

BAB II

LANDASAN TEORI

II.1. Teori Sistem

Suatu Sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variable yang terorganisir, saling berinteraksi, saling bergantung satu sama lain, dan terpadu. Secara garis besar terdapat dua kelompok pendekatan yang dilakukan yaitu :

1. Pendekatan sistem yang lebih menekankan pada elemen-elemen atau kelompoknya, didalam hal ini sistem ini didefinisikan sebagai suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu.
2. Pendekatan sistem sebagai jaringan kerja dari prosedur, yang lebih menekankan urutan operasi didalam sistem. Prosedur didefinisikan sebagai urutan operasi kerja (tuliskan-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

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 (*Environtment*)

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

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

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

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

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

Tiga fase dalam pengambilan keputusan yaitu :

a. *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.

b. *Design*

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

c. *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

II.2.5 Komponen Sistem Pendukung Keputusan

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 bahwa 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

Tujuan dari system pendukung keputusan adalah sebagai 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.

III.3 Metode Analitical Hierarchy Process (AHP)

Dalam Jurnal Ilmiah Komputer dan Informatika dengan judul “Pembangunan Sistem Pendukung Keputusan Rekrutmen Pegawai Baru di PT. ABC”, Metode AHP atau Proses Hirarki Analitik merupakan salah satu metode pengambilan keputusan dimana faktor-faktor logika, intuisi, pengalaman, pengetahuan, emosi, dan rasa dicoba untuk dioptimalkan dalam suatu proses

yang sistematis. Sistem yang dibangun didukung oleh metode AHP. Adapun langkah-langkah dalam perhitungan AHP adalah sebagai berikut:

- a. Mendefinisikan masalah dan menentukan solusi yang diinginkan.
- b. Membuat struktur hierarki yang diawali dengan tujuan utama.
- c. Menjumlahkan setiap kolom (Σ kolom) pada matriks perbandingan.
- d. Normalisasi matriks, dengan membagi setiap kolom matriks dengan jumlah kolom (Σ kolom), kemudian dijumlahkan setiap barisnya (Σ baris).
- e. Menghitung total priority value (TPV) untuk mendapatkan bobot subkriteria.
- f. Menghitung uji konsistensi

Tahapan dalam melakukan uji konsistensi adalah sebagai berikut:

- a. Mengalikan nilai TPV dengan nilai kolom matriks pada nilai matriks perbandingan kemudian jumlahkan tiap barisnya.
- b. Mencari *consistency index* (CI) dengan rumus:

$$CI = \frac{1 \max}{n - 1} \quad (2.1)$$

Dimana :

CI = *Consistency Index*

N = banyaknya elemen yang dibandingkan

1 max = *Eigen value maksimum*

- c. Mencari Consistency Ratio (CR) dengan rumus:

$$CR = \frac{CI}{RI} \quad (2.2)$$

Dimana :

CR = *Consistency Ratio*

CI = *Consistency Index*

RI = *Random Index*

Kriteria dan alternatif dinilai melalui perbandingan berpasangan. Menurut Thomas L. Saaty, untuk berbagai persoalan, skala 1 sampai 9 adalah skala terbaik dalam mengekspresikan pendapat. Nilai dan definisi pendapat kualitatif dari skala perbandingan Saaty dapat dilihat pada table berikut :

Tabel II.1 Skala Perbandingan

Intensitas Kepentingan	Definisi
1	Kedua elemen sama pentingnya
3	Elemen yang satu sedikit lebih penting daripada elemen yang lainnya
4	Elemen yang satu lebih penting daripada yang lainnya
7	Satu elemen jelas lebih mutlak penting daripada elemen lainnya
9	Satu elemen mutlak penting daripada elemen lainnya
2,4,6,8	Nilai-nilai antara dua nilai pertimbangan-pertimbangan yang berdekatan

Sumber : Astri Herdiyanti

Sebuah nilai rasio dikatakan konsisten jika bernilai $0 \leq \text{rasio} \leq 0.1$, dengan demikian hasil perhitungan data dapat dibenarkan. Untuk menentukan rasio konsisten atau tidak dapat menggunakan tabel konsistensi rasio sebagai berikut :

