

BAB II

LANDASAN TEORI

II.1. Pengertian Sistem

Sistem adalah suatu jaringan prosedur yang dibuat menurut pola yang terpadu untuk melaksanakan kegiatan pokok perusahaan Mulyadi (2001:5). Sedangkan pengertian sistem menurut Azhar Susanto (2001:2) Sistem adalah kumpulan/group dari bagian/komponen apapun baik fisik maupun non fisik yang saling berhubungan satu sama lain dan bekerjasama secara harmonis untuk mencapai satu tujuan tertentu (Widiana Mulyani dan Bambang Eka, 2015 : 16).

Mempelajari suatu sistem akan lebih mengena bila mengetahui terlebih dahulu apakah sistem itu. Pengertian tentang sistem pertama kali dapat diperoleh dari defenisi sistem itu sendiri. Jika kita perhatikan dengan seksama, diri kita juga terdiri dari berbagai sitem yang berfungsi untuk mengatur kita pada tujuan hidup kita.

Sistem merupakan kumpulan elemen yang salin berkaitan yang bertanggung jawab memproses masukan (*input*) sehingga menghasilkan keluaran (*output*). Suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajerial, dan merupakan kegiatan stategi dari suatu organisasi, serta menyediakan laporan-laporan yang diperlukan oleh pihak luar (Elena Monica, et al. 2015 : 63).

II.1.1. Konsep Dasar Sistem

Gordon B. Davis dalam bukunya menyatakan bahwa sistem bisa berupa abstrak atau fisik. Sistem yang abstrak adalah susunan gagasan-gagasan atau konsepsi yang teratur yang

saling bergantung. Sedangkan sistem yang bersifat fisik adalah serangkaian unsur yang bekerja sama untuk mencapai suatu tujuan.

Norman L. Enger menyatakan bahwa suatu sistem dapat terdiri atas kegiatan-kegiatan yang berhubungan guna mencapai tujuan-tujuan perusahaan seperti pengendalian inventaris atau penjadwalan produksi. Sedangkan Prof. Dr. Mr. S. Prajudi Atmosudirdjo menyatakan bahwa suatu sistem terdiri atas objek-objek atau unsur-unsur atau komponen-komponen yang berkaitan dan berhubungan satu sama lainnya sedemikian rupa sehingga unsur-unsur tersebut merupakan suatu kesatuan pemrosesan atau pengolahan yang tertentu.

Secara sederhana sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisasi, saling berinteraksi, saling tergantung satu sama lain dan terpadu (Tata Sutabri, 2012: 3-7).

II.1.2. Karakteristik Sistem

Model umum sebuah sistem terdiri dari *input*, proses, dan *output*. Hal ini merupakan konsep sebuah sistem yang sangat sederhana mengingat sebuah sistem dapat mempunyai beberapa masukan dan keluaran sekaligus. Selain itu sebuah sistem juga memiliki karakteristik atau sifat-sifat tertentu, yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem (Tata Sutabri, 2012: 13-14).

Adapun karakteristik yang dimaksud adalah sebagai berikut:

1. Komponen Sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang bekerja sama membentuk satu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem.

2. Batasan Sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem lainnya atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisah-pisahkan.

3. Lingkungan Luar Sistem (*Environment*)

Bentuk apapun yang ada diluar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem. Lingkungan luar sistem ini dapat menguntungkan dan dapat juga merugikan sistem tersebut.

4. Penghubung Sistem (*Interface*)

Media yang menghubungkan sistem dengan subsistem yang lain disebut dengan penghubung sistem atau *interface*. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lain. Keluaran suatu subsistem akan menjadi masukan untuk subsistem yang lain dengan melewati penghubung.

5. Masukan Sistem (*Input*)

Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*).

6. Keluaran Sistem (*Output*)

Hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain.

7. Pengolah Sistem (*Procces*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran.

8. Sasaran Sistem (*Objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat deterministik. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

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

Menurut Dadan Umar Daihani konsep Sistem Pendukung Keputusan (SPK) pertama kali diungkapkan pada awal tahun 1970-an oleh Michael S.Scott Morton yang menjelaskan bahwa sistem pendukung keputusan adalah suatu sistem yang berbasis komputer yang ditujukan untuk membantu pengambil keputusan dalam memanfaatkan data dan model tertentu untuk memecahkan berbagai persoalan yang tidak terstruktur. Definisi lain Sistem Pendukung Keputusan merupakan sebuah sistem yang dimaksudkan untuk mendukung para pengambil keputusan manajerial dalam situasi keputusan semi terstruktur .

