

BAB II

TINJAUAN PUSTAKA

II.1 Sistem Pendukung Keputusan

II.1.1. Sistem

Sistem merupakan kumpulan dari unsur atau elemen-elemen yang saling berkaitan/berinteraksi dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu. (Asbon Hendra; 2012: 157).

II.1.2. Keputusan

Keputusan merupakan kegiatan memilih suatu strategi atau tindakan dalam pemecahan masalah. Tujuan dari keputusan adalah untuk mencapai target atau aksi tertentu yang harus dilakukan. (Mohammad Taufan AZ, dkk; 2014: 14).

II.1.3. Sistem Pendukung Keputusan

Sistem Pendukung Keputusan (SPK) secara umum didefinisikan sebagai sebuah sistem yang mampu memberikan kemampuan baik kemampuan pemecahan masalah maupun kemampuan pengkomunikasian untuk masalah semi terstruktur. Secara khusus, sistem pendukung keputusan didefinisikan sebagai sebuah sistem yang mendukung kerja seorang manajer maupun sekelompok manajer dalam memecahkan masalah semiterstruktur dengan cara memberikan

informasi ataupun usulan menuju pada keputusan tertentu. (Mohammad Taufan AZ, dkk; 2014: 13).

II.2 *Simple Multi-Attribute Rating Technique (SMART)*

SMART merupakan metode pengambilan keputusan yang multiatribut. Teknik pembuatan keputusan multiatribut ini digunakan untuk membantu stakeholder dalam memilih antara beberapa alternatif. Setiap alternatif terdiri dari sekumpulan atribut dan setiap atribut mempunyai nilai-nilai, nilai ini dirata-rata dengan skala tertentu. Setiap atribut mempunyai bobot yang menggambarkan seberapa penting ia dibandingkan dengan atribut lain.

Dengan SMART pembobotan atribut dilakukan dengan dua langkah yaitu:

1. Mengurutkan kepentingan suatu atribut dari level terburuk ke level terbaik.
2. Membuat perbandingan rasio kepentingan setiap atribut dengan atribut lain dibawahnya.

SMART lebih banyak digunakan karena kesederhanaanya dalam merespon kebutuhan pembuat keputusan dan caranya menganalisa respon. Analisa yang terlibat adalah transparan sehingga metode ini memberikan pemahaman masalah yang tinggi dan dapat diterima oleh pembuat keputusan. Pembobotan pada SMART menggunakan skala antara 0 sampai 1, sehingga mempermudah perhitungan dan perbandingan nilai pada masing-masing alternatif.

Model yang digunakan dalam SMART:

$$u(a_i) = \sum_{j=1}^m w_j u_j(a_i), \quad i = 1, 2, \dots, m$$

Keterangan:

w_j = nilai pembobotan kriteria ke-j dan k kriteria

$u(a_i)$ = nilai utility kriteria ke-i untuk kriteria ke-i

Pemilihan keputusan adalah mengidentifikasi mana dari n alternatif yang mempunyai nilai fungsi terbesar. (Yeni Kustiyahningsih: 2014; 4)

II.2.1 Teknik SMART

1. Langkah 1: menentukan jumlah kriteria
2. Langkah 2: sistem secara default memberikan skala 0-100 berdasarkan prioritas yang telah diinputkan kemudian dilakukan normalisasi.

$$\text{Normalisasi} = \frac{w_j}{\sum w_j}$$

Keterangan : w_j : bobot suatu kriteria

$\sum w_j$: total bobot semua kriteria

3. Langkah 3: memberikan nilai kriteria untuk setiap alternatif.
4. Langkah 4: hitung nilai utility untuk setiap kriteria masing-masing.
5. Langkah 5: hitung nilai akhir masing-masing. (Yeni Kustiyahningsih: 2014; 5)

II.3 Microsoft Visual Basic 2010

Microsoft Visual Basic (sering disingkat sebagai VB saja) merupakan salah satu bagian dari produk pemrograman terbaru yang dikeluarkan oleh

Microsoft, yaitu bangan terintegrasi atau IDE andalan yang dikeluarkan oleh *Microsoft*, visual studio 2010 menambahkan perbaikan – perbaikan fitur dan fitur yang lebih lengkap dibandingkan versi *Visual Studio* pendahulunya, yaitu *Microsoft Visual Studio 2008*.

