

BAB II

TINJAUAN PUSTAKA

II.1. Sistem Informasi

II.1.1. Sistem

Menurut O'Brien (2006 : 29), sistem adalah sekelompok komponen yang saling berhubungan, bekerja bersama untuk mencapai tujuan bersama dengan menerima input serta menghasilkan output dalam proses transformasi yang teratur. Sistem memiliki tiga fungsi dasar yang saling berinteraksi menurut O'Brien (2006 : 29) :

1. Input melibatkan penangkapan dan perakitan berbagai elemen yang memasuki sistem untuk diproses.
2. Pemrosesan melibatkan proses transformasi yang mengubah input menjadi output.
3. Output melibatkan perpindahan elemen yang telah diproduksi oleh proses transformasi ke tujuan akhirnya.

Selain memiliki fungsi dasar, sistem mempunyai karakteristik. Gondodiyoto (2007 : 108) menyebutkan karakteristik dari sistem sebagai berikut:

1. Sistem adalah kumpulan elemen-elemen atau sumber daya yang saling berkaitan secara terpadu, terintegrasi dalam suatu hubungan hirarkis tertentu, dan bertujuan untuk mencapai tujuan tertentu.
2. Memiliki karakteristik khusus yang menjadi "jiwa atau roh" dari sistem tersebut.

3. Mempunyai sasaran yang akan dicapai.
4. Konstruksi sistem terdiri dari: masukan-proses-keluaran.
5. Sistem memerlukan pengendalian. Pengendalian merupakan proses pengaturan yang dipergunakan sistem untuk mengkoreksi setiap penyimpangan dari suatu rangkaian langkah untuk menuju sasaran.
6. Sistem memiliki pengguna. Setiap sistem harus mengarah sub sistemnya agar dapat mencapai sasaran. Sasaran sistem sebagai ukuran penentu keberhasilan suatu sistem.
7. Sistem mempunyai keterbatasan.
8. Terdiri dari subsistem-subsistem yang membentuk suatu jaringan terpadu. Setiap sistem terdiri dari lebih dari satu komponen yang saling terjalin satu sama lain disebut subsistem, yang menjalankan peran tertentu dan menjadi bagian di dalam sistem yang lebih besar. Dalam sistem, subsistem-subsistem tidak berdiri sendiri melainkan saling berinteraksi serta saling berhubungan membentuk satu kesatuan terpadu sehingga tujuan atau sasaran sistem tersebut dapat tercapai.

II.1.2. Informasi

Husein dan Amin (2002 : 9) mendeskripsikan, informasi adalah data yang telah diolah menjadi suatu bentuk yang mempunyai arti dan bermanfaat bagi manusia. Sehingga informasi lebih dikenal sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna bagi penerimanya, dimana informasi

tersebut menggambarkan tentang suatu kejadian nyata yang dapat dipahami dan dapat digunakan untuk pengambilan keputusan sekarang maupun masa depan.

Dalam perusahaan, informasi menggambarkan tentang kinerja perusahaan. Ada dua tolok ukur untuk menggambarkan kinerja karyawan dalam perusahaan, yaitu:

1. Efisiensi

Efisiensi adalah penggunaan sumber daya secara minimum guna pencapaian hasil yang optimum (Wibisono, 2010).

Efisiensi menganggap bahwa tujuan-tujuan yang benar telah ditentukan dan berusaha untuk mencari cara-cara yang paling baik untuk mencapai tujuan-tujuan tersebut. Sehingga efisien lebih mengarah melakukan sesuatu dengan benar. Efisiensi hanya dapat dievaluasi dengan penilaian-penilaian relatif, membandingkan antara masukan dan keluaran yang diterima.

2. Efektifitas

Definisi efektif adalah suatu pencapaian tujuan secara tepat atau memilih tujuan-tujuan yang tepat dari serangkaian alternatif atau pilihan cara dan menentukan pilihan dari beberapa pilihan lainnya (Wibisono, 2010). Efektifitas lebih mengarah pada hasil akhir dari kegiatan yang berjalan dalam perusahaan. Jika hasil kegiatan yang dilakukan perusahaan semakin mendekati sasaran yang ditentukan perusahaan maka semakin tinggi efektifitasnya. Sehingga efektifitas dapat juga diartikan sebagai pengukur keberhasilan dalam pencapaian tujuan yang telah ditentukan.

Ada beberapa kriteria dari informasi menurut Gondodiyoto (2007:111), antara lain:

1. Akurat, *reliable* (dapat dipercaya). Informasi tersebut kecil kesalahannya sehingga tidak menyesatkan penggunaannya.
2. Relevan (cocok atau sesuai). Informasi harus memberikan arti kepada penggunaannya atau informasi tersebut mempunyai manfaat bagi pemakainya.
3. *Timely* (tepat waktu). Informasi disajikan tepat pada saat dibutuhkan dan bisa mempengaruhi proses pengambilan keputusan. Informasi datang kepada pengguna tidak boleh terlambat.
4. *Complete* (lengkap). Informasi yang disajikan semua data-datanya sesuai dengan kenyataan.
5. *Understandable* (dimengerti). Informasi yang disajikan dalam bentuk yang mudah dimengerti oleh pembuat keputusan.
6. *Verifiable*. Informasi yang dihasilkan tidak bias, sehingga dalam pemahamannya tidak menimbulkan kesalahpahaman.
7. *Accessible*. Informasi tersedia pada saat yang diperlukan dalam format sesuai dengan kepentingannya. Informasi dikatakan bernilai apabila manfaat dari informasi lebih efektif dibandingkan dengan biaya yang dikeluarkan untuk mendapatkannya.

