

## **BAB II**

### **LANDASAN TEORI**

#### **II.1. Definisi Sistem**

Sistem dapat terdiri dari sistem-sistem bagian (*subsystem*). Sebagai misal, sistem komputer dapat terdiri dari subsistem perangkat keras dan subsistem perangkat lunak. Masing-masing subsistem dapat terdiri dari subsistem-subsistem yang lebih kecil atau terdiri dari komponen-komponen. Subsistem perangkat keras (*hardware*) dapat terdiri dari alat masukan, alat pemroses, alat keluaran dan simpanan luar, dan kemudian subsistem-subsistem tersebut akan berinteraksi sedemikian rupa sehingga dapat mencapai satu kesatuan yang terpadu.

Menurut (Jogiyanto, 2011) dalam buku *Analisa dan Design Sistem Informasi Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. “Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu ” (Deppi Linda, 2016 : 62).

##### **II.1.1. Karakteristik Sistem**

Sistem mempunyai karakteristik atau sifat-sifat tertentu, diantaranya adalah sebagai berikut:

1. **Komponen Sistem (*Components*)**

Sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk suatu kesatuan. Komponen-

komponen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.

2. Batas Sistem (*Boundary*)

Batas sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batas sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem (*Environments*)

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga merugikan suatu sistem.

4. Penghubung Sistem (*Interface*)

Penghubung merupakan media penghubung antara suatu subsistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari suatu subsistem ke subsistem lainnya.

5. Masukan Sistem (*Input*)

Masukan adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*).

6. Keluaran sistem (*Output*)

Keluaran adalah hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran dapat berupa masukan untuk subsistem yang lain.

7. Pengolah Sistem (*Process*)

Sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolah. Pengolah yang akan merubah masukan menjadi keluaran.

8. Sasaran Sistem (*Objectives*)

Sistem mempunyai tujuan atau sasaran yang akan menentukan masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem (Deppi Linda, 2016 : 62).

### II.1.2. Informasi

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. (1) Akurat (*Accurate*) Berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. (2) Tepat waktu (*Timelines*) Berarti informasi yang datang pada sipenerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai logika karena informasi merupakan landasan dalam pengambilan keputusan. (3) Relevan (*Relevance*) Berarti informasi tersebut bermanfaat bagi pemakainya (Deppi Linda, 2016 : 62-63).

### **II.1.3. Sistem Informasi**

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan laporan-laporan yang diperlukan oleh pihak luar tertentu.

Tipe sistem informasi adalah sebagai berikut:

1. Sistem informasi Akuntansi
2. Sistem informasi Pemasaran
3. Sistem informasi Manajemen Persediaan
4. Sistem informasi Personalia
5. Sistem informasi Distribusi
6. Sistem informasi Pembelian
7. Sistem informasi Kekayaan
8. Sistem informasi Analisis Kredit
9. Sistem informasi Penelitian dan Pengembangan
10. Sistem informasi Teknik

Semua sistem informasi tersebut dimaksudkan untuk memberikan informasi kepada semua tingkat manajemen, mulai manajemen tingkat bawah, manajemen tingkat menengah, hingga manajemen tingkat atas (Tata Sutabri, 2012: 38-41).

## II.2. Sistem Pendukung Keputusan

Sistem pendukung keputusan adalah suatu sistem informasi berbasis komputer interaktif yang dapat memberikan alternatif dan solusi bagi pengambil dan pembuat keputusan. Sistem Pendukung Keputusan merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan, dan pemanipulasidata. Sistem itu digunakan untuk membantu pengambilan keputusan dalam situasi semi terstruktur dan situasi yang terstruktur, dimana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat.

DSS tidak dimaksudkan untuk mengotomatisasikan pengambilan keputusan, tetapi memberikanperangkat interaktif yang memungkinkan pengambilan keputusan untuk melakukan berbagai analisis menggunakan model-model yang tersedia.

