

BAB II

LANDASAN TEORI

II.1. Teori Sistem

Menurut Kusrini (2010:5), Kata Sistem mempunyai beberapa pengertian, tergantung dari sudut mana kata tersebut didefinisikan. Secara garis besar ada dua pendekatan yang dilakukan yaitu :

1. Pendekatan sistem yang lebih menekankan pada elemen-elemen atau kelompoknya, yang didalam hal ini system ini didefinisikan sebagai suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu aturan tertentu.
2. Pendekatan sistem sebagai jaringan kerja dari prosedur, yang lebih menekankan urutan operasi di dalam sistem. Prosedur didefinisikan sebagai urutan operasi kerja (tulis-menulis), yang biasanya melibatkan beberapa orang didalam satu atau lebih departemen yang diterapkan untuk menjamin penanganan yang seragam dari transaksi bisnis yang terjadi.

Pendekatan sistem yang lebih menekankan pada elemen-elemen atau komponennya mendefinisikan sistem sebagai sekumpulan elemen-elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. Dengan demikian didalam suatu sistem, komponen-komponen ini tidak dapat berdiri sendiri, tetapi sebaliknya, saling berhubungan hingga berbentuk suatu kesatuan hingga tujuan sistem dapat tercapai.

II.1.1. Karakteristik sistem

Menurut Kusri (2010:6), Sistem mempunyai beberapa karakteristik atau sifat-sifat tertentu antara lain :

1. Komponen sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja sama untuk membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian dari sistem. Setiap sistem tidak peduli berapapun kecilnya, selalu mengandung komponen-komponen atau subsistem.

2. Batasan sistem (*Boundary*)

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya.

3. Lingkungan luar sistem (*Environment*)

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

4. Penghubung sistem (*Interface*)

Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui perhubungan ini memungkinkan sumber-sumber daya mengalir dari subsistem yang lainnya.

5. Masukan sistem (*Input*)

Merupakan 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 dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

7. Pengolah sistem (*Process*)

Suatu sistem yang dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran.

8. Sasaran sistem (*Objective*)

Suatu sistem pasti mempunyai tujuan atau sasaran. Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.

II.1.2. Klasifikasi sistem

Menurut Kusri (2010:7), Sistem dapat diklasifikasikan dari beberapa sudut pandangan, diantaranya sebagai berikut :

1. Sistem abstrak dan sistem fisik

Sistem abstrak adalah sistem yang berisi gagasan atau konsep. Misalnya, sistem teologi yang berisi gagasan tentang hubungan manusia dan Tuhan. Sistem fisik merupakan sistem yang secara fisik dapat dilihat. Misalnya sistem komputer, sistem sekolah, sistem akuntansi, dan sistem transportasi.

2. Sistem alamiah dan sistem buatan manusia

Sistem alamiah adalah sistem yang terjadi karena alam (tidak dibuat manusia). Misalnya, sistem tata surya. Sistem buatan manusia adalah sistem yang dibuat oleh manusia. Misalnya, sistem komputer dan sistem mobil.

3. Sistem tertentu dan sistem tak tentu

Sistem tertentu adalah sistem yang operasinya dapat diprediksi secara tepat. Misalnya, sistem komputer. Sistem tak tentu adalah sistem yang tak dapat diramal dengan pasti karena mengandung unsur probabilitas. Misalnya, sistem arisan dan sistem persediaan.

4. Sistem tertutup dan sistem terbuka

Sistem tertutup adalah sistem yang tidak bertukar materi, informasi, atau energi dengan lingkungan. Misalnya, reaksi kimia dalam tabung terisolasi. Sistem terbuka adalah sistem yang berhubungan dengan lingkungan dan dipengaruhi oleh lingkungan.

II.2. Sistem Pendukung Keputusan

Sistem pendukung keputusan (Inggris: *decision support systems* disingkat DSS) adalah bagian dari sistem informasi berbasis komputer (termasuk sistem berbasis pengetahuan (manajemen pengetahuan) yang dipakai untuk mendukung pengambilan keputusan dalam suatu organisasi atau perusahaan.

II.2.1. Pengertian Sistem Pendukung Keputusan

Menurut Dicky Nofriansyah, S.Kom, M.Kom (2014:1), Sistem pendukung keputusan (SPK) dibangun untuk mendukung solusi atas suatu masalah atau untuk suatu peluang. Aplikasi Sistem pendukung keputusan (SPK) digunakan untuk pengambilan keputusan. Aplikasi Sistem pendukung keputusan (SPK) menggunakan *CBIS (Computer Based Information System)* yang fleksibel, interaktif dan dapat diadaptasi yang dikembangkan untuk mendukung solusi masalah manajemen spesifik yang tidak terstruktur.