Visual Studio merupakan produk pemrograman andalan dari Microsoft Corporation, yang di dalamnya berisi beberapa jenis IDE pemrograman seperti Visual Basic, Visual C++, Visual Web Developer, Visual C#, dan Visual F#. Semua IDE pemrograman tersebut sudah mendukung penuh implementasi .Net Framework 4.0 yang merupakan pengembangan dari .Net Framework 3.5. Adapun database standar yang disertakan adalah Microsoft SQL Server 2008 express. (Wahana komputer: 2010; 2)

II.4 Microsoft SQL Server 2008

SQL Server 2008 adalah sebuah terobosan baru dari Microsoft dalam bidang *database*. SQL Server adalah 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. (Wenny Widya, dkk: 2014; 3)

II.5 UML (*Unified Modeling Language*)

Unified Modelling Language (UML) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual. Juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek.

Sejarah UML sendiri terbagi dalam dua fase; sebelum dan sesudah munculnya UML. Dalam fase sebelum, UML sebenarnya sudah mulai diperkenalkan sejak tahun 1990an namun notasi yang dikembangkan oleh para ahli analisis dan desain berbeda-beda, sehingga dapat dikatakan belum memiliki standarisasi.

Fase kedua; dilandasi dengan pemikiran untuk mempersatukan metode tersebut dan dimotori oleh *Object Management Group* (OMG) maka pengembangan UML dimulai pada akhir tahun 1994 ketika Grady Booch dengan metode OOD (*Object-Oriented Design*), Jim Rumbaugh dengan metode OMT (*Object Modelling Technique*) mereka ini bekerja pada Rational Software Corporation dan Ivar Jacobson dengan metode OOSE (*Object-Oriented Software Engineering*) yang bekerja pada perusahaan Objectory Rational.

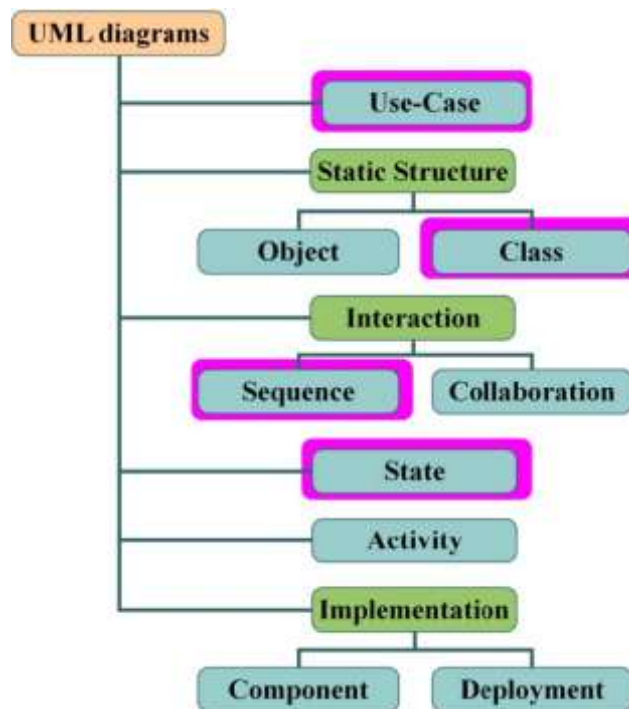
Sebagai pencetus metode-metode tersebut mereka bertiga berinisiatif untuk menciptakan bahasa pemodelan terpadu sehingga pada tahun 1996 mereka berhasil merilis UML versi 0.9 dan 0.91 melalui *Request for Proposal* (RFP) yang dikeluarkan oleh OMG.

Kemudian pada Januari 1997 IBM, *ObjecTime*, *Platinum Technology*, *Ptech*, *Taskon*, *Reich Technologies* dan *Softteam* juga menanggapi *Request for Proposal* (RFP) yang dikeluarkan oleh OMG tersebut dan menyatakan kesediaan untuk bergabung. Perusahaan-perusahaan ini menyumbangkan ide-ide mereka, dan bersama para mitra menghasilkan UML revisi 1.1. Fokus dari UML versi rilis 1.1 ini adalah untuk meningkatkan kejelasan UML Semantik versi rilis 1.0. Hingga saat ini UML versi terbaru adalah versi 2.0 (<http://www.uml.org/>).

