan sebagai berikut : “nilai merupakan perumusan terakhir yang dapat diberikan oleh guru mengenai kemajuan/prestasi belajar siswa selama masa tertentu”. Jadi, prestasi adalah hasil usaha siswa selama masa tertentu dalam melakukan kegiatan.

Prestasi belajar mempunyai hubungan erat dengan kegiatan belajar, banyak faktor yang mempengaruhi prestasi belajar baik yang berasal dari dalam individu itu sendiri maupun faktor yang berasal dari luar individu. Menurut Ngalim Purwanto (2010: 107), faktor-faktor yang mempengaruhi prestasi belajar adalah :

1. Faktor internal, yaitu faktor yang berasal dari dalam diri siswa yang meliputi faktor fisiologis dan faktor psikologis. Faktor fisiologis ini menyangkut kondisi jasmani/kondisi fisik siswa selama belajar. Sedangkan faktor psikologis meliputi aspek :
 - a. Minat belajar siswa merupakan minat belajar yang besar cenderung menghasilkan prestasi yang tinggi, sebaliknya minat belajar yang kurang akan menghasilkan prestasi belajar yang rendah
 - b. Kecerdasan/intelegensi. Seseorang yang memiliki intelegensi yang baik umumnya mudah belajar dan hasilnya pun cenderung baik.
 - c. Motivasi belajar
 - d. Bakat siswa
 - e. Kemampuan kognitif siswa
 - f. Sikap siswa terhadap mata pelajaran

2. Faktor eksternal, yaitu faktor yang berasal dari luar diri siswa yang meliputi lingkungan fisik dan sosial serta instrumen yang berupa kurikulum, program, metode mengajar, guru, sarana dan fasilitas.

Untuk mengetahui keberhasilan siswa selama proses belajar mengajar maka seorang guru mengadakan suatu penilaian kepada siswa. Dengan mengadakan penilaian tersebut seorang guru akan mengetahui sejauh mana keberhasilan siswanya dalam melakukan proses belajar mengajar.

II.4 Metode SAW

Metode *simple additive weighting* sering juga dikenal dengan istilah penjumlahan terbobot. Konsep dasar metode SAW (Rully, 2014) adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif dari semua atribut, metode SAW membutuhkan proses normalisasi matrik keputusan (x) ke skala yang dapat diperbandingkan dengan semua rating alternatif yang ada. Metode *simple additive weighting* membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada.

$$R_{ij} = \frac{X_{ij}}{\text{Max } x_{ij}} \quad \text{Jika } j \text{ adalah atribut keuntungan (benefit)}$$

$$\frac{\text{Min } x_{ij}}{X_{ij}} \quad \text{Jika } j \text{ adalah atribut biaya (cost)}$$

Dimana :

r_{ij} = rating kinerja ternormalisasi

Max_{ij} = nilai maksimum dari setiap baris dan kolom

Min_{ij} = nilai minimum dari setiap baris dan kolom

X_{ij} = baris dan kolom dari matriks

Dengan r_{ij} adalah rating kinerja ternormalisasi dari alternatif A_i pada atribut C_j ; $i=1,2,\dots,m$ dan $j = 1,2,\dots,n$. Nilai preferensi untuk setiap alternatif (V_i) diberikan sebagai:

$$V_i = \sum_{j=1}^n W_j r_{ij}$$

Dimana :

V_i = Nilai akhir dari alternatif

W_j = Bobot yang telah ditentukan

r_{ij} = Normalisasi matriks

Nilai V_i yang lebih besar mengindikasikan bahwa alternative A_i lebih terpilih.

II.5 Unified Modeling Language

Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

II.5.1 Pengenalan UML (*Unified Modeling Language*)

UML sebagai sebuah bahasa yang memberikan *vocabulary* dan tatanan penulisan kata-kata dalam '*MS Word*' untuk kegunaan komunikasi. Sebuah bahasa model adalah sebuah bahasa yang mempunyai *vocabulary* dan konsep tatanan / aturan penulisan serta secara fisik mempresentasikan dari sebuah sistem. Seperti halnya

UML adalah sebuah bahasa standard untuk pengembangan sebuah software yang dapat menyampaikan bagaimana membuat dan membentuk model-model, tetapi tidak menyampaikan apa dan kapan model yang seharusnya dibuat yang merupakan salah satu proses implementasi pengembangan software.

