

BAB II

LANDASAN TEORI

II.1. Pengertian Sistem

Sistem merupakan sekumpulan elemen-elemen yang saling terintegrasi serta melaksanakan fungsinya masing-masing untuk mencapai tujuan yang telah ditetapkan. Karakteristik sistem terdiri dari :

1. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.

2. Batasan Sistem

Batasan merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang suatu kesatuan. Batasan suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.

4. Penghubung Sistem

Penghubung merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lainnya.

5. Masukan Sistem

Masukkan sistem adalah energi yang di masukkan ke dalam sistem. Masukkan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Signal input adalah energi yang diproses untuk mendapatkan keluaran.

6. Keluaran Sistem

Keluaran sistem adalah hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

7. Pengolahan Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah akan mengubah masukan menjadi keluaran.

8. Sasaran Sistem

Suatu sistem dapat mempunyai tujuan (*goal*) atau sasaran (*objective*). Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak ada gunanya. Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.

sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.(SulindawatidanFathoni : 2010 ; 1-2).

II.2. Sistem Pendukung Keputusan

Sistem pendukung keputusan merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan dan memanipulasi data. Sistem itu digunakan untuk membantu pengambilan keputusan dalam situasi yang semi terstruktur dan situasi yang tidak terstruktur dimana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat. Sistem pendukung keputusan biasanya dibangun untuk mendukung solusi atas suatu masalah atau untuk mengevaluasi suatu peluang. Sistem pendukung keputusan yang seperti itu disebut aplikasi Sistem pendukung keputusan. Aplikasi Sistem pendukung keputusan digunakan dalam pengambilan keputusan. Aplikasi menggunakan CBIS (*Computer Based Information System*) yang fleksibel, interaktif, dan dapat diadaptasi, yang dikembangkan untuk mendukung solusi atas masalah manajemen yang tidak terstruktur.

Aplikasi Sistem pendukung keputusan menggunakan data, memberikan antar muka pengguna yang mudah, dan dapat menggabungkan pemikiran pengambilan keputusan. Sistem pendukung keputusan lebih ditujukan untuk mendukung manajemen dalam melakukan pekerjaan yang bersifat analitis dalam situasi yang kurang terstruktur dan dengan kriteria yang kurang jelas. Sistem pendukung keputusan tidak dimaksudkan untuk mengotomatisasikan pengambilan keputusan, tetapi memberikan perangkat interaktif yang memungkinkan

pengambilan keputusan untuk melakukan berbagai analisis menggunakan model-model yang tersedia.

1. Membantu manajer dalam pengambilan keputusan atas masalah semistuktur.
2. Memberikan dukungan atas pertimbangan manajer dan bukannya dimaksudkan untuk menggantikan fungsi manajer.
3. Meningkatkan efektifitas keputusan yang diambil lebih daripada perbaikan efisiensinya.
4. Kecepatan komputasi. Komputer memungkinkan para pengambil keputusan untuk melakukan banyak komputasi secara cepat dengan biaya rendah.
5. Peningkatan produktivitas.
6. Dukungan kualitas.
7. Berdaya saing.
8. Mengatasi keterbatasan kognitif dalam pemrosesan dan penyimpanan. (Saragih: 2013; 2)

II.2.1. Karakteristik SPK (Sistem Pendukung Keputusan)

Dari pengertian Sistem Pendukung Keputusan maka dapat ditentukan karakteristik antara lain :

1. Mendukung proses pengambilan keputusan, menitik beratkan pada *managment by perception*.
2. Adanya *interface* manusia atau mesin dimana manusia (*user*) tetap memegang control proses pengambilan keputusan.

3. Mendukung pengambilan keputusan untuk membahas masalah terstruktur, semi terstruktur dan tak terstruktur.
4. Memiliki kapasitas dialog untuk memperoleh informasi sesuai dengan kebutuhan.
5. Memiliki subsistem-subsistem yang terintegrasi sedemikian rupa sehingga dapat berfungsi sebagai kesatuan item.
6. Membutuhkan struktur data komprehensif yang dapat melayani kebutuhan informasi seluruh tingkatan manajemen.