Saat ini sebagian besar para perancang sistem informasi dalam menggambarkan informasi dengan memanfaatkan UML diagram dengan tujuan utama untuk membantu tim proyek berkomunikasi, mengeksplorasi potensi desain, dan memvalidasi desain arsitektur perangkat lunak atau pembuat program.

Secara filosofi UML diilhami oleh konsep yang telah ada yaitu konsep permodelan *Object Oriented* karena konsep ini menganalogikan sistem seperti kehidupan nyata yang didominasi oleh obyek dan digambarkan atau dinotasikan dalam simbol-simbol yang cukup spesifik. (Haviluddin; 2011: 01).

Berikut gambar dari diagram UML



Gambar II.1. Diagram UML

Sumber : Haviluddin; 2011 : 02

II.5.1. Tujuan Pemanfaatan UML

Tujuan dari penggunaan diagram seperti diungkapkan oleh Schmuller J. (2004), *“The purpose of the diagrams is to present multiple views of a system; this set of multiple views is called a model”*.

Berikut tujuan utama dalam desain UML adalah (Haviluddin; 2011: 02) :

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 praktik terbaik.

II.5.2. 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 (Haviluddin; 2011: 03) :

1. Struktur Diagram

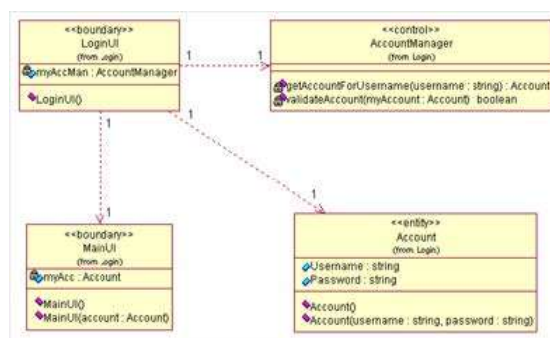
Menggambarkan elemen dari spesifikasi dimulai dengan kelas, obyek, dan hubungan mereka, dan beralih ke dokumen arsitektur logis dari suatu sistem. Struktur diagram dalam UML terdiri atas :

1) *Class diagram*

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

Class memiliki tiga area pokok :

1. Nama (dan *stereotype*)
2. Atribut
3. Metoda



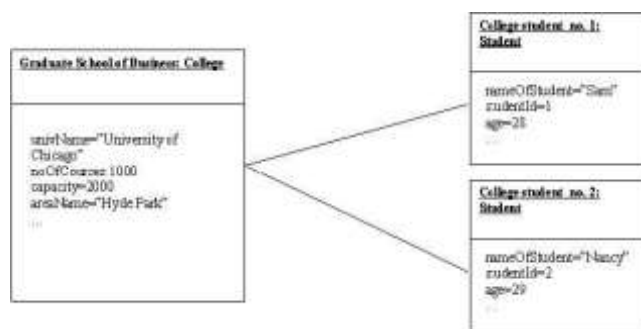
Gambar II.2. Notasi class diagram

Sumber : Haviluddin; 2011 : 03

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.

Berikut notasi *object diagram*.

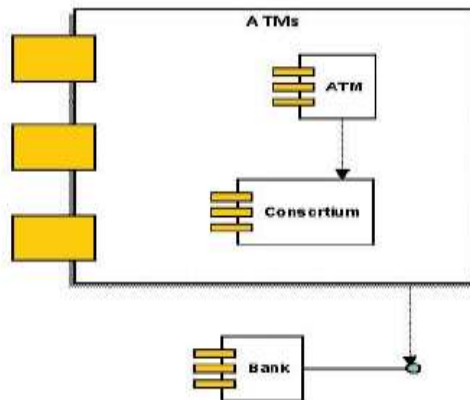


Gambar II.3. Notasi *object diagram*

Sumber : Havaluddin; 2011 : 03

3) *Component diagram*

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

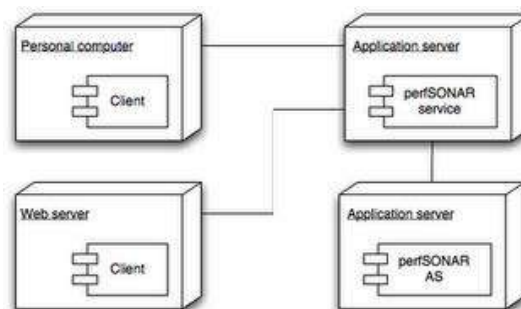


Gambar II.4. Notasi *component diagram*

Sumber : Haviluddin; 2011 : 03

4) *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.