Sistem Pendukung Keputusan lain merupakan sebuah sistem yang dimaksudkan untuk mendukung para pengambil keputusan manajerial dalam situasi keputusan semi terstruktur. SPK dimaksudkan untuk menjadi alat bantu bagi para pengambil keputusan untuk memperluas kapabilitas mereka, namun tidak untuk menggantikan penilaian.

Sistem pendukung keputusan menunjukkan sebagai sebuah sistem yang mendukung para pengambil keputusan untuk memperluas kapabilitas mereka, namun tidak untuk menggantikan penilaian mereka. Sistem pendukung keputusan merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan dan memanipulasi data. Sistem itu digunakan untuk membantu pengambilan keputusan dalam situasi yang semiterstruktural dan situasi yang tidak terstruktur, dimana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat

Dari beberapa definisi diatas, dapat dikatakan bahwa Sistem Pendukung Keputusan adalah suatu sistem informasi spesifik yang ditujukan untuk membantu manajemen dalam mengambil keputusan yang berkaitan dengan persoalan yang bersifat semi struktur dan tidak struktur. Sistem ini memiliki fasilitas untuk menghasilkan berbagai alternatif yang secara interaktif dapat digunakan oleh pemakai. Sistem ini berbasis komputer yang dirancang untuk meningkatkan efektivitas pengambilan keputusan dalam memecahkan masalah yang bersifat semi terstruktur dan tidak terstruktur (Mutammimul Ula, 2014 : 37-38).

II.2.1. Karakteristik Sistem Pendukung Keputusan

Konsep sistem pendukung keputusan pertama kali diperkenalkan pada tahun 1970-an oleh Michael S.Scott Morton dengan istilah *Management Desicion Model* (Sprague, 1982). Konsep sistem pendukung keputusan ditandai dengan sistem interaktif berbasis komputer yang membantu membentuk keputusan, memanfaatkan data dan model untuk menyelesaikan masalah- masalah yang tidak terstruktur. Pada dasarnya sistem pendukung keputusan dirancang untuk mendukung seluruh tahap pengambilan keputusan mulai dari mengidentifikasi masalah, memilih data yang relevan, menentukan pendekatan yang digunakan dalam pengambilan keputusan, sampai mengevaluasi pemilihan interaktif. Peranan sistem pendukung keputusan dalam konteks keseluruhan sistem informasi. Terdapat sepuluh karakteristik dasar sistem pendukung keputusan yang efektif (Maulidia Indapuri, 2014 : 86).

- a. Mendukung proses pengambilan keputusan, menitik beratkan pada *management by perceptio*.
- b. Adanya *interface* manusia atau mesin dimana manusia (*user*) tetap mengontrol proses pengambilan keputusan.
- c. Menggunakan model matematis dan statistik yang sesuai.
- d. Memiliki kapabilitas dialog untuk memperoleh informasi sesuai dengan kebutuhan model interaktif.
- e. *Output* ditunjukkan untuk personil organisasi dalam semua tingkatan.
- f. Memiliki subsistem – subsistem yang terintegrasi sedemikian rupa sehingga dapat berfungsi sebagai kesatuan sistem.
- g. Membutuhkan struktur data komprehensif yang dapat melayani kebutuhan tingkatan manajemen.
- h. Pendekatan *easy to use*. Ciri suatu sistem pendukung keputusan yang efektif adalah kemudahan untuk digunakan dan memungkinkan keseluruhan keleluasaan pemakai untuk memilih atau mengembangkan pendekatan-pendekatan baru dalam membahas masalah yang dihadapi.
- i. Kemampuan sistem beradaptasi secara tepat, dimana pengambil keputusan dapat menghadapi menangani dengan cara mengadaptasi sistem terhadap kondisi-kondisi dan perubahan yang terjadi.
- j. Mendukung pengambilan keputusan untuk membahas masalah-masalah terstruktur, semi terstruktur dan tidak terstruktur.

II.3. Logika *Fuzzy*

II.3.1. Pengertian 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.3.2. Metode TSukamoto

Sistem Inferensi Fuzzy merupakan suatu kerangka komputasi yang didasarkan pada teori himpunan fuzzy, aturan fuzzy berbentuk IF-THEN, dan penalaran fuzzy. Selama ini telah dikenal beberapa metode dalam FIS, seperti metode Tsukamoto, metode Mamdani dan metode Sugeno.

