BAB II

TINJAUAN PUSTAKA

II.1. Sistem Pakar

II.1.1. Pengertian Sistem Pakar

Sistem Pakar (*Expert System*) adalah sistem yang menngunakan pengetahuan manusia, dimana pengetahuan tersebut dimasukkan ke dalam sebuah komputer, dan kemudian digunakan untuk menyelesaikan masalah-masalah yang biasanya membutuhkan kepakaran atau keahlian manusia. (Muhammad Dahria, Rosindah Silalahi, dkk, Vol 12; 2013: 1).

Dapat diambil kesimpulan bahwa sistem pakar adalah sebuah sistem yang dapat menirukan keahlian seorang pakar dalam menyelesaikan masalah.

II.1.2 Manfaat Sistem Pakar

Secara garis besar, banyak manfaat yang dapat diambil dengan adanya sistem pakar, antara lain :

1. Memungkinkan orang awam bisa mengerjakan pekerjaan para ahli.
2. Bisa melakukan proses secara berulang secara otomatis.
3. Menyimpan pengetahuan dan keahlian para pakar.
4. Meningkatkan output dan produktivitas
5. Meningkatkan kualitas.
6. Mampu mengambil dan melestarikan keahlian para pakar (terutama yang termasuk keahlian langka)
7. Mampu beroperasi dalam lingkungan yang berbahaya

8. Memiliki kemampuan untuk mengakses pengetahuan.
9. Memiliki reabilitas.
10. Meningkatkan kapabilitas sistem komputer.
11. Memiliki kemampuan untuk bekerja dengan informasi yang tidak lengkap dan mengandung ketidak pastian.
12. Sebagai media pelengkap dalam pelatihan.
13. Meningkatkan kapabilitas dalam penyelesaian masalahMenghemat waktu dalam pengambilan keputusan. (Jusuf Wahyudi, Juju Jumadi, Vol. 7 No. 1 Februari 2011)

II.1.3. Kelebihan Sistem Pakar

Sistem Pakar memiliki beberapa fitur menarik yang merupakan kelebihannya, seperti :

1. Meningkatkan ketersedian (*increased availability*). Kepakaran atau keahlian menjadi tersedia dalam sistem komputer. Dapat dikatakan bahwa sistem pakar merupakan produksi kepakaran secara masal (*massproduction*).
2. Mengurangi biaya *(reduced cost).* Biaya yang diperlukan untuk menyediakan keahlian per satu orang *user* menjadi berkurang.
3. Mengurangi Bahaya *(reduced danger)*. Sistem pakar dapat digunakan di lingkungan yang mungkin berbahaya bagi manusia.
4. Permanen *(permanence).* Sistem pakar dan pengetahuan yang terdapat di dalamnya bersifat lebih permanen dibandingkan manusia

yang dapat merasa lelah, bosan, dan pengetahuannya hilang saat sang pakar meninggal dunia.

5. Keahlian Multipel *(multiple expertise).* Pengetahuan dari beberapa pakar dapat dimuat ke dalam sistem dan bekerja secara simultan dan kontinyu menyelesaikan suatu masalah setiap saat. Tingkat keahlian atau pengetahuan yang digabungkan dari beberapa pakar dapat melebihi pengetahuan satu orang pakar . (Rika Rosnelly;2011,5)

II.1.4. Kekurangan Sistem Pakar

Selain manfaat,sistem pakar juga memiliki beberapa kelemahan, diantaranya:

1. Biaya yang diperlukan untuk membuat dan memeliharanya sangat mahal.
2. Sulit dikembangkan. Hal ini tentu saja erat kaitannya dengan ketersedian pakar di bidangnya
3. Sistem Pakar tidak 100% bernilai benar. (Jusuf Wahyudi, Juju Jumadi, Vol. 7 No. 1 Februari 2011)

II.1.5. Elemen Manusia Pada Sistem Pakar

Sistem pakar tidak lepas dari elemen manusia ayng terkait di dalamnya. Personil yang terkait dengan sistem pakar ada 4, yaitu :

1. Pakar

Pakar adalah seorang individu yang memiliki pengetahuan khusus, pemahaman, pengalaman, dan metode-metode yang digunakan untuk

memecahkan persoalan dalam bidang tertentu. Pakar juga memiliki kemampuan untuk mengaplikasikan pengetahuannya dan memberikan saran serta pemecahan masalah pada domain tertentu.

