

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*). (Kusrini; 2007 :11)

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 berbentuk angka, deretan karakter, atau symbol. (Kusrini; 2007 :2).

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 hal yang sedang dipikirkan. Makna dari sebuah informasi tentu berbeda-beda antara seorang dengan lainnya. Tergantung pada tingkat kepentingannya. (Kusrini; 2007 : 11).

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.(Kusrini; 2007 :4).

II.1.4. Kualitas Informasi

Agar bisa 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. (Kusrini; 2001 :4).

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 kejadian-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. (Kusrini; 2007 :5).

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:

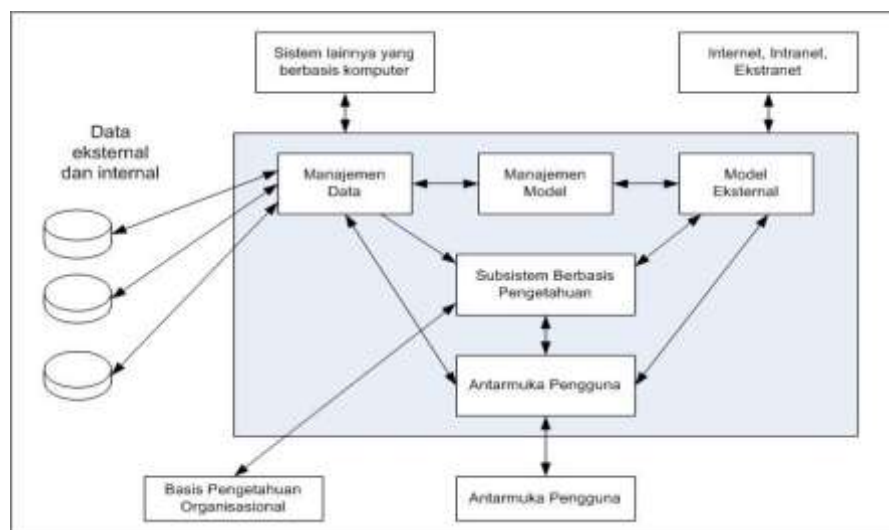
1. Sistem informasi pemrosesan transaksi(*Transaction Processing System TPS*).
2. Sistem informasi Manajemen (*Management Information System atau MIS*).
3. Sistem pendukung keputusan (*decision support system*).

II.2. Pengertian Sistem Pendukung Keputusan

Sistem pendukung 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.

Arsitektur DSS



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. (Kusrini; 2007 :19).

Karakteristik dan Kemampuan Sistem Pendukung Keputusan Menurut (Turban, 2005), ada beberapa karakteristik dari SPK, di antaranya 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 eksternal maupun internal
6. Memiliki kemampuan *what-if analysis* dan *goal seeking analysis*
7. Menggunakan beberapa model kuantitatif.

II.3. Basis Data (*Database*)

Database atau basis data adalah sekumpulan data yang memiliki hubungan secara logika dan diatur dengan susunan tertentu serta disimpan dalam media penyimpanan komputer. dat itu sendiri adalah representasi dari

semua fakta yang ada pada dunia nyata. database sering digunakan untuk melakukan proses terhadap data-data tersebut untuk menghasilkan informasi tertentu. misalnya dari data nama siswa yang berulang tahun pada hari ini. Tentu saja informasi tersebut akan anda dapatkan dari software pemroses database dengan cara anda memberikan perintah dalam bahasa tertentu yaitu SQL(*Structured Query Language*). (Wahana Komputer, 2010 : 24)

Dalam database ada sebutan-sebutan untuk satuan data yaitu:

1. Karakter, ini adalah satuan data terkecil. data terdiri atau susunan karakter yang pada akhirnya mewakili data yang memiliki arti dari sebuah fakta.
2. Field, adalah kumpulan dari karakter yang mewakili fakta tertentu misalnya seperti nama siswa, tanggal lahir, dan lain-lain. Dalam dunia perancangan database, field juga disebut atribut.
3. Record, adalah kumpulan dari field. Pada record anda dapat menemukan banyak sekali informasi penting dengan cara mengombinasikan field-field yang ada.
4. Tabel, adalah sekumpulan dari record-record yang memiliki kesamaan entity dalam dunia nyata. Kumpulan dari tabel adalah database.
5. File, adalah bentuk fisik dari penyimpanan data. File database berisi semua data yang telah disusun dan diorganisasikan sedemikian rupa sehingga memudahkan pemberian informasi. (Wahana Komputer, 2010 : 24)

II.4. VB Dan SQL

II.4.1 VB (*Visual Basic*)

Visual Basic merupakan salah satu aplikasi pemrograman yang di keluarkan oleh Microsoft. Visual basic menggunakan pendekatan Graphical User Interface (GUI) yang lebih nyaman dan lebih mudah di gunakan oleh pengguna.

