BAB II

LANDASAN TEORI

II.1. Konsep Dasar

II.1.1. Pengertian Sistem

Sistemmerupakankumpulanelemenyang saling berhubungansatu sama lain yang membentuk satu kesatuan dalam usaha mencapai suatu tujuan. Sistem bisa ditafsirkan sebagai kesatuan elemen yang memilikiketerkaitan.Beberapaelemendapat digabungkanmenjadi1unit,kelompok,atau komponen sistem tertentu (Putu Angga Septiana Putra, et al., 2016 : 3).

II.1.2. Pengertian Keputusan

Keputusan merupakan suatu reaksi terhadap beberapa solusi alternatif yang dilakukan secara sadar dengancaramenganalisakemungkinan-kemungkinandarialternatiftersebutbersama konsekuensinya. Pengambilan keputusan merupakan suatupendekatansistematisterhadaphakikatsuatu masalah,pengumpulanfakta-fakta,penentuyang matangdarialternatifyangdihadapi,danpengambilantindakanyangmenurutperhitungan merupakan tindakan yang paling tepat(Putu Angga Septiana Putra, et al., 2016 : 3).

II.1.3. Sistem Pendukung Keputusan

Sistem Pendukung Keputusan merupakan kegitan pemanfaatan sumber informasi untuk keperluan pengambilan keputusan dalam penanganan masalah khusus, atau rutin.Sistem Pendukung Keputusan secara umum didefinisikan sebagai sebuah sistem yang mampu memberikan kemampuan baik kemampuan pemecahan masalah maupun kemampuan pengkomunikasian untuk masalah semiterstruktur. Secara khusus, SPK didefinisikan sebagai sebuah sistem yang mendukung dalam memecahkan masalah semi-terstruktur dengan cara memberikan informasi ataupun usulan menuju pada keputusan tertentu.

Sistem Pendukung Keputusan memiliki tiga komponen utama yaitu *Database Management* (DBMS), *Model Base* (MBMS), dan *User Interface*.

1. *Database Management* (DBMS)

   Merupakan subsistem data yang terorganisasi dalam suatu basis data.Data yang merupakan suatu sistem pendukung keputusan dapat berasal dari luar maupun dalam lingkungan.Untuk keperluan SPK, diperlukan data yang relevan dengan permasalahan yang hendak dipecahkan melalui simulasi.

2. *Model Base* (MBMS)

   Merupakan suatu model yang merepresentasikan permasalahan kedalam format kuantitatif (model matematika sebagai contohnya) sebagai dasar simulasi atau pengambilan keputusan, termasuk didalamnya tujuan dari permaslahan (objektif), komponen-komponen terkait, batasan-batasan yang ada (*constraints*), dan hal-hal terkait lainnya.*Model Base* memungkinkan pengambil keputusan

menganalisa secara utuh dengan mengembangkan dan membandingkan solusi alternatif.

3. *User Interface*

   Terkadang disebut sebagai subsistem dialog, merupakan penggabungan antara dua komponen sebelumnya yaitu *Database Management* dan *Model Base* yang disatukan dalam komponen ketiga (*user interface*), setelah sebelumnya dipresentasikan dalam bentuk model yang dimengerti komputer. *User Interface* menampilkan keluaran sistem bagi pemakai dan menerima masukan dari pemakai kedalam Sistem Pendukung Keputusan (Putu Angga Septiana Putra, et al., 2016: 3).

II.2. Metode AHP

Analytical Hierarchy Process (AHP) merupakan metode pendukung pengambilan keputusan yang dikembangkan oleh Thomas L., Saaty pada tahun 1980.AHP merupakan alat pengambil keputusan yang menguraikan suatu permasalahan kompleks dalam struktur hirarki dengan banyak tingkatan yang terdiri dari tujuan, kriteria, dan alternatif. Hirarki didefinisikan sebagai suatu representasi dari sebuah permasalahan yang kompleks dalam suatu struktur multi level dimana level pertama adalah tujuan, yang diikuti level faktor, kriteria, sub kriteria, dan seterusnya ke bawah hingga level terakhir dari alternatif(Putu Angga Septiana Putra, et al., 2016: 4).