Seorang pakar mengetahui fakta – fakta mana yang penting, sebab akibat, fenomena – fenomena yang terkait dengan fakta, memahami arti hubungan antar fakta, juga hubungan sebab akibat, dan hubungan dengan fenomena – fenomena yang terkait serta mampu menginterprestasikan akibat – akibat yang terjadi karena sesuatu sebab terjadi (Rika Rosnelly; 2011: 10).

2. Pembangun / Pembuat Pengetahuan

Pembangun pengetahuan memiliki tugas utama menerjemahkan dan merepresentasikan pengetahuan yang diperoleh dari pakar, baik berupa pengalaman pakar dalam menyelesaikan masalah maupun sumber terdukumentasi lainnya kedalam bentuk yang bisa diterima oleh sistem pakar. Dalam hal ini pembangun pengetahuan *(knowledge engineer)* menginterpretasikan dan merepresentasikan pengetahuan yang diperoleh dalam bentuk jawaban-jawaban atas pertanyaan-pertanyaan yang diajukan pada pakar atau pemahaman, penggambaran, analogis, sistematis, konseptual yang diperoleh dari membaca beberapa dokumen cetak seperti *text book,* jurnal, makalah, dan sebagainya, Kurangnya pengalaman *Knowledge engineer* merupakan kesulitan utama dalam mengkonstruksi sistem pakar. Untuk mengatasi hal tersebut, perancang sistem pakar menggunakan

tools komersial. (Seperti pada editor-editor khusus maupun *logic debuggers*) dan usahanya akan dipusatkan pada pembangunan mesin inferensi (Rika Rosnelly; 2011: 11).

3. **Pembangun / Pembuat Sistem**

   Pembangun sistem adalah orang yang bertugas untuk merancang antarmuka pemakai sistem pakar, merancang pengetahuan yang sudah diterjemahkan oleh pembangun pengetahuan kedalam bentuk yang sesuai dan dapat diterima oleh sistem pakar dan mengimplementasikannya kedalam mesin infrensi (Rika Rosnelly; 2011: 11).

4. **Pengguna *(User)***

   Banyak sistem berbasis komputer mempunyai susunan pengguna tunggal. Hal ini berbeda jauh dengan sistem pakar yang memungkinkan mempunyai beberapa kelas pengguna. Table II.1 menunjukkan beberapa contoh hubungan antara kelas pengguna, kepentingan pengguna, dan fungsi sistem paar. (Rika Rosnelly; 2011: 12).

Table. II.1 Hubungan antara pengguna dan fungsi sistem pakar

Pengguna	**Kepentingan**	**Fungsi sistem pakar**
Klien bukan pakar	Mencari saran/nasehat	Konsultan atau penasehat
Mahasiswa	Belajar	Instruktur
Pembangun sistem	Memperbaiki/menambah basis pengetahuan	Rekan (partner)
Pakar	Membantu analisi rutin atau proses komputasi, mecari (mengklasifikasi) informasi, alat bantu diagnose	Rekan kerja atau asisten

Sumber (Rika Rosnelly;2011,12)

II.1.6. Karakteristik Sistem Pakar

Sistem Pakar umumnya dirancang untuk memenuhi beberapa karaktristik umum berikut:

1. Kinerja yang sangat baik *(high performance).* Sistem harus mampu memberikan respon berupa saran (advice) dengan tingkat kualitas yang sama dengan seorang pakar atau melebihinya.
2. Waktu respon yang baik *(adequate respon time).* Sistem juga harus mampu bekerja dalam waktu yang sama baiknya *(reasonable)* atau lebih cepat dibandingkan dengan seorang pakar dalam menghasilkan keputusan. Hal ini sangat penting terutama pada sistem waktu nyata (real-time).

3. Dapat diandalkan *(good reliability).* Sistem harus dapat diandalkan dan tidak mudah rusak/crash.
4. Dapat Dipahami *(understandable).* Sistem ini harus mampu menjelaskan langkah-langkah penalaran yang dilakukannya seperti seorang pakar.
5. Fleksibel *(flexibility).* Sistem harus menyediakan mekanisme untuk menambah, mengubah, dan menghapus pengetahuan. (Rika Rosnelly; 2011: 20)

II.1.7. Kemampuan Menjelaskan (*Explanation Capability*)

Fasilitas lain dari sistem pakar adalah kemampuannya untuk menjelaskan saran atau rekomendasi yang diberikannya. Penjelasan dilakukan dalam subsistem yang disebut subsistem penjelasan (*explanation*). Bagian dari sistem ini memungkinkan sistem untuk memerikasa penalaran yang dibuatnya sendiri dan menjelaskan operasi-operasinya.