UML tidak hanya merupakan sebuah bahasa pemrograman visual saja, namun juga dapat secara langsung dihubungkan ke berbagai bahasa pemrograman, atau bahkan dihubungkan secara langsung ke dalam sebuah object-oriented database. Begitu juga mengenai pendokumentasian dapat dilakukan seperti; *requirements*, *arsitektur*, *design*, *source code*, *project plan*, *tests*, dan *prototypes*.

II.5.2 Notasi dan Artifak Dalam UML

Menurut Julius Hermawan (2010:13), UML menyediakan beberapa notasi dan artifak standart yang biasa digunakan sebagai alat komunikasi bagi para proses analisis dan desain. Artifak didalam UML didefenisikan sebagai informasi dalam berbagai bentuk yang digunakan atau yang dihasilkan dalam proses pengembangan perangkat lunak

1. Aktor

Aktor menggambarkan segala pengguna software aplikasi (*user*). Aktor memberikan suatu gambaran jelas tentang apa yang harus dikerjakan pada software aplikasi. Sebagai contoh sebuah aktor dapat memberikan input kedalam dan menerima informasi dari software aplikasi, perlu dicatat bahwa sebuah aktor berinteraksi dengan *usecase*, tetapi tidak memiliki kontrol atas *usecase*. Sebuah aktor mungkin seorang manusia, satu *device*, hardware atau sistem informasi lainnya.



Gambar II.2 Notasi Aktor

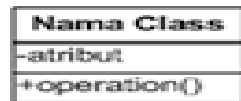
(Sumber : Adi Nugroho :2010)

2. Class

Class merupakan pembentuk utama dari sistem berorientasi obyek, karena *class* menunjukkan kumpulan obyek yang memiliki atribut dan operasi yang sama. *Class* digunakan untuk mengimplementasikan interface. *Class* digunakan untuk mengabstraksikan elemen-elemen dari sistem yang sedang dibangun. *Class* bisa merepresentasikan baik perangkat lunak maupun perangkat keras, baik konsep maupun benda nyata.

Atribut digunakan untuk menyimpan informasi. Nama atribut menggunakan kata benda yang bisa dengan jelas merepresentasikan informasi yang tersimpan

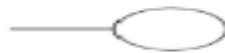
didalamnya. Operasi menunjukkan sesuatu yang bisa dilakukan oleh obyek dan menggunakan kata kerja.



Gambar II.3 Notasi Class
(Sumber : Adi Nugroho :2010)

3. *Interface*

Interface merupakan kumpulan operasi tanpa implementasi dari suatu *class*. Implementasi operasi dalam *interface* dijabarkan oleh operasi didalam *class*. Oleh karena itu keberadaan *interface* selalu disertai oleh *class* yang mengimplementasikan operasinya. *Interface* ini merupakan salah satu cara mewujudkan prinsip enkapsulasi dalam obyek.



Gambar II.4 Notasi Interface
(Sumber : Adi Nugroho :2010)

4. *Usecase*

Usecase menjelaskan uruta kegiatan yang dilakukan aktor dan sistem untuk mencapai suatu tujuan tertentu. Walaupun menjelaskan kegiatan namaun *usecase* hanya menjelaskan apa yang dilakukan aktor dan sistem, bukan bagaimana aktor dan sistem melakukan kegiatan tersebut



Gambar II.5 Notasi Usecase
(Sumber : Adi Nugroho :2010)

5. *Interaction*

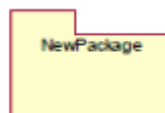
Interaction digunakan untuk menunjukkan baik aliran pesan atau informasi antar obyek maupun hubungan antar obyek. Biasanya *interaction* ini dilengkapi juga dengan teks bernama operation signature yang tersusun dari nama operasi, parameter yang dikirim dan tipe parameter yang dikembalikan.