Visual Basic merupakan even-driven prigraming(programan terkendali kejadian) artinya program menunggu sampai adanya respon dari pemakai berupa event/kejadian tertentu(tombol klik, menu dipilih, dan lain-lain) Ketika event terdeteksi, kode yang berhubungan dengan event akan dijalankan.

Visual Basic merupakan salah satu bahasa pemrograman yang andal dan banyak digunakan oleh pengembang untuk membangun berbagai macam aplikasi Windows. Visual Basic 2008 atau Visual Basic 9 adalah versi terbaru yang telah diluncurkan bersama C#, Visual C++, dan Visual Web Developer dalam satu paket Visual Studio 2008.

Visual Basic 2008 merupakan aplikasi pemrograman yang menggunakan teknologi .NET Framework. Teknologi .NET Framework merupakan komponen windows yang terintegrasi serta mendukung pembuatan, penggunaan aplikasim dan halaman web. Teknologi .NET Framework mempunyai 2 komponen utama, yaitu CLR (*Common Language Runtime*) dan *Class Library*. CLR digunakan untuk menjalankan aplikasi yang berbasis .NET sedangkan Library adalah kelas pustaka atau perintah yang digunakan untuk mengembangkan aplikasi. (Wahana Komputer, 2010 : 2)

II.4.2 Database SQL Server

SQL Server adalah sebuah terobosan baru dari Microsoft dalam bidang database. SQL Server adalah sebuah DBMS (*Database Management System*) yang dibuat oleh Microsoft untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahulunya seperti IBM dan Oracle. SQL Server dibuat pada saat kemajuan dalam bidang hardware sedemikian pesat. Oleh karena itu sudah dapat dipastikan bahwa SQL Server membawa beberapa terobosan dalam bidang pengolahan dan penyimpanan data. (Wahana Komputer, 2010 : 2)

II.5. UML

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

UML (*Unified Modeling Language*) merupakan metodologi kolaborasi antara metoda – metoda *booch*, *OMT (Object Modeling Technique)* serta *OOSE (Object oriented Engineering)* dan beberapa metoda lainnya merupakan metodologi yang paling sering digunakan saat ini untuk mengadaptasi maraknya penggunaan bahasa “pemrograman berorientasi objek” (OOP) (Adi Nograho ;2009 : 4).

II.5.2. 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:

1. Diagram Kelas (*Class Diagram*). bersifat statis. Diagram ini memperlihatkan himpunan kelas – kelas, Antarmuka – antarmuka, kolaborasi – kolaborasi, serta relasi. Diagram ini umum dijumpai pada pemodelan sistem berorientasi 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 kolaborasi 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, kolaborasi dan terutama penting pada pemodelan sistem – sistem yang reaktif.

7. Diagram Aktivitas (*Activity Diagram*). Bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan 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. (Prabowo pudjo widodo dan Herlawati, 2011:10).

II.5.3. 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.