Karakteristik dan kemampuan yang dimiliki oleh sistem pakar berbeda dengan sistem konvensional. Perbedaan ini ditunjukkan pada Tabel II.1

Tabel II.2 Perbandingan antara Sistem Konvensional dengan Sistem Pakar

Sistem konvensional	Sistem Pakar
Informasi dan pemrosesnya biasanya jadi satu dengan program	Basis pengetahuan merupakan bagian terpisah dari mekanisme inferensi
Biasanya tidak bisa menjelaskan suatu input data itu dibutuhkan atau bagaimana output itu diperoleh	Penjelasanya adalah bagian terpenting dari sistem pakar
Pengubahan program cukup sulit dan membosankan	Pengubahan aturan dapat dilakukan dengan mudah
Sistem hanya akan beroperasi jika sistem tersebut sudah lengkap	Sistem dapat beroperasi hanya dengan beberapa aturan
Eksekusi hanya dilakukan langkah demi langkah	Eksekusi dilakukan pada keseluruhan basis pengetahuan
Mengunakan data	Menggunakan pengetahuan
Tujuan utamanya adalah efisien	Tujuan utamanya adalah efektifitas

Sumber: (Jusuf Wahyudi, Juju Jumadi, Vol. 7 No. 1 Februari 2011)

II.1.8. Ciri-Ciri Sistem Pakar

Ciri dari sistem pakar yaitu terbatas pada domain keahlian tertentu, dapat memberikan penalaran untuk berbagai macamd ata yang tidak pasti, dapat mengemukakan rangkaian alasanalasan yang diberikannya dengan cara yang dapat dipahami, berdasrkan pada aturan rule tertentu, dirancang untuk dapat dikembangkan secara bertahap, pengetahuan dan mekanisme inferensi jelas terpisah, keluaranya bersifat anjuran, serta sistemnya dapat

mengaktifkan aturan secara searah yang dituntun oleh dialog dengan pemakai. Sistem pakar yang baik harus memenuhi cirri-ciri sebagai berikut : Memiliki Informasi yang handal, Mudah dimodifikasi, dapat digunakan dalam berbagai jenis computer dan Memiliki kemampuan untuk belajar beradaptasi. (Jusuf Wahyudi, Juju Jumadi, Vol. 7 No. 1 Februari 2011)

II.2. Penyakit kanker tenggorokan

Kanker tenggorokan adalah tumor yang tumbuh dan berkembang di tenggorokan, sekitar faring, laring, atau tonsil. Sama seperti kanker mulut dan lidah, sebagian besar kanker tenggorokan yang dialami pasien memiliki jenis karsinoma sel skuamosa. Semua kanker terjadi akibat adanya mutasi pada sel-sel. Mutasi inilah yang memicu pertumbuhan sel yang tidak terkendali. Itulah yang terjadi dengan kanker tenggorokan. Penyebab di balik proses mutasi tersebut belum diketahui secara pasti. Tetapi ada beberapa faktor yang diduga dapat meningkatkan risiko seseorang untuk terkena kanker tenggorokan, misalnya usia, pola hidup, serta kondisi medis. Risiko kanker tenggorokan akan meningkat seiring bertambahnya usia seseorang. Kanker ini umumnya terjadi pada orang yang berusia di atas 60 tahun. Kemungkinan munculnya kanker tenggorokan juga berhubungan erat dengan pola hidup yang kurang sehat. Beberapa contohnya meliputi: Mengonsumsi tembakau, baik dalam bentuk rokok maupun kunyah, mengonsumsi minuman keras yang berlebihan, kurang mengonsumsi buah dan sayur. Selain itu, terdapat beberapa kondisi medis tertentu yang bisa menjadi faktor pemicu kanker tenggorokan.