Untuk kegiatan pembandingan antar sepasang objek, metode AHP memberikan sebuah standar nilai pembandingan antar dua objek seperti dituangkan pada tabel II.1.

Tabel II.1. Nilai Perbandingan

Pembanding	**Nilai**
Sangat diutamakan	9
Lebih diutamakan menuju sangat diutamakan	8
Lebih diutamakn	7
Diutamakan menuju lebih diutamakan	6
Diutamakan	5
Cukup diutamakan menuju diutamakan	4
Cukup diutamakan	3
Setara menuju cukup diutamakan	2
Setara	1

(Sumber : Hilya Magdalena, 2012 : 52)

Langkah-langkah untuk melakukan perhitungan dengan metode AHP yaitu (Ni Kadek Putri, et al., 2016 : 4-5) :

1. Mendefinisikan masalah dan menentukan solusi yang diinginkan, lalu menyusun hierarki dari permasalahan yang dihadapi. Penyusunan hierarki adalah dengan menetapkan tujuan yang merupakan sasaran sistem secara keseluruhan pada level teratas.
2. Menentukan prioritas elemen

a. Langkah pertama dalam menentukan prioritas elemen adalah membuat perbandingan pasangan, yaitu membandingkan elemen secara berpasangan sesuai kriteria yang diberikan.
b. Matriks perbandingan berpasangan diisi menggunakan bilangan untuk merepresentasikan kepentingan relatif dari suatu elemen terhadap elemen lainnya.

3. Sintesis

   Pertimbangan-pertimbangan terhadap perbandingan berpasangan disintesis untuk memperoleh keseluruhan prioritas. 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. Kalikan setiap nilai pada kolom pertama dengan prioritas relatif elemen pertama, nilai pada kolom kedua dengan prioritas relatif elemen kedua dan seterusnya.

b. Jumlahkan setiap baris.

c. Hasil dari penjumlahan baris dibagi dengan elemen prioritas relatif yang bersangkutan.

d. Jumlahkan hasil bagi di atas dengan banyaknya elemen yang ada, hasilnya di sebut λ maks.

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

   CI=((λ_maks-n))/n.....(II.1)

   di mana n adalah banyakan elemen

6. Hitung Rasio Konsistensi / *Consistency Ratio* (CR) dengan rumus:

   CR=CI/IR.....(II.2)

   di mana CR = *Consistency Ratio*, CI = *Consistency Index*, IR = *Indeks Random Consistency*.

7. Memeriksa konsistensi hierarki. Jika nilainya lebih dari 10% maka penilaian dari *judgment* harus diperbaiki. Namun jika rasio konsisten kurang atau sama dengan 10% maka hasil perhitungan bisa dinyatakan benar. Daftar Indeks *Random Consistency* (IR) bisa dilihat pada Tabel II.2.

Tabel II.2.Daftar Indeks *Random Consistency* (IR)

Ukuran Matriks	Nilai IR
1.2	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
11	1.51

12	1.48
13	1.56
14	1.57
15	1.59

(Sumber : Ni Kadek Putri, et al., 2016 : 4-5)

II.3. Metode SAW

Metode SAW (*Simple Additive Weighting*) adalah sebuah metode yang sering juga dikenal istilah metode penjumlahan terbobot. Konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut yang membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada. Metode SAW ini merupakan metode yang paling dikenal dan paling banyak digunakan orang dalam menghadapi situasi MADM (*Multiple Attribute Decision Making*), dimana metode ini mengharuskan pembuat keputusan menentukan bobot bagi setiap atribut.

Konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif dari semua atribut. Metode SAW membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada. Diberikan persamaan sebagai berikut:

$$r_{ij} = \frac{x_{ij}}{Max\ x_{ij}}$$ Jika j atribut keberuntungan (benefit)……(II.3)

$$r_{ij} = \frac{Max\ x_{ij}}{x_{ij}}$$ Jika j atribut biaya (Cost)…….….(II.4)

Dimana rij adalah rating kinerja ternormalisasi dari alternatif Ai pada atribut Cj ; i=1,2,...,m dan j=1,2,...,n. Nilai preferensi untuk setiap alternatif (Vi) diberikan sebagai berikut:

$$V_i = \sum_{j=1}^{n} w_i r_{ij} \quad \text{(II.5)}$$

Keterangan:

Vi = nilai prefensi

wj = bobot rangking

rij = rating kinerja ternormalisasi

Nilai Vi yang lebih besar mengindikasikan bahwa alternatif Ai lebih terpilih(Putu Angga Septiana Putra, et al., 2016: 3-4).

II.4. Posisi Jabatan Karyawan

Karier adalah semua jabatan/pekerjaan yangdimilikiselama kehidupankerja seseorang. Perencanaan karier adalah suatu perencanaantentangkemungkinan seorang karyawan suatu organisasi atau perusahaan sebagaiindividu meniti proses kenaikan pangkat atau jabatan sesuai persyaratan dankemampuannya.

Jenjang Karier adalahsuatu kondisi yang menunjukkan adanyapeningkatan status seseorang dalam suatu orgnisasi pada jalur karier yangtelah ditetapkan dalam organisasi yang bersangkutan.Pada jenjang karierkaryawan memilikitahapan pengembangannya. Ada 3 tahapanpengembangan jenjang karier karyawan yaitu;

1. Karier awal

Karir awal/tahap pembentukan, merupakan tahap penekanan padaperhatian untuk memperoleh jaminan terpenuhinya kebutuhan ditahun-tahun awal pekerjaannya.

2. Karier Pertengahan

   Tahap karier pertengahankerap kali meliputi pengalaman baru, sepertipenugasan khusus, transfer dan promosi yang lebih tinggi, tawaran dariII – 9organisasi lain, kesempatan vasibilitas untuk jenjang organisasi yanglebih tinggi, dan pembentukan nilai seseorang bagi organisasi.

3. Karier Akhir

   Pemberian pelatihan kepada penerus, pengurangan beban kerja, ataupendelegasian tugas-tugas utama periode karier akhiradalah agar tetapproduktif dan menyiapkan diri untuk pensiun.(M. Subastian, 2014; 27).

II.4. *Visual Basic .NET 2010*

VB.Net 2010 merupakan produkpemrograman dari *MicrosoftCorporation*, dimana didalamnya berisibeberapa jenis IDE pemrograman seperti*Visual Basic, Visual C++, Visual WebDeploper, Visual C#* Dan *Visual F#*.Bahasa pemrograman *VB.Net 2010*sendiri awalnya berasal dari bahasapemrograman yang sangat populer dikalangan *programmer* komputer, yaitubahasa *Basic*, yang oleh *Microsoft*diadaptasi dalam program *microsoft quickBasic*. Seiring dengan berkembangnya teknologi komutasi desain,*Microsoft* mengeluarkan produk

yangdinamakan *Microsoft Visual Studio* dengan*Visual Basic* didalamnya (Rusli Saputra, 2016 : 233).

II.5. *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 berkecimpungdalam persaingan dunia pengolahan data menyusul pendahulunya seperti IBM dan *Oracle*.*SQl Server2008* 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.*Microsoft* merilis *SQL Server 2008* dalam beberapa versi yang disesuaikan dengan segment-segment pasar yang dituju.

Menurut cara pemrosesan data pada prosesor maka *Microsoft* mengelompokkan produk ini berdasarkan 2 jenis yaitu :

1. Versi 32-bit(x86), yang biasanya digunakan untuk komputer dengan *single*prosesor (Pentium 4) atau lebih tepatnyaprosesor 32 bit dan sistem operasi Windows XP.
2. Versi 64-bit(x64), yang biasanya digunakan untuk komputer dengan lebih dari satu prosesor (Misalnya *Core 2 Duo*) dan sistem operasi 64 bit seperti Windows XP 64, Vista, dan Windows 7.