II.2.2. Komponen-Komponen Sistem Pendukung Keputusan

Komponen-komponen sistem pendukung keputusan dapat diuraikan dalam beberapa komponen yang terdiri dari :

1. *Data Management* termasuk *database* yang mendukung data yang relevan untuk berbagai situasi dan diatur oleh *software* yang disebut *Database Management System (DBMS)*.
2. *Model Management* melibatkan model finansial, statiscal, *managment science* atau berbagai model kuantitatif lainnya, sehingga dapat memberikan ke sistem suatu kemampuan analitis dan *manajemen software* yang diperlukan.
3. Dialog subsistem (*Communication*), *user* dapat berkomunikasi dan memberikan perintah pada DSS (*Decision Support System*) melalui subsistem ini, ini berarti menyediakan antarmuka.
4. *Knowledge Management*, subsistem *optional* ini dapat mendukung subsistem lain atau bertindak sebagai komponen yang berdiri sendiri.

II.2.3. Keuntungan Sistem Pendukung Keputusan

Dengan berbagai karakter khusus yang dimiliki Sistem Pendukung Keputusan, SPK (Sistem Pendukung Keputusan) dapat memberikan berbagai manfaat dan keuntungan. Manfaat yang dapat diambil dari SPK adalah :

1. SPK memperluas kemampuan pengambilan keputusan dalam memproses data atau informasi bagi pemakainya.
2. SPK membantu pengambil keputusan untuk memecahkan masalah terutama berbagai masalah yang sangat kompleks dan tidak terstruktur.
3. SPK dapat menghasilkan solusi dengan lebih cepat serta hasilnya dapat diandalkan.
4. Walaupun suatu SPK, mungkin saja tidak mampu memecahkan masalah yang dihadapi oleh pengambil keputusan, namun ia dapat menjadi stimulan bagi pengambil keputusan dalam memahami persoalannya, karena mampu menyajikan berbagai alternatif pemecahan.

II.3. *ELECTREE*

Metode *ELECTREE* merupakan salah satu metode pengambilan keputusan multikriteria dengan konsep *outranking* dengan menggunakan perbandingan berpasangan dari alternatif-alternatif berdasarkan setiap kriteria yang sesuai [Triani: 14]. Proses perhitungan dimulai dengan inputan data peserta didik dari user yang kemudian akan dihitung dengan menggunakan metode *ELECTREE* hingga menghasilkan kelompok relevan yang sesuai kriteria yaitu kelompok

peserta didik yang diterima (Maharani Putri S.N: 2015; 3). Berikut ini merupakan langkah-langkah yang dilakukan dalam perhitungan *ELECTRE*:

1. Normalisasi Matriks Keputusan

Dalam Prosedur ini, setiap atribut menjadi nilai yang comparable.

Setiap normalisasi dari nilai x_{ij} dapat dilakukan dengan rumus.

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}} \text{ untuk } i = 1, 2, 3, \dots, m \text{ dan } j = 1, 2, 3, \dots, n.$$

Sehingga didapat matriks R hasil normalisasi,

$$R = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ r_{21} & r_{22} & \dots & r_{2n} \\ \vdots & & & \\ r_{m1} & r_{m2} & \dots & r_{mn} \end{bmatrix}$$

R adalah matriks yang telah di normalisasi, dimana m menyatakan alternatif, n menyatakan kriteria dan r_{ij} adalah normalisasi pengukuran pilihan dari alternatif ke-i dalam hubungannya dengan kriteria ke-j.