Tabel II.2 Konsistensi Rasio

N	RI
1	0.00
2	0.00
3	0.58
4	0.90
5	1.12
6	1.24
7	1.32
8	1.41
9	1.45
10	1.49
11	1.51
12	1.48
13	1.56
14	1.57
≥ 15	1.59

Sumber : Astri Herdiyanti dkk (2013:50)

II.4. *Unified Modeling Language (UML)*

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 dipelajari

dan dipahami. Dalam hal ini sasaran model sesungguhnya adalah abstraksi segala sesuatu yang ada di planet 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

UML sebagai sebuah bahasa yang memberikan *vocabulary* dan tatanan penulisan kata-kata dalam 'MS Word' untuk kegunaan komunikasi. Sebuah bahasa model adalah sebuah bahasa yang mempunyai *vocabulary* dan konsep tatanan / aturan penulisan serta secara fisik mempresentasikan dari sebuah sistem. Seperti halnya *UML* adalah sebuah bahasa *standard* untuk pengembangan sebuah *software* yang dapat menyampaikan bagaimana membuat dan membentuk model-model, tetapi tidak menyampaikan apa dan kapan model yang seharusnya dibuat yang merupakan salah satu proses implementasi pengembangan *software*.

UML tidak hanya merupakan sebuah bahasa *pemograman visual* saja, namun juga dapat secara langsung dihubungkan ke berbagai bahasa pemograman, seperti *JAVA*, *C++*, *Visual Basic*, atau bahkan dihubungkan secara langsung ke dalam sebuah *object-oriented database*.

Begitu juga mengenai pendokumentasian dapat dilakukan seperti; *requirements*, *arsitektur*, *design*, *source code*, *project plan*, *tests*, dan *prototypes*. Untuk dapat memahami *UML* membutuhkan bentuk konsep dari sebuah bahasa model, dan mempelajari 3 (tiga) elemen utama dari *UML* seperti *building block*,

aturan-aturan yang menyatakan bagaimana *building block* diletakkan secara bersamaan, dan beberapa mekanisme umum (*common*).

Alasan mengapa *UML* digunakan adalah, pertama, *scalability* dimana objek lebih mudah dipakai untuk menggambarkan sistem yang besar dan kompleks. Kedua, *dynamic modeling*, dapat dipakai untuk pemodelan sistem dinamis dan *real time*. Sebagaimana dalam tulisan pertama, penulis menjelaskan konsep mengenai obyek, *OOA&D (Obyek Oriented Analyst/ Design)* dan pengenalan *UML*, maka dalam tulisan kedua ini lebih ditekankan pada cara bagaimana *UML* digunakan dalam merancang sebuah pengembangan software yang disertai gambar atau contoh dari sebuah aplikasi.

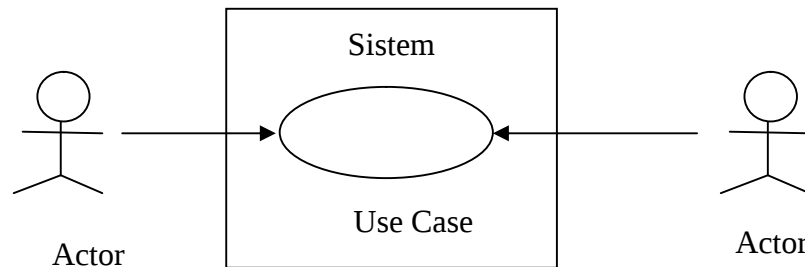
Adapun jenis-jenis dari tipe diagram *UML* adalah sebagai berikut :

1. *Use Case Diagram*

Use Case adalah deskripsi fungsi dari sebuah sistem dari *perspektif* pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Model *use case* adalah bagian dari *requirement*. *Use case* adalah alat bantu terbaik guna menstimulasi pengguna potensial untuk mengatakan tentang suatu sistem dari sudut pandangnya. Dengan demikian diharapkan akan bisa dibangun suatu sistem yang bisa membantu pengguna, perlu diingat bahwa *use case* mewakili pandangan diluar sistem. Diagram *Use Case* menunjukkan 3 aspek dari sistem yaitu : *actor*, *use case* dan *system/sub system boundary*. *Actor* mewakili peran

orang, *system* yang lain atau alat ketika berkomunikasi dengan *use case*.

