

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Sistem informasi berbasis komputer merupakan sekelompok perangkat keras dan perangkat lunak yang dirancang untuk mengubah data menjadi informasi yang bermanfaat. Jenis sistem informasi berbasis komputer

1. Pengolahan Data. Pengolahan data elektronik – *electronic data processing (EDP)* adalah pemanfaatan teknologi komputer untuk melakukan pengolahan data transaksi-transaksi dalam suatu organisasi. EDP adalah aplikasi sistem informasi akuntansi paling dasar dalam setiap organisasi. Sehubungan dengan perkembangan teknologi komputer, istilah pengolahan data mulai dikenal dan mempunyai arti yang sama dengan istilah EDP.
2. Sistem Informasi Manajemen (SIM), menguraikan penggunaan teknologi komputer untuk menyediakan informasi bagi pengambilan keputusan para manajer.
3. Sistem Pendukung Keputusan – *Decision Support Systems (DSS)*. DSS diarahkan untuk melayani permintaan informasi tertentu, khusus, dan tidak rutin dari manajemen. Contohnya adalah penggunaan *spreadsheet* untuk melakukan analisis “*what if*” dari data operasi atau anggaran.
4. Sistem Pakar – *expert systems (ES)* adalah sistem informasi berbasis pengetahuan yang memanfaatkan pengetahuannya tentang bidang aplikasi tertentu untuk bertindak seperti seorang konsultan ahli bagi pemakainya.

5. Sistem Informasi Eksekutif – *executive information systems*(EIS). EIS dibuat bagi kebutuhan informasi strategik manajemen tingkat puncak.
6. Sistem Informasi Akuntansi – sistem berbasis komputer yang dirancang untuk mengubah data akuntansi menjadi informasi (Agustinus ; 2012 : 2).

II.2. Sistem Pendukung Keputusan

Sistem pendukung keputusan (SPK) atau *Decision Support System* (DSS) adalah sistem informasi interaktif yang menyediakan informasi, pemodelan, dan manipulasi data yang digunakan untuk membantu pengambilan keputusan pada situasi yang semiterstruktur dan situasi yang tidak terstruktur di mana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat. Konsep DSS dikemukakan pertama kali oleh Scott-Morton pada tahun 1971. Beliau mendefinisikan cikal bakal DSS tersebut sebagai “Sistem berbasis komputer yang interaktif, yang membantu pengambil keputusan menggunakan data dan model untuk memecahkan persoalan-persoalan tidak terstruktur” (Abdul Kadir ; 2014 : 108).

Sistem pendukung keputusan (SPK) biasanya dibangun untuk mendukung solusi atas suatu masalah atau untuk suatu peluang. Aplikasi sistem pendukung keputusan (SPK) digunakan dalam pengambilan keputusan. Aplikasi sistem pendukung keputusan (SPK) menggunakan CBIS (*Computer Based Information System*) yang fleksibel, interaktif, dan dapat diadaptasi, yang dikembangkan untuk mendukung solusi atas masalah manajemen spesifik yang tidak terstruktur. Sistem pendukung keputusan sebagai sistem berbasis komputer yang terdiri dari tiga komponen yang saling berinteraksi, sistem bahasa (mekanisme untuk memberikan

komunikasi antara pengguna dan komponen sistem pendukung keputusan lain), sistem pengetahuan (respositori pengetahuan domain masalah yang ada pada sistem pendukung keputusan atau sebagai data atau sebagai prosedur) dan sistem pemrosesan atau lebih kapabilitas manipulasi masalah umum yang diperlukan untuk pengambilan keputusan (Nofriansyah ; 2014 : 1).

II.2.1. Definisi Sistem Pendukung Keputusan

Definisi lain mengenai sistem pendukung keputusan (SPK) yang ideal yaitu :

- a. SPK adalah sebuah sistem berbasis komputer dengan antarmuka antara mesin/komputer dan pengguna.
- b. SPK ditujukan untuk membantu pembuat keputusan dalam menyelesaikan suatu masalah dalam berbagai level manajemen dan bukan untuk mengganti posisi manusia sebagai pembuat keputusan.
- c. SPK mampu memberi alternatif solusi bagi masalah semi/tidak terstruktur baik bagi perseorangan atau kelompok dan dalam berbagai macam proses dan gaya pengambilan keputusan.
- d. SPK menggunakan data, basis data dan analisa model-model keputusan.
- e. SPK bersifat adaptif, efektif, interaktif , *easy to use* dan fleksibel
- f. SPK menyediakan akses terhadap berbagai macam format dan tipe sumber data (*data source*) (Sulistiyo ; 2012 : 2).