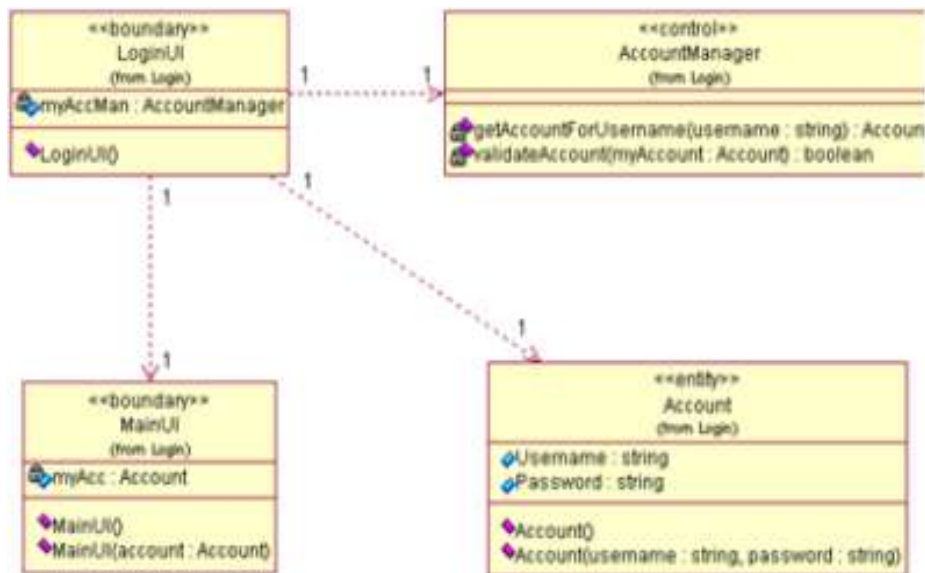
Pada UML terdiri atas tiga kategori dan memiliki 13 jenis diagram yaitu sebagai berikut:

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* dapat dilihat pada gambar II.1.



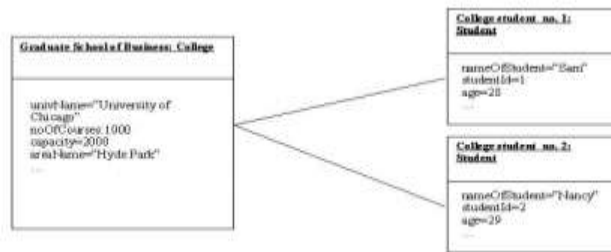
Gambar II.1 Notasi *Class diagram*

Sumber: Mulawarman; 2011: 6

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* dapat dilihat pada gambar II.2.



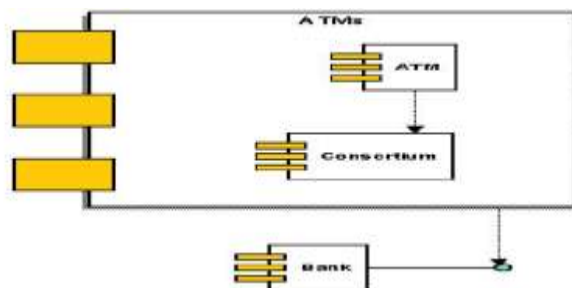
Gambar II.2 Notasi *Object diagram*

Sumber: Mulawarman; 2011: 6

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* dapat dilihat pada gambar II.3.



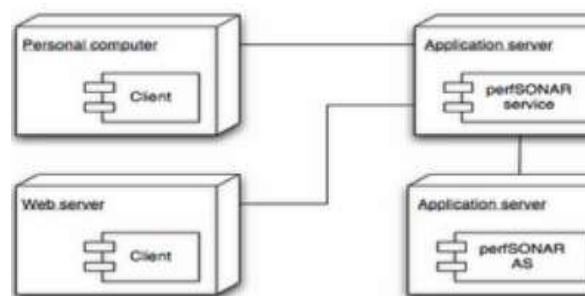
Gambar II.3 Notasi *Component diagram*

Sumber: Mulawarman; 2011: 6

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* dapat dilihat pada gambar II.4.



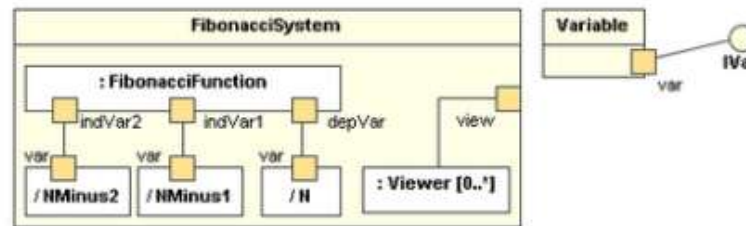
Gambar II.4 Notasi *deployment diagram*

Sumber: Mulawarman; 2011: 6

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* dapat dilihat pada gambar II.5.