Menurut Bonczek dkk, (1980) dalam buku “*Decision Support System and intelligent system* (Turban 2005:137) mendefinisikan Sistem pendukung keputusan (SPK) sebagai sistem berbasis computer yang terdiri dari tiga komponen yang saling berinteraksi, sistem bahasa (mekanisme untuk memberikan komunikasi antara pengguna dan komponen system pendukung keputusan lain), sistem pengetahuan (repository pengetahuan domain masalah yang ada pada Sistem pendukung keputusan (SPK) atau sebagai data atau sebagai prosedur), dan sistem pemrosesan masalah (hubungan antara dua komponen lainnya, terdiri dari satu atau lebih kapabilitas manipulasi masalah umum yang diperlukan untuk pengambilan keputusan).

II.2.2. Karakteristik Sistem Pendukung Keputusan

Menurut Dicky Nofriansyah, S.Kom, M.Kom (2014:2), Karakteristik Sistem pendukung keputusan (SPK) yaitu :

1. Mendukung proses pengambilan keputusan suatu organisasi atau perusahaan
2. Adanya interface manusia/mesin dimana manusia (*user*) tetap memegang kontrol proses pengambilan keputusan.
3. Mendukung pengambilan keputusan untuk membahas masalah terstruktur, semi terstruktur serta mendukung beberapa keputusan yang saling berinteraksi.
4. Memiliki kapasitas dialog untuk memperoleh informasi sesuai dengan kebutuhan.
5. Memiliki sub sistem yang terintegrasi sedemikian rupa sehingga dapat berfungsi sebagai kesatuan sistem.

II.2.3. Ciri-Ciri Sistem Pendukung Keputusan

Menurut Dicky Nofriansyah, S.Kom, M.Kom (2014:2), Kriteria atau ciri-ciri sistem pendukung keputusan adalah sebagai berikut :

1. Banyak pilihan/alternatif.
2. Ada kendala atau surat.
3. Mengikuti suatu pola atau model tingkah laku, baik yang terstruktur maupun tidak terstruktur.
4. Banyak input/variabel.
5. Ada faktor resiko, dibutuhkan kecepatan, ketepatan dan keakuratan .

II.2.4. Fase Dalam Sistem Pendukung Keputusan

Menurut Dicky Nofriansyah, S.Kom, M.Kom (2014:2), Tiga fase dalam pengambilan keputusan yaitu :

1. Intelligence

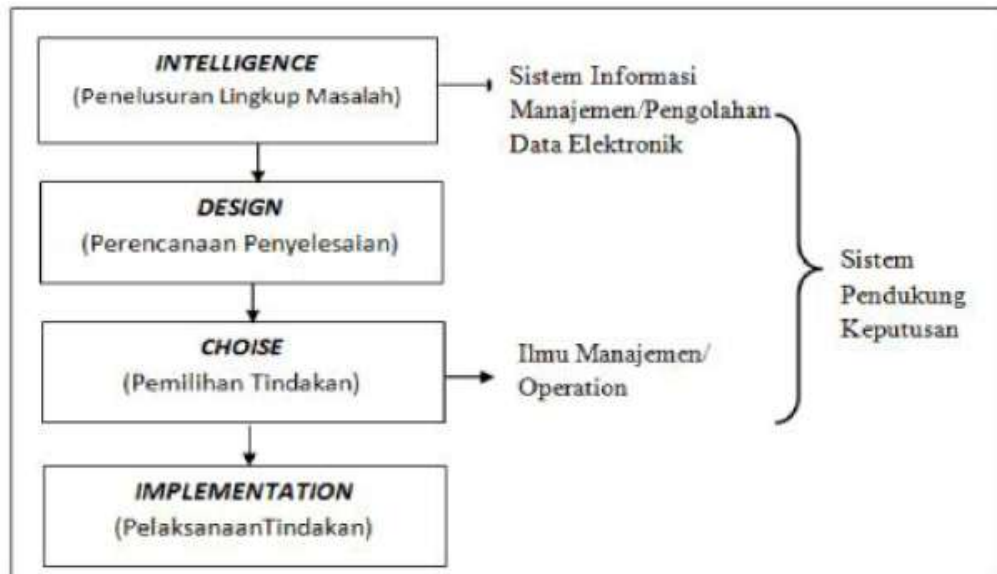
Tahap ini merupakan proses penelusuran dan pendeteksian dari ruang lingkup problematika secara proses pengenalan masalah. Data masukan diperoleh dan diuji dalam rangka mengidentifikasi masalah.

2. Design

Tahap ini merupakan proses menemukan, mengembangkan dan menganalisis alternatif tindakan yang bisa dilakukan. Tahap melakukan pengujian kelayakan solusi.

3. Choice

Pada tahap ini dilakukan proses pemilihan diantara berbagai alternatif tindakan yang mungkin dijalankan. Hasil pemilihan tersebut kemudian diimplementasikan kedalam proses pengambilan keputusan.