II.2.2. Fase Pengambilan Keputusan

Model adalah representasi/abstraksi sederhana dari realitas karena realitas terlalu kompleks dan tidak relevan untuk memecahkan masalah khusus. Proses pemodelan menggunakan empat fase pengambilan keputusan yaitu fase inteligensi, fase desain, fase pemilihan, dan fase implementasi.

a. Fase Inteligensi

Inteligensi meliputi pemindaian lingkungan pada suatu waktu tertentu maupun secara periodik yang mencakup identifikasi masalah atau peluang masalah maupun monitoring hasil dari fase implementasi. Keluaran dari fase ini adalah pernyataan masalah.

b. Fase Desain

Fase desain meliputi penemuan atau pengembangan dan menganalisis tindakan yang mungkin dilakukan. Sebuah model pengambilan keputusan dibangun, diuji dan divalidasi.

c. Fase Pemilihan

Fase pemilihan meliputi pencarian evaluasi dan rekomendasi terhadap suatu solusi yang tepat untuk model. Solusi untuk model menghasilkan sebuah solusi yang direkomendasikan untuk masalah. Masalah dianggap dipecahkan hanya jika solusi yang direkomendasikan sukses diterapkan pada fase berikutnya.

d. Fase Implementasi

Implementasi dapat diartikan membuat suatu solusi yang direkomendasikan bisa bekerja (Karismariyanti ; 2011 : 56).

II.2.3. Komponen Sistem Pendukung Keputusan

Sistem pendukung keputusan dibangun oleh lima komponen yaitu subsistem manajemen data, subsistem antarmuka pengguna, dan subsistem manajemen berbasis pengetahuan. Berdasarkan definisi, sistem pendukung keputusan harus mencakup tiga komponen utama yaitu subsistem manajemen model, subsistem manajemen moden dan antarmuka pengguna. Sistem Pendukung Keputusan disusun dari beberapa subsistem yaitu :

1. Subsistem manajemen data

Basis data yang relevan untuk suatu situasi dan dikelola oleh *database management system (DBMS)*. Subsistem ini dapat diinterkoneksi dengan data *warehouse* perusahaan yang relavan untuk pengambilan keputusan.

2. Subsistem manajemen model

Merupakan paket perangkat lunak yang menyimpan model keuangan, statistik, ilmu manajemen, atau model kuantitatif lainnya yang memberikan kemampuan analitik yang tepat. Perangkat lunak ini sering disebut *modelbase management system (MBMS)* dan dapat diimplementasikan pada sistem pengembangan web untuk berjalan pada *server* aplikasi.

3. Subsistem antarmuka pengguna

Merupakan dukungan komunikasi antara denagn pengguna. *Web browser* menjadi salah satu antarmuka yang manampilkan dalam bentuk grafis dan interaktif dengan pengguna.

4. Subsistem manajemen

Berbasis pengetahuan bertindak sebagai komponen independen yang memberikan kemampuan intelegensi untuk memperbesar pengetahuan pengambil keputusan. Perusahaan memiliki sistem manajemen pengetahuan. Keterhubungan subsistem ini dengan sistem pendukung keputusan dapat melalui interkoneksi dengan *web server* (Karismariyanti ; 2011 : 55).

II.3. Metode Weight Product

Metode Weighted Product (WP) menggunakan perkalian untuk menghubungkan rating atribut, dimana setiap atribut harus dipangkatkan dahulu dengan bobot atribut yang bersangkutan. Preferensi untuk alternatif A_i diberikan sebagai berikut :

$$S_i = \prod_{j=1}^n (X_{ij})^{w_j} \dots\dots\dots(2)$$

Keterangan :

S = Preferensi alternatif dianalogikan sebagai vektor S

X_{ij} = Nilai variabel dari alternatif pada setiap atribut

w_j = Nilai bobot kriteria

n = Banyaknya kriteria

i = Nilai alternatif

j = Nilai kriteria (Lusiana Dwi Jayanti ; 2014 ; 3).