Contohnya infeksi virus HPV (*human papillomavirus*) dan penyakit asam lambung atau GERD. (Alo dokter, 2015, pengertian kanker tenggorokan, http://www.alodokter.com/kanker-tenggorokan)

II.2.1. Jenis-Jenis kanker tenggorokan

a. Kanker nasofaring (tenggorokan bagian atas di belakang hidung). Kebanyakan kasus kanker nasofaring muncul dari terhirupnya polutan dan toksin melalui hidung dalam tempo yang lama. Paparan yang berketerusan ini bisa membentuk iritasi pada hidung. Pada awalnya iritasi ini hanya menyebabkan rasa nyeri. Namun ketika terus berkembang, maka iritasi bisa berubah menjadi infeksi yang membentuk inflamasi, abses sampai akhirnya berubah menjadi pertumbuha sel abnormal seperti kanker.

b. Kanker orofaring (tenggorokan bagian tengah, dibelakang lidah). Orofaring adalah bagian tengah ini yang berisi pembukaan dari bagian belakang mulut. Ini termasuk amandel, pangkal lidah, langit-langit lunak dan dinding tenggorokan di wilayah ini. Menghubungkan ke nasofaring (bagian atas tenggorokan) di atas dan hipofaring (bagian terendah dari tenggorokan).

c. Kanker hipofaring (tenggorokan bagian bawah). Kanker Hipofaring adalah tipe dari kanker tenggorok yang terjadi pada hipofaring, daerah yang terletak pada bagian bawah dari faring (tenggorok)

sampai bagian atas dari trakea (pipa udara) dan esofagus (pipa yang menyalurkan makanan ke lambung).

d. Kanker laring (pita suara). Kanker Laring adalah keganasan pada pita suara, kotak suara (*laring*) atau daerah lainnya di tenggorokan, Kanker laring biasanya berasal dari pita suara, menyebabkan suara serak.

II.3. *Dempster Shaffer*

II.3.1. Pengertian

Ada berbagai macam penalaran dengan model yang lengkap dan sangat konsisten, tetapi pada kenyataannya banyak permasalahan yang tidak dapat terselesaikan secara lengkap dan konsisten. Ketidak konsistennya yang tersebut adalah akibat adanya penambahan fakta baru. Penalaran yang seperti itu disebut dengan penalaran *non monotonis*. Untuk mengatasi ketidakkonsistenan tersebut maka dapat menggunakan penalaran dengan teori *Dempster Shafer*.*Dempster Shafer* adalah suatu teori matematika untuk pembuktian berdasarkan *belief function and plausible reasoning* (fungsi kepercayaan dan pemikiran yang masuk akal), yang digunakan untuk mengkombinasikan potongan informasi yang terpisah (bukti) untuk mengkalkulasi kemungkinan dari suatu peristiwa. Teori ini dikembangkan oleh Arthur P. Dempster Dan Glenn Shafer. (Muhammad Dahria, Rosindah Silalahi, dkk, Vol 12; 2013: 1).

Secara umum teori *Dempster Shafer* ditulis dalam suatu *interval Belief* dan *Plausibility*.

a. *Belief (Bel)* adalah ukuran kekuatan *evidence* dalam mendukung suatu himpunan proposisi. Jika bernilai 0 maka mengindikasikan bahwa tidak ada *evidence*, dan jika bernilai 1 menunjukkan adanya kepastian.

b. Plausibility (Pl) dinotasikan sebagai :Pl(s) = 1 – Bel (¬s)

Plausibility juga bernilai 0 sampai 1. Jika yakin akan ¬s, maka dapat dikatakan bahwa Bel(¬s)=1, dan Pl(¬s)=0. Pada teori *Dempster Shafer* dikenal adanya *frame of discrement* yang dinotasikan dengan θ. *Frame* ini merupakan semesta pembicaraan dari sekumpulan hipotesis.

Tujuannya adalah mengaitkan ukuran kepercayaan elemen-elemen θ. Tidak semua *evidence* secara langsung mendukung tiap-tiap elemen. Untuk itu perlu adanya probabilitas fungsi densitas (m). Nilai m tidak hanya mendefinisikan elemen-elemen θ saja, namun juga semua subsetnya. Sehingga jika θ berisi n elemen, maka subset θ adalah 2n . Jumlah semua m dalam subset θ sama dengan 1.

Apabila tidak ada informasi apapun untuk memilih hipotesis, maka nilai : m{θ} = 1,0.Apabila diketahui X adalah subset dari θ, dengan m1 sebagai fungsi densitasnya, dan Y juga merupakan subset dari θ dengan m2 sebagai fungsi densitasnya, maka dapat dibentuk fungsi kombinasi m1 dan m2 sebagai m3, yaitu:

$$M_3(Z) = \frac{\sum x \cap y = z^m 1(x).^m 2(y)}{1 - \sum x \cap y = \emptyset^m 1(x).^m 2(y)} \qquad \text{.....(1)}$$

II.3.2. Kelebihan dan Kekurangan *Dempster Shaffer*

Kelebihan dari metode *Dempster Shaffer* adalah:

a. Kesulitan dalam menentukan nilai *prior probability*.
b. Aturan kombinasi dapat digunakan untuk menggabungkan bukti-bukti.
c. Dalam keadaan atau situasi tidak pasti, *ignorance* dapat ditentukan.
d. Mudah untuk menentukan bukti-bukti dengan tingkat abstraksi yang berbeda-beda.