2. Pembobotan pada Matriks yang Telah Dinormalisasi

Setelah dinormalisasi, setiap kolom dari matriks R dikalikan dengan bobot-bobot (W_j) yang ditentukan oleh pembuat keputusan. Sehingga, weighted normalized matrix adalah $V = RW$ yang di tulis sebagai :

$$W = R.W$$

$$\begin{bmatrix} v_{11} & v_{12} & \dots & v_{1n} \\ v_{21} & v_{22} & \dots & v_{2n} \\ \vdots & & & \\ v_{m1} & v_{m2} & \dots & v_{mn} \end{bmatrix} = \begin{bmatrix} w_1 r_{11} & w_2 r_{12} & \dots & w_n r_{1n} \\ w_1 r_{21} & w_2 r_{22} & \dots & w_n r_{2n} \\ \vdots & & & \\ w_1 r_{m1} & w_2 r_{m2} & \dots & w_n r_{mn} \end{bmatrix}$$

Dimana W adalah

$$W = \begin{bmatrix} w_1 & 0 & \dots & 0 \\ 0 & w_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & w_n \end{bmatrix}$$

3. Menentukan Aggregate Dominance Matrix.

Matriks E sebagai aggregate dominance matriks adalah matriks yang setiap elemennya merupakan perkalian antara elemen matriks F dengan elemen matriks G yang bersesuaian, secara matematis dapat dinyatakan sebagai berikut :

$$E_{kl} = F_{kl} \times g_{kl}$$

4. Eliminasi Alternatif Less Favourable

Matriks E memberikan urutan pilihan dari setiap alternatif, yaitu bila $E_{kl} = 1$ maka alternatif A_k merupakan alternatif yang lebih baik dari pada A_l . Sehingga, baris dalam matriks E yang memiliki jumlah $E_{kl} = 1$ paling sedikit dapat dieeliminasi. Dengan demikian, alternatif terbaik adalah alternatif yang mendominasi alternatif lainnya (Syekril : 2010; 3-6).

II.4. *Online Shop*

Online Shop merupakan bentuk perubahan yang di sajikan oleh internet dari segi inovasi dalam berbelanja. *Online Shop* memberikan berbagai kemudahan dalam proses transaksinya apakah hal itu yang membuat masyarakat untuk menggunakan *Online Shop* atukah hal lain yang membuat masyarakat memilih online shop sebagai cara belanja. Fenomena *Online Shop* dikalangan

masyarakat memilih cara belanja dengan menggunakan *Online Shop*. (Pratiwi : 2013; 8)

II.5. *Microsoft Visual Basic 2010*

Padaakhir tahun 1999, Teknologi *NET* diumumkan *Microsoft* memosisikan teknologi tersebut sebagai platform untuk membangun *XML Web Services*. *XML Web Service* memungkinkan aplikasi tipe manapun dan dapat mengambil data yang tersimpan pada *server* dengan tipe apapun melalui internet (PriyantoHidayatullah ; 2012 : 4).

Visual Basic NET adalah *Visual Basic* yang direkayasa kembali untuk digunakan pada platform *NET* sehingga aplikasi yang dibuat menggunakan *Visual Basic NET* dapat berjalan pada sistem computer apa pun, dan dapat mengambil data dari server dengan tipe apapun asalkan terinstal *NET Framework* (PriyantoHidayatullah ; 2012 : 4).

II.6. *Microsoft SQL Server 2008 R2*

dibawah ini : (Dita Monita ; 2013 : 32).

$$a_w = \sqrt{a_1 \times a_2 \times \dots \times a_n}$$

Keterangan :

a_i = penilaian responden ke - i

a_w = penilaian gabungan

n = banyaknya responden

II.6. *Microsoft Visual Studio 2010*

Visual Basic diturunkan dari bahasa BASIC. *Visual Basic* terkenal sebagai bahasa pemrograman yang mudah untuk digunakan terutama untuk membuat aplikasi yang berjalan di atas *platform* Windows. Pada tahun 90an, *Visual Basic* menjadi bahasa pemrograman yang paling populer dan menjadi pilihan utama untuk mengembangkan program berbasis windows . Versi *Visual Basic* terakhir sebelum berjalan diatas *.NET Framework* adalah VB6 (*Visual Studio* 1998). (Erick Kurniawan ; 2011 : 1).

