BAB II

TINJAUAN PUSTAKA

II.1. Konsep Dasar Sistem Informasi

Konsep dasar sistem akan menguraikan beberapa pengertian sistem, karakteristik sistem, pengertian dan komponen sistem informasi (Kusrini;2009:11).

II.1.1. Pengertian Sistem

Sistem merupakan kumpulan elemen yang saling berkaitan yang bertanggung jawab memproses masukan *(input*) sehingga menghasilkan keluaran (*output*).

II.1.2. Data

Data merupakan representasi dari fakta atau gambaran mengenai suatu objek atau kejadian, ambil contoh fakta mengenai biodata mahasiswa yang meliputi nama, jenis kelamin, agama yang dianut, dan lain-lain. Contoh lain dari fakta mengenai kejadian /transaksi dalam sebuah perusahaan dagang adalah meliputi waktu transaksi, pelaku transaksinya(pelanggan, kasir), barang yang ditransaksikan, serta jumlah dan harganya. Data dinyatakan dengan nilai yang berbentukm angka, deretan karakter, atau simbol.

II.1.3. Informasi

Informasi merupakan hasil olah data, di mana data tersebut sudah diproses dan diinterpretasikan menjadi sesuatu yang bermakna untuk pengambilan keputusan. Informasi juga diartikan sebagai himpunan dari data yang relevan

dengan satu atau beberapa orang dalam suatu waktu. Suatu informasi berguna bagi pembuat keputusan karena informasi bisa menurunkan ketidakpastian (meningkatkan pengetahuan) tentang hala yang sedang dipikirkan. Makna dari sebuah informasi tentu berbeda-beda anatara seorang dengan lainnya. Tergantung pada tingkat kepentingannya.

Kegunaan informasi bagi seseorang tergantung pada waktu. Pada suatu waktu tertentu informasi tersebut mungkin sangat diperlukan dilain hari, mungkin saja hal tersebut sudah tidak berguna sama sekali. Contohnya, informasi pebandingan harga barang akan sangat dibutuhkan oleh seseorang yang akan membeli barang tersebut. Namun saat ini dia sedang tidak mempertimbangkan untuk membeli barang tersebut, informasi tersebut menjadi kurang bermakna.

II.1.4. Kualitas Informasi

Agar bias menyediakan keluaran yang berguna untuk membantu manager atau para pengambil keputusan, sebuah sistem informasi harus mampu mengumpulkan data dan mentransformasikan data tersebut kedalam informasi yang memiliki kualitas-kualitas tersebut.

Berikut karakteristik informasi yang berkualitas:

1. **Relevan**. Informasi yang disajikan sebaiknya terkait dengan keputusan yang akan diambil oleh pengguna informasi tersebut. Misalnya, seorang manager yang akan memberikan kredit kepada pelanggan bias melihat laporan keuangan pelanggan tersebut karena laporan tersebut terkait dengan keputusan yang dibuat, yaitu memberikan atau tidak memberikan kredit kepada pelanggan.

2. **Akurat** . kecocokan antara informasi dengan kejadia-kejadian atau objek-objek yang diwakilinya.
3. **Lengkap**. merupakan derajat sampai seberapa jauh informasi menyertakan kejadian-kejadian atau objek-objek yang berhubungan.
4. **Tepat waktu**. Infoemasi yang tidak tepat waktu akan menjadi informasi yang tidak berguna atau tidak dapat digunakan untuk membantu mengambil keputusan.
5. **Dapat dipahami**. Hal tersebut terkait dengan bahasa dan cara penyajian informasi agar pengguna lebih mudah mengambil keputusan.
6. **Dapat dibandingkan**. Sebuah informasi yang memungkinkan seorang pemakai untuk mengidentifikasi persamaan dan perbedaan antara dua objek atau kejadian yang mirip.

II.1.5. Pengertian Sistem Informasi

Suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi besifat manajerial, dan merupakan kegiatan strategi dari suatu organisasi, serta menyediakan laporan-lapoaran yang diperlukan oleh pihak luar. (Kusrini ;2007 :11).

Berdasarkan dukungan kepada pemakainya, sistem informasi dibagi menjadi:

1. Sistem informasi pemrosesan transaksi(*Transaction Processing System TPS).*
2. Sistem informasi Manajemen (*Management Information System atau MIS).*
3. Sistem pendukung keputusan *(decision support system).*

II.2. Pengertian Sistem Penunjang Keputusan

Sistem penunjang keputusan merupakan sistem interaktif yang menyediakan informasi, pemodelan dan pemanipulasian data. Sistem itu digunakan untuk membantu pengambil keputusan dalam situasi yang semiterstruktur dan situasi yang tidak terstruktur, di mana tak seorang pun secara pasti bagaimana keputusan seharusnya dibuat. (Kusrini ; 2007 :15).