Sedangkan secara keseluruhan terdapat versi-versi seperti berikut ini:

1. Versi *Compact*, ini adalah versi “Tipis” dari semua versi yang ada. Versi ini seperti versi *desktop* pada *SQL Server 2000*. Versi ini juga digunakan pada *handleddrvice* seperti Pocket PC, PDA, *SmartPhone*, Tablet PC.
2. Versi *Express*, ini adalah versi “Ringan” dari semua versi yang ada(tetapi versi ini berbeda dengan versi *compact*) dan paling cocok untuk latihan. *Express Manager* standar, integrasi dengan CLR dan XML (Agus Tinus Setiawan dan Rika Putri Permadani, 2016 : 54-55).

II.6. *Database*

*Database*merupakankumpulandaridatayang salingberhubungansatudenganyanglainnya, tersimpandiperangkatkeraskomputerdan digunakan untuk memanipulasinya.*Database*merupakan salah satu komponen yang pentingdalamsisteminformasi,karenamerupakan basisdalammenyediakaninformasibagipara pemakai.Penerapan*database*dalamsistem informasidisebutdengan*databasesystem*.Sistem basisdata(*database system*)adalahsuatusistem informasi yang mengitegrasikan kumpulan dari data yangsalingberhubungansatudenganyanglainnya danmembuatnyatersediauntukbeberapaaplikasi yangbermacam-macamdidalamsuatuorganisasi.Dengansistemdasardatainitiap-tiaporangatau

bagiandapatmemandang*database*daribeberapa sudut pandangan yang berbeda.Bagian kredit dapat memandangnyasebagaidatapiutang.Bagian penjualandapatmemandangnyasebagaidata penjualan.Bagianpersonaliadapatmemandangnya sebagaidatarelawan.Semuaterintegrasidalam sebuahdatayangumum.Berbedadengansistem pengolahandatatradisional(*traditional processing system*),sumberdataditanganisendiri untuk tiap-tiap aplikasi (Samsuri Ridwan, et al., 2014 : 45).

II.7. Normalisasi

Secara berturut-turut masing-masing level normalisasi dibahas berikut ini, dimulai dari bentuk tidak normal.(Eddy Sutanta; 2011: 176-179)

1. Relasi bentuk tidak normal (*Un Normalized Form* / UNF)

   Relasi-relasi yang dirancang tanpa mengindahkan batasan dalam defisi basis data dan karakteristik *Relational Database Management System* (RDBM) menghasilkan relasi *Un Normalized Form* (UNF).Bentuk ini harus di hindari dalam perancangan relasi dalam basis data. Relasi *Un Normalized Form* (UNF) mempunyai kriteria sebagai berikut.

   a. Jika relasi mempunyai bentuk *non flat file* (dapat terjadi akibat data disimpan sesuai dengan kedatangannya, tidak memiliki struktur tertentu, terjadi duplikasi atau tidak lengkap)

b. Jika relasi membuat *set atribut* berulang *(non single values)*

c. Jika relasi membuat *atribut non atomic value*

2. Relasi bentuk normal pertama (*First Norm Form* / 1NF)

Relasi disebut juga *First Norm Form* (1NF) jika memenuhi kriteria sebagai berikut.

a. Jika seluruh atribut dalam relasi bernilai *atomic (atomic value)*

b. Jika seluruh atribut dalam relasi bernilai tunggal *(single value)*

c. Jika relasi tidak memuat set atribut berulang

d. Jika semua record mempunyai sejumlah atribut yang sama.

Permasalahan dalam *First Norm Form* (1NF) adalah sebagai berikut.

a. Tidak dapat menyisipkan informasi parsial

b. Terhapusnya informasi ketika menghapus sebuah *record*

3. Bentuk normal kedua *(Second Normal Form /* 2NF)

Relasi disebut sebagai *Second Normal Form (*2NF) jika memenuhi kriteria sebagai berikut

a. Jika memenuhi kriteria *First Norm Form (*1NF).

b. Jika semua atribut nonkunci *Functional Dependence* (FD) pada *Primary Key* (PK).