Berikut gambar notasi *use case* :



Gambar II.2. Notasi Use Case Diagram

Sumber : Adi Nugroho (2010:6)

2. Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class* memiliki tiga area pokok :

- a. Nama (dan *stereotype*)
- b. Atribut
- c. Metoda

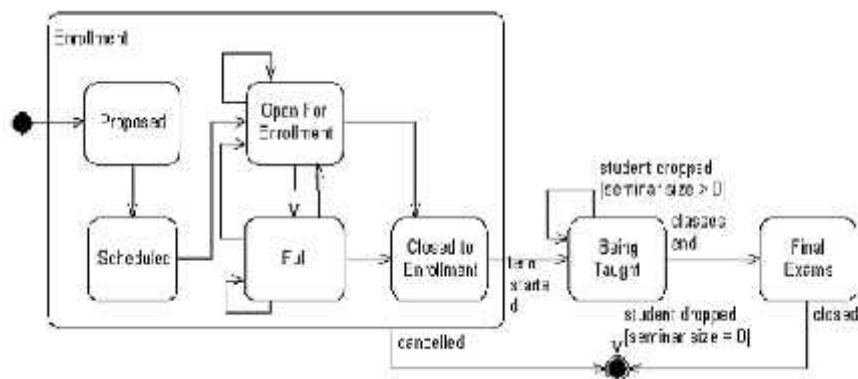
Atribut dan metoda dapat memiliki salah satu sifat berikut :

- a. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan.
- b. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya.
- c. *Public*, dapat dipanggil oleh siapa saja

Class dapat merupakan implementasi dari sebuah *interface*, yaitu *class* *abstrak* yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class*. Dengan demikian *interface* mendukung resolusi metoda pada saat *run-time*.

3. Statechart Diagram

Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari *stimuli* yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*).



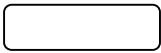
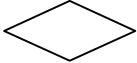
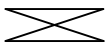
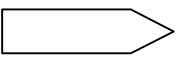
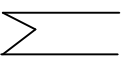
Gambar II.3 Statechart Diagram

Sumber : Adi Nugroho (2010:6)

4. Activity Diagram

Activity Diagram adalah teknik untuk mendiskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. *Activity Diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa. Berikut adalah simbol yang ada pada *activity diagram*.

Tabel II.3. Simbol Yang Ada Pada *Activity Diagram*

SIMBOL	KETERANGAN
	Titik Awal
	Titik Akhir
	<i>Activity</i>
	Pilihan untuk pengambilan keputusan.
	<i>Fork</i> : digunakan untuk menunjukkan kegiatan yang dilakukan secara 31cenario atau untuk menggabungkan dua kegiatan 31cenario menjadi satu.
	<i>Rake</i> ; menunjukkan adanya dekomposisi
	Tanda waktu
	Tanda pengiriman
	Tanda penerimaan
	Aliran Akhir (<i>Flow Final</i>)

Sumber : Adi Nugroho (2010:6)

5. *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display* , dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu.

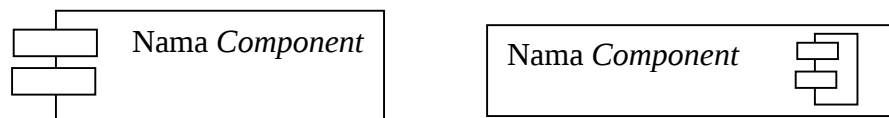
6. Collaboration Diagram

Collaboration diagram juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*

7. Component Diagram

Component Diagram mengandung *component*, *interface* dan *relationship*.

Notasi *component Diagram* dapat dilihat seperti gambar di bawah ini :



Gambar II.4. Notasi Component Diagram

Sumber : Adi Nugroho (2010:6)

8. Deployment Diagram

Deployment Diagram menunjukkan tata letak sebuah sistem secara fisik, menampilkan bagian-bagian *software* yang berjalan pada bagian-bagian *hardware*. Sistem terdiri dari node-node dimana setiap node diwakili untuk sebuah kubus. Garis yang menghubungkan antara 2 kubus menunjukkan hubungan di antara kedua node tersebut. Tipe node bisa berupa *device* yang berwujud *hardware* dan bisa juga *processor*.