DSS biasanya dibangun untuk mendukung solusi atas suatu masalah atau untuk mengevaluasi suatu peluang. DSS yang seperti itu disebut aplikasi DSS. Aplikasi DSS digunakan dalam pengambilan keputusan.

Apalikasi DSS menggunkan data, memberikan antar muka pengguna yang mudah, dan dapat menggabungkan pemikiran pengambil keputusan. DSS tidak dimaksudkan untuk mengotomatisasikan pengambilan keputusan, tetapi memberikan perangkat interaktif yang memungkinkan pengambil keputusan untuk melakukan berbagai analisis menggunakan model-model yang tersedia.

Keputusan yang diambil untuk menyelesaikan suatu masalah dilihat dari keterstrukturannya yang biasa dibagi menjadi:

1. Keputusan terstruktur (*structured decision)* merupakan keputusan secara berulang-ulang dan bersifat rutin. Prosedur pengambilan keputusan sangatlah jelas. Keputusan tersebut terutama dilakukan pada manajemen tingkat bawah. Misalnya, keputusan pemesanan barang dan keputusan penagihan piutang.
2. Keputusan semiterstruktur (*semistructured decision)* merupakan yang memiliki dua sifat. Sebagian keputusan bias ditangani oleh komputer dan yang lain harus tetap dilakukan oleh pengambil keputusan.

3. Keputusan tak terstruktur (*unstructured decision*) merupakan keputusan yang penanganannya rumit karena tidak terjadi berulang-ulang atau tidak selalu terjadi.

Karakteristik dan Kemampuan Sistem Penunjang Keputusan Menurut (Turban, 2005), ada beberapa karakteristik dari SPK, di antarannya adalah sebagai berikut:

1. Mendukung seluruh kegiatan organisasi
2. Mendukung beberapa keputusan yang saling berinteraksi
3. Dapat digunakan berulang kali dan bersifat konstan
4. Terdapat dua komponen utama, yaitu data dan model
5. Menggunakan baik data ekternal maupun internal
6. Memiliki kemampuan *what-if analysis* dan *goal seeking analysis*
7. Menggunakan beberapa model kuantitatif.

II.3. Java dan Mysql

II.3.1 Java

Java adalah sebuah bahasa pemrograman yang diciptakan oleh *James Gosling*, seorang *developer* dari *Sun Microsystem* pada tahun 1991. Selanjutnya Java dikembangkan *Sun Microsystem* dan banyak digunakan untuk menciptakan *Executable Content* yang dapat didistribusikan melalui *network*.

Java adalah bahasa pemrograman *Object – Oriented* dengan unsur-unsur seperti bahasa C++ dan bahasa-bahasa lainnya yang memiliki *libraries* yang cocok untuk lingkungan internet. *Java* dapat melakukan banyak hal dalam melakukan pemrograman, seperti membuat animasi halaman *web*,

pemrograman *Java* untuk Ponsel dan aplikasi interaktif. *Java* juga dapat digunakan untuk handphone, internet dan lain-lain. (jurnal Informatika Mulawarman, 2010:18).

II.3.2. Mysql

MySQL termasuk jenis *Relational Database Management (RDBMS)*. Itulah sebabnya istilah seperti table, baris, kolom, digunakan pada *MySQL*. Jadi dapat ditarik kesimpulan bahwa *MySQL* merupakan sebuah *database* yang berfungsi sebagai penyimpanan dan manajemen data. Dan *MySQL* ini bisa berjalan di banyak system operasi salah satunya yaitu system operasi *windows.*

MySQL merupakan salah satu aplikasi yang dipergunakan untuk mengelola database yang terdapat dalam server database MySQL. Aplikasibini dikembangkan oleh produsen MySQL itu sendiri, yaitu SUN Microsystem beserta komunitas yang tegabung didalamnya. MySQL sebenarnya merupakan aplikasi hasil itegtasi ntara 3 buah aplikasi MySQL Administrator, MySQL Control Centre, dan MySQL Schematic Diagran.

Aplikasi ini memiliki kemampuan untuk mengelola segala aspek yang terdapat dalam database MySQL, mulai dari administarasi database, pembuatan database beserta unsure didalamnya, sampai dengan pembuatan diagram konek dari database tersebut (Abdul Kadir, 2001:353).