Permasalahan dalam *Second Normal Form /* 2NF adalah sebagai berikut

a. Kerangkapan data *(data redundancy)*

b. Pembaharuan yang tidak benar dapat menimbulkan inkonsistensi data *(data inconsistency)*

c. Proses pembaharuan data tidak efisien

Kriteria tersebut mengidentifikasikan bahwa antara atribut dalam *Second Normal Form* masih mungkin mengalami *Third Norm Form*.Selain itu, relasi *Second Normal Form* (2NF) menuntut telah didefinisikan atribut *Primary Key* (PK) dalam relasi. Mengubah relasi *First Norm Form* (1NF) menjadi bentuk *Second Normal Form* (2NF) dapat dilakukan dengan mengubah struktur relasi dengan cara :

a. Identifikasikan *Functional Dependence* (FD) relasi *First Norm Form* (1NF)
b. Berdasarkan informasi tersebut, dekomposisi relasi *First Norm Form* (1NF) menjadi relasi-relasi baru sesuai *Functional Dependence* nya. Jika menggunakan diagram maka simpul-simpul yang berada pada puncak diagram ketergantungan data bertindak *Primary Key* (PK) pada relasi baru

4. Bentuk normal ketiga (*Third Norm Form* / 3NF)

Suatu relasi disebut sebagai *Third Norm Form* jika memenuhi kriteria sebagai berikut.

a. Jika memenuhi kriteria *Second Normal Form* (2NF)
b. Jika setiap atribut nonkunci tidak (TDF) *(Non Transitive Dependeny)* terhadap *Primary Key* (PK)

Permasalahan dalam *Third Norm Form* (3NF) adalah keberadaan penentu yang tidak merupakan bagian dari *Primary Key* (PK)

menghasilkan duplikasi rinci data pada atribut yang berfungi sebagai *Foreign Key* (FK) (duplikasi berbeda dengan keterangan data).

Mengubah relasi *Second Normal Form* (2NF) menjadi bentuk *Third Norm Form* (3NF) dapat dilakukan dengan mengubah struktur relasi dengan cara :

a. Identifikasi TDF relasi *Second Normal Form* (2NF)
b. Berdasarkan informasi tersebut, dekomposisi relasi *Second Normal Form* (2NF) menjadi relasi-relasi baru sesuai TDF-nya.

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

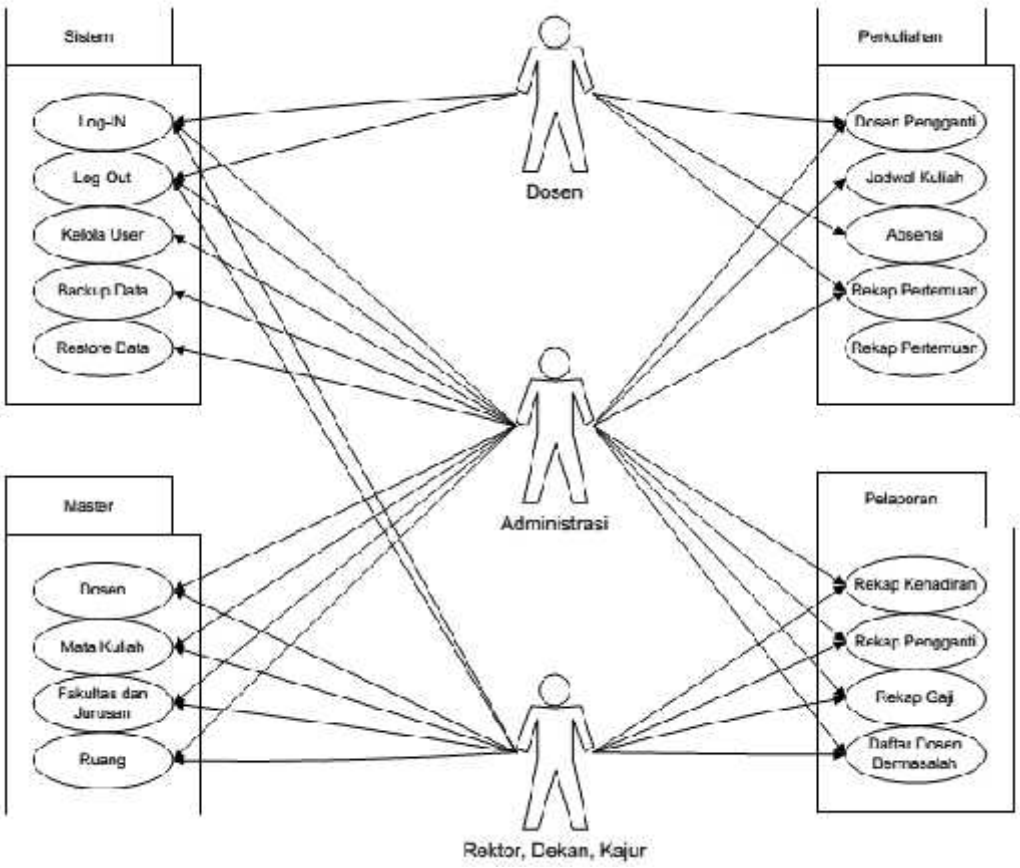
UML adalah bahasa untuk mengspesifikasi, menvisualisasi, membangun dan mendokumentasikan *artefact* (bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya). UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan peranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C#, atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C (I Made Budi Adnyana, 2016 : 51-52).