II.3.1. Langkah – Langkah Perhitungan Metode Weighted Product

Langkah-langkah dalam perhitungan metode *Weighted Product* (WP) adalah sebagai berikut :

1. Siapkan terlebih dahulu bobot dan lakukan perbaikan bobot dengan $W_j =$

$$W_j / W_j$$

2. Normalisasi matriks X berdasarkan persamaan

$$S_i = \prod_{j=1}^n (X_{ij})^{W_j} \text{ sehingga diperoleh matriks ternormalisasi } S$$

3. w_j adalah pangkat bernilai positif untuk atribut keuntungan, dan bernilai negatif untuk atribut biaya.

4. Lakukan perangkingan dengan menggunakan persamaan :

$$V_i = \frac{\prod_{j=1}^n x_{ij}^{w_j}}{\prod_{j=1}^n (x_j^*)^{w_j}}$$

5. Ditemukan urutan alternatif terbaik yang akan menjadi keputusan

II.3.2. Studi Kasus Weighted Product

Tabel II.1 Alternatif

Produsen	Kriteria	Sub Kriteria
PT. Lateks Sejahtera	Tebal (mm)	0,5 - 1mm
	Tegangan Putus (N/mm ²)	> 20 N/mm ²
	Perpanjangan Putus (%)	> 200 %
	Ketahanan Sobek (N/mm ²)	> 15 N/mm ²
	Pengembangan (%)	> 20%
PT. Jaya Lateks	Tebal (mm)	> 1mm
	Tegangan Putus (N/mm ²)	17 - 20 N/mm ²
	Perpanjangan Putus (%)	160 - 200 %
	Ketahanan Sobek (N/mm ²)	> 15 N/mm ²
	Pengembangan (%)	> 20%
PT. M&S	Tebal (mm)	0,5 - 1mm
	Tegangan Putus (N/mm ²)	17 - 20 N/mm ²
	Perpanjangan Putus (%)	160 - 200 %

	Ketahanan Sobek (N/mm ²)	10 - 15 N/mm ²
	Pengembangan (%)	10 - 20 %
PT. Health	Tebal (mm)	> 1mm
	Tegangan Putus (N/mm ²)	> 20 N/mm ²
	Perpanjangan Putus (%)	> 200 %
	Ketahanan Sobek (N/mm ²)	> 15 N/mm ²
	Pengembangan (%)	> 20%

Tabel II.2 Perbaikan Bobot

Kriteria	Bobot Lama	Bobot Baru
Tebal (mm)	5	0.3333333333333333
Tegangan Putus (N/mm ²)	4	0.2666666666666667
Perpanjangan Putus (%)	2	0.1333333333333333
Ketahanan Sobek (N/mm ²)	3	0.2
Pengembangan (%)	1	0.0666666666666667

Tabel III.3 Nilai Setiap Alternatif Di Setiap Kriteria

Alternatif	Kriteria				
	K01	K02	K03	K04	K05
PT. Lateks Sejahtera	2	3	3	3	3
PT. Jaya Lateks	3	2	2	3	3
PT. M&S	2	2	2	2	2
PT. Health	3	3	3	3	3

Tabel II.4. Perhitungan Vektor S

0.2508

Alternatif	Kriteria					Vektor S
	K01	K02	K03	K04	K05	
PT. Lateks Sejahtera	1.2599	1.3404	1.1578	1.2457	1.076	0.2577
PT. Jaya Lateks	1.4422	1.203	1.0968	1.2457	1.076	0.2508
PT. M&S	1.2599	1.203	1.0968	1.1487	1.0473	0.1966
PT. Health	1.4422	1.3404	1.1578	1.2457	1.076	0.2949

Tabel II.5. Perangkingan

Rangking	Alternatif	Vektor S
1	PT. Health	0.2949
2	PT. Lateks Sejahtera	0.2577
3	PT. Jaya Lateks	0.2508
4	PT. M&S	0.1966