Sedangkan kekurangan metode *Dempster Shaffer* adalah:

a. Perhitungan komputasi yang kompleks.
b. Teori Pengambilan keputusan yang kurang
c. Eksperimen perbandingan antara teori *Dempster Shaffer* dengan teori *Probabilitas* sulit untuk dilakukan.
d. Tidak adanya keuntungan yang dapat terlihat dengan jelas pada teori *Dempster Shaffer.*

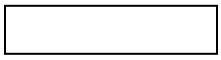
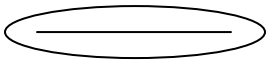
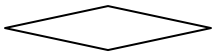
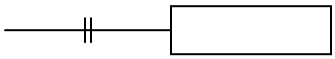
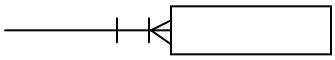
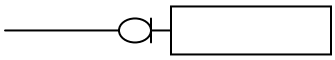
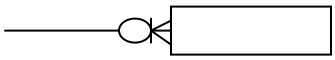
II.4. Basis Data

Basis data adalah kumpulan data (arsip, atau file) yang saling berhubungan yang disimpan dalam media penyimpanan elektronis agar dapat dimanfaatkan kembali dangan cepat, dan mudah. Sedangkan sistem basis data adalah kumpulan file, atau tabel yang saling berhubungan yang memungkinkan beberapa pemakai, atau program lain untuk mengakses, dan memanipulasi file-file (tabel) tersebut (Eka Kurniawan, 2015)

II.5. ERD (*Entity Relationship Diagram*)

Entity Relationship Diagram (*ERD*) adalah bagian yang menunjukkan hubungan antara entity yang ada dalam sistem. Simbol-simbol yang digunakan dapat dilihat dari tabel II.6. (Yuhendra, M.T, Dr. Eng dan Riza Eko Yulianto, 2015, Hal : 70).

Tabel II.6. Simbol Yang Digunakan Pada *Entity Relationship Diagram (ERD)*

SIMBOL	KETERANGAN
	*Entity*
	Atribut Dan *Entity*
	Atribut Dan *Entity* Dengan *Key* (Kunci)
	Relasi Atau Aktifitas Antar *Entity*
	Hubungan Satu Dan Pasti
	Hubungan Banyak Dan Pasti
	Hubungan Satu Tapi Tidak Pasti
	Hubungan Banyak Tapi Tidak Pasti

(Sumber : Yuhendra, M.T, Dr. Eng dan Riza Eko Yulianto; 2015)

II.6. Kamus Data

Kamus data merupakan sebuah daftar yang terorganisasi dari elemen data yang berhubungan dengan sistem, dengan definisi yang tegar dan teliti sehingga pemakai dan analis sistem akan memiliki pemahaman yang umum mengenai *input*, *output*, komponen penyimpanan. (Yusi Ardi Binarso, 2012, Hal : 74).

Contoh kamus data untuk entitas wisuda dan alumni yang digunakan berdasarkan sistem informasi alumni teknik informatika adalah sebagai berikut:

1. Data Wisuda

   Wisuda = @idWisuda + bulan + tahun + jumlahPeserta

   @Wisuda = { integer }

   Bulan = [januari|…|Desember]

   Tahun = [2000|…|2999]

   Jumlah Peserta = { integer }

   Integer = [0-9]

2. Data Alumni

   Alumni = @NIM + namaLengkap + password + email + tglLahir + jenisKelamin + noTelp + alamatAsal + kotaAsal + alamatSekarang + kotaSekarang + instansi + jabatan + judulTA + lamaTA + tglLulus + lamaStudi + IPK + foto + jawabanPK + kunciAktivasi + statusAktif + idWisuda + idPertanyaan