Visual Basic .NET dirilis pada bulan februari tahun 2002 bersamaan dengan *platform .NET Framework* 1.0. Kini sudah ada beberapa versi dari *Visual Basic* yang berjalan pada *platform .NET* , yaitu VB 2002 (VB7), VB 2005 (VB8), VB 2008 (VB9), dan yang terakhir adalah VB 2010 (VB10) yang dirilis bersamaan dengan *Visual Studio* 2010. Selain *Visual Basic* 2010, *Visual Studio* 2010 juga mendukung beberapa bahasa lain, yaitu C#, C++, F# (bahasa baru untuk *functional programming*), *IronPhyton*, dan *IronRuby* (bahasa baru untuk *dynamic programming*). (Erick Kurniawan ; 2011 : 1).

SQL (*Structured Query Language*) adalah sebuah bahasa yang dipergunakan untuk mengakses data dalam basis data relasional. Bahasa ini secara *de facto* merupakan bahasa standar yang digunakan dalam manajemen basis data relasional. Saat ini hampir semua server basis data yang ada mendukung bahasa ini untuk melakukan manajemen datanya.

SQL terdiri dari dua bahasa, yaitu *Data Definition Language* (DDL) dan *Data Manipulation Language* (DML). Implementasi DDL dan DML berbeda untuk setiap bahasa ini memiliki bentuk standar yang ditetapkan oleh ANSI.

II.7. Sistem Basis Data

Sistem basis data dapat didefinisikan sebagai sekumpulan subsistem yang terdiri atas basis data dengan para pemakai yang menggunakan basis data secara bersama-sama, personal-personal yang merancang dan mengelola basis data, teknik-teknik untuk merancang dan mengelola basis data, serta sistem komputer untuk mendukungnya. Dari definisi tersebut dapat disimpulkan bahwa sistem basis data mempunyai beberapa elemen penting, yaitu :

1. Basis data sebagai inti dari sistem basis data.
2. Perangkat lunak (*software*) untuk perancangan dan pengelolaan basis data.
3. Perangkat keras (*hardware*) sebagai pendukung operasi pengolahan data.
4. Manusia (*brainware*) yang mempunyai peran penting dalam sistem tersebut, yaitu sebagai pemakai atau para spesialis informasi yang

mempunyai fungsi sebagai perancang atau pengelola. (Tri Puji Prakoso: 2015; 13-14)

II.8. Kamus Data

Kamus data adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Kamus data selain digunakan untuk dokumentasi dan mengurangi redundansi, juga dapat digunakan untuk :

1. Memvalidasi diagram aliran data dalam hal kelengkapan dan keakuratan.
2. Menyediakan suatu titik awal untuk mengembangkan layar dan laporan-laporan.
3. Menentukan muatan data yang disimpan dalam *file-file*.
4. Mengembangkan logika untuk proses-proses diagram aliran data.

Kamus data dibuat pada tahap analisis sistem dan digunakan baik pada tahap analisis maupun pada tahap perancangan sistem. Pada tahap analisis sistem, kamus data dapat digunakan sebagai alat komunikasi antara analisis sistem dengan pemakai sistem tentang data yang mengalir di sistem, yaitu tentang data yang masuk ke sistem dan tentang informasi yang dibutuhkan oleh pihak pemakai sistem. Pada tahap perancangan sistem, kamus data digunakan untuk merancang input, merancang laporan-laporan dan *database*.