Gambar II.1. Fase Proses Pengambilan Keputusan
(Sumber : Dicky Nofriansyah ; 2014)

II.2.5. Komponen Sistem Pendukung Keputusan

Menurut Dicky Nofriansyah, S.Kom, M.Kom (2014:3), Secara garis besar sistem pendukung keputusan dibangun oleh tiga komponen utama yaitu :

1. Sub Sistem Data (*Database*)

Sub sistem data merupakan komponen sistem pendukung keputusan yang berguna sebagai penyedia data bagi sistem. Data tersebut disimpan untuk diorganisasikan dalam sebuah basis data yang diorganisasikan oleh suatu sistem yang disebut dengan Sistem Manajemen Sistem Basis Data (*Database Management System*)

2. Sub sistem Model

Model adalah suatu tiruan dari alam nyata. Kendala yang sering dihadapi dalam merancang model adalah bawah model yang dirancang tidak mampu mencerminkan seluruh variabel alam nyata, sehingga keputusan yang diambil tidak sesuai dengan kebutuhan. Oleh karena itu, dalam menyimpan berbagai

model harus diperhatikan dan harus dijaga fleksibilitasnya. Hal lain yang harus diperhatikan adalah pada setiap model yang disimpan hendaknya ditambahkan rincian, keterangan dan penjelasan yang komprehensif mengenai model yang dibuat.

3. Sub sistem dialog (*User System Interface*)

Sub sistem dialog adalah fasilitas yang mampu mengintegrasikan sistem yang terpasang dengan pengguna secara interaktif, yang dikenal dengan sub sistem dialog. Melalui sub sistem dialog sistem diimplementasikan sehingga pengguna dapat berkomunikasi dengan sistem yang dibuat.

II.2.6. Tujuan Sistem Pendukung Keputusan

Menurut Dicky Nofriansyah, S.Kom, M.Kom (2014:4), Tujuan dari system pendukung keputusan adalah sebaagi berikut :

1. Membantu dalam pengambilan keputusan atas masalah yang terstruktur
2. Memberikan dukungan atas pertimbangan manager dan bukannya dimaksudkan untuk menggantikan fungsi manager.
3. Meningkatkan efektifitas keputusan yang diambil lebih dari perbaikan efesiensinya.
4. Kecepatan komputasi komputer memungkinkan para pengambil keputusan untuk banyak melakukan komputasi secara cepat dengan biaya yang sangat rendah.
5. Peningkatan produktifitas membangun suatu kelompok pengambil keputusan, terutama para pakar bisa sangat mahal. Sistem pendukung keputusan komputerisasi mengurangi ukuran kelompok dan memungkinkan para anggotanya untuk berada diberbagai lokasi yang berbeda-beda (menghemat

biaya perjalanan). Selain itu produktifitas staf pendukung (misalnya analisis keuangan dan hokum) bisa ditingkatkan. Produktivitas juga ditingkatkan menggunakan peralatan optimalisasi yang menjalankan sebuah bisnis.

II.3. Metode *Certainty Factor* (CF)

Menurut Kusrini (2008:15), Faktor Kepastian (*Certainty Factor*) diperkenalkan oleh Shortlife Buchanan dalam pembuatan MYCIN. *Certainty Factor* merupakan nilai parameter klinis yang diberikan MYCIN untuk menunjukkan besarnya kepercayaan. *Certainty Factor* menurut Giarrantano dan Riley didefenisikan sebagai berikut:

$$CF (Pk,G) = MB (Pk,G) - MD (Pk,G)$$

Dengan :

$$MB(Pk,G) = \begin{cases} 1 & , P(Pk) = 1 \\ \frac{\max [P(Pk | G), P(Pk)] - P(Pk)}{\max [1,0] - P(Pk)} & , \text{yang lain} \end{cases}$$

$$MD(Pk,G) = \begin{cases} 1 & , P(Pk) = 0 \\ \frac{\max [P(Pk | G), P(Pk)] - P(Pk)}{\max [1,0] - P(Pk)} & , \text{yang lain} \end{cases}$$

Dimana :

- CF (Pk,G) tingkat kepastian penyakit Pk, berdasarkan gejala G.
- MB (Pk,G) Pengukuran tingkat kepastian Pk, karena adanya gejala G.

- MD (Pk,G) pengukuran tingkat ketidakpercayaan penyakit Pk, berdasarkan gejala G.
- $P(Pk|G)$ probabilitas penyakit Pk dengan diketahui gejala G telah terjadi.
- $P(Pk)$ probabilitas penyakit Pk.