Tujuan dari sistem pendukung keputusan adalah :

1. Membantu manajer dalam pengambilan keputusan atas masalah semi terstruktur.
2. Memberikan dukungan atas pertimbangan manajer dan bukannya dimasukkan untukmenggantikan fungsi manajer.
3. Meningkatkan efektivitas keputusan yang diambil manajer lebih dari pada perbaikan efisiensinya.
4. Kecepatan komputasi. Komputer memungkinkan para pengambil keputusan untuk melakukan banyak kompuasi secara cepat dengan biaya yang rendah.

5. Peningkatan produktivitas. Membangun suatu kelompok pengambil keputusan, terutama parapakar, bisa sangat mahal. Pendukung terkomputerisasi bisa mengurangi ukuran kelompok dan memungkinkan para anggotanya untuk berada diberbagai lokasi yang berbeda-beda (menghematbiaya perjalanan).
6. Dukungan kualitas. Komputer bisa meningkatkan kualitas keputusan yang dibuat.
7. Berdaya saing. Manajemen dan pemberdayaan sumberdaya perusahaan.
8. Mengatasi keterbatasan kognitif dalam pemrosesan dan penyimpanan (Kiki Yasdomi, 2015 : 93).

### **II.3. Logika *Fuzzy***

Logika *fuzzy* adalah logika yang menggunakan konsep sifat kesamaran. Sehingga logika *fuzzy* adalah logika dengan tak hingga banyak nilai kebenaran yang dinyatakan dalam bilangan real dalam selang  $(0,1)$ . Logika *fuzzy* adalah suatu cara yang tepat untuk memetakan suatu ruang input kedalam suatu ruang output, mempunyai nilai kontinyu. *Fuzzy* dinyatakan dalam derajat dari suatu keanggotaan dan derajat dari kebenaran. Oleh sebab itu sesuatu dapat dikatakan sebagian benar dan sebagian salah pada waktu yang sama (Mutammimul Ula, 2014 : 38-39).

Berikut ini adalah alasan digunakan logika *fuzzy* antara lain:

- a. Konsep logika *fuzzy* mudah dimengerti, karena logika *fuzzy* menggunakan dasar teori himpunan, maka konsep matematis yang mendasari penalaran *fuzzy* tersebut cukup mudah untuk dimengerti
- b. Logika *fuzzy* sangat fleksibel, artinya mampu beradaptasi dengan perubahan-perubahan dan ketidakpastian yang menyertai permasalahan.
- c. Logika *fuzzy* memiliki toleransi terhadap data yang tidak tepat
- d. Logika *fuzzy* didasarkan pada bahasa alami, menggunakan bahasa sehari-hari sehingga mudah dimengerti.

### II.3.1. Metode Tsukamoto

Metode *Tsukamoto* merupakan perluasan dari penalaran monoton, Pada Metode *Tsukamoto*, setiap konsekuen pada aturan yang berbentuk *IF THEN* harus direpresentasikan dengan suatu himpunan *fuzzy* dengan fungsi keanggotaan yang monoton. Sebagai hasilnya, *output* hasil inferensi dari tiap-tiap aturan diberikan secara tegas (*crisp*) berdasarkan *-predikat (fire strength)*. Hasil akhirnya diperoleh dengan menggunakan rata-rata terbobot (Ratih Kumala Sari dan Hamdani, 2012 : 111).

Bentuk model *fuzzy Tsukamoto* adalah (Mutammimul Ula, 2014 : 41) :

***IF (X IS A) and (Y IS B) Then (Z IS C)***

Dimana : A,B dan C adalah himpunan *fuzzy*.

Misalkan diketahui 2 rule berikut .