Pada metode Tsukamoto, setiap konsekuen pada aturan yang berbentuk IF-THEN harus direpresentasikan dengan suatu himpunan fuzzy dengan fungsi keanggotaan monoton. Sebagai hasilnya, keluaran hasil inferensi dari tiap-tiap aturan diberikan secara tegas (*crisp*) berdasarkan α -predikat (*fire strength*). Hasil akhir menggunakan rata-rata terbobot. 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 fungsi implikasi MIN untuk mendapatkan nilai α -predikat tiap-tiap rule ($\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_n$) Kemudian masing-masing nilai α -predikat ini digunakan untuk menghitung keluaran hasil inferensi secara tegas (*crisp*) masing-masing *rule* ($z_1, z_2, z_3, \dots, z_n$)

- d. Defuzzyfikasi

Menggunakan metode rata-rata (*Average*)

$$Z^* = \frac{\sum \alpha_i Z_i}{\sum \alpha_i} \dots\dots\dots(1)$$

Keterangan :

Z= Variabel *output*

α_i = Nilai α predikat

z_i = Nilai variabel *output*

Tsukamoto yaitu setiap 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 A_1 dan A_2 , sedangkan variabel y terbagi atas himpunan B_1 dan B_2 . Variabel z juga terbagi atas dua himpunan yaitu C_1 dan C_2 . Tentu saja himpunan C_1 dan C_2 harus merupakan himpunan yang bersifat monoton. Ada 2 aturan yang digunakan, yaitu:

$$[R1] \text{ IF } (x \text{ is } A_1) \text{ And } (y \text{ is } B_2) \text{ THEN } (z \text{ is } C_1) (5)$$

$$[R1] \text{ IF } (x \text{ is } A_2) \text{ And } (y \text{ is } B_2) \text{ THEN } (z \text{ is } C_2) (6)$$

Keterangan:

R1 : Aturan *fuzzy*

X : variabel pinjaman

α_1 : Himpunan pinjaman tertinggi

α_2 : Himpunan pinjaman terendah

And : Operator yang digunakan

y : Variabel jaminan

B_1 : Himpunan jaminan tertinggi

B_2 : 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. 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.4.1. Tujuan Pengembangan Basis Data

Sebagaimana usaha-usaha pada bidang yang lainnya, pelaksanaan perancangan dan penyusunan basis data tentu mempunyai tujuan. Tujuan basis data telah diperinci dengan cara yang berbeda-beda.

Sebuah tinjauan tentang tujuan pengembangan basis data yang cukup terperinci disampaikan oleh James Martin (1975). Ia membedakan tujuan pengembangan basis data menjadi dua kelompok, yaitu tujuan primer dan tujuan sekunder. Tujuan primer dimaksudkan sebagai tujuan utama yang ingin dicapai dalam setiap usaha perancangan dan pengembangan basis data. Tujuan sekunder merupakan tujuan tambahan yang dimaksudkan untuk mencapai tujuan primer (Edhy Sutanta, 2011).

II.5. 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).

Proses normalisasi menghasilkan relasi yang optimal, yaitu

1. Memiliki struktur *record* yang konsisten secara logik;
2. Memiliki struktur *record* yang mudah untuk dimengerti;
3. Memiliki struktur *record* yang sederhana dalam pemeliharaan;

4. Memiliki struktur *record* yang mudah ditampilkan kembali untuk memenuhi kebutuhan pengguna;
5. Minimalisasi kerangkapan data guna meningkatkan kinerja sistem.

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

- a. Identifikasikan *Functional Dependence (FD)* relasi *First Normal Form (1NF)*
 - b. Berdasarkan informasi tersebut, dekomposisi relasi *First Normal 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 Normal Form / 3NF*)

Suatu relasi disebut sebagai *Third Normal 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 Normal 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.
5. Bentuk normal *Boyce-Codd* (*Boyce-Codd Norm Form* / BCNF)

Bentuk normal *Boyce-Codd Norm Form* (BCNF) dikemukakan oleh R.F. Boyce dan E.F. Codd. Suatu relasi disebut sebagai *Boyce-Codd Norm Form* (BCNF) jika memenuhi kriteria sebagai berikut.

- a. Jika memenuhi kriteria *Third Normal Form* (3NF)
 - b. Jika semua atribut penentu (determinan) merupakan CK
6. Bentuk normal keempat (*Forth Norm Form* / 4NF)

Relasi disebut sebagai *Forth Norm Form* (4NF) jika memenuhi kriteria sebagai berikut.