Berdasarkan Perhitungan Metode Weighted Product Diatas Maka **PT. Health** adalah Produsen Yang Mempunyai Kriteria Sarung Tangan Lateks Terbaik Dengan Nilai **0.2949**

II.4. Pengertian Java

Java Language Specification adalah definisi teknis dari bahasa pemrograman java yang di dalamnya terdapat aturan penulisan sintakx dan semantij java. API adalah *application programming interface* yaitu sebuah *layer* yang berisi *class-class* yang sudah didefinisikan dan antarmuka pemrograman yang akan membantu para pengembang aplikasi dalam perancangan sebuah aplikasi (Wahana Komputer ; 2010 : 1)


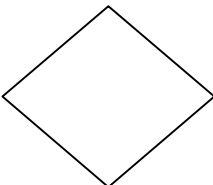
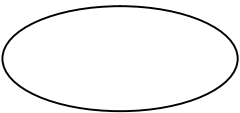

II.5. Pengertian MySQL

MySQL merupakan salah satu sistem database yang sangat handal karena menggunakan sistem *SQL*. Pada awalnya *SQL* berfungsi sebagai bahasa penghubung antara program database dengan bahasa pemrograman yang kita gunakan. Dengan adanya *SQL* maka para pemrogram jaringan dan aplikasi tidak mengalami kesulitan sama sekali di dalam menghubungkan aplikasi yang mereka buat. Setelah itu *SQL* dikembangkan lagi menjadi sistem database dengan munculnya *MySQL*. *MySQL* merupakan database yang sangat cepat, beberapa user dapat menggunakan secara bersamaan dan lebih lengkap dari *SQL*. *MySQL* merupakan salah satu *software* gratis yang dapat di-*download* melalui situsnya. *MySQL* merupakan sistem manajemen *database*, relasional sistem database dan *software open source*(Stendy B. Sakur ; 2015 : 57).

II.6. Entity Relationship Diagram (ERD)

Entity Relationship Diagram atau ERD adalah alat pemodelan data utama dan akan membantu mengorganisasi data dalam suatu proyek ke dalam entitas-entitas dan menentukan hubungan antarentitas. Proses memungkinkan analisis menghasilkan struktur basisdata yang baik sehingga data dapat disimpan dan diambil secara efisien (Simarmata ; 2010 : 67).

Tabel II.6. Simbol ERD

Simbol	Keterangan
	Entitas, adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai.
	Relasi, menunjukkan adanya hubungan di antara sejumlah entitas yang berbeda.
	Atribut, berfungsi mendeskripsikan karakter entitas (atribut yang berfungsi sebagai key diberi garis bawah)
	Garis, sebagai penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut

(Sumber : Janner Simarmata ; 2010 : 67)

II.7. Teknik Normalisasi

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basisdata relasional. Pada dasarnya, normalisasi adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari tabel rasional.

Teori normalisasi didasarkan pada konsep bentuk normal. Sebuah tabel relasional dikatakan berada pada bentuk normal tertentu jika tabel memenuhi himpunan batasan tertentu. Ada lima bentuk normal yang telah ditemukan.

II.7.1. Bentuk-bentuk Normalisasi

1. Bentuk normal tahap pertama (1st Normal Form)

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 tersendiri.

Contoh normalisasi 1NF adalah seperti pada tabel berikut :

Tabel 7. Tabel Bentuk Normal Pertama (1NF)

p#	Status	kota	b#	qty
p1	20	Yogyakarta	b1	300
p1	20	Yogyakarta	b2	200
p1	20	Yogyakarta	b3	400
p1	20	Yogyakarta	b4	200
p1	20	Yogyakarta	b5	100
p1	20	Yogyakarta	b6	100
p2	10	Medan	b1	300
p2	10	Medan	b2	400
p3	10	Medan	b2	200
p4	20	Yogyakarta	b2	200
p4	20	Yogyakarta	b4	300
p4	20	Yogyakarta	b5	400

2. Bentuk normal tahap kedua (2nd normal form)

Definisi bentuk normal kedua menyatakan bahwa tabel dengan kunci utama gabungan hanya dapat berada pada 1NF, tetapi tidak pada 2NF. Sebuah tabel relasional berada pada bentuk normal kedua jika dia berada pada bentuk normal kedua jika dia berada pada 1NF dan setiap

kolom bukan kunci yang sepenuhnya tergantung pada seluruh kolom yang membentuk kunci utama.