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 Wahana Komputer (2010:140), Database diartikan sebagai refresentasi fakta dunia nyata yang mewakili sebuah objek, misalnya manusia, hewan, barang, peristiwa, konsep dan lain sebagainya yang direkam dalam bentuk huruf, teks, symbol, angka, suara, gambar dan lainnya. Sedangkan basis data dapat diartikan sebagai tempat berkumpul, sarang atau gudang untuk menyimpan sesuatu. Dengan demikian basis data/database dapat diartikan sebagai tempat berkumpul, menyimpan data-data suatu benda atau kejadian yang saling berhubungan.

Menurut Kusrini (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 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

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 tool 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/Kelebihan Basis Data

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 metematika. 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

Beberapa operasi dasar basis data yaitu :

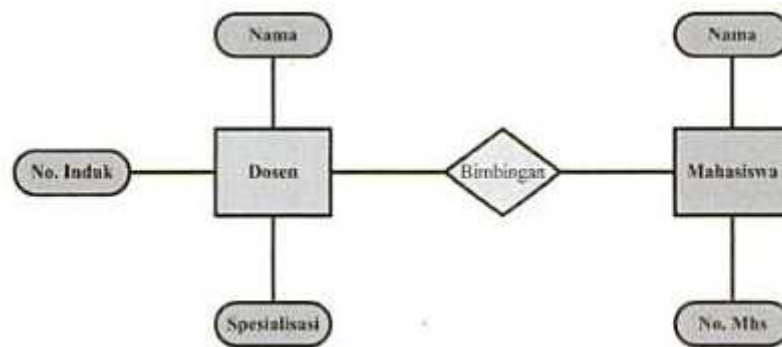
1. Pembuatan basis data
2. Penghapusan basis data
3. Pembuatan file/tabel
4. Penghapusan file/tabel
5. Pengubahan tabel
6. Penambahan/pengisian
7. Pengambilan data
8. Penghapusan data

II.5.3. Pemodelan Basis Data

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 ERD. *Entity Relationship Model* adalah abstraksi konseptual yang mewakili struktur dari suatu basis data.



Gambar II.5. Diagram Dengan Notasi Chen ERD

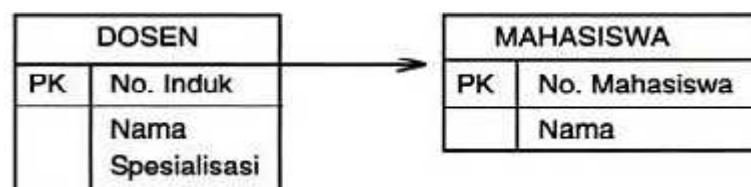
Sumber : Kusrini (2010, p9)

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



Gambar II.6. Diagram Dengan Notasi Crows Foot

Sumber : Kusrini (2010, p9)



Gambar II.7. Diagram Dengan Notasi Relational

Sumber : Kusrini (2010, p9)

II.6. Normalisasi

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 :

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.7. Internet

Internet pertama kali digunakan sebagai proyek penelitian yang ditemukan oleh *Advanced Research Project Agency (ARPA) Department Of Defense (DOD)* di Amerika Serikat. Pada dasarnya internet digunakan untuk menghubungkan komputer. Versi yang pertama disebut ARPANET. Pada tahun 1972, ARPA berubah menjadi DARPA dengan tetap mempromosikan proyek ARPANET.

Pengembangan internet dengan jenis peralatan yang berbeda, namun bisa berhubungan satu sama lain merupakan tantangan terbesar pada saat ini. Pada tahun 1973-1974, peneliti merancang sebuah *Transmission Control Protocol/Internet Protocol (TCP/IP)*. Pada awalnya *TCP/IP* dimaksudkan untuk menyediakan dukungan untuk kebutuhan berikut :

- a. Interporabilitas antar sistem heterogen
- b. Komunikasi *end-to-end* berbagai jaringan yang berbeda.
- c. Operasi otomatis dan sempurna didalam menghadapi terjadinya kegagalan hubungan data

Pada saat itu aplikasi yang digunakan masih sangat sederhana daripada yang digunakan saat ini. Aplikasi yang paling banyak digunakan mungkin adalah telnet untuk *login remote* dan FTP untuk perpindahan file dan e-mail.