- a. Jika memenuhi kriteria *Boyce-Codd Norm Form*.
 - b. Jika setiap atribut didalamnya tidak mengalami ketergantungan pada banyak nilai.
7. Bentuk normal kelima (*Fifth Norm Form / 5NF*)
- Suatu relasi memenuhi kriteria *Fifth Norm Form (5NF)* jika kerelasian antar data dalam relasi tersebut tidak dapat direkonstruksi dari struktur relasi yang sederhana.
8. Bentuk normal kunci domain (*Domain Key Norm Form / DKNF*)
- Relasi disebut sebagai *Domain Key Norm Form (DKNF)* jika setiap batasan dapat disimpulkan secara sederhana dengan mengetahui sekumpulan nama atribut dan domainnya selama menggunakan sekumpulan atribut pada kuncinya.

II.6. Entity Relationship Diagram (ERD)

ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. Didalam pembuatan ERD perlu di perhatikan penentuan sesuatu konsep apakah merupakan suatu entity, atribut, atau relationship. Model data *ERD* dibangun berdasarkan persepsi dari dunia nyata yang mengandung himpunan dari objek-objek yang disebut entitas dan hubungan antara objek-objek tersebut. Model *ERD* ini digunakan untuk memfasilitasi perancangan basis data dari sebuah skema organisasi, dengan mentransformasi kebutuhan suatu *basis data* dari suatu organisasi ke dalam bentuk skema konseptual yang akan menghasilkan struktur logika dari suatu *basis data*. Setiap objek yang terbentuk di dalam suatu organisasi bersifat unik (Mutammimul Ula, 2014 : 38).

II.6.1. Kelebihan dan Kelemahan ERD

Jika diterapkan dengan benar/tepat maka penggunaan ERD dalam pemodelan data memberikan keuntungan bagi perancangan maupun pengguna basis data antara lain (Edhy Susanta, 2011).

- a. Memudahkan perancang dalam hal menganalisis sistem yang akan dikembangkan.
- b. Memudahkan perancang saat merancang basis data.
- c. Rancangan basis data yang dikembangkan berdasarkan ERD umumnya telah berada dalam bentuk optimal.
- d. Dalam banyak kesempatan, penggunaan simbol-simbol grafis (termasuk ERD) lebih mudah dipahami oleh para pengguna dibandingkan bentuk naratif.
- e. Dengan menggunakan ERD, pengguna umumnya mudah memahami sistem dan basis data yang dirancang oleh perancang.

Kelemahan ERD diantaranya adalah

- a. Kebutuhan media yang sangat luas.
- b. Sering kali ERD tampil sangat ruwet.

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

Unified Modelling Language (UML) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual (Braun, *et. al.* 2001). Juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek (Whitten, *et. al.* 2004) (Haviluddin, 2011 : 1).

II.7.1. Sejarah UML (*Unified Modelling Language*)

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 Softeam juga menanggapi Request for Proposal (RFP) yang dikeluarkan oleh OMG tersebut dan menyatakan kesediaan untuk bergabung.

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 : 1-2).

II.7.2. Diagram UML (*Unified Modeling Language*)

UML memiliki beberapa jenis diagram, diagram – diagram yang termasuk di dalam UML, yaitu :

- a. *Use Case Diagram*
- b. *Activity Diagram*
- c. *Sequence Diagram*
- d. *Class Diagram*

II.7.3. Komponen-komponen *Unified Modeling Language* (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 (Sugrue J. 2009).

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, diantaranya : (Haviluddin, 2011 : 3).

1. Struktur Diagram

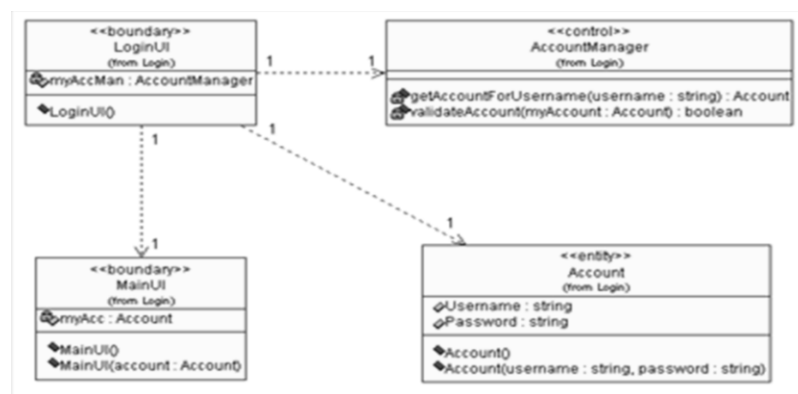
Menggambarkan elemen dari spesifikasi dimulai dengan kelas, obyek, dan hubungan mereka, dan beralih ke dokumen arsitektur logis dari suatu sistem. Beberapa struktur diagram dalam UML terdiri atas :