Menurut Windu Gata, Grace (2013:4), *Unified Modeling Language* (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membanngun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga

merupakan alat untuk mendukung pengembangan sistem (Ade Hendini, 2016 : 108).

II.8.1. *Use Case* Diagram

Use case diagramdigunakan untuk memodelkan bisnis proses berdasarkan perspektif penggunasistem.*Use case*diagramterdiriatasdiagramuntuk*use case*dan*actor*.*Actor*merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi.*Use case* merepresentasikanoperasi-operasiyangdilakukanoleh*actor*.*Use case* digambarkanberbentukelipsdengannamaoperasidituliskandidalamnya.*Actor*yang melakukan operasi dihubungkan dengan garis lurus ke *use case*(Arif Yulianto, 2015 : 218).

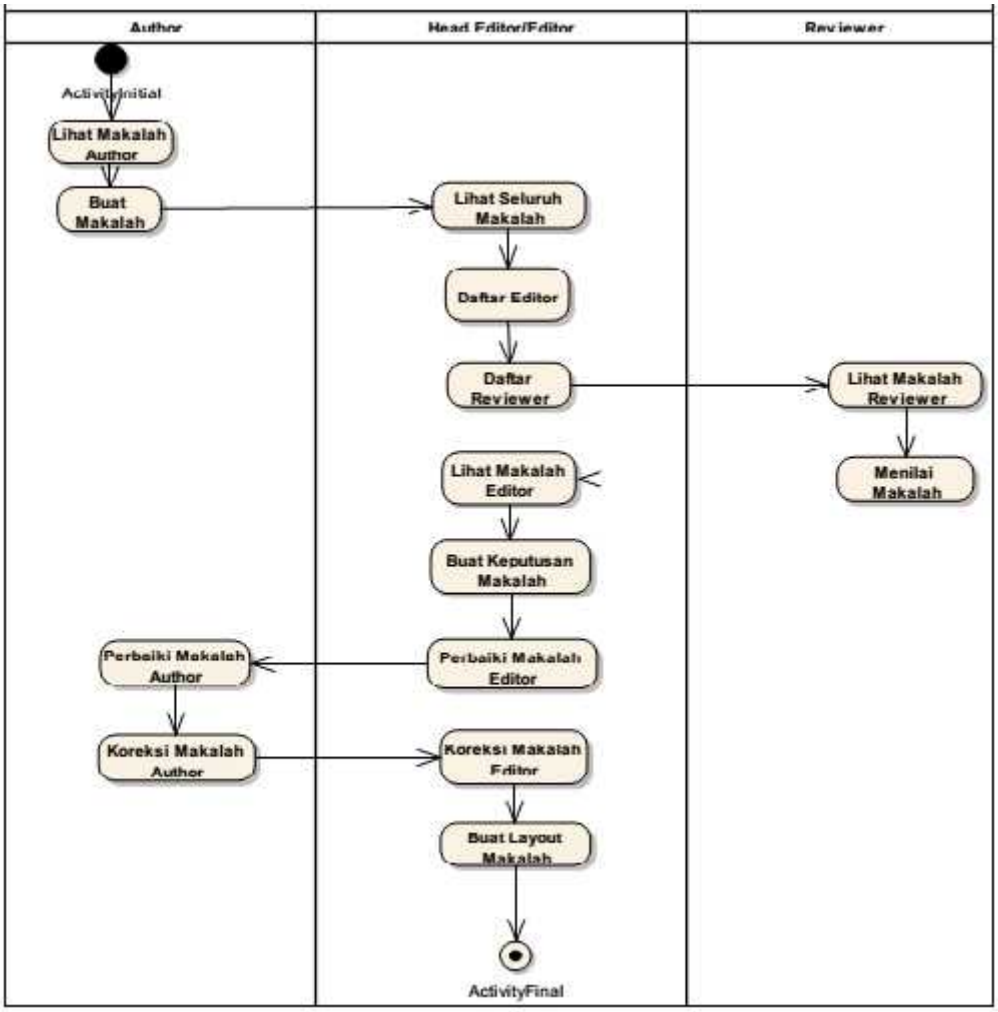


Gambar II.1. *Use Case* Diagram
(Sumber :Arif Yulianto, 2015 : 219)

II.8.2. *Activity* Diagram

*Activity*diagramadalahdiagramyang menggambarkan sifat dinamis secara alamiah sebuah sistemdalambentukmodelalirandankontroldari aktivitas ke aktivitas lainnya (Yosua P.W Simeremere, et al.,2013 : 471).

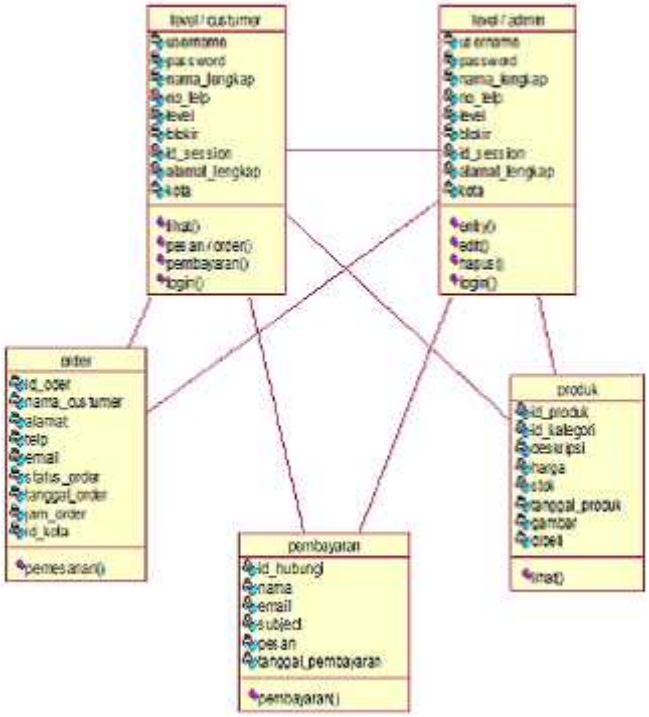
*Activity*diagrammenggambarkanberbagai aliran aktivitas (*work flow*) dalam sistem yang sedangdirancang,bagaimanamasing-masing aliran berawal, *decision*yang mungkin terjadi, dan bagaimana mereka berakhir (Abulwafa Muhammad, et al., 2013 : 57-58).