***IF (X is A1) AND (Y is B1) THEN (Z is C1)***

*IF (X is A2) AND (Y is B2) THEN (Z is C2)*

Dalam inferensinya, metode tsukamoto menggunakan tahapan berikut :

- a. *Fuzzyfikasi*
- b. Pembentukan basis pengetahuan *Fuzzy(Rule* dalam bentuk *IF....THEN)*
- c. Mesin Inferensi, menggunakan fungsiimplikasi MIN untuk mendapatkan nilai -predikat tiap-tiap rule ( 1, 2, 3,..... n)Kemudian masing-masing nilai -predikatini digunakan untuk menghitung keluaranhasil inferensi secara tegas (crisp)masing-masing rule (z1, z2, z3,..... zn )
- d. *Defuzzyfikasi*

Menggunakan metode rata-rata (*Average*)

$$Z^* = \frac{\sum_i Z_i}{i} \dots\dots\dots(1)$$

Keterangan :

Z= Variabel *output*

i = Nilai predikat

zi = Nilai variabel *output*

*Tsukamoto* yaitusetiap konsekuen pada aturan berbentuk *IF-THEN* harus dipresentasikan dengan suatu himpunan *fuzzy*, dengan fungsi keanggotaan yang monoton. Sebagai hasilnya, *output* hasil inferensi dari tiap-tiap aturan diberikan dengan berdasarkan predikat (*fire strength*). Hasil akhir diperoleh dengan menggunakan rata-rata terbobot. Misalkan ada 2 variabel *input*, yaitu x dan y serta

satu variabel output z. Variabel x terbagi atas dua himpunan yaitu A1 dan A2, sedangkan variabel y terbagi atas himpunan B1 dan B2. Variabel z juga terbagi atas dua himpunan yaitu C1 dan C2. Tentu saja himpunan C1 dan C2 harus merupakan himpunan yang bersifat monoton. Ada 2 aturan yang digunakan, yaitu:

[R1] IF (xis A<sub>1</sub>) And (yis B<sub>2</sub>) THEN (zis C<sub>1</sub>)

[R1] IF (xis A<sub>2</sub>) And (yis B<sub>2</sub>) THEN (zis C<sub>2</sub>)

Keterangan:

R1 : Aturan *fuzzy*

X : variabel pinjaman

1 : Himpunan pinjaman tertinggi

2 : Himpunan pinjaman terendah

And : Operator yang digunakan

y : Variabel jaminan

B1 : Himpunan jaminan tertinggi

B2 : Himpunan jaminan terendah

THEN : Operator yang digunakan

Z : Variabel Penghasilan (nilai *crisp*)

C1 : Himpunan penghasilan tertinggi (harus monoton)

C2 : Himpunan penghasilan terendah (harus monoton)

#### II.4. *Insentif*

*Insentif* merupakan suatu bentuk motivasi yang dinyatakan dalam bentuk uang. *Insentif* adalah untuk meningkatkan motivasi karyawan dalam upaya

mencapai tujuan–tujuan organisasi. Penghargaan berupa insentif atas dasar prestasi kerjanya yang tinggi merupakan rasa pengakuan dari organisasi terhadap prestasi karyawan dan kontribusi kepada organisasi.

Sistem insentif dapat diterapkan untuk hampir semua jenis pekerjaan-pekerjaan profesional, manajerial dan eksekutif. Terdapat beberapa bentuk insentif sebagai berikut (Aditya Febrianto, et al., 2016 : 3-4) :

1. *Piece work* (upah per *output*)

Sistem insentif yang memberikan imbalan bagi pekerja atas tiap unit yang dihasilkan. Upah harian atau ditentukan dengan mengalikan jumlah unit yang dihasilkan dengan tarif per unit.

2. *Production bonus* (bonus produksi)

*Insentif* yang dibagikan kepada pekerja melebihi *output* yang ditetapkan. Para pekerja biasanya menerima upah pokok bila mereka dapat menghasilkan *output* di atas standar mereka memperoleh bonus, yang jumlahnya biasanya ditentukan atas dasar tarif per unit produktivitas di atas standar.

3. *Commissions* (komisi)

*Insentif* ini diberikan atas dasar jumlah unit yang terjual. Sistem ini biasanya diberlakukan untuk pekerjaan seperti wiraniaga, agen *real estate*.

4. *Maturity curve* (curva kematangan)

Bentuk *insentif* ini diberikan untuk mengakomodasi para pekerja yang memiliki unjuk gigi tinggi, dilihat dari aspek produktivitas atau pekerja yang telah berpengalaman.