Pada tahun 1990, DOD memutuskan untuk membubarkan ARPANET dan menggantikannya dengan pendukung (*backbone*) NSFNET bekerja sama dengan agen jaringan lain. Hal inilah yang menjadi prinsip jaringan internet. Berdasarkan keterangan diatas maka pengertian Internet (kependekan dari *interconnection-networking*) adalah seluruh jaringan komputer yang saling terhubung menggunakan standar sistem global *Transmission Control Protocol/Internet Protocol Suite* (TCP/IP) sebagai protokol pertukaran paket (*packet switching communication protocol*) untuk melayani miliaran pengguna di seluruh dunia. Rangkaian internet yang terbesar dinamakan Internet. Cara menghubungkan rangkaian dengan kaidah ini dinamakan *internetworking* ("antarjaringan") (Janner Simarmata, 2010 : 50)

II.8. Web

Aplikasi Web adalah aplikasi yang sejak awal dirancang untuk dieksekusi didalam program yang berbasis web. Defenisi ini mengungkapkan 2 aspek penting dari aplikasi yaitu :

- a. Suatu aplikasi web dirancang agar dapat berjalan dilingkungan berbasis web. Artinya aspek-aspek hipermedia dalam kaitanya dengan hiperteks dan multimedia didalan kombinasi dengan logika aplikasi tradisional harus diperhitungkan di seluruh siklus hidup aplikasi, yang membuatnya berbeda dengan aplikasi konvensional.
- b. Aplikasi web adalah suatu aplikasi yang tidak berupa sekumpulan halaman-halaman web.P

- c. Secara khusus aplikasi web menguatkan notasi sesi membedakannya dari paradigma web permintaan respons (*request reponse*) yang biasa. Dalam konteks ini, web services secara dinamis akan menghasilkan halaman yang tidak mungkin dipertimbangkan aplikasi web. Contoh, suatu layanan yang jadwal yang diberi tujuan dan keberangkatan, dan tempat yang diinginkan akan mengembalikan sekumpulan halaman yang berisi koneksi dan kereta yang tersedia.

II.9. PHP

PHP atau *Hipertext Processor* merupakan bahasa berbentuk *script* yang ditempatkan dalam server dan dieksekusi dalam server dan selanjutnya ditransfer dan dibaca oleh client. PHP juga disisipkan dalam Bahasa HTML.

PHP diciptakan oleh seorang pria berkewarganeraan Denmark yang bernama Rasmus Leodorf pada tahun 1995. Banyak programmer yang tertarik untuk mengembangkan PHP karena bersifat *open source*. Pada awal peluncurannya, PHP hanya dibuat untuk diintegrasikan dengan *web server apache*. Namun sekarang PHP dapat juga bekerja dengan *web server* seperti PWS (*Personal Web Server*), IIS (*Internet Information Server*) dan Xitami

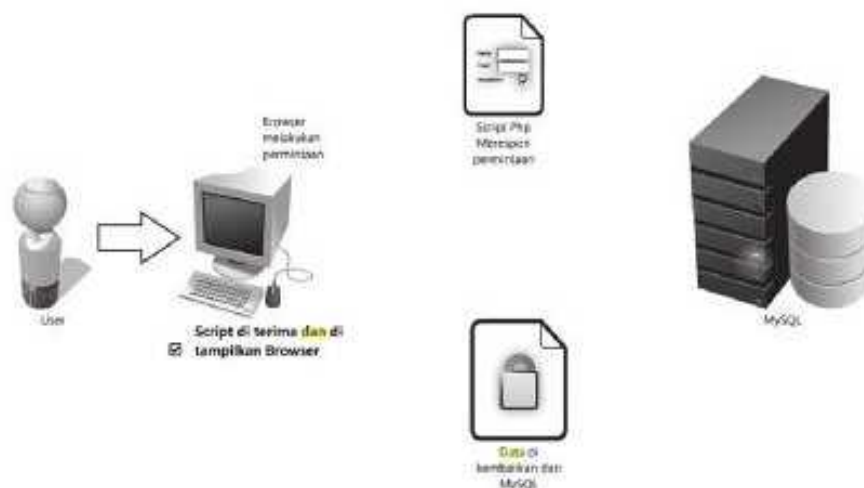


Gambar II.8. Logo PHP

Sumber : Andrea Edelheid dan Khairil Nasution (2012:1)

