

BAB II

TINJAUAN PUSTAKA

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 :

- a. Pendekatan system 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.
- b. Pendekatan system sebagai jaringan kerja dari prosedur, yang lebih menekankan urutan operasi didalam system. 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 system yang lebih menekankan pada elemen-elemen atau komponennya mendefinisikan system sebagai sekumpulan elemen-elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. Dengan demikian didalam suatu system, komponen-komponen ini tidak dapat berdiri sendiri, tetapi sebaliknya, saling berhubungan hingga berbentuk suatu kesatuan hingga tujuan system dapat tercapai.

II.1.1. Karakteristik sistem

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

a. 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 berupaya suatu subsistem atau bagian dari sistem. Setiap sistem tidak peduli berapapun kecilnya, selalu mengandung komponen-komponen atau subsistem.

b. Batasan sistem (*Boundary*)

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

c. 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

d. 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.

e. Masukan sistem (*Input*)

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

f. Keluaran sistem (*Output*)

Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

g. Pengolah sistem (*Process*)

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

h. 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 Kusrini (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 sediaan.

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 (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 “*Decission Support System and intelligent system*” (Turban 2005:137) mendefinisikan Sistem pendukung keputusan (SPK) sebagai system berbasis computer yang terdiri dari tiga komponen yang saling berinteraksi, system bahasa (mekanisme untuk memberikan komunikasi antara pengguna dan komponen system pendukung keputusan lain), system pengetahuan (repository pengetahuan domain masalah yang ada pada Sistem pendukung keputusan (SPK) atau sebagai data atau sebagai prosedur), dan system 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 (2014:2), Karakteristik Sistem pendukung keputusan (SPK) yaitu :

- a. Mendukung proses pengambilan keputusan suatu organisasi atau perusahaan
- b. Adanya interface manusia/mesin dimana manusia (*user*) tetap memegang kontrol proses pengambilan keputusan.
- c. Mendukung pengambilan keputusan untuk membahas masalah terstruktur, semi terstruktur serta mendukung beberapa keputusan yang saling berinteraksi.
- d. Memiliki kapasitas dialog untuk memperoleh informasi sesuai dengan kebutuhan.
- e. Memiliki subsistem yang terintegrasi sedemikian rupa sehingga dapat berfungsi sebagai kesatuan sistem.

II.2.3 Ciri-Ciri Sistem Pendukung Keputusan

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

- a. Banyak pilihan/alternatif
- b. Ada kendala atau surat
- c. Mengikuti suatu pola atau model tingkah laku, baik yang terstruktur maupun tidak terstruktur.
- d. Banyak input/Variabel
- e. Ada faktor resiko, dibutuhkan kecepatan, ketepatan dan keakuratan .

II.2.4 Fase Dalam Sistem Pendukung Keputusan

Menurut Dicky Nofriansyah (2014:2), Tiga fase dalam pengambilan keputusan yaitu :

a. Inteligence

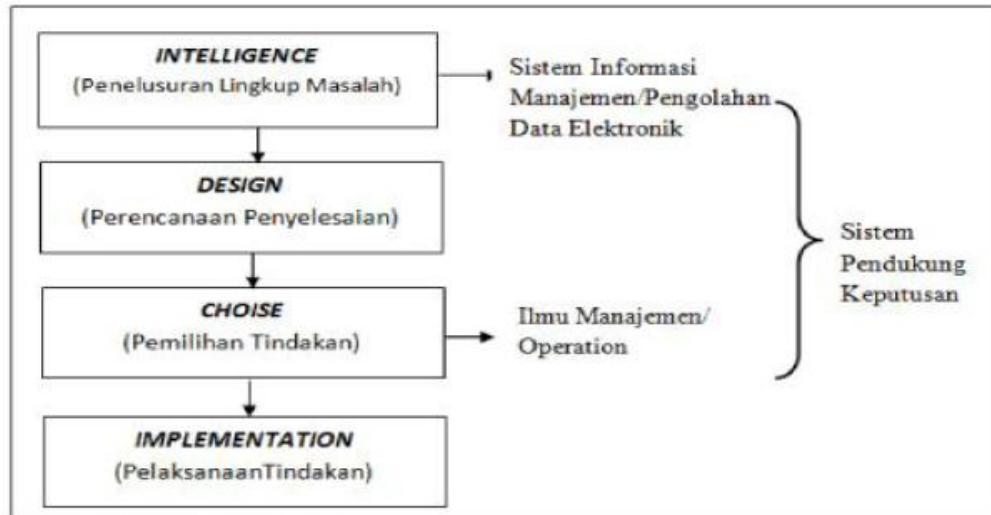
Tahap ini merupakan proses penelusuran dan pendekripsi dari ruang lingkup problematika secara proses pengenalan masalah. Data masukan diperoleh dan diuji dalam rangka mengidentifikasi masalah.

b. Design

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

c. Choice

Pada tahai 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, S.Kom, M.Kom ; 2014)

II.2.5 Komponen Sistem Pendukung Keputusan

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

a. Subsistem Data (*Database*)

Subsistem 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*)

b. Subsistem Model

Model adalah suatu tiruan dari alam nyata. Kendala yang sering diahadapi 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 diajaga fleksibilitasnya. Hal lain yang harus diperhatikan adalah pada setiap model yang disimpan hendaknya ditambahkan rincian, keterangan dan penjelasan yang komprehensif mengenai model yang dibuat.

c. Subsistem dialog (*User System Interface*)

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

II.2.6 Tujuan Sistem Pendukung Keputusan

Menurut Dicky Nofriansyah (2014:4), Tujuan dari system pendukung keputusan adalah sebagai berikut :

- a. Membantu dalam pengambilan keputusan atas masalah yang terstruktur
- b. Memberikan dukungan atas pertimbangan manager dan bukannya dimaksudkan untuk menggantikan fungsi manager.
- c. Meningkatkan efektifitas keputusan yang diambil lebih dari perbaikan efisiensinya.
- d. Kecepatan komputasi komputer memungkinkan para pengambil keputusan untuk banyak melakukan komputasi secara cepat dengan biaya yang sangat rendah.
- e. 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 hukum) bisa ditingkatkan. Produktivitas juga ditingkatkan menggunakan peralatan optimalisasi yang menjalankan sebuah bisnis.

II.3 Metode SAW (*Simple Additive Weighting*)

Menurut Youllia Indrawaty (2011:33), Konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut. Metode SAW membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada.:

$$r_{ij} = \begin{cases} \frac{x_{ij}}{\text{Max}_x} & \text{Jika Atribut keuntungan (Benefit)} \end{cases} \quad (1)$$

$$r_{ij} = \begin{cases} \frac{\text{Min}_x - x_{ij}}{x} & \text{Jika Atribut biaya (Cost)} \end{cases} \quad (2)$$

Keterangan :

R_{ij} = nilai rating kinerja normalisasi

X_{ij} = nilai atribut yang dimiliki dari setiap kriteria

$\text{Max } x_{ij}$ = nilai terbesar dari setiap kriteria

$\text{Min } x_{ij}$ = nilai terkecil dari setiap kriteria

Benefit = nilai terbesar adalah terbaik

Cost = nilai terkecil adalah terbaik

dimana r_{ij} adalah rating kinerja ternormalisasi dari alternatif A_i pada atribut C_j ; $i=1,2,\dots,m$ dan $j=1,2,\dots,n$. Nilai *preferensi* untuk setiap alternatif (V_i) diberikan sebagai:

$$V_i = \sum_{j=1}^n W_j r_{ij} \quad (3)$$

Keterangan :

V_i = rangking untuk setiap alternatif

W_j = nilai bobot dari setiap kriteria

r_{ij} = nilai rating kinerja ternormalisasi.

Nilai V_i yang lebih besar mengindikasi bahwa alternatif A_i lebih terpilih

Langkah-langkah perhitungan dengan menggunakan metode SAW ialah sebagai berikut:

1. Merubah nilai kriteria ke dalam bilangan angka yang parameteranya sudah ditentukan didalam rubrik penilaian.
2. Menghitung matriks ternormalisasi untuk setiap kriterianya.
3. Mengalikan masing-masing nilai kriteria dengan vector bobot.
4. Lalu nilai kriteria yang sudah didapat dijumlahkan untuk hasil nilai setiap guru.

II.4. Unified Modeling Language (UML)

Adi Nugroho (2010:6), *Unified Modeling Language (UML)* adalah bahasa pemodelan untuk system atau perangkat lunak yang berparadigma “berorientasi objek”. Pemodelan (*Modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajaran dan dipahami. Dalam hal ini sasaran model sesungguhnya adalah abstraksi segala sesuatu yang ada diplanet bumi menjadi gambaran-gambaran umum yang lebih mudah dipahami dan dipelajari. Adapun tujuan pemodelan (dalam rangka pengembangan system/perangkat lunak aplikasi)

sebagai sarana analisis, pemasahaman visualisasi dan komunikasi antar anggota tim pengembang.

II.4.1. 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 system. 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 system dan mempertahankan konsistensi antara desain dan implementasi dalam bahasa pemrograman.

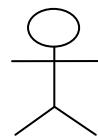
Namun karena UML hanya merupakan bahasa pemodelan maka UML bukanlah rujukan bagaimana melakukan analisis dan desain berorientasi objek. Untuk mengetahui bagaimana melakukan analisis dan desain berorientasi objek secara baik, sudah terdapat beberapa metodologi yang bias diikuti.

II.4.2. 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 didalam UML didefinisikan sebagai informasi dalam berbagai bentuk yang digunakan atau dihasilkan dalam proses pengembangan perangkat lunak.

1. Aktor

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



Gambar II.2. Notasi Aktor

(Sumber : Adi Nugroho ; 2010)

2. Class

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

Mahasiswa
-nim : String
-nama : String
-ipk : Double
+registrasi() : Boolean

Gambar II.3. Notasi Class

(Sumber : Adi Nugroho ; 2010)

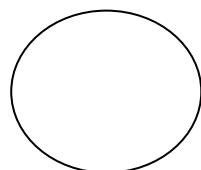
Class digunakan untuk mengabstraksikan elemen-elemen dari system yang dibangun. *Class* bisa untuk direpresentasikan baik perangkat lunak maupun perangkat keras, baik konsep maupun benda nyata. Atribut digunakan untuk menyimpan informasi. Nama atribut menggunakan kata benda yang bisa dengan jelas direpresentasikan informasi yang disimpan didalamnya. 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.

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 : Adi Nugroho ; 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 system 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. Didalam *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 system 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.

f. Pre-Condition

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

g. 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 : Adi Nugroho ; 2010)

6. Package

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



Gambar II.7. Notasi *Package*

(Sumber : Adi Nugroho ; 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 : Adi Nugroho ; 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 kumpulan-kumpulan *file* yang saling berkaitan.

Menurut Kusrini (2010:2), 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 symbol).

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 Kusrini (2010:2), Basis data bertujuan untuk mengatur data sehingga diperoleh kemudahan, ketepatan dan kecepatan dalam pengambilan kembali. Untuk mencapat tujuan, syarat basisdata yang baik adalah sebagai berikut :

- a. Tidak adanya redundansi dan inkonsistensi data

Redundansis terjadi jika suatu informasi disimpan di beberapa 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.

- b. Kesulitan pengaksesan data

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

- c. 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/Kelebihan Basis Data

Menurut Kusrini (2010:5), Banyak manfaat yang diperoleh dengan menggunakan basis data, Manfaat/Kelebihan Basis Data dan kelebihan basis data diantaranya adalah :

- a. 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 metematika. Dengan perancangan yang benar maka penyajian informasi dapat dilakukan dengan cepat dan mudah.

- b. 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.

- c. Pemusatan kontrol data

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

- d. 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.

e. 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.

f. 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.

g. 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.

h. 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.

i. Pemakaian secara langsung

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

j. Kebebasan data

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

k. User View

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

II.5.3. Operasi Dasar Database

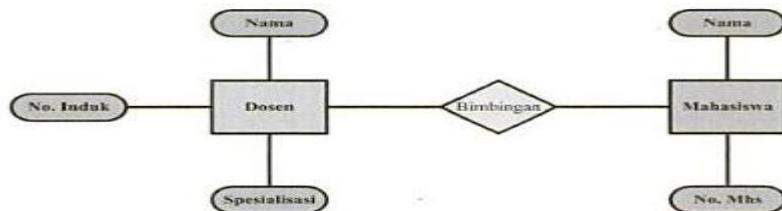
Menurut Kusrini (2010:9), Beberapa operasi dasar basis data yaitu :

- a. Pembuatan basis data
- b. Penghapusan basis data
- c. Pembuatan file/tabel
- d. Penghapusan file/tabel
- e. Pengubahan tabel
- f. Penambahan/pengisian
- g. Pengambilan data
- h. Penghapusan data

II.5.4. Pemodelan Basis Data

Menurut Samiaji Sarosa (2010:4), Model diperlukan untuk mendapatkan penyederhanaan dari kenyataan dan memungkinkan desainer program program aplikasi berekspresi 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 : Adi Nugroho ; 2010)

Dalam perkembangannya banyak diciptakan notasi ERD yang berbeda-beda seperti telihat gambar dibawah ini.



Gambar II.10. Diagram Dengan Notasi Crows Foot

(Sumber : Adi Nugroho ; 2010)



Gambar II.11. Diagram Dengan Notasi Relational

(Sumber : Adi Nugroho ; 2010)

II.5.5. Normalisasi

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

a. *Insertion Anomaly*

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

b. *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 :

a. *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.

b. *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.

c. *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

Java telah hadir dalam dunia pemrograman selama satu dekade lebih. Sudah ada beberapa bahasa pemrograman lainnya yang berusaha menyamai bahkan menggantikan kedudukan Java sebagai bahasa pemrograman yang pertama kali memperkenalkan pemrograman lintas platform secara independen dan tidak tergantung dalam sebuah mesin.

II.6.1. Sekilas Sejarah Java

Menurut Wahana Komputer (2010:1), Bahasa pemrograman Java dari sebuah tim pengembangan *software* dari Sun Microsystem yang dipimpin oleh James Gosling dan Patrick Naughton. Pada tahun 1991, Sun Microsystem mengembangkan sebuah bahasa pemrograman yang berukuran kecil untuk diimplementasikan pada alat elektronik rumah tangga.

Oleh karena itu keharusan sebuah bahasa pemrograman yang kecil, menghasilkan kode yang kecil pula dan harus platform independen membuat tim pada proyek terinspirasi oleh ide pemrograman yang sama yang telah ditemukan oleh Niklaus Wirth, penemu Pascal. Jadi penemu Bahasa Pascal juga memiliki pemikiran tentang sebuah software bahasa pemrograman yang portable dan tidak tergantung pada sebuah platform atau mesin.

Booming Bahasa Java dimulai pada tahun 1995 ketika Netscape memutuskan untuk menggunakan Java pada web browsernya, yaitu *Netscape Navigator* pada Januari 1996. Hal ini kemudian diikuti oleh raksasa-raksasa *software* seperti IBM, Symantec, Inprise dan masih banyak lagi termasuk Microsoft dengan *internet explorer*-nya.

II.6.2. Karakteristik Java

Menurut Wahana Komputer (2010:6), Karakteristik Java adalah sebagai berikut :

1. *Java is simple*

Sebenarnya tidak ada satupun bahasa pemrograman yang disebut sederhana, akan tetapi bila dibandingkan dengan pendahulunya seperti Bahasa C++ yang merupakan bahasa pemrograman yang mendominasi dunia pemrograman Java menjadi lebih sederhana.

2. *Java is object oriented*

Pemrograman berorientasi objek adalah pendekatan atau metodologi perancangan system berdasarkan objek. Metode ini mengantikan metode procedural yang telah lama dilakukan. Dalam pemrograman berorientasi objek semua hal dapat dianggap objek.

3. *Java is distributed*

Distributed Computing adalah metode komputerisasi dengan menggunakan dan beberapa komputer yang dihubungkan dengan jaringan untuk mengelola tugas-tugas tertentu. Java telah memiliki kemampuan *Networking* yang bagus, yang menjadikan menulis program *networking* seperti mengirim dan menerima data dari sebuah file.

4. *Java is interpreted*

Karakteristik yang satu ini penting untuk dimengerti oleh pengguna Java yang baru pertama kali mengenal Java. Java adalah bahasa pemrograman yang menggunakan interpreter atau penerjemah supaya dapat menjalankan program

5. *Java is robust*

Robust artinya dapat diandalkan Java merupakan bahasa pemrograman yang dapat diandalkan.

6. *Java is Secure*

Sebuah bahasa pemrograman internet, Java digunakan pada lingkungan *networking* dan terdistribusi.

7. *Java is Architecture Neutral*

Program yang dihasilkan oleh Java tidak tergantung oleh arsitektur komputer tertentu.

8. *Java is Portable*

Program Java dapat dibawa kemana-mana dan dijalankan dimana-mana.

9. *Java is Performance*

Kinerja dan performa pemrograman Bahasa Java ini sering mendapatkan kritikan atau dianggap lambat beberapa developer.

II.7. Mysql

Menurut Wahana Komputer (2010:5), Mysql adalah program database yang mampu mengirim dan menerima data dengan sangat cepat dan *multi user*. Mysql memiliki dua bentuk lisensi yaitu *free software* dan *shareware*. Mysql sudah cukup lama dikembangkan, beberapa fase penting dalam pengembangan mysql adalah sebagai berikut :

- a. Mysql dirilis pertama kali secara internal pada 3 Mei 1995
- b. Versi Windows dirilis pada 8 Januari 1998 untuk windows 95 dan windows 1995.
- c. Versi 2.32, beta dari Juni 2000, dan dirilis pada Januari 2001.

- d. Versi 4.0 : Beta dari bulan Juni 2004, dirilis pada bulan Oktober 2005
- e. Versi 5.0 : Beta dari Bulan Maret 2005, dirilis pada Bulan Oktober 2005
- f. Sun Microsystem membeli Mysql AB pada tanggal 26 Pebruari 2008.
- g. Versi 5.1 : Dirilis 27 November 2008

Mysql database server adalah *RDBMS (Relational Database Management System)* yang dapat menangani data yang bervolume besar. Meskipun begitu, tidak menuntut *resource* yang besar. Mysql adalah database yang paling popular diantara database-database lainnya.

II.7.1 Kelebihan dan Keuntungan Memakai Mysql

Menurut Wahana Komputer (2010:6), Dalam dunia programming ada beberapa database yang sering digunakan antara lain Mysql, Oracle, PostgreSQL, Microsoft Sql Server dan lain-lain. Ketika dibandingkan antara Mysql dengan system manajemen database yang lain, Mysql memiliki beberapa kelebihan dan keuntungan dibandingkan database lain diantaranya adalah :

- a. Banyak ahli berpendapat bawah Mysql merupakan server tercepat
- b. Mysql merupakan system manajemen database yang *open source* yaitu *software* ini bersifat *free* atau bebas digunakan oleh perseorangan atau instansi tanpa harus membeli atau membayar kepada pembuatnya.
- c. Mysgl mempunyai ferforma yang tinggi namun simpel
- d. Database Mysql mengerti bahasa SQL (*Structured Query Languange*)
- e. Mysql dapat diakses melalaui protocol ODBC (*Open Database Connectivity*) buatan Microsoft. Ini menyebabkan Mysql dapat diakses oleh banyak software.

- f. Semua *client* dapat mengakses *Server* dalam satu waktu tanpa harus menunggu yang lain mengakses database.
- g. Mysql dapat diakses dari semua tempat di internet dengan hak akses tertentu
- h. Mysql merupakan database yang mampu menyimpan data berkapasitas besar, sampai berukuran gigabyte
- i. Mysql dapat berjalan diberbagai *operating system* seperti Linux, Windows, Solaris dan lain-lain.