II.4. UML

II.4.1 Pengenalan UML (*Unified Modeling Languange)*

UML *(Unified Modeling Language)* merupakan metodologi koloborasi antara metoda – metoda *booch, OMT (Object Modeling Technique)* serta OOSE (*Object oriented Engineering)* dan beberapa metoda lainnya merupakan metodololgi yang paling sering digunakan saat ini untuk mengadaptasi marak nya penggunaan bahasa "pemograman berorientasi objek" (OOP) (Adi Nugroho, 2009 : 4).

II.4.2. Diagram – Diagram UML

Beberapa literature menyebutkan bahwa UML menyediakan Sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa diagram yang digabung menjadi diagram interaksi. Namun demikian model – model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain:(Prabowo pudjo widodo dan Herlawati, 2011:10).

1. Diagram Kelas (*Class Diagram).* bersifat statis. Diagram ini memperlihatkan himpunan kelas – kelas, Antarmuka – antarmuka, koloborasi – koloborasi, serta relasi. Diagram ini umum dijumpai pada pemodelan sistem berientasi objek.
2. Diagram paket *(Package Diagram).* Bersifat statis. Diagram ini memperlihatkan kumpulan kelas – kelas , merupakan bagian dari diagram komponen.
3. Diagram *Use Case.* Bersifat statis. Diagram ini memperlihatkan himpunan *use-case* dan aktor – aktor (suatu jenis khusus dari kelas). Diagram ini

terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna.

4. Diagram interaksi dan *Sequence* (urutan). Bersifat dinamis. Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.
5. Diagram Komunikasi *(Communication Diagram).* Bersifat dinamis. Diagram koloborasi UML yang menekankan organisasi structural dari objek – objek yang menerima serta mengirim pesan.
6. Diagram *Statechart (Statechart Diagram). Bersifat dinamis.* Diagram status memperlihatkan keadaan – keadaan pada sistem, memuat status *(state),*transisi, kejadian serta aktifitas. Diagram ini terutama penting untuk memperlihatkan sifat dinamis dari antar muka *(interface),* kelas, koloborasi dan terutama penting pada pemodelalan sistem – sistem yang reaktif.
7. Diagram Aktivitas *(Activity Diagram).* Barsifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihat aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi – fungsi suatu sistem dan member tekanan pada aliran kendali antar objek
8. Diagram Komponen *(Component Diagram).* Bersifat statis. Diagram komponen ini memperlihatkan hubungan dengan supplier dan pelanggan yang bersifat eksternal harus diperhatikan.

II.4.3. Jenis – jenis UML

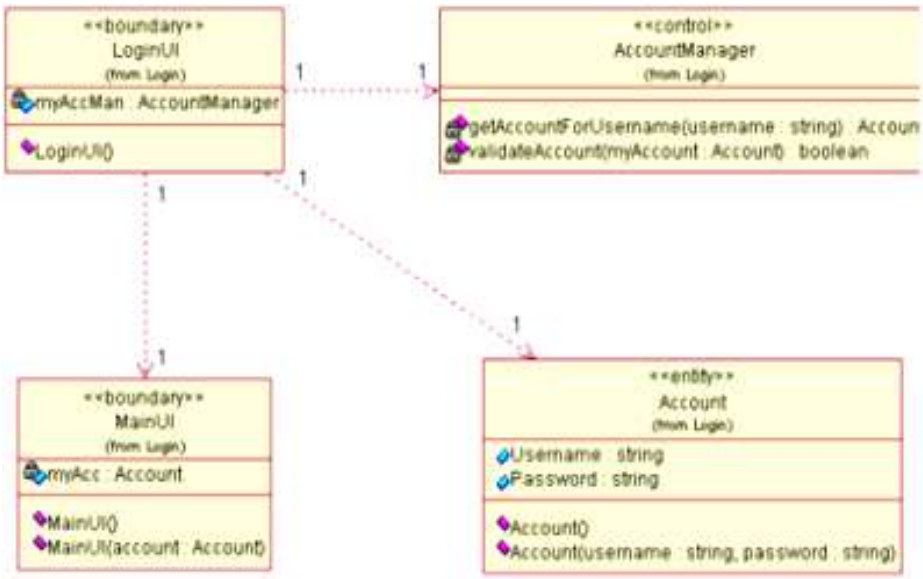
Sejauh ini para pakar merasa lebih mudah dalam menganalisa dan mendesain atau memodelkan suatu sistem karena UML memiliki seperangkat aturan dan notasi dalam bentuk grafis yang cukup spesifik.