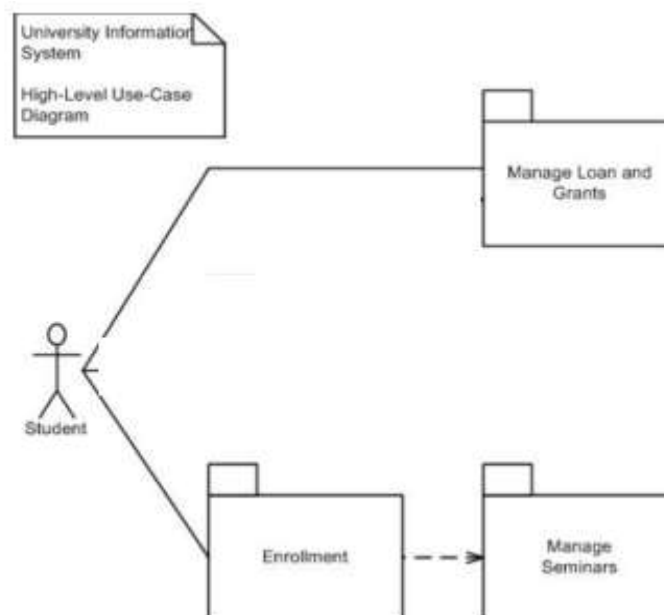
Gambar II.5 Notasi *Composite structure diagram*

Sumber: Mulawarman; 2011: 6

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* dapat dilihat pada gambar II.6.



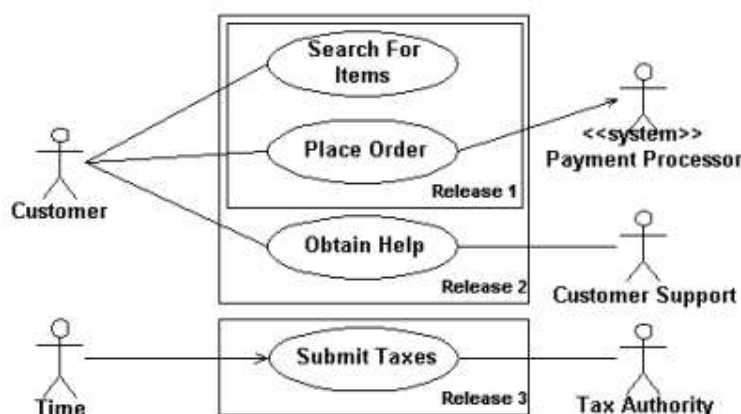
Gambar II.6 Notasi *Package diagram*

Sumber: Mulawarman; 2011: 4

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 gambar notasi *use case diagram* dapat dilihat pada gambar II.7.



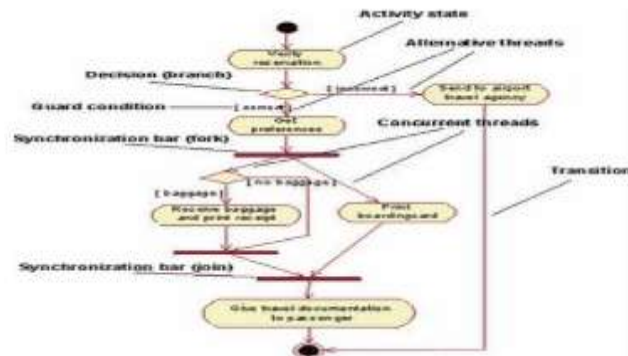
Gambar II.7 Notasi *use case diagram*

Sumber: Mulawarman; 2011: 4

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 gambar notasi *Activity diagram* dapat dilihat pada gambar II.8.



Gambar II.8 Notasi *Activity diagram*

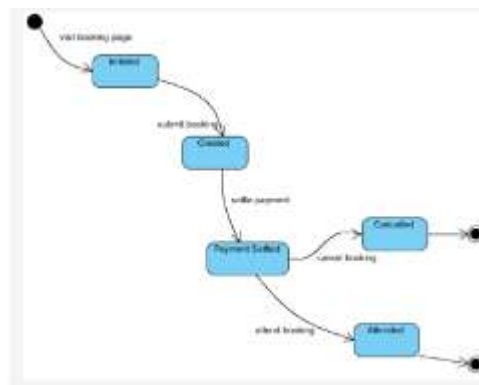
Sumber: Mulawarman; 2011: 4

9. State Machine diagram

Menggambarkan state, transisi state dan event.