Apabila terdapat gejala-gejala yang berbeda menyebabkan penyakit yang sama maka misalnya gejala $G(G_1, G_2, \dots, G_n)$ menyebabkan penyakit Pk, maka terdapat nilai $E(E_1, E_2, \dots, E_n)$ juga menyebabkan penyakit Pk, maka terdapat nilai $CF_1(Pk,G)$ dan $CF_2(Pk,E)$. Tingkat kepastian yang dihasilkan oleh sistem dalam menentukan diagnosa adalah CF kombinasi seperti yang dirumuskan pada persamaan :

$$CF_{Kombinasi}(CF_1, CF_2) = \begin{cases} CF_1 + CF_2(1-CF_1), & \text{Keduanya} > 0 \\ \frac{CF_1 + CF_2}{1-\min(|CF_1|, |CF_2|)}, & \text{Salah Satu} < 0 \\ CF_1 + CF_2(1-CF_1) & \text{Keduanya} < 0 \end{cases}$$

Dengan menggali dari hasil wawancara dengan pakar. Nilai CF (Rule) didapat dari interpretasi 'term' dari pakar menjadi nilai CF tertentu.

Tabel II.1. CF Value Interpretation

Certain Term	MD/MB
Tidak ada	0- 0.2
Mungkin	0.4
Kemungkinan Besar	0.6
Hampir Pasti	0.8
Pasti	1.0

(Sumber : Kusri ; 2008)

II.4. Pengenalan UML

Menurut Julius Hermawan (2010:7), UML (*Unified Modeling Language*) adalah bahasa standard yang digunakan untuk menjelaskan dan memvisualisasikan artifak dan proses analisis dan desain berorientasi objek. UML Menyediakan standar pada notasi dan diagram yang bias digunakan untuk memodelkan suatu sistem. UML dikembangkan oleh tiga pendekar “berorientasi objek” yaitu Gradi Booch, Jim Rumbaugh dan Ivar Jacobson. UML menjadi bahasa yang bias digunakan untuk berkomunikasi dalam prespektif objek antara *user* dengan *developer*, antara *developer* analisis dengan *developer* desain dan antara *developer* desain dengan *developer* pemrograman.

UML memungkin *developer* melakukan pemodelan secara visual, yaitu penekanan pada penggambaran, bukan didominasi oleh narasi. Pemodelan visual membantu untuk menangkap struktur dan kelakuan dari si objek, mempermudah penggambaran interaksi antara elemen dalam sistem dan mempertahankan konsistensi antara desain dan implementasi dalam bahasa pemrograman.

Namun karena UML hanya merupakan bahasa pemodelan maka UML bukanlah rujukan bagaimana melakukan analisis dan desain berorientasi objek. Untuk mengetahui bagaimana melakukan analisis dan desain berorientasi objek secara baik, sudah terdapat beberapa metodologi yang bias diikuti.

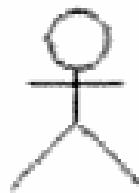
II.4.1. Notasi dan Artifak dalam UML

Menurut Julius Hermawan (2010:13), UML menyediakan beberapa notasi dan artifak standard yang bias digunakan sebagai alat komunikasi bagi para proses analisis dan desain. Artifak di dalam UML didefenisikan sebagai informasi

dalam berbagai bentuk yang digunakan atau dihasilkan dalam proses pengembangan perangkat lunak.

1. Aktor

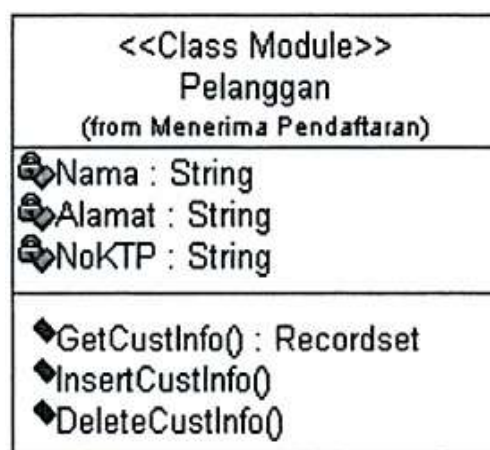
Aktor adalah segala sesuatu yang berinteraksi dengan sistem aplikasi komputer. Jadi aktor ini bisa berupa orang, perangkat keras atau juga objek lain dalam sistem yang sama. Biasanya yang dilakukan oleh Aktor adalah memberikan informasi pada sistem dan\atau memerintahkan sistem untuk melakukan sesuatu.



Gambar II.2. Notasi Aktor
(Sumber : Andi ; 2010)

2. Class

Class merupakan pembentuk utama dari sistem berorientasi objek karena *class* menunjukkan kumpulan objek yang memiliki atribut dan operasi yang sama. *Class* digunakan untuk mengimplementasikan *interface*



Gambar II.3. Notasi Class
(Sumber : Andi ; 2010)

Class digunakan untuk mengabstraksikan elemen-elemen dari sistem yang dibangun. *Class* bisa untuk direpresentasikan baik perangkat lunak maupun perangkat keras, baik konsep maupun benda nyata. Atribut digunakan untuk menyimpan informasi. Nama atribut menggunakan kata benda yang bisa dengan jelas direpresentasikan informasi yang disimpan di dalamnya. Operasi menunjukkan sesuatu yang bisa dilakukan oleh objek, dan menggunakan kata kerja.