5. *Merit Raises* (Upah kontribusi)

Kenaikan gaji atau upah yang diberikan sesudah penilaian unjuk kerja. Kenaikan ini biasanya diputuskan oleh atasan langsung pekerja, seringkali dengan bekerja sama dengan atasan yang lebih tinggi.

6. *Nomenetary Incentives* (insentif non materi)

*Insentif* seperti ini diberikan sebagai penghargaan atas unjuk kerja yang berkaitan dengan pekerjaan, saran yang diberikan kepada perusahaan atau kegiatan pengabdian kepada masyarakat misalnya banyak perusahaan yang memiliki program pemberian seperti plakat, sertifikat, liburan, cuti, dan insetif lain yang tidak berbentuk uang.

7. *Eksecutives Incentives* (insentif eksekutif)

Bentuk-bentuk *insentif* bagi eksekutif antara lain bonus uang tunai, *stock option* (hak untuk memberi saham perusahaan dengan harga tertentu), *performance objectives*.

## II.5. Pengertian Basis Data

Basis data (*database*) adalah kumpulan dari berbagai data yang saling berelasi atau berhubungan satu dengan yang lainnya, data dinyatakan dengan nilai. Basis data tersimpan di perangkat keras, serta dimanipulasi dengan menggunakan perangkat lunak. Pendefinisian basis data meliputi spesifikasi dari

tipe data, struktur dan batasan dari data atau informasi yang akan disimpan.

Istilah-istilah dalam basis data (Mutammimul Ula, 2014 : 38) :

1. *Enterprise*: suatu bentuk organisasi seperti Bank, Sekolah, Rumah Sakit, Pabrik, Kantor dan sebagainya.
2. Entitas: suatu objek yang dapat dibedakan dari lainnya yang dapat di wujudkan dalam basis data. Kumpulan dari entitas disebut himpunan entitas.
3. Atribut dan elemen data: karakteristik dari suatu entitas.
4. *Record* data: kumpulan suatu elemen data yang saling berhubungan.
5. Tabel: kumpulan data atau informasi.

Basis data mempunyai peran penting dalam sistem informasi, yaitu sebagai sumber penyedia data utama untuk memenuhi kebutuhan-kebutuhan informasi seluruh pemakai atau informasi bagi para pengambil keputusan (Edhy Sutanta, 2011 : 18).

## **II.6. Normalisasi**

Normalisasi diartikan sebagai suatu teknik yang menstrukturkan mendekomposisi data dalam cara-cara tertentu untuk mencegah timbulnya permasalahan pengolahan data dalam basis data. Permasalahan yang dimaksud adalah berkaitan dengan penyimpangan-penyimpangan (*anomallies*) yang terjadi akibat adanya kerangkapan data dalam relasi dan in-efisiensi pengolahan (Martin, 1975) : (Edy Sutanta ; 2011 : 174-175).

Secara berturut-turut masing-masing level normal tersebut dibahas berikut ini, dimulai dari bentuk tidak normal. (Edy 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

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 Dependency*) 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 berfungsi sebagai *Foreign Key (FK)* (duplikasi berbeda dengan keterangan data).

Mengubah relasi *Second Normal Form* (2NF) menjadi bentuk *Third Normal 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.7. *Unified Modeling Language* (UML)**

*Unified Modelling Language* (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak sebuah sistem. UML lebih mengedepankan penggunaan diagram untuk menggambarkan aspek dari sistem, karena tergolong bahasa visual yang lebih mudah dan lebih cepat dipahami dibandingkan dengan bahasa pemrograman. *Unified Modelling Language* (UML) biasa digunakan untuk :