   @NIM = 1 { character } 14

   namaLengkap = 1 { character } 50

   password = 1 { character } 50

email = 1 { character } 25

tglLahir = date

jenisKelamin = { L | P }

noTelp = 1 { character } 20

alamatAsal = 1 { character } 100

kotaAsal = 1 { character } 20

alamatSekarang = 1 { character } 100

kotaSekarang = 1 { character } 20

instansi = 1 { character } 50

jabatan = 1 { character } 50

judulTA = 1 { character } 250

lamaTA = { integer }

tglLulus = date

lamaStudi = { integer }

IPK = decimal

Foto = 1 { character } 50

JawabanPK = 1 { character } 20

kunciAktivasi = 1 { character } 65

statusAktif = [aktif | non aktif]

idWisuda = *dapat dilihat pada data Wisuda

idPertanyaan = *dapat dilihat pada data Pertanyaan_Keamanan

character = [A-Z | a-z | 0-9 | |]

integer = [0-9]

II.7. Normalisasi

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

Proses normalisasi menghasilkan relasi yang optimal, yaitu (Martin, 1975) : (Edy Sutanta ; 2011 : 175)

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 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.

5. Bentuk normal *Boyce-Cood (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 Norm Form* (3NF)
   b. Jika semua atribut penentu (determinan) merupakan CK

6. Bentuk normal keempat (*Forth Norm Form* / 4NF)

   Relasi disebut sebagi *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 menggunkan sekumpulan atribut pada kuncinya.

II.8. SQL Server 2008

SQL (Structured Query Language) adalah sebuah bahasa yang mengakses data dalam basis data relasional. Bahasa ini secara bahasa standar yang digunakan dalam manajemen basis data relashampir semua server basis data yang ada mendukung bahasa manajemen datanya. *SQL* terdiri dari dua bahasa, yaitu *Data Definition Language Manipulation Language (DML)*. Implementasi DDL dan DML sistem manajemen basis data *(SMBD)*, namun secara umum

implemen bahasa ini memiliki bentuk standar yang ditetapkan oleh *ANSI*. (Jimmy Setiawan, Vol. 6, No.2, 2011)

II.9. Bahasa Pemrograman *VB.NET*

Bahasa Pemrograman *VB.NET Microsoft* Visual Basic (sering disingkat sebagai VB saja) merupakan sebuah bahasa pemrograman yang bersifat event driven dan menawarkan *Integrated Development Environment (IDE) visual* untuk membuat program aplikasi berbasis sistem operasi *Microsoft Windows* dengan menggunakan model pemrograman *Common Object Model (COM). Visual Basic* merupakan turunan bahasa *basic* dan menawarkan pengembangan aplikasi komputer berbasis grafik dengan cepat, akses ke basis data menggunakan *Data Access Objects (DAO), Remote Data Objects (RDO),* atau *ActiveX Data Object (ADO),* serta menawarkan pembuatan kontrol *ActiveX* dan objek *ActiveX*.

Visual Basic merupakan turunan bahasa *basic* dan menawarkan pengembangan aplikasi komputer berbasis grafik dengan cepat, akses ke basis data menggunakan *Data Access Objects (DAO), Remote Data Objects (RDO),* atau *ActiveX Data Object (ADO),* serta menawarkan pembuatan kontrol *ActiveX* dan objek *ActiveX*. Beberapa bahasa skrip seperti *Visual Basic for Applications (VBA)* dan *Visual Basic Scripting Edition (VBScript),* mirip seperti halnya Visual Basic, tetapi cara kerjanya yang berbeda. Para *programmer* dapat membangun aplikasi dengan menggunakan komponen-komponen yang disediakan oleh *Microsoft Visual Basic* Program-program

yang ditulis dengan *Visual Basic* juga dapat menggunakan *Windows API*, tapi membutuhkan deklarasi fungsi eksternal tambahan. (Jimmy Setiawan, Vol. 6, No.2, 2011)

II.10. UML (*Unified Modelling Language*)

UML singkatan dari *Unified Modelling Language* yang berarti bahasa permodelan standar. UML diaplikasikan untuk maksud tertentu, biasanya antara lain untuk :

1. Merancang perangkat lunak
2. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasi sistem yang ada, proses-proses dan organisasinya.

Sejauh ini para pakar merasa lebih mudah dalam menganalisa dan mendesain atau memodelkan suatu sistem karena UML memiliki seperangkat aturan dan notasi dalam bentuk grafis yang cukup spesifik.