3. *Interface*

Interface merupakan kumpulan informasi tanpa implementasi dari suatu *class*. Implementasi operasi dari suatu *interface* dijabarkan oleh operasi didalam *class*. Oleh karena itu keberadaan *interface* selalu disertai oleh *class* yang mengimplementasikan operasinya. *Interface* ini merupakan salah satu cara mewujudkan prinsip enkapsulasi dalam objek.



Gambar II.4. Notasi Interface
(Sumber : Andi ; 2010)

4. *Use Case*

Use case menjelaskan urutan kegiatan yang dilakukan Aktor dan system untuk mencapai tujuan tertentu. Walaupun menjelaskan kegiatan namun *use case* hanya menjelaskan apa yang dilakukan oleh actor dan system, bukan bagaimana system melakukan kegiatan tersebut.



Gambar II.5. Notasi Use Case
(Sumber : Andi ; 2010)

Didalam *use case* terdapat teks untuk menjelaskan urutan kegiatan yang disebut *use case specification*. *use case specification* terdiri dari :

a. Nama *Use Case*

Mencantumkan nama dari use case yang bersangkutan. Sebaiknya diawali dengan kata kerja untuk menunjukkan suatu aktivitas

b. Deskripsi singkat

Menjelaskan secara singkat dalam 1 atau 2 kalimat tentang tujuan dari use case ini.

c. Aliran Normal (*Basic Flow*)

Ini adalah jantung dari *use case*. Menjelaskan interaksi antara actor dan sistem dalam kondisi normal, yaitu segala sesuatu berjalan dengan lancar tiada halangan atau hambatan dalam mencapai tujuan dalam *use case*.

d. Aliran Alternatif (*Alternative Flow*)

Merupakan pelengkap dari *basic flow* tidak ada yang sempurna dalam setiap kali *use case* berlangsung. Di dalam *Alternative Flow* ini dijelaskan dalam apa yang terjadi bila suatu halangan atau hambatan terjadi sewaktu *use case* berlangsung. Ini terutama berhubungan dengan *error* yang mungkin terjadi terutama karena sistem kekurangan data untuk diolah.

e. *Special Requirement*

Berisi kebutuhan lain yang belum tercakup dalam kebutuhan normal dan alternatif. Biasanya secara tegas dibedakan bahwa *basic flow* dan *alternate flow* menangani kebutuhan fungsional dari *use case* sementara *Special Requirement* yang tidak berhubungan dengan kebutuhan fungsional, misalnya

kecepatan transaksi maksimum artinya berapa cepat dan berapa lama, kapasitas akses yaitu jumlah *user* yang akan mengakses dalam waktu bersamaan.

c. *Pre-Condition*

Menjelaskan persyaratan yang harus dipenuhi sebelum *use case* bisa dimulai.

d. *Post-Condition*

Menjelaskan kondisi yang berubah atau terjadi saat *use case* selesai dieksekusi.

5. *Interaction*

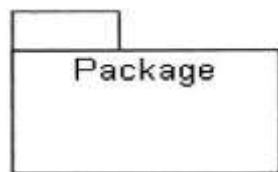
Digunakan untuk menunjukkan baik aliran pesan maupun informasi antara objek maupun antara hubungan objek. Biasanya *Interaction* dilengkapi juga dengan teks bernama *operation signature* yang tersusun dari mana operasi, parameter yang dikirim dan *type* parameter yang dikembalikan.



Gambar II.6. Notasi *Interaction*
(Sumber : Andi ; 2010)

6. *Package*

Package adalah kontainer atau wadah konseptual yang digunakan untuk mengelompokkan elemen-elemen dari sistem yang sedang dibangun, sehingga bisa dibuat model menjadi lebih sederhana. Tujuannya adalah untuk mempermudah pengelihatian dari model yang sedang dibangun.



Gambar II.7. Notasi *Package*
(Sumber : Andi ; 2010)

7. *Note*

Note digunakan untuk memberikan keterangan dan komentar tambahan dari suatu elemen sehingga bisa langsung terlampir dalam model. *Note* ini bisa ditempelkan ke semua elemen notasi yang lain.

8. *Dependency*

Dependency merupakan relasi yang menunjukkan bahwa perubahan pada salah satu elemen member pengaruh pada elemen lain.



Gambar II.8. Notasi *Dependency*
(Sumber : Andi ; 2010)

II.5. Pengertian Basis Data (*Database*)

Basis data merupakan kumpulan dari data-data yang saling terkait dan saling berhubungan satu dengan yang lainnya. Basis data adalah kumpulankumpulan *file* yang saling berkaitan.

Menurut Kusri (2010, p2), pengertian Basis Data adalah kumpulan data yang saling berelasi. Data sendiri merupakan fakta mengenai obyek, orang, dan lain-lain. Data dinyatakan dengan nilai (angka, deretan karakter, atau simbol).