Gambar II.6 Notasi *Interaction*
(Sumber : Adi Nugroho :2010)

6. *Package*

Package adalah kontainer atau wadah konseptual yang digunakan untuk mengelompokkan elemen-elemen dari sistem yang sedang dibangun, sehingga bisa dibuat jadi model yang lebih sederhana. Tujuannya adalah untuk mempermudah penglihatan dari model yang sedang dibangun.



Gambar II.7 Notasi *Package*
(Sumber : Adi Nugroho :2010)

7. *Note*

Note digunakan untuk memberikan keterangan dan komentar tambahan dari suatu elemen sehingga bisa langsung terlampir dalam model. *Note* ini bisa ditempelkan ke semua elemen notasi yang lain.



Gambar II.8 Notasi *Note*
(Sumber : Adi Nugroho :2010)

8. *Depedency*

Depedency merupakan relasi yang menunjukkan bahwa perubahan pada salah satu elemen memberi pengaruh pada elemen lain. Elemen yang adapada pada bagian tanda panah adalah elemen yang tergantung pada elemen yang ada dibagian tanpa tanda panah



Gambar II.9 Notasi *Defedency*
(Sumber : Adi Nugroho :2010)

II.6. *Use Case Diagram*

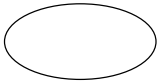
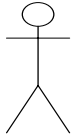
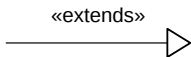
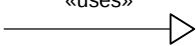
Use case atau diagram *use case* merupakan pemodelan untuk kelakuan (*behaviour*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat.

Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

- a. Aktor merupakan 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 aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.

- b. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. Berikut adalah simbol-simbol yang ada pada diagram *use case* :

Tabel II.1. Simbol-simbol Diagram *Use Case*

Simbol	Keterangan
<i>Use Case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>
Aktor / 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 aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama aktor
Asosiasi / Association 	Komuniikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
Ekstensi / <i>Extends</i> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan
Generalisasi / <i>Generalization</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
Menggunakan / <i>Include</i> / <i>Uses</i> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini

(Sumber : Rosa A.S., M. Shalahuddin, 2014 : 156-158)

II.7. Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut adalah simbol-simbol yang ada pada diagram aktivitas :

Tabel II.2. Simbol-simbol Diagram Aktivitas

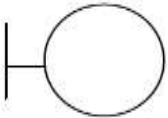
Simbol	Keterangan
Status Awal 	Status awal aktivitas sistem sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan / <i>Decision</i> 	<i>Decision</i> , atau pilihan untuk mengambil keputusan.
Penggabungan / <i>Join</i> 	Arah tanda panah alur proses.
Status Akhir 	Titik akhir atau akhir dari aktivitas.

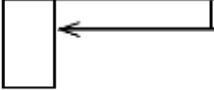
(Sumber : Rosa A.S., M. Shalahuddin, 2014 : 162)

II.8. Sequence Diagram

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*. Berikut adalah simbol-simbol yang ada pada diagram sekuen :

Tabel II.3. Simbol Sequence Diagram

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

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

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar, 215 : 95)

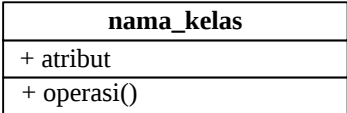
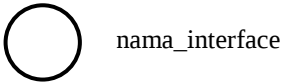
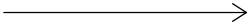
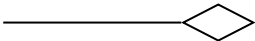
II.9. Class Diagram

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.

- Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas
- Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas

Berikut adalah simbol-simbol yang ada pada diagram kelas :

Tabel II.4. Simbol-Simbol Diagram Kelas

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Antarmuka / <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi / <i>association</i> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Asosiasi berarah / <i>directed association</i> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Generalisasi / <i>Generalization</i> 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus)
Kebergantungan / <i>dependency</i> 	Relasi antarkelas dengan makna kebergantungan antarkelas
Agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua bagian (<i>whole-part</i>)

(Sumber : Rosa A.S., M. Shalahuddin, 2014 : 141-147)

II.10. Pengertian Basis Data (*Database*)

Beberapa pengertian *database* menurut para ahli:

1. Menurut Kustiyaningsih (2011:146), “*Database* adalah Struktur penyimpanan data. Untuk menambah, mengakses dan memperoses data yang disimpan dalam sebuah *database* komputer, diperlukan sistem manajemen *database* seperti *MYSQL Server*”.
2. Menurut Anhar (2010:45), “*Database* adalah sekumpulan tabel-tabel yang berisi data dan merupakan kumpulan dari *field* atau kolom. Struktur file yang menyusun sebuah *database* adalah *Data Record* dan *Field*”.
3. Menurut Martono, (2009:32), “*Database* adalah sebagai kumpulan data dari penempatan tenaga kerja yang saling terkait dan mempengaruhi sesuai dengan tingkat kepentingannya sehingga data tersebut terintegrasi dan *independence*

Berdasarkan beberapa pendapat para ahli yang dikemukakan di atas dapat 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.

II.10.1 Tujuan Basis Data

Menurut Kusrini (2010:2), basis data bertujuan untuk mengatur data sehingga diperoleh kemudahan, ketepatan, dan kecepatan dalam pengambilan keputusan. Untuk mencapai tujuan, syarat basis data yang baik adalah sebagai berikut :

1. Tidak adanya redundansi dan inkonsistensi

Redudansi terjadi jika suatu informasi disimpan di beberapa tempat, misalnya kita punya data siswa yang memuat no.induk, nama, alamat dan atribut lainnya, sementara kita punya data lain tentang penilaian siswa yang isinya terdapat no.induk, nama, kehadiran, mata pelajaran dan nilai. Pada kedua data tersebut kita temukan atribut nim dan nama.

2. Kesulitan Pengaksesan Data

Basis data memiliki fasilitas untuk melakukan pencarian informasi dengan menggunakan query ataupun dari tool yang melibatkan tabelnya. Dengan fasilitas ini, bisa segera langsung melihat data dari software DBMS

3. *Multiple User*

Basis data memungkinkan penggunaan data secara bersama-sama oleh banyak pengguna pada saat yang bersamaan atau pada saat yang berbeda. Dengan melakukan basis data pada bagian server yang bisa diakses dari banyak *client*, sudah menyediakan akses kesemua pengguna dari komputer client ke sumber informasi yaitu basis data.

II.10.2 Manfaat/Kelebihan Basis Data

Menurut Kusri (2010:5), banyak manfaat yang diperoleh menggunakan basis data, manfaat/kelebihan basis data dan kelebihan basis data diantaranya adalah :

1. Kecepatan dan Kemudahan

Dengan menggunakan basis data pengambilan keputusan dapat dilakukan dengan cepat dan mudah. Basis data memiliki kemampuan dalam

mengelompokkan, mengurutkan bahkan perhitungan dengan matematika. Dengan perancangan yang benar maka penyajian informasi dapat dilakukan dengan cepat dan mudah

2. Kebersamaan Pemakai

Sebuah basis data dapat digunakan oleh banyak user dan banyak aplikasi. Untuk data yang diperlukan oleh banyak bagian/orang, tidak perlu dilakukan pencatatan dimasing-masing bagian/orang, tetapi cukup dengan satu basis data untuk dipakai bersama

3. Pemusatan *Control* Data

Cukup satu data untuk keperluan, pengontrolan terhadap data juga cukup dilakukan disatu tempat saja

4. Efisiensi Ruang Penyimpanan

Dengan pemakaian bersama, tidak perlu menyediakan tempat penyimpanan diberbagai tempat tetapi cukup satu saja, sehingga ini dapat menghemat ruang penyimpanan yang dimiliki oleh sebuah organisasi

5. Keakuratan (*Accuracy*)

Penerapan secara tepat acuan tipe data, domain data, keunikan data, hubungan antar data, dan lainnya, dapat menekankan ketidak akuratan dalam pemasukan/penyimpanan data

6. Ketersediaan (*Availability*)

Dengan basis data, semua data dapat di backup memilih data mana yang masih diperlukan yang perlu disimpan ditempat lain. Hal ini mengingat

pertumbuhan transaksi sebuah organisasi dari lain waktu kewaktu membutuhkan penyimpanan yang semakin besar