Pada UML terdiri atas tiga kategori dan memiliki 13 jenis diagram yaitu sebagai berikut:

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. Class memiliki tiga area pokok : Nama (dan stereotype) , Atribut, Metoda.

Adapun gambar notasi *Class diagram* yaitu seperti dibawah ini:



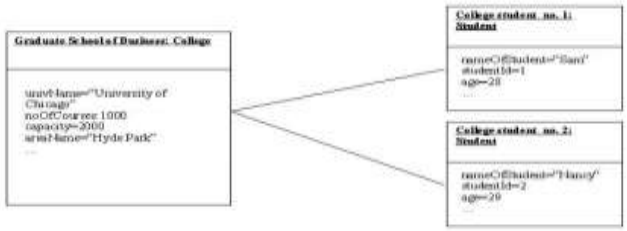
Gambar II.1 Notasi *Class diagram*

Sumber : Haviluddin, 2011

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.

Adapun gambar notasi *Objek diagram* yaitu seperti dibawah ini:



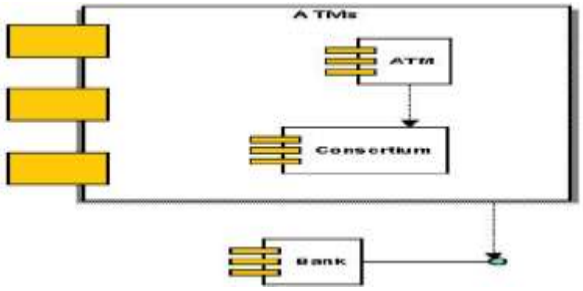
Gambar II.2 Notasi *Object diagram*

Sumber : Haviluddin, 2011

3. *Component diagram*

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

Adapun gambar notasi *Component diagram* yaitu seperti dibawah ini:



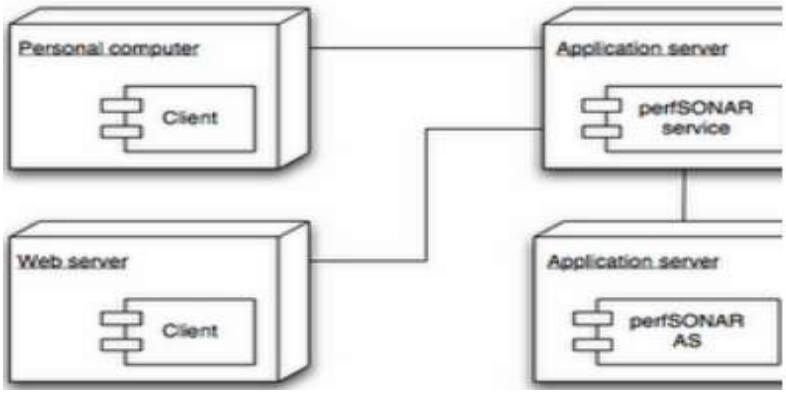
Gambar II.3 Notasi *Component diagram*

Sumber : Haviluddin, 2011

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.

Adapun gambar notasi *deployment diagram* yaitu seperti dibawah ini:



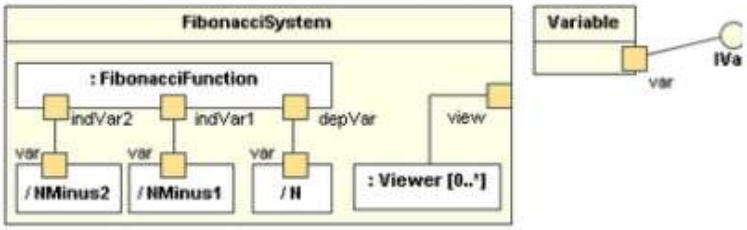
Gambar II.4 Notasi *deployment diagram*

Sumber : Haviluddin, 2011

5. *Composite structure diagram*

Composite structure diagram Sebuah diagram struktur komposit mirip dengan diagram kelas, tetapi menggambarkan bagian individu, bukan seluruh kelas. Kita dapat menambahkan konektor untuk menghubungkan dua atau lebih bagian dalam atau ketergantungan hubungan asosiasi.