Adapun gambar notasi *State Machine diagram* dapat dilihat pada gambar

II.9.



Gambar II.9 *State Machine diagram*

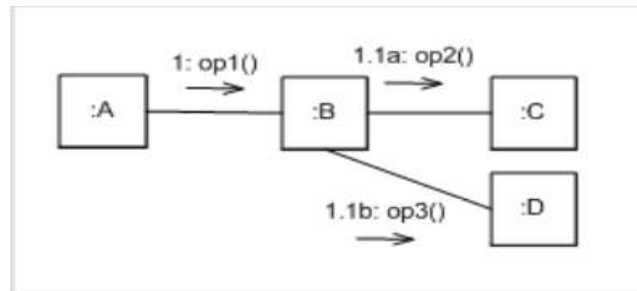
Sumber: Mulawarman; 2011: 6

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 gambar notasi *Communication diagram* dapat dilihat pada gambar II.10.



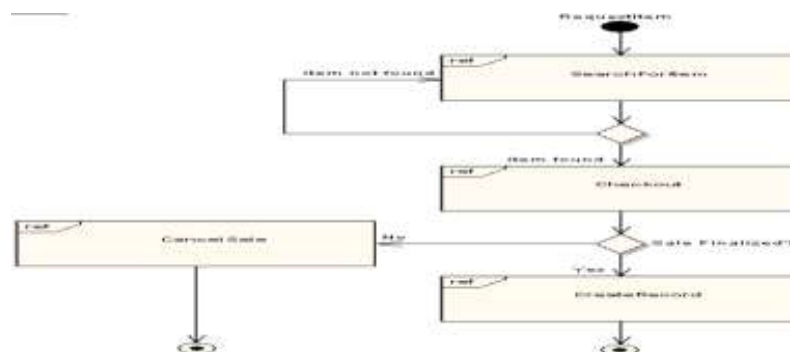
Gambar II.10 Notasi *Communication diagram*

Sumber: Mulawarman; 2011: 4

11. *Interaction Overview diagram*

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

Adapun gambar notasi *Interaction OverView diagram* dapat dilihat pada gambar II.11.



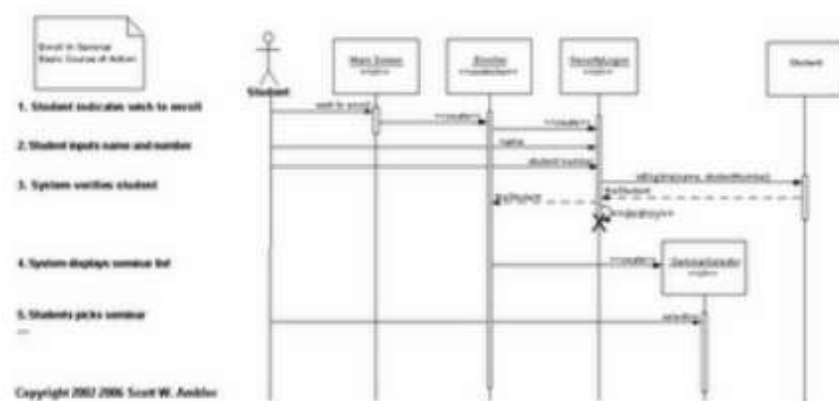
Gambar II.11 Notasi *Interaction Overview diagram* Sumber:

Sumber: Mulawarman; 2011: 5

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 gambar notasi *Sequence diagram* dapat dilihat pada gambar II.12.