7. Keamanan (*Security*)

Kebanyakan DBMS dilengkapi dengan fasilitas manajemen pengguna. Pengguna diberi hak akses yang berbeda-beda sesuai dengan kepentingan dan posisinya. basis data bisa diberikan password untuk membatasi orang yang diaksesnya

8. Kemudahan Dalam Pembuatan Program Aplikasi Baru

Penggunaan basis data merupakan bagian dari perkembangan teknologi. Dengan adanya basis data pembuatan aplikasi bisa memanfaatkan kemampuan dari DBMS, sehingga membuat aplikasi tidak perlu mengurus penyimpanan data, tetapi cukup mengatur interface untuk pengguna

9. Pemakaian Secara Langsung

Basis data memiliki fasilitas yang lengkap untuk melihat datanya secara langsung dengan tools yang disediakan DBMS

10. Kebebasan Data

Perubahan dapat dilakukan pada level DBMS tanpa harus membongkar kembali aplikasinya

11. *User View*

Basis data menyediakan pandangan yang berbeda-beda

II.10.3. Operasi Dasar Database

Menurut Kusrini (2010:9), beberapa operasi dasar basis data yaitu :

1. Pembuatan basis data
2. Penghapusan basis data
3. Pembuatan file/tabel
4. Penghapusan file/tabel
5. Pengubahan tabel
6. Penambahan/pengisian
7. Pengambilan data
8. Penghapusan data

II.10.4 Pemodelan Basis Data

Model basis data menyatakan hubungan antar rekaman yang tersimpan dalam basis data. Beberapa literatur menggunakan istilah struktur data logis untuk menyatakan keadaan ini. Menurut Abdul Kadir (2008:22), model dasar yang paling umum ada 3 macam yaitu :

1. Model Hirarkis

Model hirarkis biasanya disebut model pohon karena menyerupai pohon yang dibalik. Model ini menggunakan pola hubungan orang tua dengan anak. Setiap simpulan menyatakan sekumpulan medan. Contoh produk DBMS yang menggunakan Model Hirarkis adalah IMS (*Information Management Sytem*), yang dikembangkan oleh 2 perusahaan yaitu *IBM* dan *Rockell International Corporation*.

2. Model Jaringan

Model jaringan distandarisasikan pada tahun 1971 oleh *Data Base Task Group/DBTG*. Model ini juga disebut model CODASYL (*Confrence on Data Systems Languages*), karena DBTG adalah bagian dari CODASYL.

Model ini menyerupai model hirarkis dengan perbedaan suatu simpul anak biasa memiliki lebih dari satu orang tua. Oleh karena sifatnya yang demikian, model ini biasa menyatakan hubungan 1:1 (satu orang tua punya satu anak) dan 1:M (satu orang tua punya banyak anak) maupun N:M (beberapa anak bisa mempunyai beberapa orang tua).

3. Model Relasional

Model Relasional merupakan model yang paling sederhana sehingga mudah digunakan dan dipahami oleh pengguna, serta yang paling populer saat ini. Model ini menggunakan sekumpulan *table* berdimensi dua, dengan masing-masing relasi tersusun atas *tupel* atau baris dan atribut. Relasi ini dirancang sedemikian rupa sehingga dapat menghilangkan kemubaziran data dan menggunakan kunci tamu untuk berhubungan dengan relasi lain. Contoh produk DBMS yang menggunakan model relasional adalah CAIDBMS/DB, dari *Computer Associates International Inc.*

II.10.5 Normalisasi

Menurut samiaji Sarosa (2010:5), normalisasi adalah teknik yang dirancang untuk merancang tabel basisdata relasional untuk meminimalkan duplikasi data dan menghindarkan basis data tersebut anomali (keanehan).