Adapun gambar notasi *Composite structur diagram* yaitu seperti dibawah ini:



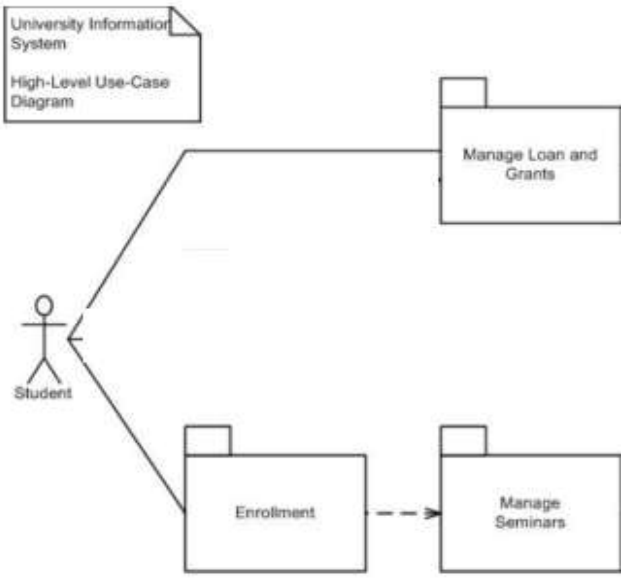
Gambar II.5 Notasi *Composite structure diagram*

Sumber : Haviluddin, 2011

6. *Package diagram*

Paket diagram biasanya digunakan untuk menggambarkan tingkat organisasi yang tinggi dari suatu proyek software. Atau dengan kata lain untuk menghasilkan diagram ketergantungan paket untuk setiap paket dalam Pohon Model.

Adapun contoh gambar notasi *package diagram* yaitu seperti dibawah ini:



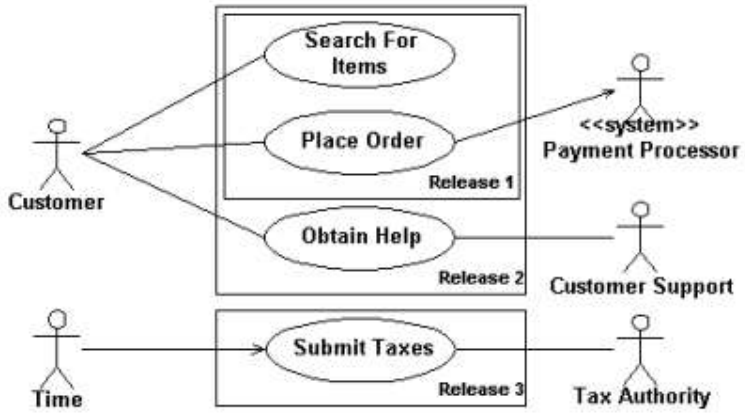
Gambar II.6 Notasi *Package diagram*

Sumber : Haviluddin, 2011

7. *Use case 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*.

Adapun contoh gambar notasi *use case diagram* yaitu seperti dibawah ini:



Gambar II.7 Notasi *use case diagram*

Sumber : Haviluddin, 2011

8. *Activity diagram*

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

Adapun contoh gambar notasi *Activity diagram* yaitu seperti dibawah ini:

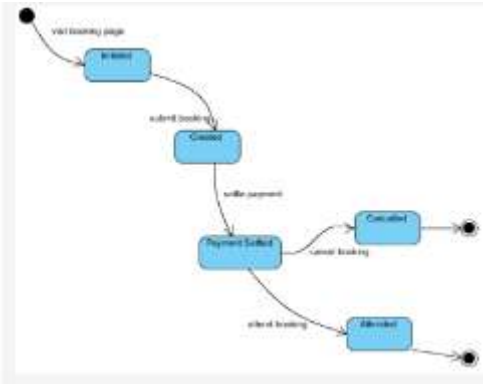
Gambar II.8 Notasi *Activity diagram*

Sumber : Haviluddin, 2011

9. State *Machine diagram*

Menggambarkan state, transisi state dan event.

Adapun contoh gambar notasi *State Mchine diagram* yaitu seperti dibawah ini:



Gambar II.9 *State Machine diagram*

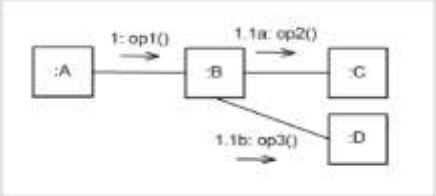
Sumber : Haviluddin, 2011

10. *Communication diagram*

Serupa dengan *sequence diagram*, tetapi diagram komunikasi juga digunakan untuk memodelkan perilaku dinamis dari use case. Bila dibandingkan

dengan *Sequence diagram*, diagram komunikasi lebih terfokus pada menampilkan kolaborasi benda daripada urutan waktu.