Kamus data mendefinisikan elemen data dengan fungsi sebagai berikut :

1. Menjelaskan arti aliran data dan penyimpanan data dalam DFD.

2. Mendeskripsikan komposisi paket data yang bergerak melalui aliran (misalnya alamat diuraikan menjadi kota, negara dan kode pos)
3. Mendeskripsikan komposisi penyimpanan data.
4. Menspesifikasikan nilai dan satuan yang relevan bagi penyimpanan dan aliran.
5. Mendeskripsikan hubungan detail antar penyimpanan (yang akan menjadi titik perhatian dalam *entity-relationship diagram*). (Arhami : 2010 ; 7)

II.9. Normalisasi

Normalisasi adalah suatu teknik untuk memecah/ mendekomposisi data dalam cara-cara tertentu untuk mencegah timbulnya permasalahan pengolahan data dalam basis data dalam bisnis data. Proses normalisasi terdiri dari beberapa level, yaitu :

1. Bentuk Tidak Normal (*Un Normalized Form/UNF*). Kriteria dari bentuk ini adalah :
 - a. Jika relasi mempunyai bentuk nonflat file.
 - b. Jika relasi memuat set atribut berulang.
 - c. Jika relasi memuat atribut non atomic value.
2. Bentuk Normal Pertama (*First Normal Form/1NF*). Kriteria dari bentuk ini adalah :
 - a. Jika seluruh atribut dalam relasi bernilai atomaik.
 - b. Jika seluruh atribut dalam relasi bernilai tunggal.

- c. Jika relasi tidak memuat set atribut berulang.
- d. Jika semua *record* mempunyai sejumlah atribut yang sama.

Permasalahan dalam adalah :

- a. Tidak dapat menyisipkan informasi parsial.
- b. Terhapusnya informasi ketika menghapus sebuah *record*.
- c. Pembaharuan atribut non kunci mengakibatkan sejumlah *record* harus diperbaharui.

Untuk mengubah relasi UNF menjadi bentuk 1NF dilakukan dengan cara :

- a. Melengkapi nilai-nilai dalam atribut.
- b. Mengubah struktur relasi.

3. Bentuk Normal Kedua (*Second Normal Form/2NF*). Kriteria dari bentuk ini adalah :

- a. Jika memenuhi kriteria 1NF
- b. Jika semua atribut non kunci memiliki ketergantungan fungsional terhadap atribut kunci.

Permasalahan dalam 2NF adalah :

- a. Kerangkapan data.
- b. Pembaharuan yang tidak benar dapat menimbulkan inkonsistensi data.
- c. Proses pembaharuan data tidak efisien.
- d. Penyimpanan pada saat penyisipan, penghapusan dan pembaharuan.

Untuk mengubah relasi 1NF menjadi bentuk 2NF dilakukan dengan cara :

- a. Identifikasi ketergantungan fungsional pada relasi 1NF.

- b. Berdasarkan ketergantungan fungsional, dekomposisi relasi 1NF menjadi relasi-relasi baru sesuai dengan ketergantungan fungsionalnya.
4. Bentuk normal Ketiga (*Third Normal Form/3NF*). Kriteria dari bentuk ini adalah :
- a. Jika memenuhi kriteria 3NF.
 - b. Jika semua atribut non kunci tidak memiliki ketergantungan transitif kunci. (Prihatin: 2012; 5-6).

II.10. *Unified Modelling Language(UML)*

Pengembangan sistem adalah aktivitas manusia. Tanpa adanya kemudahan untuk memahami sistem notasi, proses pengembangan kemungkinan besar akan mengalami kesalahan. UML adalah sistem notasi yang sudah dibakukan di dunia pengembangan sistem, hasil kerja sama dari Grady Booch, James Rumbaugh dan Ivar Jacobson. UML yang terdiri dari serangkaian diagram memungkinkan bagi sistem analis untuk membuat cetak biru sistem yang komperhesif kepada klien, programmer dan tiap orang yang terlibat dalam proses pengembangan tersebut. Dengan UML akan dapat menceritakan apa yang seharusnya dilakukan oleh sistem bukan bagaimana yang seharusnya dilakukan oleh sebuah sistem.