Basis data dapat didefinisikan dalam berbagai susut pandang seperti berikut:

1. Himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian rupa sehingga kelak dapat dimanfaatkan dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa tanpa pengulangan (*redundancy*) yang tidak perlu, untuk memenuhi kebutuhan.
3. Kumpulan *file*/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpan elektronik.

II.5.1. Tujuan Basis Data

Menurut Kusri (2010, p2), Basis data bertujuan untuk mengatur data sehingga diperoleh kemudahan, ketepatan dan kecepatan dalam pengambilan kembali. Untuk mencapai tujuan, syarat basisdata yang baik adalah sebagai berikut :

1. Tidak adanya redudansi dan inkonsistensi data

Redudansis terjadi jika suatu informasi disimpan dibeberapa tempat. Misalnya ada data mahasiswa yang memuat nim, nama, alamat dan atribut lainnya, sementara kita punya data lain tentang data KHS mahasiswa yang isinya terdapat NIM, nama, mata kuliah dan nilai. Pada kedua data tersebut kita temukan atribut nama.

2. Kesulitan pengaksesan data

Basis data memiliki fasilitas untuk melakukan pencarian informasi dengan menggunakan *query* ataupun dari *tools* yang melibatkan tabelnya. Dengan fasilitas ini, bisa segera langsung melihat data dari *software* DBMnnya.

3. Multiple *user*

Basis data memungkinkan penggunaan data secara bersama-sama oleh banyak pengguna pada saat yang bersamaan atau pada saat yang berbeda. Dengan meletakkan basis data pada bagian server yang bisa diakses dari banyak client, sudah menyediakan akses kesemua pengguna dari komputer *client* ke sumber informasi yaitu basis data.

II.5.2. Manfaat dan Kelebihan Basis Data

Menurut Kusriani (2010, p5), Banyak manfaat yang diperoleh dengan menggunakan basis data, Manfaat/Kelebihan Basis Data dan kelebihan basis data diantaranya adalah :

1. Kecepatan dan kemudahan

Dengan menggunakan basis data pengambilan informasi dapat dilakukan dengan cepat dan mudah. Basis data memiliki kemampuan dalam mengelompokkan, mengurutkan bahkan perhitungan dengan matematika. Dengan perancangan yang benar maka penyajian informasi dapat dilakukan dengan cepat dan mudah.

2. Kebersamaan pemakai (*sharability*)

Sebuah basis data dapat digunakan oleh banyak user dan banyak aplikasi. Untuk data yang diperlukan oleh banyak bagian/orang, tidak perlu dilakukan pencacatan dimasing-masing bagian/orang, tetapi cukup dengan satu basis data untuk dipakai bersama.

3. Pemusatan kontrol data

Karena cukup satu basis data untuk banyak keperluan, pengontrolan terhadap data juga cukup dilakukan disatu tempat saja.

4. Efisiensi ruang penyimpanan

Dengan pemakaian bersama, tidak perlu menyediakan tempat penyimpanan diberbagai tempat tetapi cukup satu saja, sehingga ini dapat menghemat ruang penyimpanan yang dimiliki oleh sebuah organisasi.

5. Keakuratan (*Accuracy*)

Penerapan secara tepat acuan tipe data, domain data, keunikan data, hubungan antar data, dan lain-lain, dapat menekan ketidakakuratan dalam pemasukan/penyimpanan data.

6. Ketersediaan (*Availability*)

Dengan basis data, semua data dapat dibackup, memilah-milah data mana yang masih diperlukan yang perlu disimpan ke tempat lain. Hal ini mengingat pertumbuhan transaksi sebuah organisasi dari lain waktu ke waktu membutuhkan penyimpanan yang semakin besar.

7. Keamanan (*Security*)

Kebanyakan DBMS dilengkapi dengan fasilitas manajemen pengguna. Pengguna diberi hak akses yang berbeda-beda sesuai dengan kepentingan dan posisinya. Basis data bisa diberikan password untuk membatasi orang yang diaksesnya.

8. Kemudahan dalam pembuatan program aplikasi baru

Penggunaan basis data merupakan bagian dari perkembangan teknologi. Dengan adanya basis data pembuatan aplikasi bisa memanfaatkan kemampuan dari DBMS. Sehingga membuat aplikasi tidak perlu mengurus penyimpanan data, tetapi cukup mengatur interface untuk pengguna.

9. Pemakaian secara langsung

Basis data memiliki fasilitas yang lengkap untuk melihat datanya secara langsung dengan tools yang disediakan oleh DBMS.

10. Kebebasan data

Perubahan dapat dilakukan pada level DBMS tanpa harus membongkar kembali program aplikasinya.

11. User View

Basis data menyediakan pandangan yang berbeda-beda untuk tiap-tiap pengguna.