Adapun contoh gambar notasi *Communicatin diagram* yaitu seperti dibawah ini:



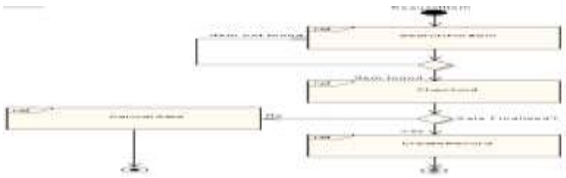
Gambar II.10 Notasi *Communication diagram*

Sumber : Haviluddin, 2011

11. *Interaction Overview diagram*

Interaksi overview diagram berfokus pada gambaran aliran kendali interaksi dimana node adalah interaksi atau kejadian interaksi.

Adapun contoh gambar notasi *Interaction OverView diagram* yaitu seperti dibawah ini



Gambar II.11 Notasi *Interaction Overview diagram*

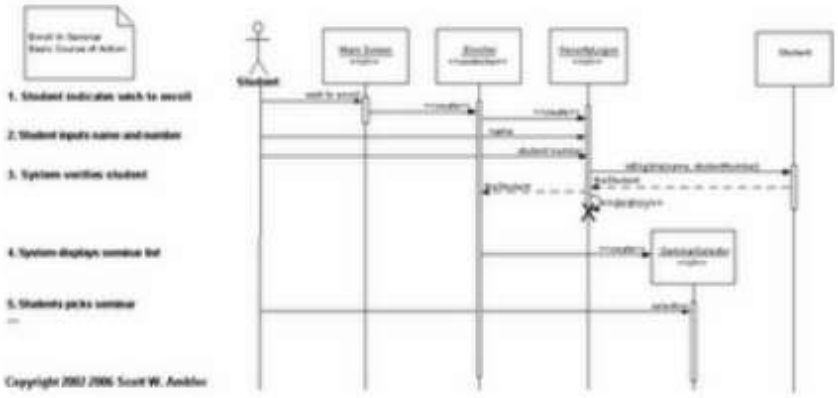
Sumber : Haviluddin, 2011

12. *Sequence diagram*

Sequence diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya sequence diagram adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara

logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case diagram.*

Adapun contoh gambar notasi *Sequence diagram* yaitu seperti dibawah ini:



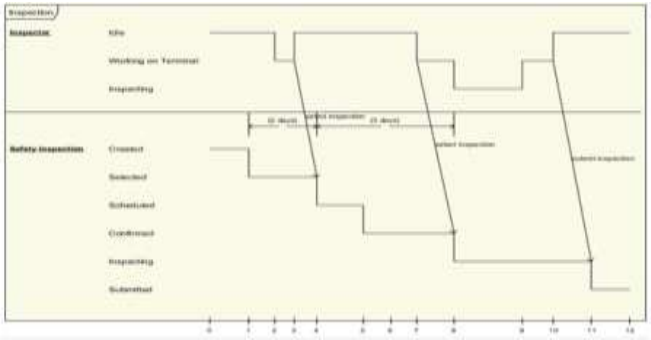
Gambar II.12 Notasi *Sequence diagram*

Sumber : Haviluddin, 2011

13. *Timing diagram*

Timing diagram di UML didasarkan pada diagram waktu *hardware* awalnya dikembangkan oleh para insinyur listrik.

Adapun contoh gambar notasi *Timing diagram* yaitu seperti dibawah ini:



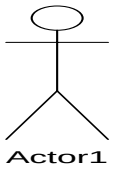
Gambar II.13 Notasi *Timing diagram*

Sumber : Haviluddin, 2011

II.4.3.1. Notasi – Notasi UML

UML memiliki seperangkat notasi yang akan digunakan ke dalam tiga kategori diatas yaitu struktur diagram, behaviour diagram dan interaction diagram. Berikut beberapa notasi dalam UML diantaranya :

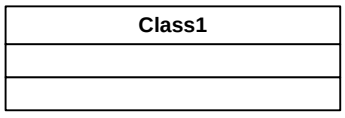
1. *Actor;* menentukan peran yang dimainkan oleh user atau sistem lain yang berinteraksi dengan subjek. *Actor* adalah segala sesuatu yang berinteraksi langsung dengan sistem aplikasi komputer, seperti orang, benda atau lainnya.



Gambar II.14. Notasi *actor* pada UML

Sumber : Haviluddin, 2011

2. *Class* diagram; Notasi utama dan yang paling mendasar pada diagram UML adalah notasi untuk mempresentasikan suatu class beserta dengan atribut dan operasinya. *Class* adalah pembentuk utama dari sistem berorientasi objek.