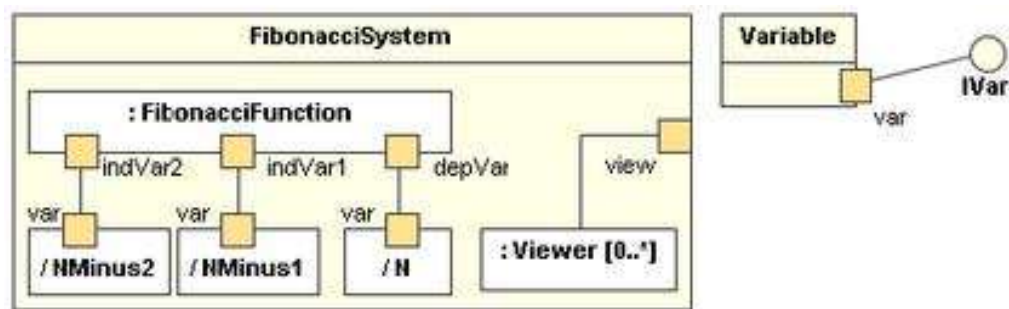


Gambar II.5. Notasi *deployment diagram*

Sumber : Haviluddin; 2011 : 04

5) *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.

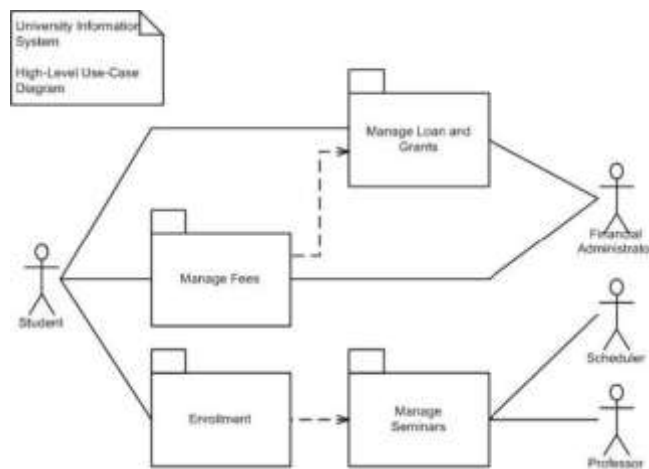


Gambar II.6. Notasi *composite diagram*

Sumber : Havaluddin; 2011 : 04

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.



Gambar II.7. Notasi *Package diagram*

Sumber : Haviluddin; 2011 : 04

2. Behavior Diagram

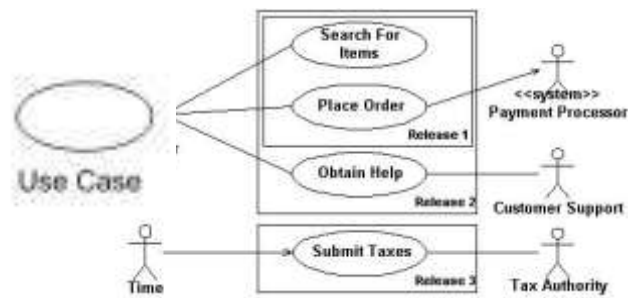
Menggambarkan ciri-ciri behavior/metode/ fungsi dari sebuah sistem atau business process. Behavior diagram dalam UML terdiri atas :

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*.

Use Case memiliki dua istilah :

1. *System use case*; interaksi dengan sistem.
2. *Business use case*; interaksi bisnis dengan konsumen atau kejadian nyata



Gambar II.8. Notasi Use case diagram

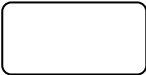
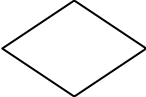
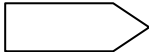
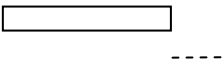
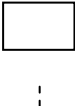
Sumber : Haviluddin; 2011 : 04