Tabel 8. Tabel Bentuk Normal Kedua (2NF)

Pemasok2			Barang		
p#	Status	Kota	p#	b#	qty
P1	20	Yogyakarta	p1	b1	300
P2	10	Medan	p1	b2	200
P3	10	Medan	p1	b3	400
P4	20	Yogyakarta	p1	b4	200
P5	30	Bandung	p1	b5	100
			p1	b6	100
			p2	b1	300
			p2	b2	400
			p3	b2	200
			p4	b2	200
			p4	b4	300
			p4	b5	400

3. Bentuk normal tahap ketiga (3rd normal form)

Bentuk normal ketiga mengharuskan semua kolom pada tabel relasional tergantung hanya pada kunci utama. Secara definisi, sebuah tabel berada pada bentuk normal ketiga (3NF) jika tabel sudah berada pada 2NF dan setiap kolom yang bukan kunci tidak tergantung secara transitif pada kunci utamanya.

Tabel 9. Tabel Bentuk Normal Ketiga (3NF)

Pemasok Kota		Kota Status	
p#	Kota	Kota	status
P1	Yogyakarta	Yogyakarta	20
P2	Medan	Medan	10
P3	Medan	Yogyakarta	20
P4	Yogyakarta	Bandung	30
P5	Bandung		

4. *Boyce Code Normal Form (BCNF)*

Setelah 3NF, semua masalah normalisasi hanya melibatkan tabel yang mempunyai tiga kolom atau lebih dan semua kolom adalah kunci. Banyak praktisi berpendapat bahwa menempatkan entitas pada 3NF sudah cukup karena sangat jarang entitas yang berada pada 3NF bukan merupakan 4NF dan 5NF.

5. **Bentuk Normal Keempat (4NF)**

Sebuah tabel rasional berada pada bentuk normal keempat (4NF) jika dia dalam BCNF dan semua ketergantungan multivalued merupakan ketergantungan fungsional. Bentuk normal keempat (4NF) didasarkan pada konsep ketergantungan multivalued (MVD). Sebuah ketergantungan multivalued tiga kolom, satu kolom mempunyai banyak baris bernilai sama, tetapi kolom lain bernilai berbeda.

Tabel 10. Tabel Bentuk Normal Keempat (4NF)

Pegawai Proyek		Pegawai Ahli	
peg#	Pry#	Peg#	Ahli
1211	P1	1211	Analisis
1211	P3	1211	Perancangan
		1211	Pemrograman

6. **Bentuk Normal Kelima**

Sebuah tabel berada pada bentuk normal kelima (5NF) jika ia tidak dapat mempunyai dekomposisi lossless menjadi sejumlah tabel lebih kecil. Empat bentuk normal pertama berdasarkan pada konsep ketergantungan fungsional, sedangkan bentuk normal kelima

berdasarkan pada konsep ketergantungan gabungan (*join dependence*) (Janner Simarmata ; 2010 : 78).

Tabel 11. Tabel Bentuk Normal Kelima (5NF)

peg#	Pry#	Ahli
1211	11	Perancangan
1211	28	Pemrograman

II.8. UML (*Unified Modeling Language*)

UML menyediakan beberapa notasi dan artifak standar yang bisa digunakan sebagai alat komunikasi bagi para pelaku dalam proses analisis dan disain. Artifak di dalam UML didefinisikan seabagi informasi dalam berbagai bentuk yang digunakan atau dihasilkan dalam proses pengembangan perangkat lunak. Contohnya adalah *source code* yang dihasilkan oleh proses pemrograman. Yang perlu diperhatikan untuk menjaga konsistensi antar artifak selama proses analisis dan disain adalah bahwa setiap perubahan yang terjadi pada satu artifak harus juga dilakukan pada artifak sebelumnya (Evi Triandini ; 2012 : 15).