Gambar II.15 Notasi *class* pada UML

Sumber : Haviluddin, 2011

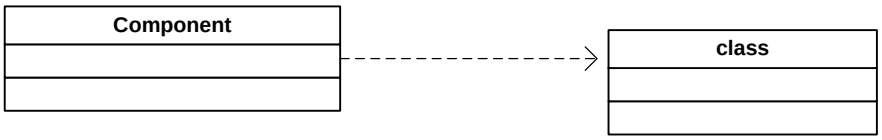
3. *Use case* adalah deskripsi fungsi dari sebuah sistem perspektif pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara

user (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai.

Gambar II.16. Notasi *Use case* pada UML

Sumber : Haviluddin, 2011

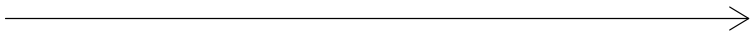
4. *Realization* menunjukkan hubungan bahwa elemen yang ada di bagian tanpa panah akan merealisasikan apa yang dinyatakan oleh elemen yang ada di bagian dengan panah.



Gambar II.17 Notasi *Realization* pada UML

Sumber : Haviluddin, 2011

5. *Interaction* digunakan untuk menunjukkan baik aliran pesan atau informasi antar obyek maupun hubungan antar obyek.

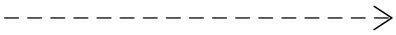


Gambar II.18 Notasi *Interaction* pada UML

Sumber : Haviluddin, 2011

6. *Dependency* merupakan relasi yang menunjukan bahwa perubahan pada salah satu elemen memberi pengaruh pada elemen lain. Terdapat 2 *stereotype* dari *dependency*, yaitu include dan *extend*. *Include* menunjukkan bahwa suatu

bagian dari elemen (yang ada digaris tanpa panah) memicu eksekusi bagian dari elemen lain (yang ada di garis dengan panah).



Gambar II.18 Notasi *Dependency* pada UML

Sumber : Haviluddin, 2011

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

Gambar II.19 Notasi *Note* pada UML

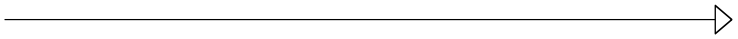
Sumber : Haviluddin, 2011

8. *Association* menggambarkan navigasi antar *class* (navigation), berapa banyak obyek lain yang bisa berhubungan dengan satu obyek (*multiplicity* antar *class*) dan apakah suatu class menjadi bagian dari class lainnya (aggregation).

Gambar II.20 Notasi *Association* pada UML

Sumber : Haviluddin, 2011

9. *Generalization* menunjukkan hubungan antara elemen yang lebih umum ke elemen yang lebih spesifik.



Gambar II.21 Notasi *Generalization* pada UML

Sumber : Haviluddin, 2011

10. *Package* adalah mekanisme pengelompokkan yang digunakan untuk menandakan pengelompokkan elemen-elemen model.

Gambar II.22 Notasi *Package* pada UML

Sumber : Haviluddin, 2011

11. *Interface* merupakan kumpulan operasi berupa implementasi dari suatu *class*. Atau dengan kata lain implementasi operasi dalam *interface* dijabarkan oleh operasi di dalam *class*.

Gambar II.23 Notasi *Interface* pada UML

Sumber : Haviluddin, 2011

II.4.3.2. Pengertian ERD

Entity relationship (ER) data model didasarkan pada persepsi terhadap dunia nyata yang tersusun atas kumpulan objek-objek dasar yang disebut entitas dan hubungan antar objek). Entitas adalah sesuatu atau objek dalam dunia nyata yang dapat dibedakan dari objek lain. Entitas digambarkan dalam basis data dengan kumpulan atribut. Misalnya: nim, nama, alamat, dan kota. Relasi adalah hubungan antara beberapa entitas. (Tri Octafian, 2011: 150).

Struktur logis (skema database) dapat ditunjukkan secara grafis dengan diagram ERD yang dibentuk dari komponen-komponen berikut : (Tri Octafian, 2011: 150).

II.4.3.3 Pemetaan Kardinalitas

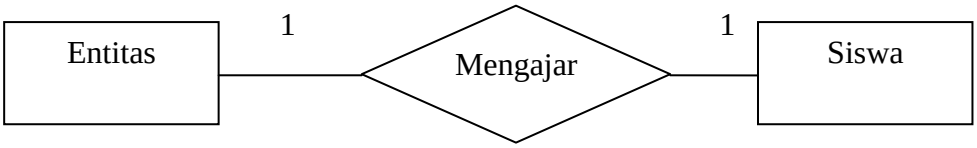
Pemetaan kardinalitas menyatakan jumlah entitas di mana entitas lain dapat dihubungkan ke entitas tersebut melalui sebuah himpunan relasi. (Tri Octafian, 2011:150-151).