UML adalah bahasa untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya. UML tidak hanya

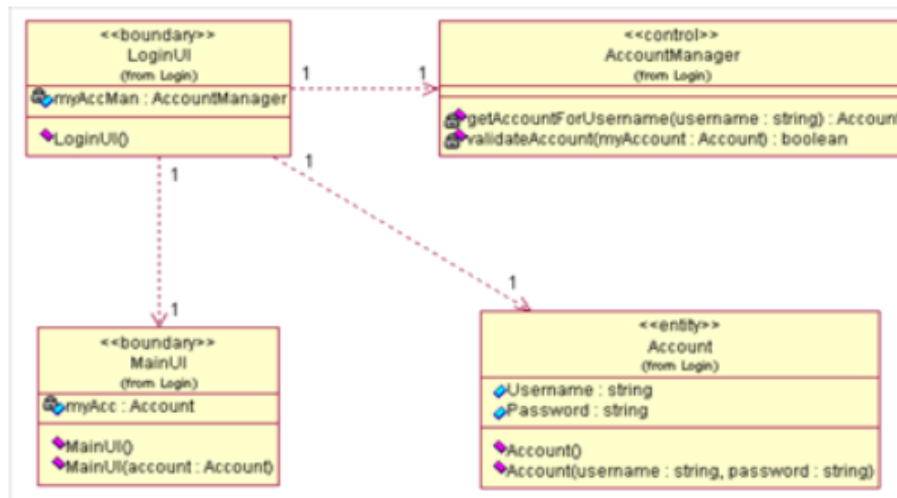
merupakan sebuah bahasa pemrograman visual saja, namun juga dapat secara langsung dihubungkan ke berbagai bahasa pemrograman seperti JAVA, C++, Visual Basic atau bahkan dihubungkan secara langsung ke dalam sebuah *object oriented database*. Begitu juga mengenai pendokumentasian dapat dilakukan seperti *requirements*, arsitektur, *design*, *source*, *project plan*, *tests* dan *prototypes*. UML memiliki 8 tipe diagram, namun pada penulisan kripsi ini penulis akan menggunakan 4 tipe diagram UML yaitu *use case diagram*, *sequence diagram*, *activity diagram* dan *class diagram*.

II.10.1. Tipe Diagram UML

Adapun 8 tipe diagram UML adalah :

1. *Class Diagram*

Class diagram menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas. *Class diagram* membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Selama tahap desain, *class diagram* berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat. *Class* memiliki tiga area pokok : nama, atribut, metoda.



Gambar II.1 Use Case Model
(Sumber : Haviluddin:2011; 3)

2. Object Diagram

Object diagram menggambarkan kejelasan kelas dan warisan dan kadang-kadang diambil ketika merencanakan kelas, atau untuk membantu pemangku kepentingan non-program yang mungkin menemukan diagram kelas terlalu abstrak.

3. Component Diagram

Component diagram menggambarkan struktur fisik dari kode, pemetaan pandangan logis dari kelas proyek untuk kode aktual di mana logika ini dilaksanakan.



Gambar II.2. Notasi Component Diagram
(Sumber : Haviluddin:2011; 3)

4. *Deployment Diagram*

Deployment diagram memberikan gambaran dari arsitektur fisik perangkat lunak, perangkat keras, dan artefak dari sistem. *Deployment* diagram dapat dianggap sebagai ujung spektrum dari kasus penggunaan, menggambarkan bentuk fisik dari sistem yang bertentangan dengan gambar konseptual dari pengguna dan perangkat berinteraksi dengan sistem.

5. *Use Diagram*

Diagram yang menggambarkan *actor*, *use case* dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. Sebuah *use case* digambarkan sebagai elips horizontal dalam suatu diagram UML *use case*.



Gambar II.3. Notasi Component Diagram
(Sumber : Haviluddin:2011; 4)

6. *Activity Diagram*

Menggambarkan aktifitas-aktifitas, objek, *state*, transisi *state* dan *event*. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas.



Gambar II.4. Notasi Component Diagram
(Sumber : Haviluddin:2011; 4)