II.5.3. Operasi Dasar Database

Menurut Kusrini (2010, p9), Beberapa operasi dasar basis data yaitu :

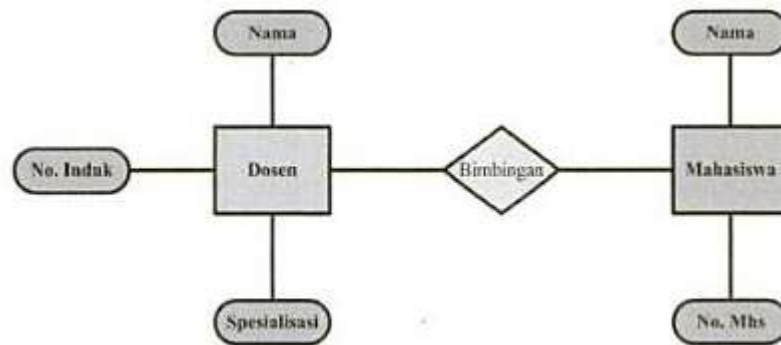
1. Pembuatan basis data
2. Penghapusan basis data
3. Pembuatan file/tabel
4. Penghapusan file/tabel
5. Pengubahan tabel
6. Penambahan/pengisian
7. Pengambilan data
8. Penghapusan data

II.5.4. Pemodelan Basis Data

Menurut Samiaji Sarosa (2010:4), Model diperlukan untuk mendapatkan penyederhanaan dari kenyataan dan memungkinkan desainer program aplikasi bereksperimen dengan berbagai macam *variable* sebelum diaplikasikan ke *system*

yang berjalan. Untuk merancang suatu aplikasi basis data alat yang biasa digunakan adalah *Entity Relationship Diagram (ERD)*. ERD didasarkan dari artikel yang dipublikasikan oleh Peter Phin Shan Chen.

Ada beberapa case tool menamakan notasi ERD yang digunakan sebagai chen ERD. *Entity Relationship Model* adalah abstraksi konseptual yang mewakili struktur dari suatu basis data.

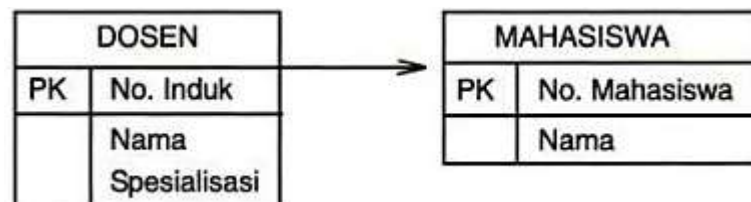


Gambar II.9. Diagram Dengan Notasi Chen ERD
(Sumber : Grasindo ; 2010)

Dalam perkembangannya banyak diciptakan notasi ERD yang berbeda-beda seperti terlihat pada gambar 2.11.



Gambar II.10. Diagram Dengan Notasi Crows Foot
(Sumber : Grasindo ; 2010)



Gambar II.11. Diagram Dengan Notasi Relational
(Sumber : Grasindo ; 2010)

II.5.4. Normalisasi

Menurut Samiaji Sarosa (2010:5), Normalisasi adalah teknik yang dirancang untuk merancang tabel basis data relasional untuk meminimalkan duplikasi data dan menghindari basis data tersebut anomali. Suatu basis data dikatakan tidak normal jika terjadi 3 (tiga) anomali berikut :

1. *Insertion Anomaly*

Anomali yang terjadi jika ada data yang tidak bisa disisipkan kedalam table.

2. *Update/Modification anomaly*

Anomali yang terjadi jika ada perubahan pada suatu item data maka harus mengubah lebih dari satu baris data.

Langkah-langkah normalisasi sampai pada bentuk 3NF adalah sebagai berikut :

1. *First Normal Form (1NF)*

Untuk menjadi 1NF suatu table harus memenuhi dua syarat. Syarat pertama tidak ada kelompok data atau *field* yang berulang. Syarat kedua harus ada *primary key (PK)* atau kunci unik, atau kunci yang membedakan satu baris dengan baris yang lain dalam satu table. Pada dasarnya sebuah table selamat tidak ada kolom yang sama merupakan bentuk table dengan 1NF.

2. *Second Normal Form (2NF)*

Untuk menjadi 2NF suatu table harus berada dalam kondisi 1NF dan tidak memiliki *partial dependencies*. *Partial dependencies* adalah suatu kondisi jika atribut non kunci (Non PK) tergantung sebagian tetapi bukan seluruhnya pada PK.

3. *Third Normal Form (3NF)*

Untuk menjadi 3NF suatu table harus berada dalam kondisi 2NF dan tidak memiliki *transitive dependencies*. *Transitive dependencies* adalah suatu kondisi

dengan adanya ketergantungan fungsional antara 2 atau lebih atribut non kunci (Non PK).

II.6. Java NetBeans

Menurut jurnal pemrograman dekstop java Badiyanto S.Kom. M.Kom (1), NetBeans adalah suatu tool untuk membuat program dengan menggunakan bahasa pemrograman java berbasis grafis. Cara membuat program dengan merancang dengan tampilan menggunakan komponen *visual* dan proses diletakan pada *event driven*.

NetBeans menyediakan sekumpulan perangkat lunak modular yang disebut modul yang dipakai untuk membangun suatu aplikasi. Sebuah modul adalah merupakan arsip java (*Java Archive*) yang memuat kelas-kelas Java yang berinteraksi dengan NetBeans Open API. Kemudian rancangan tampilan yang dibuat menggunakan NetBeans programmnya secara otomatis akan digenerate menjadi kode.

II.7. SQL Server 2008 Express Edition

Menurut Wahana Komputer (2010:2), SQL Server 2008 *Express Edition* sebuah terobosan baru dalam bidang database, SQL Server adalah sebuah DBMS (*Database Management System*) yang dibuat oleh Microsoft untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahulunya seperti IBM dan Oracle. SQL Server 2008 *Express Edition* dibuat pada saat kemajuan dalam bidang hardware semakin pesat. Oleh karena itu sudah dapat dipastikan bahwa SQL Server 2008 *Express Edition* membawa terobosan dalam bidang pengolahan dan penyimpanan data.

II.7.1. Kebutuhan Hardware

Adapun *hardware* yang diperlukan untuk instalasi SQL Server 2008 *Express Edition* minimal adalah sebagai berikut :

1. Prosesor minimal 1 GHz
2. Memori minimal 512 MB
3. Sistem Operasi Windows

Biar dapat diinstal pada sistem komputer dengan memori 512 MB, tetapi disarankan menggunakan memori 1 GB. Sedangkan untuk jaringannya diperlukan adalah :

1. *Sharer Memory*
2. TCP/IP
3. *Named Pipes*
4. *Virtual Interface Adapter (VIA)*(Wahana Komputer, 2010:2).

II.7.2. Versi SQL Server 2008 Express Edition

Microsoft merilis SQL Server 2008 *Express Edition* dalam beberapa versi yang disesuaikan dengan segmen-segmen pasar yang dituju. Versi-versi tersebut adalah sebagai berikut :

1. Menurut cara pemrosesan data pada prosesor maka Microsoft mengelompokkan produk ini berdasarkan dua jenis yaitu :
 - a. Versi 32 Bit (x86), yang biasanya digunakan untuk komputer *single processor* (Pentium 4) atau lebih tepatnya prosesor 32 bit atau Windows XP.

- b. Versi 64 Bit (x64), yang biasanya digunakan oleh computer yang lebih sari satu processor (Misalnya *Core 2 duo*) dan system operasi 64 bit, Vista dan Windows 7.
2. Sedangkan secara keseluruhan terdapat versi-versi seperti berikut :
 - a. Versi *Compact* ini adalah versi “tipis” dari semu versi yang ada
 - b. Versi *Express* ini adalah versi “ringan”

II.7.3 Instalasi SQL Server 2008 Express Edition

Proses instalasi SQL Server 2008 *Express Edition* tidak sama dengan instalasi versi-versi sebelumnya. Proses SQL Server 2008 *Express Edition* agak panjang melalui beberapa tahapan. Tahapan yang dilakukan akan membawa beberapa pilihan yang akan diisi dalam *setting* sebuah *server database*. Berikut ini adalah pilihan-pilihan yang akan dijumpai dalam proses instalasi SQL Server 2008 *Express Edition*.

1. Tempat direktori utama dan penyimpanan file database

Direktori utama adalah direktori dimana semua file program akan ditempatkan dan file-file tersebut tidak akan berubah selama anda menjalankan SQL server. Direktori utama secara standard akan berada dalam direktori “C:\Program Files\Microsoft SQL Server”.

2. Penggunaan *Multiple instance*

Instance adalah sebuah turunan dari server database SQL Server. Karena sebuah tiruan maka sebuah *Instance* memiliki fungsi yang sama dengan database server aslinya. Arti sebenarnya *Instance SQL Server* adalah sebuah server database yang tidak men-*sharing* sistemnya dan database user dengan database server lainnya yang ada dalam komputer yang sama.

3. Jasa *Autentification User* (Menggunakan Windows atau mixed)

Autentification User diperlukan supaya server tidak dapat dipergunakan oleh orang yang tidak bertanggungjawab dan tidak berhak. Dalam SQL server ada dua *Autentification User* yang dapat digunakan yaitu :

- a. Mode Windows, Pada mode ini SQL Server akan melakukan autentifikasi dengan menggunakan level login pada system operasi.
- b. Mode Mixel atau campuran, mode ini menginjinkan *user* untuk masuk kedalam system SQL server dengan menggunakan *Account* yang dibuat di system operasi windows atau juga menggunakan *account* yang di *set up* pada SQL Server (Wahana Komputer, 2010:2)