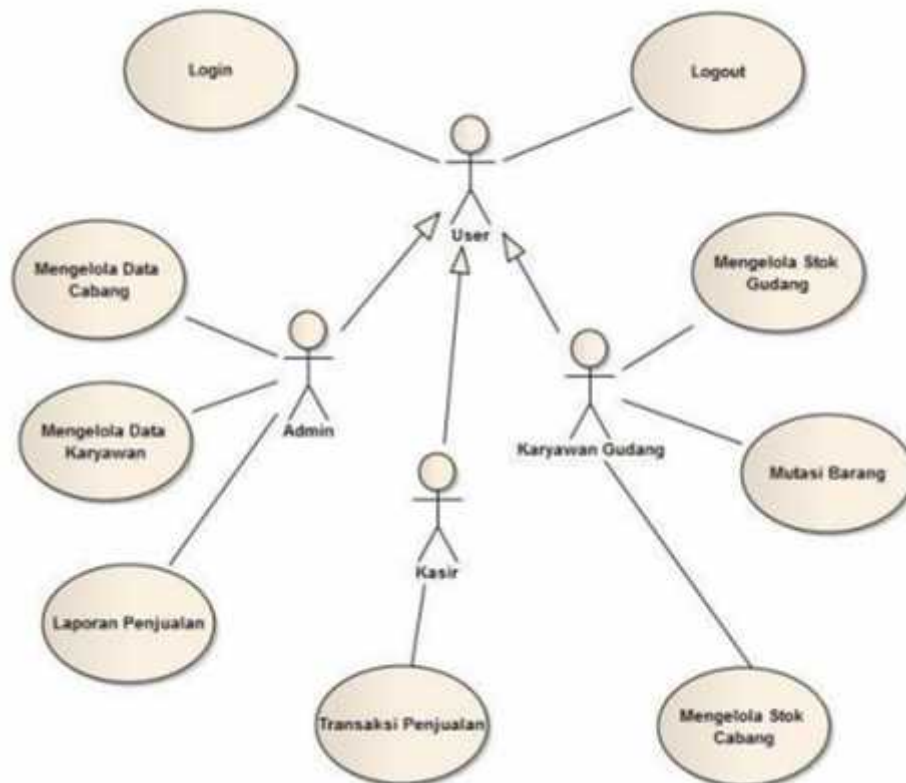
1. Menggambarkan batasan sistem dan fungsi-fungsi sistem secara umum, dibuat dengan *use case* dan aktor.
2. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan *interaction diagrams*.
3. Menggambarkan representasi struktur static sebuah sistem dalam bentuk *class diagram*.
4. Membuat model *behavior* yang menggambarkan kebiasaan atau sifat sebuah sistem dengan *state transition diagrams* UML.

5. Menyatakan arsitektur implementasi fisik menggunakan *component and development diagrams*.
6. Menyampaikan atau memperluas *functionalty* dengan *stereotypes*.

Pemodelan penggunaan UML merupakan metode pemodelan berorientasi objek dan berbasis visual. Karenanya pemodelan objek yang fokus pada pendefinisian struktur statis dan model sistem informasi yang dinamis daripada mendefinisikan data dan model proses yang tujuannya adalah pengembangan tradisional. UML menawarkan diagram yang dikelompokkan menjadi lima perspektif berbeda untuk memodelkan suatu sistem. Seperti satu set *blue print* yang digunakan untuk membangun sebuah rumah (Saipul Anwar, et al., 2016 : 75-76).

### **II.7.1. Use Case Diagram**

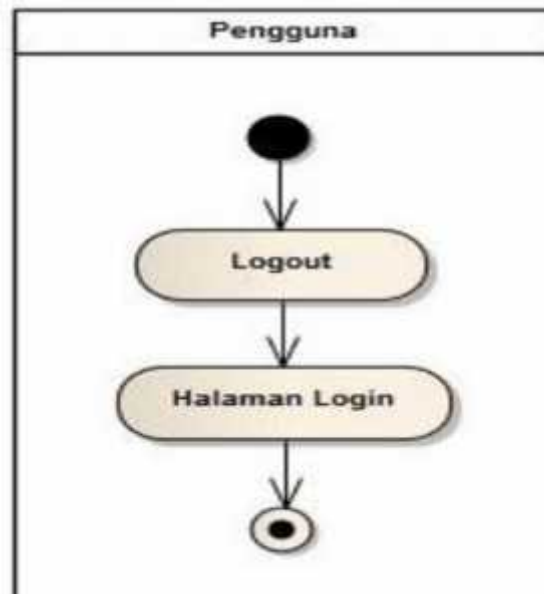
*Use case* diagram merupakan pemodelan untuk kelakuan(*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (Ade Hendini, 2016 : 108).