Gambar II.2. *Activity* Diagram
(Sumber :Yosua P.W Simeremere, et al., 2013 : 472)

II.8.3. *Class* Diagram

*Class*Diagrammenggambarkan*class*sdalam sebuahsistemdanhubungannyaantarasatu denganyanglain,sertadimasukkanpula atribut dan operasi. Umumnya*class*diagram darisuatusistemakanmenggambarkanjuga bagaimana struktur *database* yang dibutuhkan untuk membangun sistem tersebut (Abulwafa Muhammad, et al., 2013 : 57).

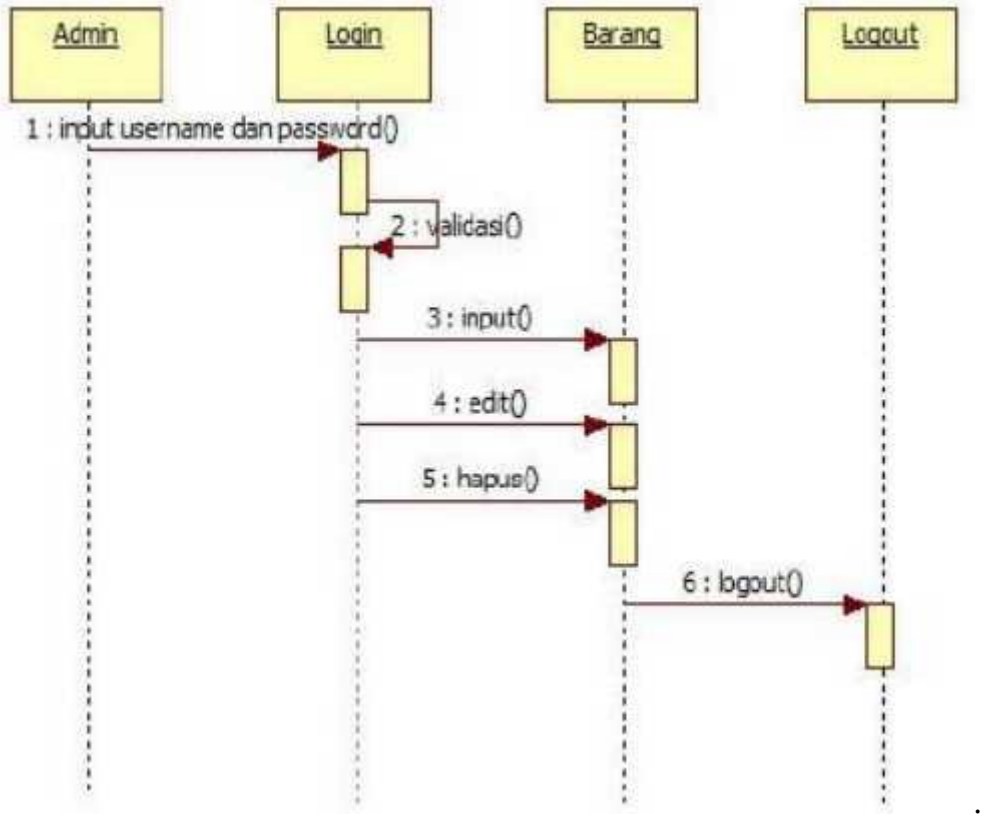


Gambar II.3.*Class* Diagram
(Sumber :Abulwafa Muhammad, et al., 2013 : 57)

II.8.4. *Sequence* Diagram

*Sequence*diagramadalahsuatudiagramyang memperlihatkan/menampilkaninteraksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutanataurangkaianwaktu.Interaksiantarobjek tersebut termasuk pengguna,*display*, dan sebagainya berupa pesan/*message*(Yosua P.W Simeremere, et al., 2013 : 471).

*Sequence*diagramdigunakanuntukmenggambarkanperilakuaktorpadasebua h sistemsecaradetailmenurutwaktu.Diagram inimenunjukkansejumlahcontohobjekdan *message*(pesan)yangdiletakkandiantara objek-objek di dalam *use case*(Abulwafa Muhammad, et al., 2013 : 58).



Gambar II.4. *Sequence* Diagram
(Sumber :Abulwafa Muhammad, et al., 2013 : 58)