Komponen atau notasi UML diturunkan dari 3 (tiga) notasi yang telah ada sebelumnya yaitu Grady Booch, OOD (*Object-Oriented Design*), Jim Rumbaugh, OMT (*Object Modelling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*).

Pada UML ada beberapa jenis diagram antara lain yaitu :

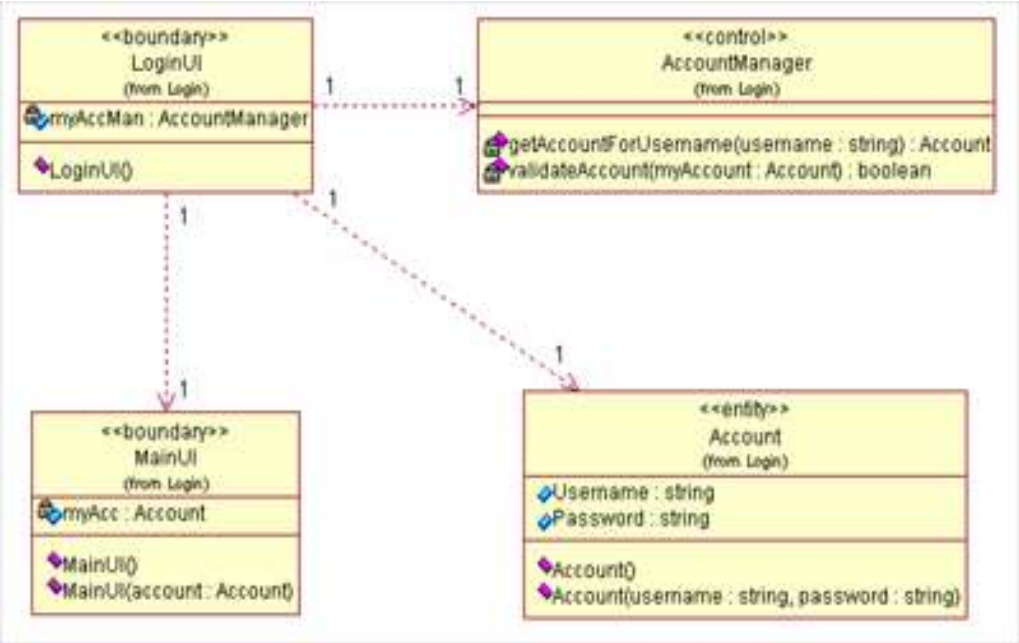
1. Struktur Diagram

Menggambarkan elemen dari spesifikasi dimulai dengan kelas, obyek, dan hubungan mereka, dan beralih ke dokumen arsitektur logis dari suatu sistem. Struktur diagram dalam UML terdiri atas (Haviluddin ; 2011 : 3) :

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 :

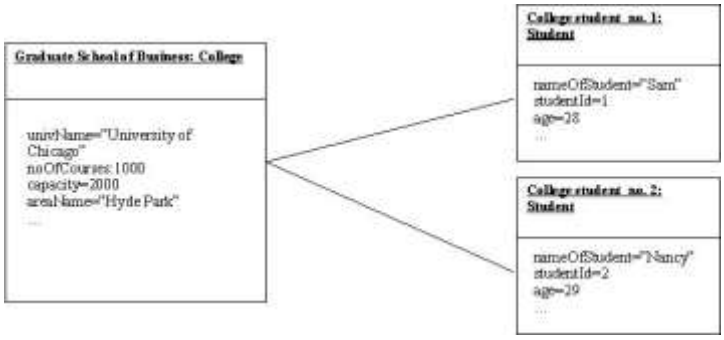
   1) Nama (dan *stereotype*)
   2) Atribut
   3) Metoda



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

b. *Object* diagram

Object diagram menggambarkan kejelasan kelas dan warisan dan kadang-kadang diambil ketika merencanakan kelas, atau untuk membantu pemangku kepentingan non-program yang mungkin menemukan diagram kelas terlalu abstrak. Berikut notasi *object diagram.*