**Gambar II.1. Use Case Diagram**  
(Sumber : Ade Hendini, 2016 : 112)

### II.7.2. Activity Diagram

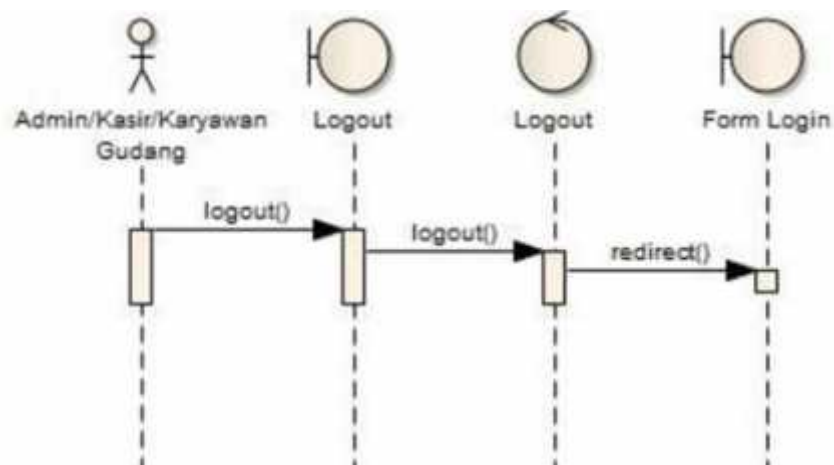
Activity diagram menggambarkan *workflow* (alirankerja) atau aktivitas dari sebuah sistem atau proses bisnis (Ade Hendini, 2016 : 109).



**Gambar II.2. Activity Diagram**  
(Sumber : Ade Hendini, 2016 : 112)

### II.7.3. Sequence Diagram

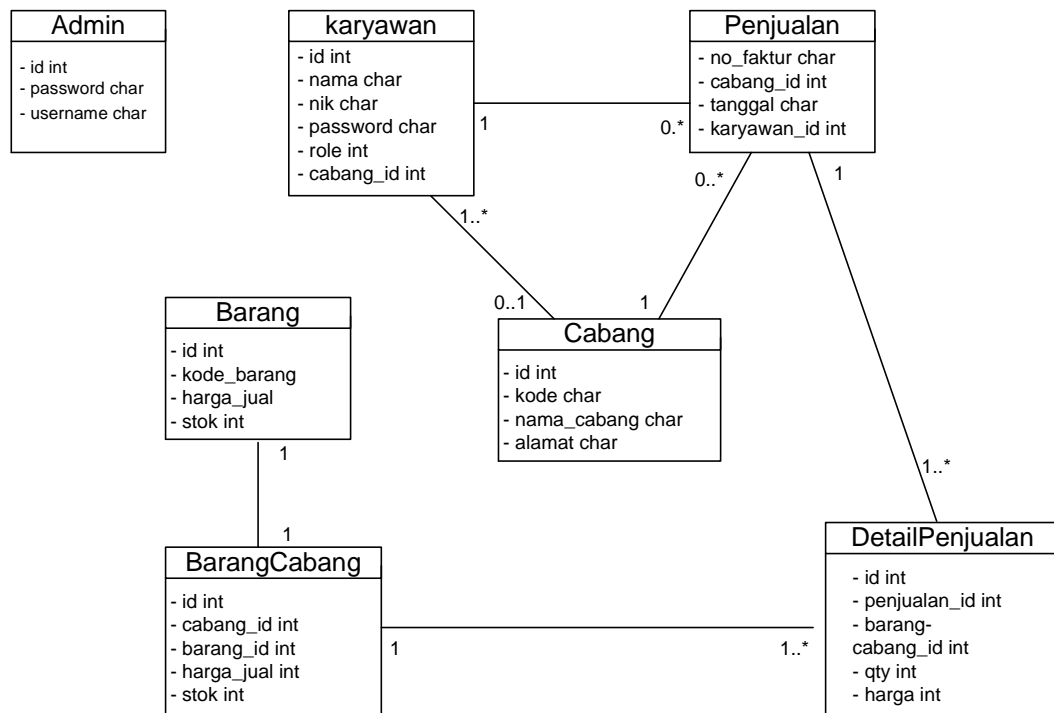
*Sequence* diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek (Ade Hendini, 2016 : 110).