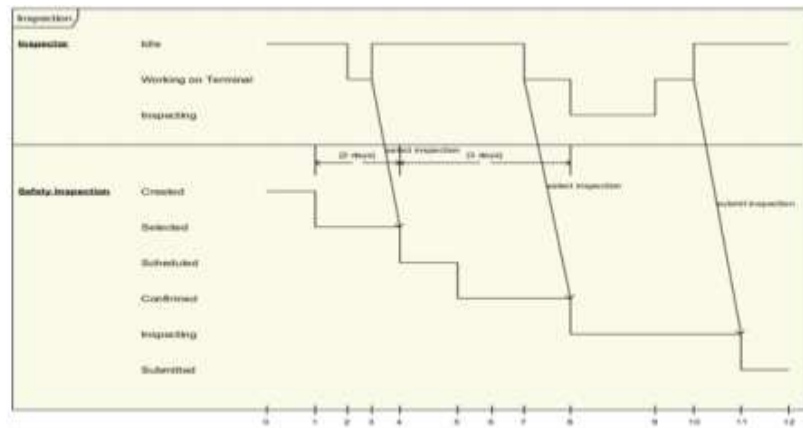
Gambar II.12 Notasi *Sequence diagram*

Sumber: Mulawarman; 2011: 5

13. *Timing diagram*

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

Adapun gambar notasi *Timing diagram* dapat dilihat pada gambar II.13.



Gambar II.13 Notasi *Timing diagram*

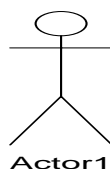
Sumber: Mulawarman; 2011: 6

II.5.3.1. 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.

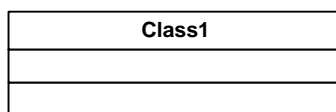
Adapun gambar *Actor* dapat dilihat pada gambar II.14.



Gambar II.14. Notasi *actor* pada UML

Sumber: Mulawarman; 2011: 6

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.
3. Adapun gambar *Class diagram* dapat dilihat pada gambar II.15.



Gambar II.15 Notasi *class* pada UML

Sumber: Mulawarman; 2011: 6

4. *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.
- Adapun gambar *use case* dapat dilihat pada gambar II.16.



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

Sumber: Mulawarman; 2011: 6

5. *Realization* menunjukkan hubungan bahwa elemen yang ada di bagian tanpa panah akan merealisasikan apa yang dinyatakan oleh elemen yang ada di bagian dengan panah.
- Adapun gambar *Realization* dapat dilihat pada gambar II.17.



Gambar II.17 Notasi *Realization* pada UML

Sumber: Mulawarman; 2011: 6

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

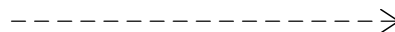
Adapun *Interaction* gambar dapat dilihat pada gambar II.18.



Gambar II.18 Notasi *Interaction* pada UML

Sumber: Mulawarman; 2011: 6

7. *Dependency* merupakan relasi yang menunjukkan 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).
8. Adapun *Dependency* gambar dapat dilihat pada gambar II.18.



Gambar II.19 Notasi *Dependency* pada UML

Sumber: Mulawarman; 2011: 6

9. *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.

Adapun gambar *Note* dapat dilihat pada gambar II.19.



Gambar II.20 Notasi *Note* pada UML

Sumber: Mulawarman; 2011: 6

10. *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).

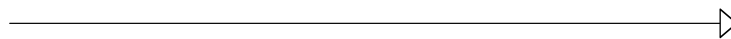
Adapun gambar *Association* dapat dilihat pada gambar II.20.

Gambar II.21 Notasi *Association* pada UML

Sumber: Mulawarman; 2011: 7

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

Adapun gambar *Association* dapat dilihat pada gambar II.21.

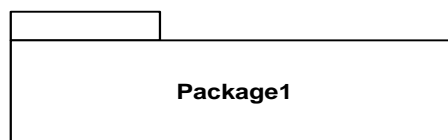


Gambar II.22 Notasi *Generalization* pada UML

Sumber: Mulawarman; 2011: 7

12. *Package* adalah mekanisme pengelompokan yang digunakan untuk menandakan pengelompokan elemen-elemen model.

Adapun gambar *Package* dapat dilihat pada gambar II.22.

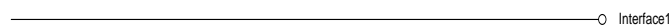


Gambar II.23 Notasi *Package* pada UML

Sumber: Mulawarman; 2011: 7

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

Adapun gambar *Association* dapat dilihat pada gambar II.23.



Gambar II.24 Notasi *Interface* pada UML

Sumber: Mulawarman; 2011: 7

II.5.3.2. 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.5.3.3 Pemetaan Kardinalitas

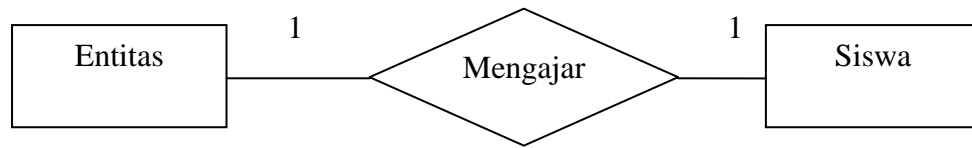
Pemetaan kardinalitas menyatakan jumlah entitas di mana entitas lain dapat dihubungkan ke entitas tersebut melalui sebuah himpunan relasi. (Tri Octafian; 2011:150-151).