Dengan normalisasi kita ingin mendesain database relasi yang terdiri dari tabel berikut :

1. Basis data diperlukan
2. Memiliki sedikit redudasi
3. Mengkomodosi banyak nilai untuk tipe data yang diperlukan
4. Mengefisienkan update
5. Menghindari kemungkinan kehilangan data secara tidak disengaja/tidak diketahui

Alasan utama dari normalisasi database minimal sampai dengan bentuk normal ketiga yaitu menghilangkan kemungkinan adanya “*insertion anomalies*”, “*deletion anomalies*”, dan “*update anomalies*”. Tipe-tipe kesalahan tersebut sangat mungkin terjadi pada database yang tidak normal. Jenis – Jenis Normalisasi sebagai berikut :

1. *First Normal Form (1NF)*

Untuk menjadi 1NF suatu table harus memenuhi dua syarat. Syarat pertama tidak ada kelompok data atau field yang berulang. Syarat yang kedua harus ada *primary key* (PK) atau kunci unik, atau kunci yang membedakan satu baris dengan baris yang lain dalam satu table

2. *Second Normal Form (2NF)*

Untuk menjadi suatu table harus berada dalam kondisi 1NF dan tidak memiliki *partial dependencies*. *partial dependencies* adalah suatu kondisi jika atribut *non primary key* (Non PK) tergantung sebagian tetapi bukan seluruhnya pada PK.

3. *Third Normal Form (3NF)*

Untuk menjadi bentuk Normal Ketiga maka relasi haruslah dalam bentuk Normal Kedua dan semua atribut bukan primer tidak punya hubungan yang transitif. Artinya setiap atribut bukan kunci harus bergantung hanya pada kunci primer secara menyeluruh. Contoh pada bentuk Normal kedua termasuk juga bentuk Normal Ketiga karena seluruh atribut yang ada di situ bergantung penuh pada kunci primernya.

Boyce-Codd Normal Form (BCNF)

Boyce-Codd Normal Form mempunyai paksaan yang lebih kuat dari bentuk Normal ketiga. Untuk menjadi BCNF, relasi harus dalam bentuk Normal Kesatu dan setiap atribut dipaksa bergantung pada fungsi pada atribut *super key*.

II.11. PHP

PHP merupakan singkatan dari “*Hypertext Preprocessor*”, yang merupakan sebuah bahasa scripting tingkat tinggi yang dipasang pada dokumen HTML. Sebagian besar sintaks dalam PHP mirip dengan bahasa C, Java dan Perl, namun pada PHP ada beberapa fungsi yang lebih spesifik. Sedangkan tujuan utama dari penggunaan bahasa ini adalah untuk memungkinkan perancang web yang dinamis dan dapat bekerja secara otomatis.

II.12. MySQL

MySQL adalah Relational Database Management System (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial. 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. Menurut cara pemrosesan data pada prosesor maka Microsoft mengelompokkan produk ini berdasarkan 2 jenis yaitu :

- Versi 32-bit(x86), yang biasanya digunakan untuk komputer dengan single prosesor (Pentium 4) atau lebih tepatnya prosesor 32 bit dan sistem operasi Windows XP.
- Versi 64-bit(x64), yang biasanya digunakan untuk komputer dengan lebih dari satu prosesor (Misalnya Core dual) dan system operasi 64 bit seperti Windows XP 64, Vista, dan Windows 7.

II.12.1 Kelebihan dan Keuntungan Memakai MySQL

Menurut Wahana Komputer (2010-6), dalam dunia programming ada beberapa database yang sering digunakan antara lain mysql, oracle, postgresql, microsoft sql server dan lain-lain. Ketika dibandingkan antara mysql dengan sistem manajemen database yang lain, mysql memiliki beberapa kelebihan dan

keuntungan dibandingkan database lain diantaranya adalah :

1. Banyak ahli berpendapat bahwa mysql merupakan server tercepat
2. Mysql merupakan sistem manajemen database yang open source yaitu software ini bersifat free atau bebas digunakan oleh perseorangan atau tanpa harus membeli atau membayar kepada pembuatnya
3. Mysql mempunyai performa yang tinggi dan simple
4. Database mysql mengerti bahasa sql (*structured query language*)
5. Semua *client* dapat mengakses server dalam satu waktu tanpa harus menunggu yang lain mengakses database.
6. Mysql merupakan database yang mampu menyimpan data berkapasitas besar, sampai berukuran gigabyte.
7. Mysql dapat berjalan pADA operating sistem seperti linux, windows dan lainnya.



BAB III

ANALISIS DAN DESAIN SISTEM