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 :

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



Gambar II.1. Notasi Class Diagram
(Sumber : Haviluddin, 2011 : 3)

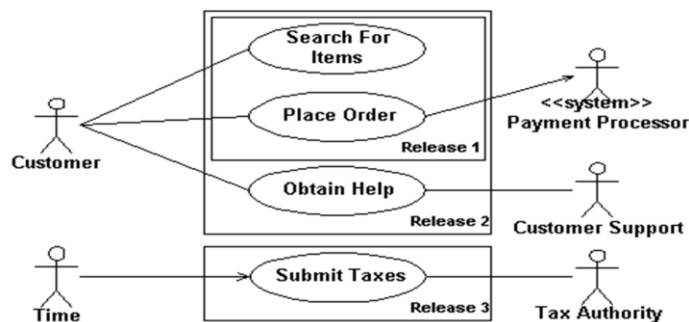
2. *Behavior Diagram*

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

a. *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, yaitu :

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



Gambar II.2. Notasi Use Case Diagram
(Sumber : Haviluddin, 2011 : 4)

b. *Activity diagram*

Menggambarkan aktifitas-aktifitas, objek, *state*, *transisi state* dan *event*. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas. Berikut notasi *object diagram* dapat dilihat pada Gambar II.4. di bawah ini.

Gambar II.4. Notasi Sequence Diagram
(Sumber : Haviluddin, 2011 : 5)

Untuk menggambarkan analisa dan desain diagram, UML memiliki seperangkat notasi yang akan digunakan ke dalam tiga kategori diatas yaitu struktur diagram, behaviour diagram, dan interaction diagram. Berikut beberapa notasi dalam UML diantaranya :

- 1) *Actor*, menentukan peran yang dimainkan oleh user atau sistem lain yang berinteraksi dengan subjek. *Actor* adalah segala sesuatu yang berinteraksi langsung dengan sistem aplikasi komputer, seperti orang, benda atau lainnya. Tugas actor adalah memberikan informasi kepada sistem dan dapat memerintahkan sistem untuk melakukan sesuatu tugas.
- 2) *Class diagram*, Notasi utama dan yang paling mendasar pada diagram UML adalah notasi untuk mempresentasikan suatu class beserta dengan atribut dan operasinya. Class adalah pembentuk utama dari sistem berorientasi objek.
- 3) *Use Case* dan *use case specification*, *Use case* adalah deskripsi fungsi dari sebuah sistem perspektif pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara user (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem disebut skenario. *Use case* merupakan awal yang sangat baik untuk setiap fase pengembangan berbasis objek, design, testing, dan dokumentasi yang menggambarkan kebutuhan sistem dari sudut pandang di luar sistem. Perlu diingat bahwa *use case* hanya menetapkan apa yang seharusnya dikerjakan oleh

sistem, yaitu kebutuhan fungsional sistem dan tidak untuk menentukan kebutuhan nonfungsional, misalnya: sasaran kinerja, bahasa pemrograman dan lain sebagainya.

- 4) *Interaction, Interaction* digunakan untuk menunjukkan baik aliran pesan atau informasi antar obyek maupun hubungan antar obyek.
- 5) *Association, Association* menggambarkan navigasi antar *class (navigation)*, berapa banyak obyek lain yang bisa berhubungan dengan satu obyek (*multiplicity antar class*) dan apakah suatu class menjadi bagian dari class lainnya (*aggregation*).

II.8. Microsoft Visual Basic 2010

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 platform NET, 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 .NET Framework. Microsoft .NET ialah sebuah platform untuk membangun, menjalankan, dan meningkatkan generasi lanjut dari aplikasi terdistribusi, memperluas klien, server dan service-service (Widiana Mulyani, Bambang Eka Purnama : 2015 : 16).

II.9. Microsoft SQL Server 2008

SQL (*Structured Query Language*) adalah sebuah bahasa yang dipergunakan untuk mengakses data dalam basis data relasional. Bahasa ini secara *de facto* merupakan bahasa standar yang dipergunakan dalam manajemen basis data relasional. Saat ini hampir semua server basis data yang ada mendukung bahasa ini untuk melakukan manajemen datanya.

SQL terdiri dari dua bahasa, yaitu *Data Definition Language Manipulation Language (DDL)*. Implementasi DDL dan DML sistem manajemen basis data (*SMBD*), namun secara umum implemen bahasa ini memiliki bentuk standar yang ditetapkan oleh *ANSI*. (Adelia, Jimmy Setiawan, 2011 : 115).