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.25 Hubungan *One To One*.

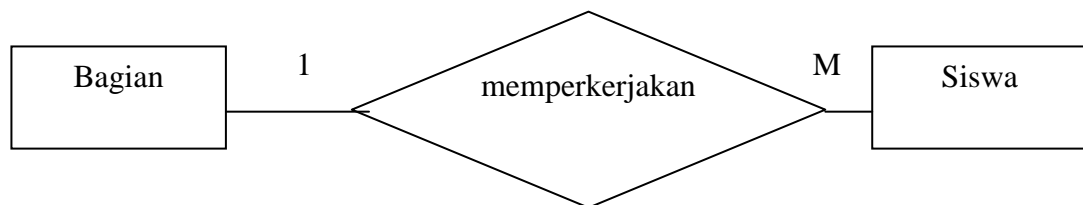
Sumber: Tri Octafian; 2011:150

2. *One to Many/ Many to One*

Sebuah entitas pada A berhubungan dengan lebih dari satu entitas pada B dan sebuah 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.26 *One To Many*

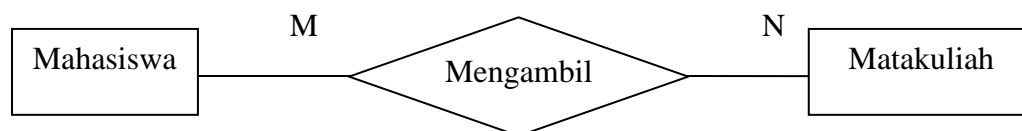
Sumber: Tri Octafian; 2011:151

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.27 Hubungan *Many To Many*

Sumber: Tri Octafian ; 2011:152

II.6. Pengertian Metode AHP (*Analitycal Hirearcy Process*)

Pada dasarnya, proses pengambilan keputusan adalah memilih suatu alternatif. Peralatan utama AHP adalah sebuah hirarki fungsional dengan input utamanya persepsi manusia. Keberadaan hirarki memungkinkan dipecahnya masalah kompleks atau tidak terstruktur dalam sub-sub masalah, lalu menyusunnya menjadi suatu bentuk hirarki. (Kusrini ; 2007 : 133).

AHP memiliki banyak keunggulan dalam menjelaskan proses pengambilan keputusan. Salah satunya adalah dapat digambarkan secara grafis sehingga mudah dipahami oleh semua pihak yang terlibat dalam pengambilan keputusan.

II.6.1. Prinsip Dasar AHP

Dalam menyelesaikan permasalahan dengan metode AHP ada beberapa prinsip yang harus dipahami, diantaranya:

1. Membuat hirarki.

Sistem yang kompleks bisa dipahami dengan memecahkannya menjadi elemen-elemen pendukung, menyusun elemen secara hirarki, dan menggabungkannya atau mensintesisnya.

2. Penilaian kriteria dan alternatif.

Kriteria dan alternatif dilakukan dengan perbandingan berpasangan. Menurut Saaty (1988), untuk berbagai persoalan, skala 1 sampai 9 adalah skala terbaik untuk mengekspresikan pendapat. Nilai dan definisi pendapat kualitatif dari skala perbandingan Saaty bisa diukur menggunakan tabel analisis.

3. *Synthesis of priority* (menentukan Prioritas).

Untuk setiap kriteria dan alternatif, perlu dilakukan perbandingan berpasangan (*pairwise comparisons*). Nilai-nilai perbandingan relatif dari seluruh alternatif kriteria bisa disesuaikan dengan *judgement* yang telah ditentukan untuk menghasilkan bobot dan prioritas. Bobot dan prioritas dihitung dengan memanipulasi matriks atau melalui penyelesaian persamaan matematika.

4. *Logical Consistency* (Konsistensi Logis)

Konsistensi memiliki dua makna. Pertama, objek-objek yang serupa bisa dikelompokkan sesuai dengan keseragaman dan relevansi. Kedua, menyangkut tingkat hubungan antarobjek yang didasarkan pada kriteria tertentu.

II.6.2. Prosedur AHP

Pada dasarnya, prosedur atau langkah-langkah dalam metode AHP meliputi:

1. Mendefinisikan masalah dan menentukan solusi yang diinginkan, lalu menyusun hierarki dari permasalahan yang dihadapi.
2. Menentukan prioritas elemen.
 - a. Langkah pertama adalah membuat perbandingan pasangan, yaitu membandingkan elemen secara berpasangan sesuai dengan kriteria yang diberikan.
 - b. Matriks perbandingan berpasangan diisi menggunakan bilangan untuk merepresentasikan kepentingan relatif dari suatu elemen terhadap elemen lainnya.

Adapun tabel yang digunakan dalam menilai perbandingan pasangan adalah sebagai berikut :

Tabel.II.1 Skala Perbandingan Pasangan

Intensitas Kepentingan	Keterangan
1	Kedua elemen sama pentingnya
3	Elemen yang satu sedikit lebih penting daripada elemen yang lainnya
5	Elemen yang satu sedikit lebih penting daripada elemen yang lainnya
7	Satu elemen jelas lebih mutlak penting daripada lainnya
9	Satu elemen jelas lebih mutlak penting daripada lainnya
2,4,6,8	Nilai – nilai antara dua nilai pertimbangan yang berdekatan
Kebalikan	Jika aktivitas i mendapat suatuangka dibandingkan dengan aktivitas i, maka i memiliki nilai kebalikannya dibandingkan dengan i

Sumber: Kusrini; 2007: 134

3. Sintesis Hal-hal yang dilakukan dalam langkah ini adalah :

- a. Menjumlahkan nilai-nilai dari setiap kolom pada matriks.
- b. Membagi setiap nilai dari kolom dengan total kolom yang bersangkutan untuk memperoleh normalisasi matriks.
- c. Menjumlahkan nilai-nilai dari setiap baris dan membaginya dengan jumlah elemen untuk mendapatkan nilai rata-rata.

4. Mengukur konsistensi

Dalam pembuatan keputusan, penting untuk mengetahui seberapa baik konsistensi yang ada karena kita tidak menginginkan keputusan berdasarkan pertimbangan dengan konsistensi yang rendah. Hal-hal yang dilakukan dalam langkah ini adalah:

- a. Mengalikan setiap nilai pada kolom pertama dengan prioritas relative elemen pertama, nilai pada kolom kedua dengan prioritas relative elemen kedua, dan seterusnya.
- b. Menjumlahkan setiap baris.
- c. Hasil dari penjumlahan baris dibagi dengan elemen prioritas relative yang bersangkutan.
- d. Menjumlahkan hasil bagi diatas dengan banyaknya elemen yang ada, hasilnya disebut λ maks.

5. Menghitung *Consistency Index* (CI) dengan rumus :

$$CI = (\lambda \text{ maks} - n) / n$$

Dimana n = banyaknya elemen

6. Menghitung *Consistency Ratio* (CR) dengan rumus : $CR = CI / RC$

Dimana CR = Consistency Ratio

CI = Consistency Index

IR = Indeks Random Consistency

7. Memeriksa konsistensi hierarki. Jika nilainya lebih dari 10%, maka penilaian dari data judgment harus diperbaiki. Namun jika rasio konsistensi (CI/IR) kurang atau sama dengan 0,1, maka hasil perhitungan bisa dinyatakan benar.

Penentuan indeks random konsistensi mengacu pada tabel berikut ini:

Tabel.II.2 Daftar Random Konsistensi

Ukuran Matriks	Konsistensi acak
1	0,00
2	0,00
3	0,58
4	0,90
5	1,12
6	1,24
7	1,32
8	1,41
9	1,45
10	1,49

Sumber: Kusri; 2007: 136

Menghitung CI dengan rumus:

$$CI = \frac{\alpha \max}{n-1} .$$

Menghitung CR dengan rumus:

$$CR = \frac{CI}{RC}$$

Menghitung Lamda max dengan rumus :

$$\alpha \max = \frac{\sum \alpha}{n}$$