1. *Use Case* Diagram

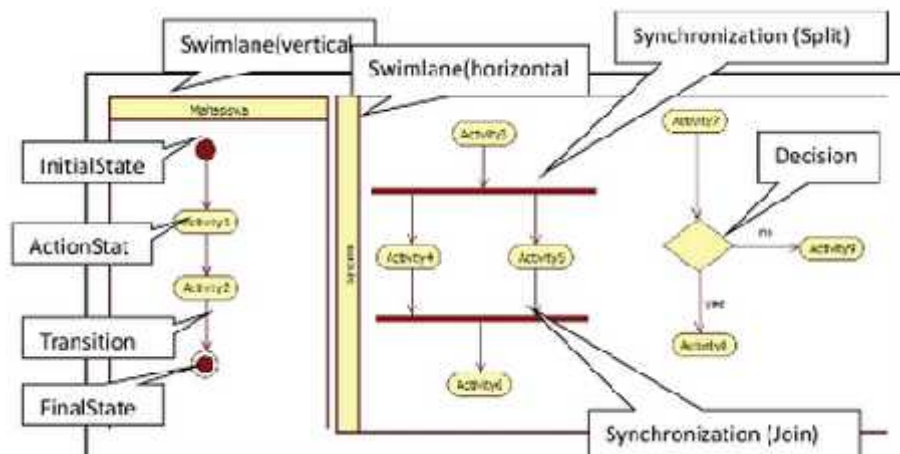
Usecase diagram menjelaskan apa yang akan dilakukan oleh sistem yang akan dibangun dan siapa yang berinteraksi dengan sistem.



Gambar II.1. Notasi Use Case Diagram
(Sumber : Evi triandini ; 2012 : 23)

2. Activity Diagram

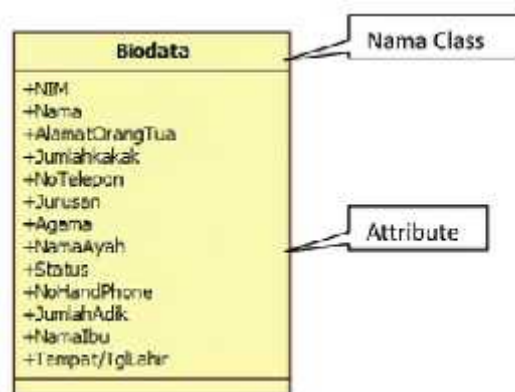
Activity diagram adalah sebuah diagram alur kerja yang menjelaskan berbagai kegiatan pengguna (atau sistem), orang yang melakukan masing-masing aktivitas, dan aliran sekuensial dari aktivitas-aktivitas tersebut.



Gambar II.2. Notasi Activity Diagram
(Sumber : Evi triandini ; 2012 : 37)

3. Class Diagram

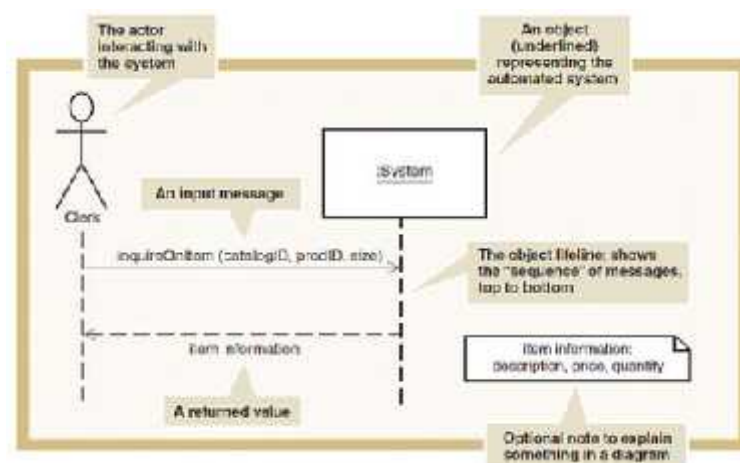
Class diagram merupakan diagram yang selalu ada di permodelan sistem berorientasi obyek. *Class* diagram menunjukkan hubungan antar class dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan.



Gambar II.3. Notasi Class Diagram
(Sumber : Evi triandini ; 2012 : 28)

4. Sequence Diagram

Sequence diagram menjelaskan secara detail urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari use case.



Gambar II.4. Notasi Sequence Diagram
(Sumber : Evi triandini ; 2012 : 24)