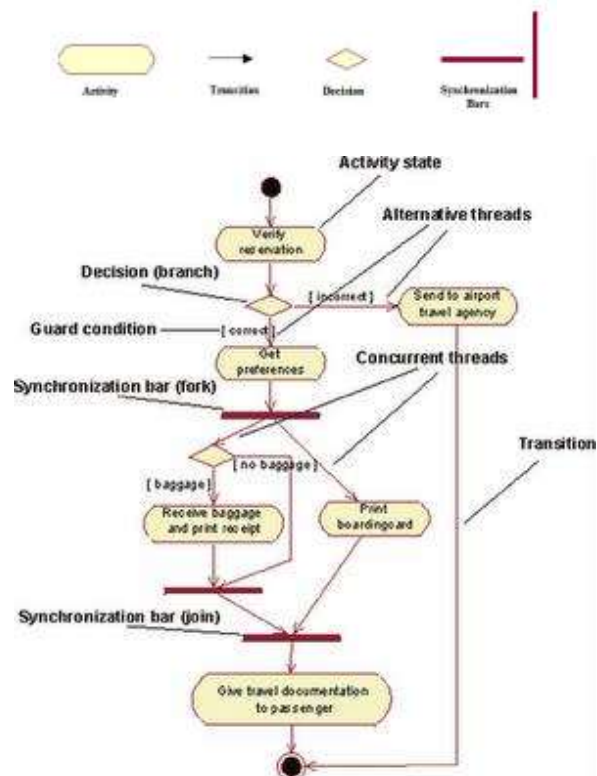
8) Activity diagram

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

Tabel II.1. Simbol-simbol dalam UML

Simbol	Keterangan
	Actor; mewakili peran orang
	Use Case; Interaksi antara system dan actor
	Sistem
	Class
	Message

	Titik awal
	Titik akhir
	Activity
	Pilihan untuk pengambilan keputusan
	Fork; digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu
	Rake; menunjukkan adanya dekomposisi
	Tanda waktu
	Tanda pengiriman
	Tanda penerimaan
	Aliran akhir (Flow Final)
	Activation (Mewakili sebuah eksekusi operasi dari participant)
	Lifeline (Setiap participant terhubung dengan garis titik-titik)
	Participant

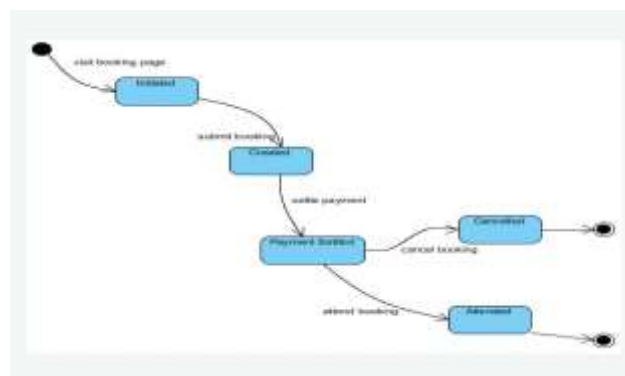


Gambar II.9. Notasi Activity diagram

Sumber : Haviluddin; 2011 : 04

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

Menggambarakan *state*, transisi *state* dan *event*.



Gambar II.10. Notasi State Machine diagram

Sumber : Haviluddin; 2011 : 05

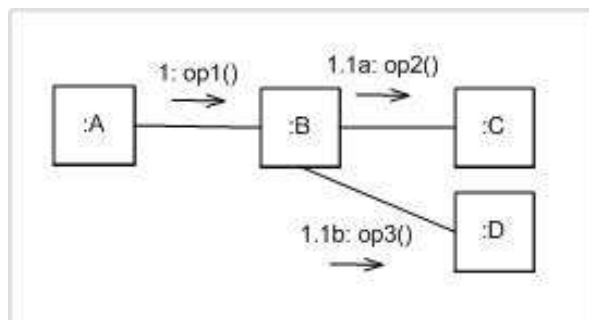
3. Interaction diagram

Bagian dari behavior diagram yang menggambarkan interaksi objek.

Interaction diagram dalam UML terdiri atas :

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.