1. *One to One*

Sebuah entitas pada A berhubungan dengan paling banyak satu entitas pada B dan sebuah entitas pada B berhubungan dengan paling banyak satu entitas pada A.

Contoh:

Pada pengajaran privat, satu guru satu siswa. Seorang guru mengajar seorang siswa, seorang siswa diajar oleh seorang guru.



Gambar II.24 Hubungan *One To One.*

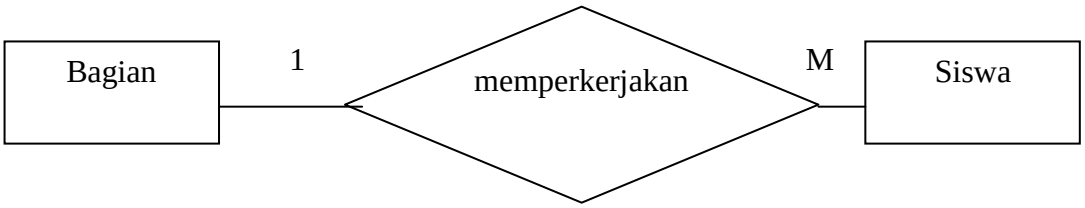
Sumber : Tri Octafian, 2011:151

2. *One to Many/ Many to One*

Sebuah entitas pada A berhubungan dengan lebih dari satu entitas pada B dansebuah entitas pada B berhubungan dengan paling banyak satu entitas pada A,atau sebaliknya (Many to One).

Contoh:

Dalam satu perusahaan, satu bagian mempekerjakan banyak pegawai. Satu bagian mempekerjakan banyak pegawai, satu pegawai kerja dalam satu bagian.



Gambar II.25 *One To Many*

Sumber : Tri Octafian, 2011:152

3. *Many To Many*

Sebuah entitas pada A berhubungan dengan lebih dari satu entitas pada B dan sebuah entitas pada B berhubungan dengan lebih dari satu entitas pada A.

Contoh:

Dalam universitas, seorang mahasiswa dapat mengambil banyak mata kuliah.Satu mahasiswa mengambil banyak mata kuliah dan satu mata kuliah diambil banyak mahasiswa.

Gambar II.26 Hubungan *Many To Many*

Sumber : Tri Octafian, 2011:152

II.5. Pengertian Metode *Fuzzy SAW* (*Simple Additive Weighting)*

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

Konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut (Fishburn, 1967) (MacCrimmon, 1968). Metode SAW membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada. Metode ini merupakan metode yang paling terkenal dan paling banyak digunakan dalam menghadapi situasi Multiple Attribute Decision Making (MADM). MADM itu sendiri merupakan suatu metode yang digunakan untuk mencari alternatif optimal dari sejumlah alternatif dengan kriteria tertentu.

(Asep Kamaludin, 2012:3)

II.5.1. Langkah-Langkah Penyelesaian SAW (*Simple Additive Weighting*)

Langkah-Langkah Penyelesaian Fuzzy SAW sebagai berikut :

1. Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan, yaitu Ci.
2. Menentukan rating kecocokan setiap alternatif pada setiap kriteria.
3. Membuat matriks keputusan berdasarkan kriteria(Ci), kemudian melakukan normalisasi matriks berdasarkan persamaan yang disesuaikan dengan jenis atribut (atribut keuntungan ataupun atribut biaya) sehingga diperoleh matriks ternormalisasi R.
4. Hasil akhir diperoleh dari proses perankingan yaitu penjumlahan dari perkalian matriks ternormalisasi R dengan vektor bobot sehingga diperoleh nilai terbesar yang dipilih sebagai alternatif terbaik (Ai)sebagai solusi.

Formula untuk melakukan normalisasi tersebut adalah :

Dimana rij adalah rating kinerja ternormalisasi dari alternatif Ai pada atribut Cj; i=1,2,...,m dan j=1,2,...,n. Nilai preferensi untuk setiap alternatif (Vi) diberikan sebagai: $V_i = \sum_{j=1}^{n} w_j r_{ij}$

Keterangan :

Vi = rangking untuk setiap alternatif

wj = nilai bobot dari setiap kriteria

rij = nilai rating kinerja ternormalisasi

Nilai Vi yang lebih besar mengindikasikan bahwa alternatif Ai lebih terpilih.