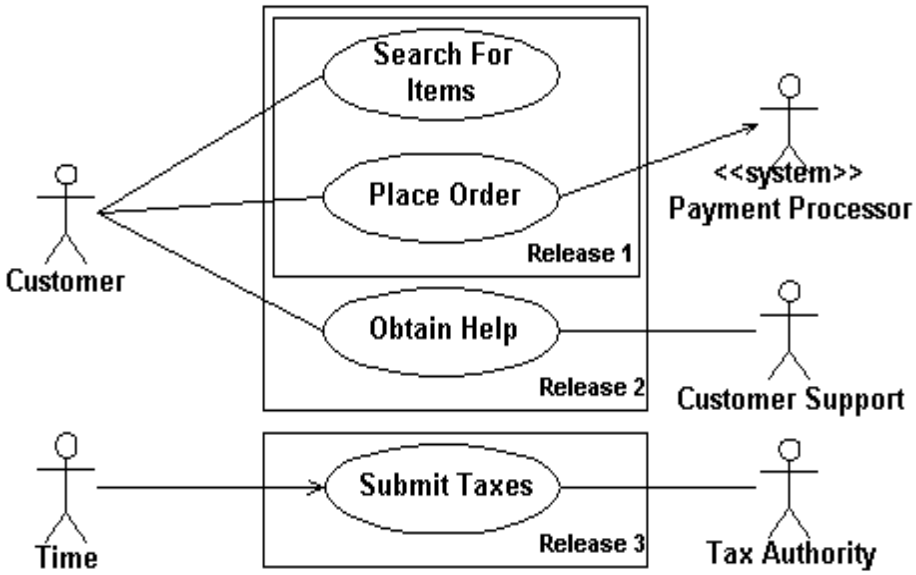
Gambar II.2. Notasi *Object Diagram*
Sumber : (Haviluddin ; 2011 : 3)

c. *Use case* diagram

Diagram yang menggambarkan *actor*, *use case* dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. Sebuah *use case* digambarkan sebagai elips horizontal dalam suatu diagram UML *use case*. *Use Case* memiliki dua istilah :

1). *System use case*; interaksi dengan sistem.

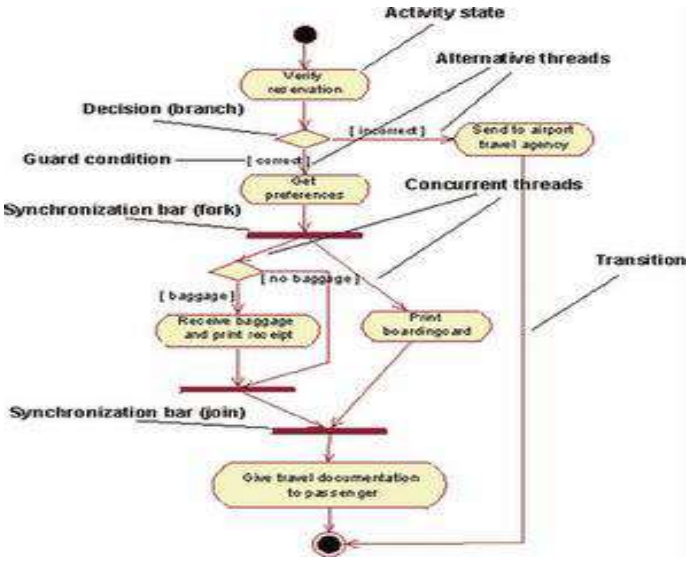
2). *Business use case;* interaksi bisnis dengan konsumen atau kejadian nyata



Gambar II.3. Notasi *Use Case Diagram*
Sumber : (Haviluddin ; 2011 : 4)

d. *Activity* diagram

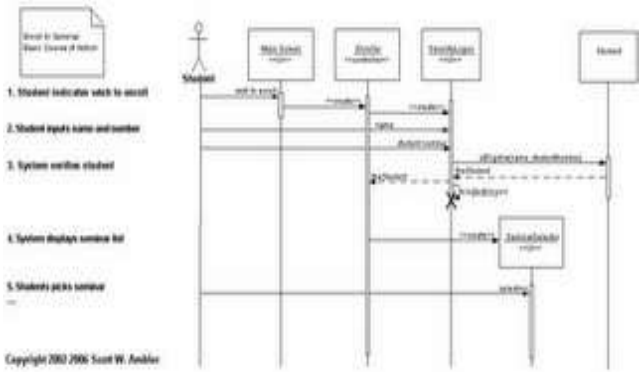
Menggambarkan aktifitas-aktifitas, objek, *state*, transisi *state* dan *event*. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas



Gambar II.4. Notasi *Activity Diagram*
Sumber : (Haviluddin ; 2011 : 4)

e. *Sequence* diagram

Sequence diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *sequence* diagram adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case* diagram.



Gambar II.5. Notasi *Sequence Diagram*
Sumber : (Haviluddin ; 2011 : 5)