**Gambar II.3. Sequence Diagram**  
(Sumber : Ade Hendini, 2016 : 114)

#### II.7.4. Class Diagram

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class* diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class* diagram secara khas meliputi: Kelas (*Class*), Relasi *Associations*, *Generalitiation* dan *Aggregation*, atribut (*Attributes*), operasi (*operation/method*) dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality* (Ade Hendini, 2016 : 111).



**Gambar II.4. Class Diagram**  
(Sumber : Ade Hendini, 2016 : 115)

## II.8. *Microsoft Visual Basic .NET*

*Microsoft Visual Basic .NET (VB .NET)* adalah suatu pengembangan aplikasi bahasa pemrograman berbasis *Visual Basic* dan merupakan bahasa pemrograman terbaru buatan *Microsoft* setelah *Microsoft Visual Basic 6.0*. Pengembangan yang signifikan dari *VB .NET* ialah kemampuannya memanfaatkan *platformNET*, sehingga pengguna dapat membuat aplikasi *Windows*, aplikasi konsol, pustaka kelas, layanan NT, aplikasi *web form*, dan *XML Web Service*, yang secara keseluruhan memungkinkan integrasi tanpa batas dengan bahasa pemrograman lain sehingga berpeluang untuk berintegrasi dengan *web*.

Beberapa keunggulan lainnya yang dimiliki *VB .NET*, seperti memiliki penanganan *debug* yang baik sehingga pembangun aplikasi dapat mengetahui kesalahan kode yang terjadi secara cepat dan memiliki *Windows form design* yang memungkinkan pembangun/*developer* memperoleh aplikasi *desktop* dalam waktu singkat. *VB .NET* memiliki *Interface Development Environment (IDE)* yang lebih lengkap dan mudah bagi *user* pemula untuk mencari komponen atau objek yang kita inginkan, seperti menempelkan kontrol-kontrol yang terdapat pada *toolbox*, mampu memformat secara otomatis ukuran *textbox*, serta mengatur *property* dari masing-masing kontrol. *VB .NET* juga memiliki *.NETFramework*. *Microsoft .NET* ialah sebuah *platform* untuk membangun, menjalankan, dan meningkatkan generasi lanjut dari aplikasi terdistribusi, memperluas *clien*, *server* dan *service-service* (Widiana Mulyani dan Bambang Eka Purnama, 2015 : 16).

## II.9. *SQL Server 2008*

*Microsoft* telah merilis generasi berikutnya dari *SQL Server 2008*, yang nantinya akan memasuki tahap *SQL Server 2008 R2*, yang mana proyek ini sendiri disebut sebagai Kilimanjaro. Tepatnya sekitar tanggal 6 Mei 2010 pihak *Microsoft* meresmikan produk baru ini. Dalam hal ini, *SQL Server 2008 R2* secara umum tersedia kira-kira satu tahun setelah perusahaan Redmond resmi mengumumkan label untuk produk yang dikembangkan sejak itu sebagai *SQL Server Kilimanjaro*. *Microsoft* mengklaim bahwa *SQL Server 2008 R2* ini lebih memiliki banyak manfaat. *SQL Server 2008 R2* milik *Microsoft* ini terus menampilkan komitmennya. Sejak *Microsoft* membuat versi rilis ini pada *Community Technology Preview CTP*) pada bulan Agustus 2009 lalu, produk ini disambut baik oleh komunitas dan telah diunduh sebanyak 150 ribu kali (Mukadar, et al., 2014 : 70).