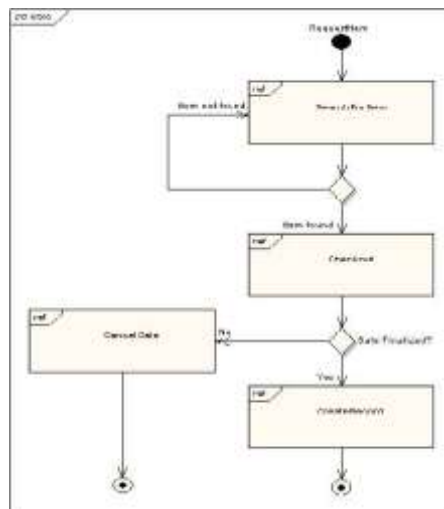


Gambar II.11. Notasi *communication diagram*

Sumber : Haviluddin; 2011 : 05

11) *Interaction Overview diagram*

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

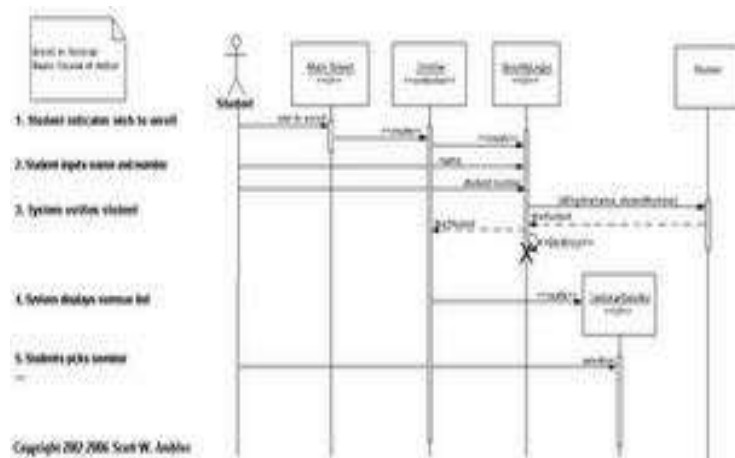


Gambar II.12. Notasi overview diagram

Sumber : Haviluddin; 2011 : 05

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*.

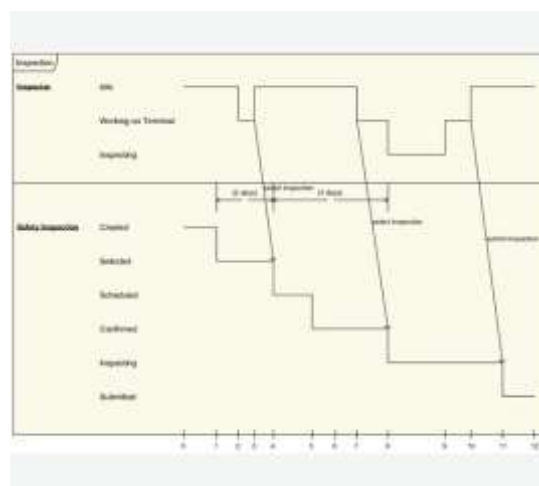


Gambar II.13. Notasi *Sequence diagram*

Sumber : Haviluddin; 2011 : 05

13) *Timing diagram*

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



Gambar II.14. Notasi *Sequence diagram*

Sumber : Haviluddin; 2011 : 06

II.6 Handphone

Handphone adalah perangkat telekomunikasi elektronik yang mempunyai kemampuan dasar yang sama dengan telepon konvensional saluran tetap, namun dapat dibawa ke mana-mana (portabel, *mobile*) dan tidak perlu disambungkan dengan jaringan telepon menggunakan kabel (nirkabel; *wireless*). Saat ini Indonesia mempunyai dua jaringan telepon nirkabel yaitu sistem GSM (*Global System for Mobile Telecommunications*) dan sistem CDMA (*Code Division Multiple Access*). Selain berfungsi untuk melakukan dan menerima panggilan telepon, ponsel umumnya juga mempunyai fungsi pengiriman dan penerimaan pesan singkat (*short message service*, SMS).

Penemu sistem telepon genggam yang pertama adalah Martin Cooper, seorang karyawan Motorola pada tanggal 03 April 1973, walaupun banyak disebut-sebut penemu telepon genggam adalah sebuah tim dari salah satu divisi Motorola (divisi tempat Cooper bekerja) dengan model pertama adalah DynaTAC. Ide yang dicetuskan oleh Cooper adalah sebuah alat komunikasi yang kecil dan mudah dibawa bepergian secara fleksibel. (<http://www.wikipedia.org/>)

II.6.1 Kriteria-kriteria Handphone

Kriteria yang digunakan dalam pendukung keputusan terdiri dari:

- a. Harga, kaitannya dengan layanan handphone.
- b. Fasilitas, kaitannya dengan layanan handphone.
- c. Bentuk/Model, kaitannya dengan kenyamanan pengguna.

- d. Kondisi, kaitannya dengan keadaan dan kondisi handpone. (Fitri Hanung Wibowo, dkk; 2014: 2)