II.1.3. Sistem Informasi

Sistem informasi dapat secara manual maupun terkomputerisasi. Menurut O'Brien (2006 : 5), sistem informasi merupakan kombinasi teratur apa pun dari

orang-orang, *hardware*, *software*, jaringan komunikasi, dan sumber daya data yang mengumpulkan, mengubah, dan menyebarkan informasi dalam sebuah organisasi. Organisasi menyesuaikan sistem informasinya dengan kebutuhan pemakainya. Tiga tujuan utama bagi sistem informasi, yaitu (Hall, 2001:18):

1. Untuk mendukung fungsi kepengurusan manajemen. Sistem informasi menyediakan informasi tentang kegunaan sumber daya ke pemakai eksternal melalui laporan keuangan tradisional dan laporan-laporan yang diminta lainnya.
2. Untuk mendukung pengambilan keputusan manajemen. Sistem informasi memberikan para manajer informasi yang mereka perlukan untuk melakukan tanggung jawab pengambilan keputusan.

Untuk mendukung kegiatan operasi perusahaan hari demi hari. Sistem informasi menyediakan informasi bagi personel operasi untuk membantu mereka melakukan tugas mereka setiap hari dengan efisien dan efektif. Menurut Hall (2007 : 62) bagian perusahaan yang terlibat dalam pengembangan sistem adalah praktisi sistem, pengguna akhir, dan pemegang kepentingan dalam perusahaan.

II.2. Visual Studio 2008

Visual Studio 2008 merupakan suatu perangkat lunak yang dapat digunakan untuk mengembangkan berbagai macam aplikasi yang memiliki berbagai macam tipe antara lain aplikasi *desktop* (*windows form*, *command line* (*console*)), aplikasi *web*, *windows mobile* (*Pocket PC*).

Visual Studio 2008 diluncurkan *Microsoft* pada tanggal 19 November 2007, dan memiliki lebih dari satu *kompiler*, *SDK (Software Development Kit)*, dan dokumentasi tutorial (*MSDN Library*). *Kompiler* yang dimasukkan ke dalam paket *Visual Studio 2008* antara lain *Visual C++*, *Visual C#*, *Visual Basic*, *Visual InterDev*, *Visual J++*, *Visual J#*, *Visual FoxPro*, dan *Visual SourceSafe*. Dan kesemuanya ditujukan untuk *platform .Net Framework 3.5* (Muhammad Sadeli ; 2009 : 2).

II.3. Database

Database atau basis data adalah sekumpulan data yang memiliki hubungan secara logika dan diatur dengan susunan tertentu serta disimpan dalam media penyimpanan komputer. Data 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 (Wahana Komputer ; 2010 : 24).

II.4. SQL

SQL Server 2008 adalah 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 2008* dibuat pada saat kemajuan dalam bidang hardware sedemikian pesat. Oleh

karena itu sudah dapat dipastikan bahwa SQL Server 2008 membawa beberapa terobosan dalam bidang pengolahan dan penyimpanan data.

1. Kebutuhan Hardware

Adapun hardware minimal yang dibutuhkan adalah sebagai berikut:

- a. Processor \geq 1GHz
- b. Memory \geq 512 MB
- c. OS = Windows XP 32 bit-Windows

Sedangkan untuk jaringannya yang diperlukan adalah:

- a. Shared memory
- b. TCP/IP
- c. Named Pipes
- d. VIA (Virtual Interface Adapter)

2. Versi-versi SQL Server 2008

Microsoft merilis SQL Server 2008 dalam beberapa versi yang disesuaikan dengan segment-segment pasar yang dituju. Versi-versi tersebut adalah sbb:

- a. Versi 32-bit (x86), yang biasanya digunakan untuk komputer dengan single processor 32 bit dan sistem operasi windows XP.data
- b. Versi 64-bit (x64), yang biasanya digunakan untuk komputer dengan lebih dari satu processor (Misalnya: Core 2 Duo) dan system operasi 64 bit seperti Windows Xp 64, Vista, dan Windows 7 (Wahana Komputer ; 2010 : 2-3).

II.5. Normalisasi

Normalisasi adalah suatu proses yang digunakan untuk menentukan pengelompokan atribut-atribut dalam sebuah relasi sehingga diperoleh relasi yang berstruktur baik. Dalam hal ini yang dimaksud dengan relasi yang berstruktur baik adalah relasi yang memenuhi dua kondisi berikut:

1. Mengandung redundansi sesedikit mungkin, dan
2. Memungkinkan baris-baris dalam relasi disisipkan, dimodifikasi, dan dihapus tanpa menimbulkan kesalahan atau ketidak konsistenan (Abdul Kadir ; 2009 : 116).

Normalisasi sendiri dilakukan melalui sejumlah langkah. Setiap langkah berhubungan dengan bentuk normal (*normal form*) tertentu. Dalam hal ini yang disebut bentuk normal adalah “suatu keadaan relasi yang dihasilkan oleh penerapan aturan-aturan sederhana yang berhubungan dengan dependensi fungsional terhadap relasi tersebut” (Hoffer, dkk ; 2005). Yang dimaksud dengan aturan-aturan tersebut dan juga istilah dependensi fungsional akan dibahas belakangan. Bentuk normal dalam normalisasi dapat berupa:

- a. Bentuk normal pertama (1NF / *First normal form*)
- b. Bentuk normal kedua (2NF / *Second normal form*)
- c. Bentuk normal ketiga (3NF / *Third normal form*)
- d. Bentuk normal Boyce-Codd (BCNF / *Boyce-Codd normal form*)
- e. Bentuk normal keempat (4NF / *Fourth normal form*)
- f. Status Bentuk normal kelima (5NF / *Fifth normal form*) (Abdul Kadir ; 2009 : 116-117).

1. Bentuk Normal Pertama (1NF)

Contoh yang kita gunakan di sini adalah sebuah perusahaan yang mendapatkan barang dari sejumlah pemasok. Masing-masing pemasok berada pada satu kota. Sebuah kota dapat mempunyai lebih dari satu pemasok dan masing-masing kota mempunyai kode status sendiri. Masing-masing pemasok bisa menyediakan banyak barang. Tabel relasionalnya dapat ditulis sebagai berikut:

PEMASOK (p#, status, kota, b#, qty) dimana

P# : kode pemasok (kunci utama)

Status : kode status kota

Kota : nama kota

b# : barang yang dipasok

qty : jumlah barang yang dipasok

Supaya bisa digabungkan jumlah barang yang dipasok (qty) secara unik dengan barang (b#) dan pemasok (p#), kita menggunakan kunci utama gabungan yang tersusun dari b# dan p#.

Sebuah tabel relasional secara defenisi selalu berada dalam bentuk normal pertama. Semua nilai pada kolom-kolomnya adalah atomik. Ini berarti kolom-kolom tidak mempunyai nilai berulang.

Meskipun berada pada 1NF, tabel pemasok mengandung data berulang. Sebagai contoh, informasi tentang lokasi pemasok dan status lokasi harus diulang untuk setiap barang yang dipasok. Perulangan menyebabkan apa yang disebut

update anomalies. *Update anomalies* adalah masalah yang timbul ketika informasi ditambahkan, dihapus, atau di-*update*.

2. Bentuk Normal Kedua (2NF)

Defenisi bentuk normal kedua menyatakan bahwa tabel dengan kunci utama gabungan hanya dapat berada pada 1NF, tetapi tidak ada pada 2NF. Sebuah tabel relasional berada pada bentuk normal kedua jika dia berada pada 1NF dan setiap kolom bukan kunci yang sepenuhnya tergantung pada kunci utama. Ini berarti setiap kolom bukan kunci harus tergantung pada seluruh kolom yang membentuk kunci utama. Proses mengubah tabel 1NF menjadi 2NF adalah:

- a. Tentukan senbarang kolom penentu selain kunci gabungan dan kolom-kolom yang ditentukannya.
- b. Buat dan beri tabel baru untuk masing-masing penentu dan kolom-kolom yang ditentukannya.
- c. Pindahkan kolom-kolom yang ditentukan dari tabel asal ke tabel baru. Penentu akan menjadi kunci utama pada tabel baru.
- d. Hapus kolom yang baru dipindahkan dari tabel asal, kecuali penentu yang berfungsi sebagai kunci tamu.
- e. Tabel asal bisa diberi nama baru.

3. Bentuk Normal Ketiga (3NF)

Bentuk normal ketiga mengharuskan semua kolom pada tabel relasional tergantung hanya pada kunci utama. Secara defenisi, sebuah tabel berada pada normal ketiga (3NF) jika tabel sudah berada pada 2NF dan setiap kolom bukan

kunci tidak tergantung secara transitif dengan kunci utamanya. Dengan kata lain, semua atribut bukan kunci tergantung secara fungsional hanya pada kunci utamanya (Janer Simarmata & Imam Paryudi ; 2006 : 79-82).

II.6. Entity Relationship Diagram (ERD)

ERD adalah alat pemodelan data utama dan akan membantu mengorganisasi data dalam suatu proyek ke dalam entitas-entitas dan menentukan hubungan antar entitas. Proses memungkinkan analis menghasilkan struktur basis data yang baik sehingga data dapat disimpan dan diambil secara efisien (Janner Simarmata & Imam Paryudi ; 2006 : 67). Pada dasarnya ada 3 macam simbol yang digunakan, yaitu :

1. Entitas (*Entity*)

Entitas adalah sesuatu yang nyata atau abstrak dimana kita akan menyimpan data. Ada 4 kelas entitas yaitu misalnya pegawai, pembayaran, kampus, dan buku. Contoh suatu entitas disebut instansi, misalnya pegawai Adi, pembayaran Joko, dan lain sebagainya.

2. Relasi (*Relationship*)

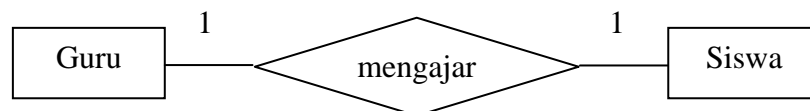
Relasi adalah hubungan alamiah yang terjadi antara satu atau lebih entitas, misalnya proses pembayaran pegawai. Kardinalitas menentukan kejadian suatu entitas untuk satu kejadian pada entitas yang berhubungan. Misalnya mahasiswa bisa mengambil banyak mata kuliah.

3. Atribut (*Attribute*)

Atribut adalah ciri umum semua atau sebagian besar instansi pada entitas tersebut. Sebutan lain atribut adalah properti, elemen data, dan field. Misalnya nama, alamat, nomor pegawai, dan gaji adalah atribut entitas pegawai. Sebuah atribut atau kombinasi atribut yang mengidentifikasi satu dan hanya satu instansi suatu entitas disebut kunci utama atau pengenal. Misalnya, nomor pegawai adalah kunci utama untuk pegawai (Janner Simarmata & Imam Paryudi ; 2006 : 67).

Jenis-jenis hubungan:

- a. *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.



Gambar II.1. Relational 1 to 1

Sumber : (Janner Simarmata & Imam Paryudi ; 2006 : 63-64)

- b. *One to Many*

Sebuah entitas pada A berhubungan dengan nol atau lebih entitas pada B. Sebuah entitas pada B dapat dihubungkan dengan paling banyak satu entitas pada A.



Gambar II.2. Relational One to Many

Sumber : (Janner Simarmata & Imam Paryudi ; 2006 : 64-65)

c. *Many to One*

Sebuah entitas pada A berhubungan dengan paling banyak satu entitas pada B. Sebuah entitas pada B dapat dihubungkan dengan nol atau lebih entitas pada A.

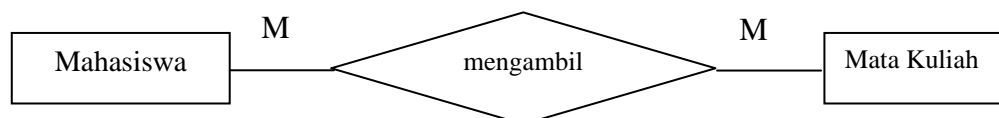


Gambar II.3. Relational Many to One

Sumber : (Janner Simarmata & Imam Paryudi ; 2006 : 65)

d. *Many to Many*

Sebuah entitas pada A berhubungan dengan nol atau lebih entitas pada B dan sebuah entitas pada B dapat dihubungkan nol atau lebih entitas pada A.



Gambar II.4. Relational Many to Many

Sumber : (Janner Simarmata & Imam Paryudi ; 2006 : 65-66)

II.7. UML (*Unified Modelling Language*)

UML (Unified Modelling Language) adalah bahasa pemodelan standar. (Chonoles, 2003: bab 1) mengatakan sebagai bahasa, berarti *UML* memiliki sintaks dan semantik. Ketika kita membuat model menggunakan konsep *UML* ada aturan - aturan yang harus diikuti. Bagaimana elemem pada model – model yang dibuat berhubungan satu dengan yang lainnya harus mengikuti standar yang ada. *UML* bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya. *UML* diaplikasikan untuk maksud tertentu, biasanya antara lain untuk:

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasi sistem yang ada, proses – proses dan organisasinya (Prabowo Pudjo Widodo, Herlawati ; 2011 : 6-7).

II.7.1. Tujuan Pemanfaatan UML

Berikut tujuan utama dalam desain UML adalah:

1. Menyediakan bagi pengguna (analisis dan desain sistem) suatu bahasa pemodelan visual yang ekspresif sehingga mereka dapat mengembangkan dan melakukan pertukaran model data yang bermakna.
2. Menyediakan mekanisme yang spesialisasi untuk memperluas konsep inti.
3. Karena merupakan bahasa pemodelan visual dalam proses pembangunannya maka UML bersifat independen terhadap bahasa pemrograman tertentu.
4. Memberikan dasar formal untuk pemahaman bahasa pemodelan.

5. Mendorong pertumbuhan pasar terhadap penggunaan alat desain sistem yang berorientasi objek (OO).
6. Mendukung konsep pembangunan tingkat yang lebih tinggi seperti kolaborasi, kerangka, pola dan komponen terhadap suatu sistem.
7. Memiliki integrasi praktek terbaik (Haviluddin ; 2011 : 2).

II.7.2. Komponen-komponen 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.

Komponen atau notasi UML diturunkan dari 3 (tiga) notasi yang telah ada sebelumnya yaitu Grady Booch, OOD (*Object-Oriented Design*), Jim Rumbaugh, OMT (*Object Modelling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*).

Pada UML versi 2 terdiri atas tiga kategori dan memiliki 13 jenis diagram yaitu :

1. Struktur Diagram

Menggambarkan elemen dari spesifikasi dimulai dengan kelas, obyek, dan hubungan mereka, dan beralih ke dokumen arsitektur logis dari suatu sistem.

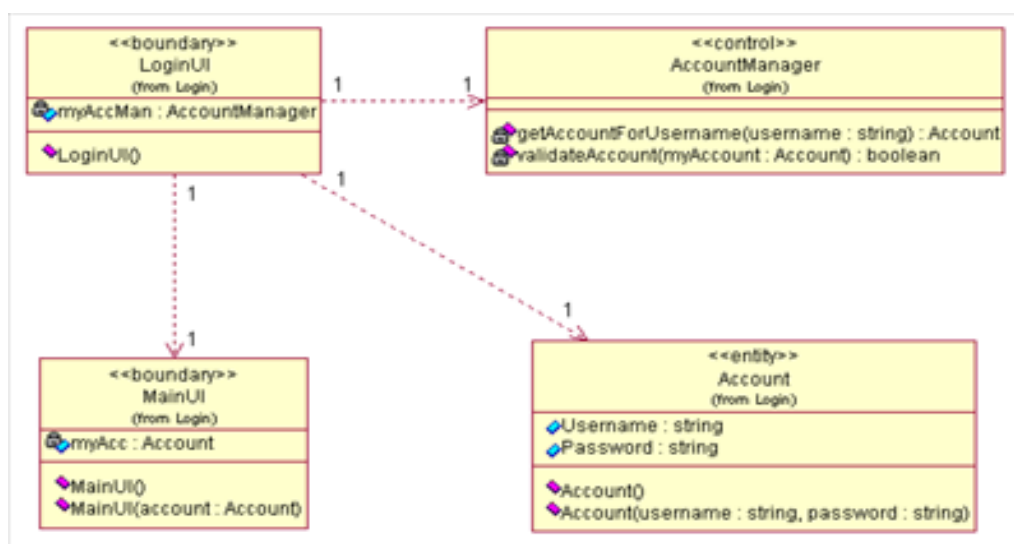
Struktur diagram dalam UML terdiri atas (Haviluddin ; 2011 : 3) :

a. 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 :

- a. Nama (dan *stereotype*)
- b. Atribut
- c. Metoda

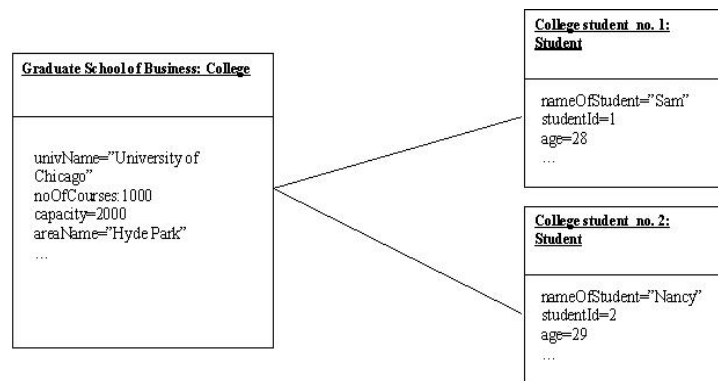


Gambar II.5. Notasi *Class Diagram*

Sumber : (Haviluddin ; 2011 : 3)

b. *Object diagram*

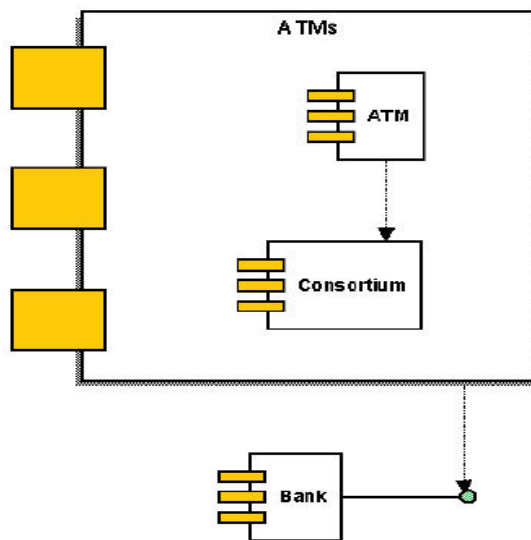
Object diagram menggambarkan kejelasan kelas dan warisan dan kadang-kadang diambil ketika merencanakan kelas, atau untuk membantu pemangku kepentingan non-program yang mungkin menemukan diagram kelas terlalu abstrak. Berikut notasi *object diagram*.



Gambar II.6. Notasi Object Diagram
 Sumber : (Haviluddin ; 2011 : 3)

c. *Component diagram*

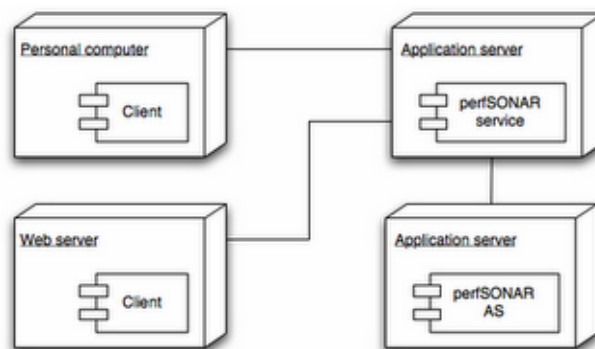
Component diagram menggambarkan struktur fisik dari kode, pemetaan pandangan logis dari kelas proyek untuk kode aktual di mana logika ini dilaksanakan.



Gambar II.7. Notasi Component Diagram
 Sumber : (Haviluddin ; 2011 : 3)

d. *Deployment diagram (Collaboration diagram in version 1.x)*

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.

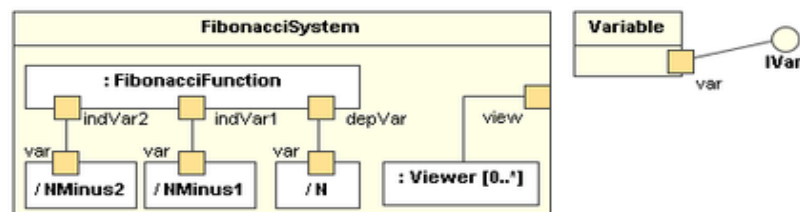


Gambar II.8. Notasi *Deployment Diagram*

Sumber : (Haviluddin ; 2011 : 4)

e. *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.

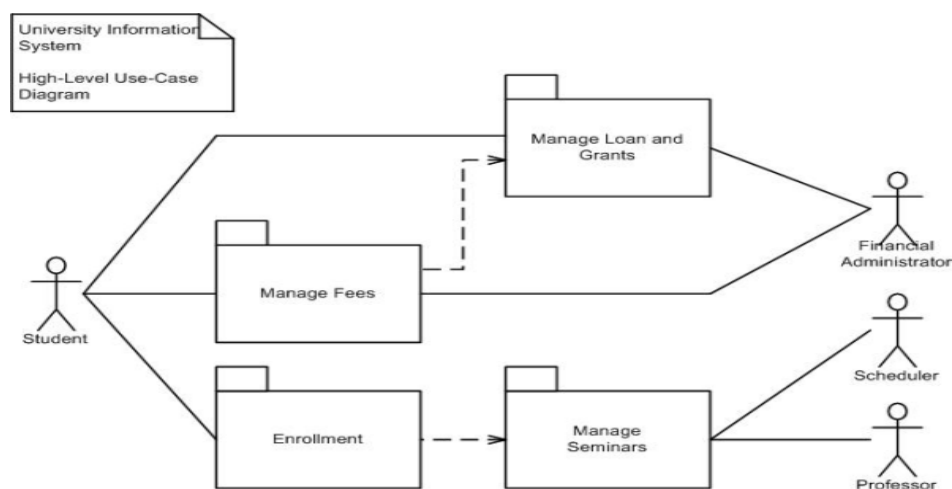


Gambar II.9. Notasi *Composite Diagram*

Sumber : (Haviluddin ; 2011 : 4)

f. *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.



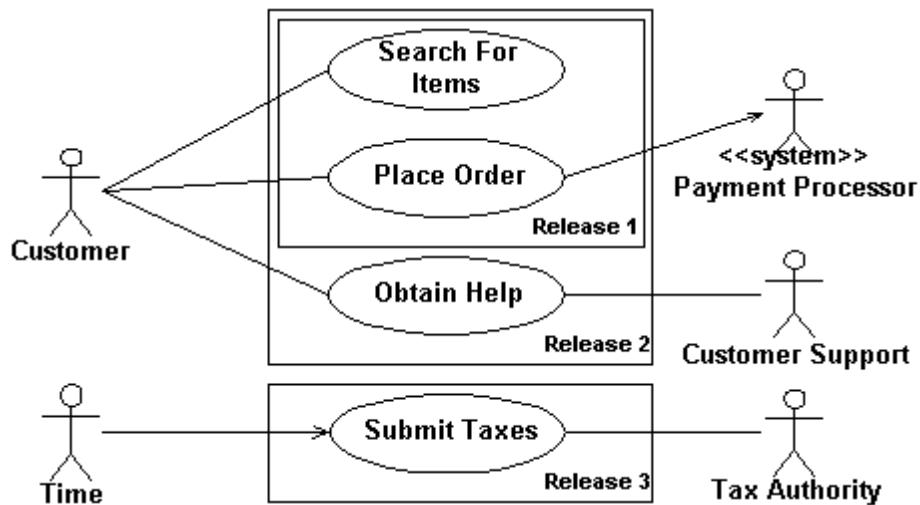
Gambar II.10. Notasi *Package Diagram*

Sumber : (Havaluddin ; 2011 : 4)

g. *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*. *Use Case* memiliki dua istilah :

- a. *System use case*; interaksi dengan sistem.
- b. *Business use case*; interaksi bisnis dengan konsumen atau kejadian nyata



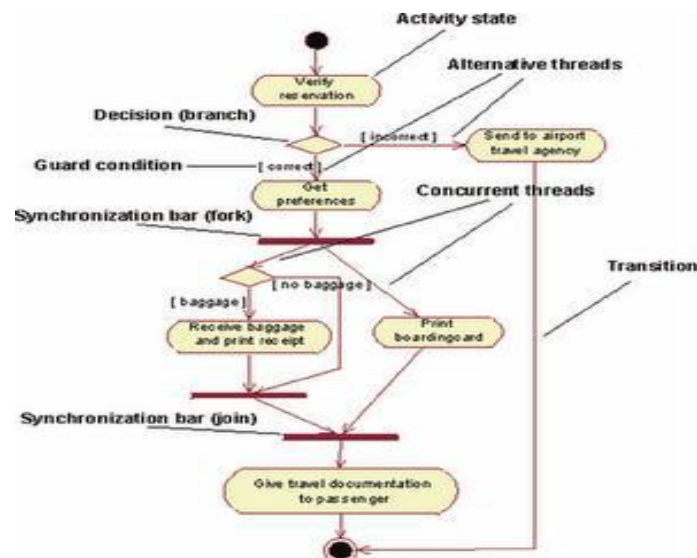
Gambar II.11. Notasi Use Case Diagram

Sumber : (Haviluddin ; 2011 : 4)

h. Activity diagram

Menggambarkan aktifitas-aktifitas, objek, *state*, transisi *state* dan *event*.

Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas

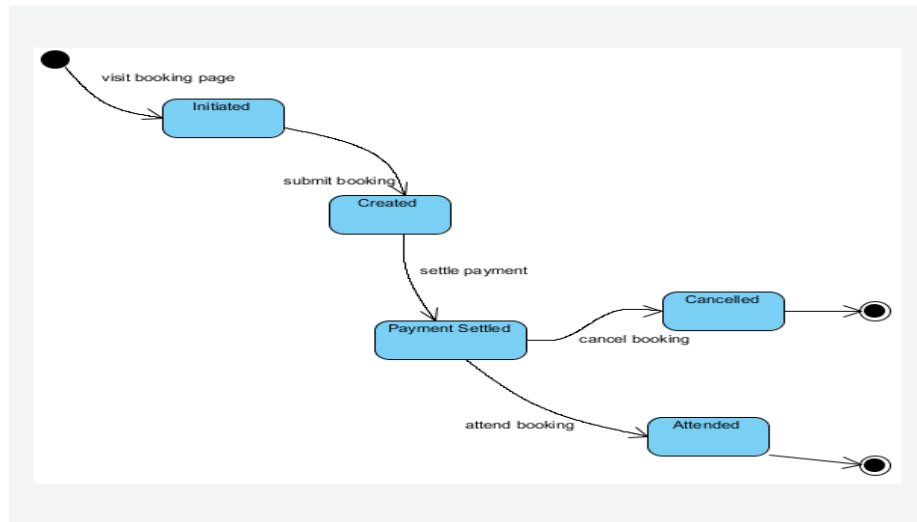


Gambar II.12. Notasi Activity Diagram

Sumber : (Haviluddin ; 2011 : 4)

i. *State Machine* diagram (*State chart* diagram in diagram in version 1.x)

Menggambaran *state*, transisi *state* dan *event*.

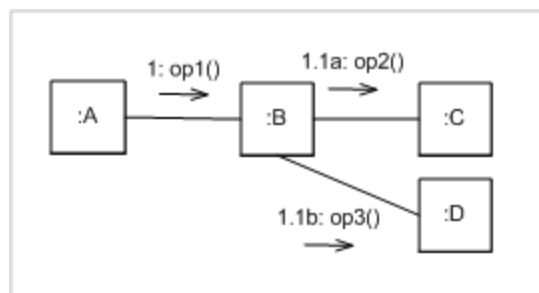


Gambar II.13. Notasi *State Machine* Diagram

Sumber : (Haviluddin ; 2011 : 5)

j. *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.

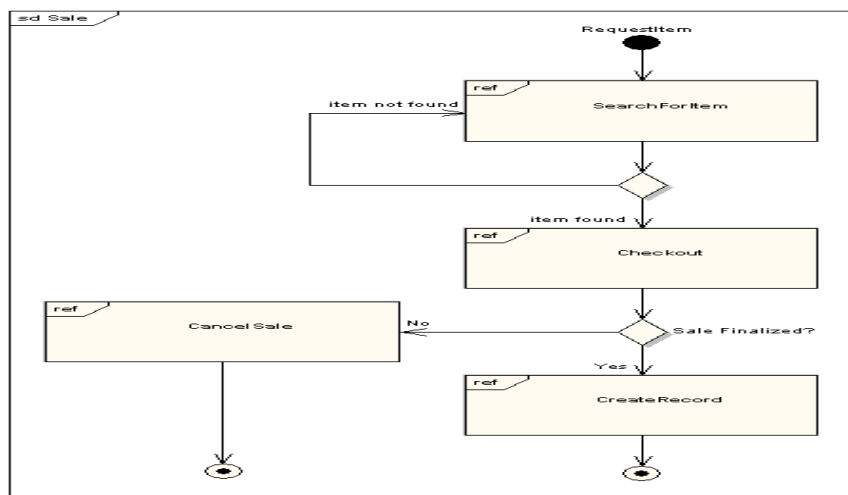


Gambar II.14. Notasi *Communication* Diagram

Sumber : (Haviluddin ; 2011 : 3)

k. *Interaction Overview* diagram

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

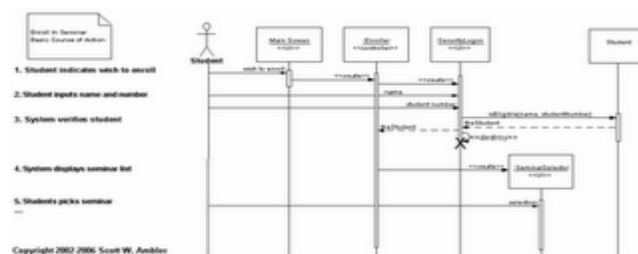


Gambar II.15. Notasi *Overview Diagram*

Sumber : (Haviluddin ; 2011 : 3)

l. *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.

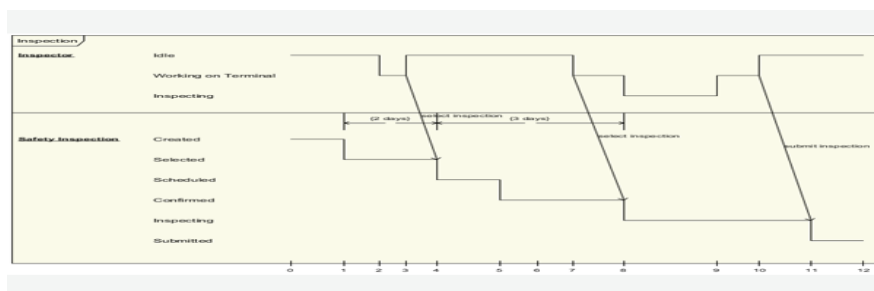


Gambar II.16. Notasi *Sequence Diagram*

Sumber : (Haviluddin ; 2011 : 5)

m. *Timing* diagram

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



Gambar II.17. Notasi *Timing Diagram*
Sumber : (Haviluddin ; 2011 : 6)

II.8. K-Means

Data mining merupakan ekstraksi informasi yang tersirat dalam sekumpulan data. Menurut Turban dkk (Kusrini & Luthfi, 2009:3), data mining merupakan sebuah proses untuk menggali kumpulan data dan menemukan informasi di dalamnya. Penggalan data ini dapat dilakukan dengan menggunakan metode statistik, matematika, kecerdasan buatan, maupun machine learning, di mana proses ini terjadi secara otomatis. Penggalan data ini dilakukan pada sekumpulan data yang besar untuk menemukan pola atau hubungan yang ada dalam kumpulan data tersebut (Kusrini & Luthfi, 2009:3). Hasil penemuan yang diperoleh setelah proses penggalan data ini, kemudian dapat digunakan untuk analisis yang lebih lanjut, dalam kaitannya dengan pengambilan keputusan yang harus dilakukan pihak manajemen suatu organisasi atau untuk melakukan pengecekan, apakah tujuan organisasi telah tercapai atau belum.

Kegiatan data mining dapat dikategorikan ke dalam 2 golongan yaitu prediktif dan deskriptif (Kantardzic, 2003 : 2). Perbedaan dari kedua golongan tersebut terletak pada hasil akhir proses, di mana predictive data mining merupakan kegiatan untuk menemukan nilai baru yang mungkin akan muncul dari sebuah himpunan data yang dikehendaki, sementara descriptive data mining digunakan untuk menemukan pola data yang dapat dimengerti oleh manusia dari sebuah himpunan data. Beberapa tugas dari sebuah proses data mining antara lain sebagai berikut (Kantardzic, 2003 : 2):

1. *Classification* – penggalian data prediktif untuk menggolongkan data ke dalam satu dari beberapa kelas data yang telah didefinisikan
2. *Regression* – penggalian data prediktif untuk memetakan data ke variabel prediksi yang bersifat real-value
3. *Clustering* – penggalian data deskriptif untuk mengidentifikasi himpunan finit dari berbagai kategori atau cluster untuk menggambarkan data yang dianalisis
4. *Summarization* – penggalian data deskriptif yang melibatkan metode-metode untuk menemukan deskripsi dari sebuah himpunan atau subset data
5. *Dependency modeling* – penggalian data untuk menemukan model lokal yang mendeskripsikan dependensi yang signifikan antar masing-masing variabel atau nilai atau fitur yang ada dalam sebuah himpunan data
6. *Change and deviation detection* – untuk menemukan perubahan signifikan yang terjadi dalam sebuah himpunan data.

Clustering merupakan sebuah teknik penggalian data yang bersifat *unsupervised*, di mana data-data yang ada dikelompokkan ke dalam beberapa *cluster*. Pengelompokan data ini didasarkan pada kesamaan karakter atau kriteria dari data-data yang dianalisis. Data-data yang ada dalam *cluster* yang sama memiliki karakter atau kriteria yang sama, sementara data-data yang berada dalam *cluster* yang berbeda juga memiliki karakter atau kriteria yang berbeda. Ada beberapa kategori metode *clustering*, yang antara lain sebagai *hierarchical clustering* misalnya metode yang didasarkan pada penggabungan dua *cluster* yang terdekat, misalnya metode *agglomerative hierarchical clustering*. Pada kondisi dasar kelompok algoritma ini, masing-masing data yang ada dianggap sebagai sebuah *cluster*. Dan kategori lain yaitu kategori *non-hierarchical clustering* misalnya metode *k-means clustering* (Agusta, 2007 : 47). Metode *hierarchical clustering* merupakan sebuah metode yang tidak didasarkan pada optimisasi. Dalam *hierarchical clustering*, satu data dapat menjadi anggota dari beberapa *cluster* sekaligus. Sedangkan dalam metode *non-hierarchical*, keanggotaan suatu data bersifat pasti dan tegas dalam 1 *cluster* saja.

K-Means clustering merupakan salah satu metode *clustering non hirarki*. Metode ini bekerja dengan mengelompokkan data-data dengan kriteria yang sama ke dalam *cluster* yang sama, dan data dengan kriteria yang berbeda ke dalam *cluster* yang berbeda pula, dengan tujuan untuk meminimalkan variasi dalam satu *cluster* yang sama dan memaksimalkan variasi antar *cluster* (Agusta, 2007 : 47).

K-Means merupakan salah satu metode data *clustering non hirarki* yang berusaha mempartisi data yang ada ke dalam bentuk satu atau lebih *cluster*/kelompok. Metode ini mempartisi data ke dalam *cluster*/kelompok sehingga data yang memiliki karakteristik yang sama dikelompokkan ke dalam satu *cluster* yang sama dan data yang mempunyai karakteristik yang berbeda dikelompokkan ke dalam kelompok yang lain (Yudi Agusta, PhD ; 2007 : 47).

Data clustering menggunakan metode K-Means ini secara umum dilakukan dengan algoritma dasar sebagai berikut :

1. Tentukan jumlah *cluster*.
2. Alokasikan data ke dalam *cluster* secara *random*.
3. Hitung *centroid*/rata-rata dari data yang ada di masing-masing *cluster*.
4. Alokasikan masing-masing data ke *centroid*/rata-rata terdekat.
5. Kembali ke step 3, apabila masih ada data yang berpindah *cluster* atau apabila perubahan nilai *centroid*, ada yang di atas nilai *threshold* yang ditentukan atau apabila perubahan nilai pada *objective function* yang digunakan di atas nilai *threshold* yang ditentukan.

Beberapa permasalahan yang sering muncul pada saat menggunakan metode K-Means untuk melakukan pengelompokkan data adalah :

1. Ditemukannya beberapa model *clustering* yang berbeda.
2. Pemilihan jumlah *cluster* yang paling tepat.
3. Kegagalan untuk *converge*.
4. Pendeteksian *outliers*.
5. Bentuk masing-masing *cluster*.

6. Masalah *overlapping*. (Yudi Agusta, PhD ; 2007 : 51)

K-means memiliki banyak kelemahan, antara lain:

1. Bila jumlah data tidak terlalu banyak, mudah untuk menentukan *cluster* awal.
2. Jumlah *cluster*, sebanyak K, harus ditentukan sebelum dilakukan perhitungan.
3. Tidak pernah mengetahui *real cluster* dengan menggunakan data yang sama, namun jika dimasukkan dengan cara yang berbeda mungkin dapat memproduksi *cluster* yang berbeda jika jumlah datanya sedikit.
4. Tidak tahu kontribusi dari atribut dalam proses pengelompokan karena dianggap bahwa setiap atribut memiliki bobot yang sama.

Salah satu cara untuk mengatasi kelemahan itu adalah dengan menggunakan K-means clustering namun hanya jika tersedia banyak data.