II.9.1. Cara Kerja PHP

Andrea Edelheid dan Khairil Nasution (2012:12), Dalam bukunya "Buku Pintar Menguasai PHP dan Mysql", menjelaskan cara kerja PHP Secara teori, cara kerja PHP berjalan dari database dalam server kemudian menyalurkannya dalam *script PHP* untuk kemudian ditampilkan kedalam *webbrowser* untuk bisa dibaca *user*



Gambar II.9. Flowchart Cara Kerja PHP

Sumber : Andrea Edelheid dan Khairil Nasution (2012:1)

II.9.2. Kelebihan PHP

Menurut Anhar (2010:3), Kelebihan PHP adalah sebagai berikut :

a. Kesederhaan

Kesederhaan akan memudahkan untuk belajar PHP. User yang sedikit tahu atau bahkan tidak tahu sama sekali, tidak mengerti pemrograman PHP bisa dengan cepat belajar dan mencoba membuat aplikasi web PHP. Selain itu, PHP memiliki banyak sekali fungsi *built-in* untuk menangani kebutuhan

standard pembuatan aplikasi web. Dengan adanya fungsi-fungsi tersebut, maka tentu saja proses belajar PHP terutama dalam pengembangan aplikasi akan jauh lebih mudah karena semua sudah tersedia.

- b. Dalam sisi pemahamannya, PHP adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.
- c. PHP adalah bahasa pemrograman yang *open source* yang dapat digunakan di berbagai system operasi seperti : Linux, Unix, Macintosh dan Windows. PHP dapat dijalankan secara *runtime* melalui *console* serta dapat menjalankan perintah-perintah system. *open source* artinya kode-kode PHP terbuka untuk umum dan tidak harus membayar biaya pembelian atas keaslian lisensi yang biasanya cukup mahal. Karena *source code* PHP tersedia secara gratis, maka hal tersebut memungkinkan komunitas milis-milis dan developer untuk selalu melakukan perbaikan, pengembangan dan menemukan bug dalam Bahasa PHP.
- d. Webserver yang mendukung PHP dapat ditemukan dimana-mana mulai dari Apache, IIS, Lighttpd hingga Xitami dengan konfigurasi yang cukup mudah.
- e. PHP juga dilengkapi dengan berbagai macam pendukung lain seperti support langsung keberbagai macam database yang populer seperti Mysql, PostgreSQL, Mysql dan lain-lain.

II.10. 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 populer diantara database-database lainnya.

II.10.1 Kelebihan dan Keuntungan Memakai Mysql

Menurut Wahana Komputer (2010:6), Dalam dunia programming ada beberapa database yang sering digunakan antara lain Mysql, Oracle, PostgreSQL, MSql, 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 :

1. Banyak ahli berpendapat bawah Mysql merupakan server tercepat

2. 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.
3. Mysql mempunyai ferporma yang tinggi namun simpel
4. Database Mysql mengerti bahasa SQL (*Structured Query Language*)
5. Mysql dapat diakses melalui protocol ODBC (*Open Database Connectivity*) buatan Microsoft. Ini menyebabkan Mysql dapat diakses oleh banyak *software*.
6. Semua *client* dapat mengakses *Server* dalam satu waktu tanpa harus menunggu yang lain mengakses database.
7. Mysql dapat diakses dari semua tempat di internet dengan hak akses tertentu
8. Mysql merupakan database yang mampu menyimpan data berkapasitas besar, sampai berukuran gigabyte
9. Mysql dapat berjalan diberbagai *operating system* seperti Linux, Windows, Solaris dan lain-lain.

II.11. Xampp

Pengertian XAMPP adalah perangkat lunak (free software) bebas, yang mendukung untuk banyak sistem operasi, yang merupakan kompilasi dari program.

Fungsi XAMPP sendiri adalah sebagai server yang berdiri sendiri (localhost), yang terdiri beberapa program antara lain : Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP.