BAB 2

TINJAUAN PUSTAKA

2.1. Tinjauan Perusahaan

2.1.1. Sejarah Perusahaan

Ibrahim Art adalah salah satu perusahaan yang bergerak di industri furniture Jepara. Ibrahim Art menjual berbagai furniture dan ukiran yang berasal dari kota Jepara. Sejak pertama kali dibuka pada tahun 2000 di Jepara, Ibrahim Art telah mempunyai banyak pelanggan. Sampai saat ini Ibrahim Art telah memiliki beberapa cabang yang berfungsi sebagai *showroom* dan terletak di beberapa kota, diantaranya adalah :

1. Jl. Raya Lembang 177, Bandung.
2. Jl. Kaliurang KM. 7, Yogjakarta.
3. Jl. Joglo Raya, Jakarta.
4. Jl. Cileduk Penceng, Tangerang.
5. Baranangsiang Tanah Baru, Bogor.
6. Jl. Urip Sumoharjo (Pulutan), Pemalang.

2.1.2. Visi dan Misi

2.1.2.1. Visi

Menjadi salah satu perusahaan furniture Jepara yang dapat memberikan pelayanan terbaik terhadap seluruh konsumen di Indonesia dan Mancanegara.

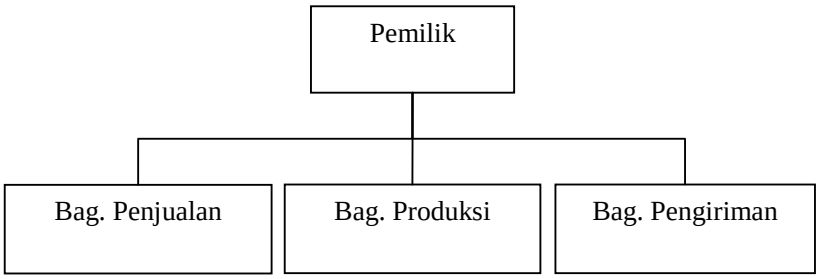
2.1.2.2. Misi

1. Menyediakan produk dengan kualitas terbaik.
2. Menjamin kualitas produk yang dibeli konsumen.
3. Menyediakan layanan purnajual bagi setiap konsumen.

2.1.3. Struktur Organisasi

Struktur Organisasi merupakan susunan yang terdiri dari fungsi-fungsi dan hubungan hubungan yang menyatakan keseluruhan kegiatan untuk mencapai suatu sasaran. Secara fisik struktur organisasi dapat dinyatakan dalam bentuk bagan yang memperlihatkan hubungan unit-unit organisasi dan garis-garis wewenang yang ada.

Struktur keorganisasian yang ada di Ibrahim Art dapat dilihat pada gambar 2.1 berikut ini:



Gambar 2.1 *Struktur Organisasi Ibrahim Art*

2.1.4. Deskripsi Tugas

Deskripsi tugas yang ada di perusahaan adalah sebagai berikut :

1. Pemilik
   a. Sebagai pengambil keputusan.
   b. Sebagai koordinator semua kegiatan yang dilaksanakan.

c. Mengevaluasi semua kegiatan yang telah dilaksanakan.
d. Menerapkan dan mengesahkan kebijakan yang menyangkut eksistensi perusahaan.
e. Melaksanakan pemeriksaan yang meliputi seluruh aspek kegiatan agar pengelolaan dapat dilaksanakan secara efektif dan efisien.

2. Bagian Penjualan
   a. Bertanggung jawab langsung kepada pemilik perusahaan.
   b. Melayani pembelian/pemesanan secara langsung.
   c. Melaporkan hasil penjualan harian.
   d. Melaporkan pemesanan harian.
   e. Melakukan pemeliharaan terhadap stok barang yang ada di *showroom.*
3. Bagian Produksi
   a. Bertanggung jawab langsung kepada pemilik perusahaan.
   b. Bertanggung jawab atas pengadaan barang.
   c. Bertanggung jawab atas barang yang ada di tempat produksi dan pengiriman barang ke bagian penjualan.
   d. Melaporkan jumlah dan jenis barang yang diproduksi.
4. Bagian Pengiriman
   a. Bertanggung jawab langsung kepada pemilik perusahaan.
   b. Bertanggung jawab atas pengiriman barang kepada pelanggan.
   c. Melaporkan pengiriman barang yang dilakukan.

2.2. Landasan Teori

2.2.1. Konsep Sistem

2.2.1.1. Data

Data sebenarnya merupakan fakta-fakta atau kejadian yang dapat berupa angka angka atau kode-kode tertentu[1]. Data dapat berupa angka-angka, huruf, atau simbol-simbol khusus atau gabungan darinya. Data mentah masih belum bisa bercerita banyak, sehingga perlu diolah lebih lanjut. Pengolahan data (*Data Processing*) adalah manipulasi dari data ke dalam bentuk yang lebih berguna dan lebih berarti berupa suatu informasi.

2.2.1.2. Informasi

Informasi (*Information*) adalah hasil dari kegiatan pengolahan data yang memberikan bentuk yang lebih berarti dari suatu kejadian. Jadi pengolahan data elektronik (PDE) atau *electronic data processing* (EDP) adalah manipulasi dari data kedalam bentuk yang lebih berarti berupa suatu informasi dengan menggunakan suatu alat elektronik, yaitu komputer.

Suatu proses pengolahan data terdiri dari tiga tahapan dasar yang disebut dengan siklus pengolahan data, yaitu input, proses dan output.



Gambar 2.2 *Proses Pengolahan Data*

INPUT : Tahap ini merupakan proses memasukkan data kedalam proses komputer lewat data alat input (*input device*).

PROSES : Tahap ini merupakan proses pengolahan dari data yang sudah dilakukan oleh alat pemproses (*processing device*) yang dapat berupa proses menghitung, membandingkan, mengendalikan atau mencari di *storage* (media penyimpanan).

OUTPUT : Tahap ini merupakan proses menghasilkan output dari hasil pengolahan data ke alat output (*output device*) yaitu berupa informasi.

2.2.1.3. Sistem

Sistem berasal dari bahasa Yunani “systema” yang mempunyai arti :

1. Suatu keseluruhan yang tersusun dari sekian banyak bagian.
2. Hubungan yang berlangsung diantara satuan-satuan atau komponen-komponen secara teratur.

Terdapat dua kelompok dalam pengertian sistem, yaitu pengertian yang menekankan pada prosedur dan yang menekankan pada komponen atau elemennya. Dengan pendekatan prosedur sistem didefinisikan sebagai kumpulan dari beberapa prosedur yang mempunyai tujuan tertentu[1]. Yang dimaksud prosedur disini adalah urutan-urutan yang tepat dari tahapan-tahapan instruksi yang menerangkan apa yang harus dikerjakan, siapa yang mengerjakannya, kapan dikerjakan dan bagaimana mengerjakannya.

Dengan pendekatan komponen, sistem merupkan kumpulan dari komponen komponen yang saling berkaitan untuk mencapai tujuan tertentu[1].

Dengan adanya sistem ini, kita berpandangan bahwa sistem merupakan suatu komponen yang saling berhubungan dan dengan keterkaitannya

tersebut dapat menimbulkan suatu yang disebut informasi sebagai output untuk mencapai tujuan.

Menurut William S. Davis :

"*Sistem adalah suatu penggabungan metode, prosedur atau teknik yang telah diatur membentuk kesemuanya itu menjadi terorganisasi secara baik dan benar*".

Menurut Indriyono Gitosumarno :

"*Sistem adalah suatu agresi (kumpulan) elemen yang dinamis, yang berhubungan satu sama lain dan saling tergantung dan berjalan sesuai dengan hukum-hukum tertentu*".

Menurut Dj. A. Simarmata :

"*Sistem adalah keseluruhan elemen yang mendukung pencapaian tujuan yang mempunyai kegiatan fungsional satu sama lain*".

Menurut H.A. Harding :

"*Sistem adalah bagian yang mempunyai kaitan satu sama lain, yang bersama-sama bereaksi menurut pola tertentu terhadap input dengan tujuan untuk menghasilkan output. Biasanya pola tindakannya dibuat dengan tujuan untuk mengoptimalkan faktor-faktor dan sifat-sifat tertentu*".

Menurut Gordon B. Davis :

"*Sistem dapat membentuk abstrak maupun fisik. Sebuah sistem abstrak adalah suatu susunan teratur, gagasan atau konsepsi yang saling*

bergantung, sedang sistem yang bersifat fisik adalah serangkaian unsur yang bekerjasama untuk mencapai suatu tujuan bersama".

Menurut S. Prajudi Atmasudiro :

"*Sistem adalah setiap sesuatu yang terdiri dari objek-objek tau unsur-unsur yang saling berhubungan satu sama lainnya sehingga unsur tersebut merupakan satu keatuan dalam pemrosesan atau pengolahan tertentu*".

Berdasarkan pendapat para ahli tersebut dapat diambil beberapa kesimpulan yang merupakan ciri-ciri dari sistem yaitu sebagai berikut :

1. Sistem mempunyai tujuan tertentu.
2. Sistem terdiri dari bagian-bagian atau elemen-elemen.
3. Bagian-bagian yang berhubungan membentuk struktur yang mengikat keseluruhan dalam satu kesatuan.

Karakteristik Sistem yang terdiri dari beberapa karakter yaitu :

1. Komponen-komponen, suatu sistem berisi komponen yang saling berinteraksi atau saling bekerja sama.
2. Batas sistem, yaitu pemisah antara sistem dengan lingkungan luarnya.
3. Lingkungan Luar, yaitu merupakan sistem yang ada diluar batas sistem ini tetapi lingkungan ini ada yang memberi pengaruh positif dan negatif.
4. Penghubung, yaitu keluaran dari suatu sub sistem dapat sebagai penghubung ke sub sistem lainnya.

5. Masukan, yaitu masukan yang diberikan pada sistem ada 2 macam yaitu masukan perawatan (sebagai program dalam *compiler* agar sistem beroperasi) dan masukan sinyal (sebagai data dalam komputer agar menghasilkan).
6. Proses, yaitu pengolahan input sehingga menghasilkan output yang diinginkan.
7. Sasaran, tujuannya dibuat suatu sistem.
8. Keluaran, yaitu hasil dari operasi sistem ada yang memenuhi sasaran dan ada yang tidak.

Klasifikasi sistem diantaranya :

1. *Transaction Processing Systems* (TPS)

   Sistem informasi komputerisasi yang dikembangkan untuk memproses data-data dalam jumlah besar untuk transaksi bisnis rutin seperti daftar gaji dan inventarisasi.

2. *Office Automation Systems (OAS)*

   Mendukung pekerja data yang biasanya tidak menciptakan pengetahuan baru melainkan hanya menganalisis informasi sedemikian rupa untuk mentransformasikan data atau memanipulasikannya dengan cara tertentu sebelum membaginya atau menyebarkannya secara keseluruhan dengan organisasi dan kadang-kadang diluar itu.

3. *Knowledge Work Systems (KWS)*

   Mendukung para pekerja profesional seperti ilmuan, insinyur dan dokter dengan membantu mereka menciptakan pengetahuaan baru dan

memungkinkan mereka mengkontribusikannya ke organisasi atau masyarakat.

2.2.1.4. *System Development Life Cycle* (SDLC)

Fase *System Development Life Cycle* :

1. Identifikasi, seleksi, dan perencanaan sistem
    a. Mengidentifikasi kebutuhan user
    b. Menyeleksi kebutuhan user dari proses identifikasi dengan
    c. Melihat kapasitas teknologi dan efisiensi
    d. Merencanakan sistem

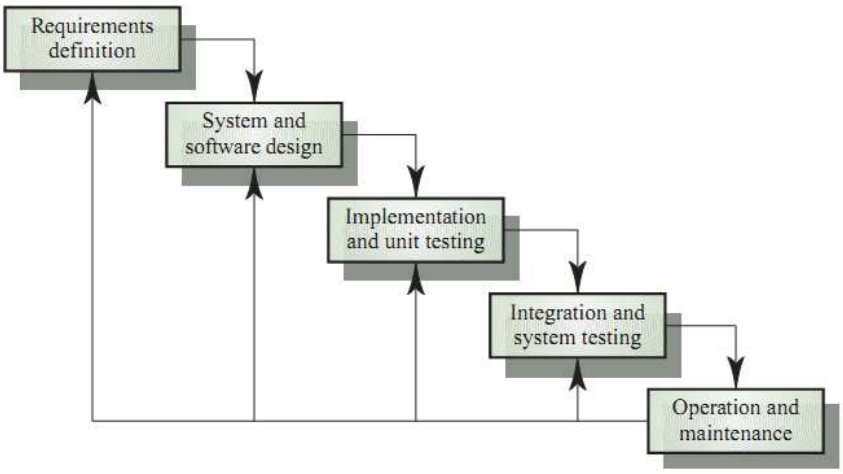
    Kebutuhan :

    a. Kebutuhan fungsional and non-fungsional
    b. Kebutuhan User (*customer*)
    c. Kebutuhan Sistem (kontrak dengan klien)
    d. Kebutuhan dokumen dan perangkat lunak (*developer*)
2. Analisis sistem
    a. Pemodelan Data
    b. Pemodelan Proses
3. Desain sistem
    a. Desain form dan laporan (*report*)
    b. Desain Antarmuka dan dialog (*message*)
    c. Desain basis data dan file (*framework*)
    d. Desain proses (struktur proses)

4. Implementasi sistem
    a. Pemrograman dan pengetesan perangkat lunak(*software*)
        i. *Developmental* (*error testing* per modul oleh *programmer*)
        ii. *Alpha testing* (*error testing* ketika sistem digabungkan dengan antarmuka *user* , oleh *software tester*)
        iii. *Beta testing* (*testing* dengan lingkungan dan data sebenarnya)
    b. Konversi sistem
        i. Mengaplikasikan perangkat lunak pada lingkungan yang sebenarnya untuk digunakan oleh organisasi
    c. Dokumentasi
    d. Pelatihan
5. Pemeliharaan sistem (*maintenance*)
    a. *Corrective*, memperbaiki desain dan error pada program
    b. *Adaptive*, memodifikasi sistem untukberadaptasi dengan perubahan lingkungan
    c. *Perfective*, melibatkan sistem untuk menyelesaikan masalah baru atau mengambil kesempatan (penambahan fitur)
    d. *Preventive*, menjaga sistem darikemungkinan masalah di masa yang akan datang

Jenis *System Development Life Cycle* :

1. *Waterfall*



Gambar 2.3 *Waterfall*

Kelebihan :

a. Proses-prosesnya mudah dipahami dan jelas

b. Mudah dalam pengelolaan proyek

   i. Dokumen dihasilkan setiap akhir fase

   ii. Sebuah fase dijalankan setelah fase sebelumnya selesai

c. Struktur sistem jelas

d. Kondisi tepat SDLC Waterfall

   i. Kebutuhan user telah sangat dipahami

   ii. Kemungkinan terjadinya perubahan kebutuhan user kecil

Kelemahan :

a. ‰ Proyek dunia nyata jarang mengikuti alur proses

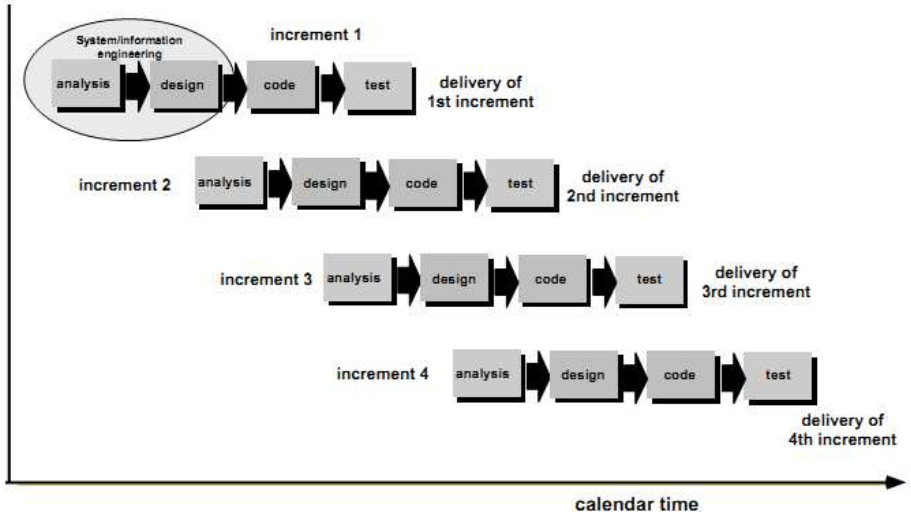
b. ‰ Kesulitan jika terjadi perubahan kebutuhan

   i. Waktu pengerjaan bertambah

ii. Ada anggota tim yang harus menunggu pekerjaan pekerja lain

iii. Kesabaran customer/klien

2. *Incremental Model*



Gambar 2.4 *Incremental Model*

a. Memberikan implementasi sebagai inisial proses kepada user untuk mendapatkan masukan balik

b. Pengembangan dengan terus melakukan eksplorasi

c. Per prototipe

d. Permasalahan

i. Batasan proses tidak jelas

ii. Sistem kurang terstruktur

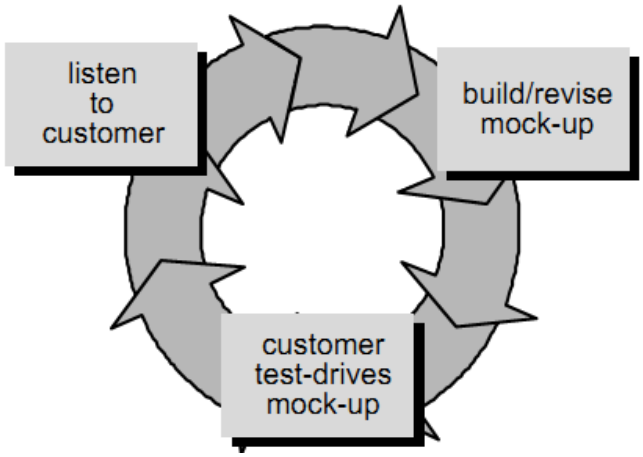
e. Kemampuan aplikasi

i. Untuk sistem dengan interaksi skala kecil dan medium

ii. Untuk antarmuka user

iii. Untuk sistem dengan masa penggunaan pendek

3. *Prototypes*

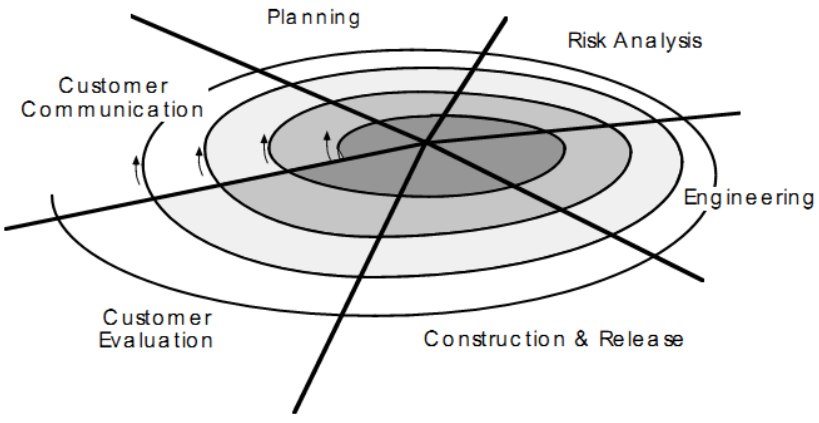


Gambar 2.5 *Prototypes*

Kelemahan :

a. Harus ada versi yang dapat dijalankan sebagai prototipe sebelum sistem dikembangkan (bisa berupa contoh sistem lain)
b. Harus ada implementasi sistem yang dikembangkan sebelum dibuat sebuah sistem final

4. *Spiral Life Cycle*



Gambar 2.6 *Spiral Life Cycle*

a. Mendefinisikan kebutuhan dengan sedetail mungkin
b. Pembuatan desain untuk sistem yang baru
c. Pembuatan prototipe dari pembuatan desain, pembuatan prototipe selanjutnya berdasarkan evaluasi prototipe sebelumnya
d. Proses prototipe dilakukan berulang-ulang sampai customer puas
e. Sistem dibuat berdasarkan prototipe yang memuaskan customer
f. Sistem di tes dan dievaluasi

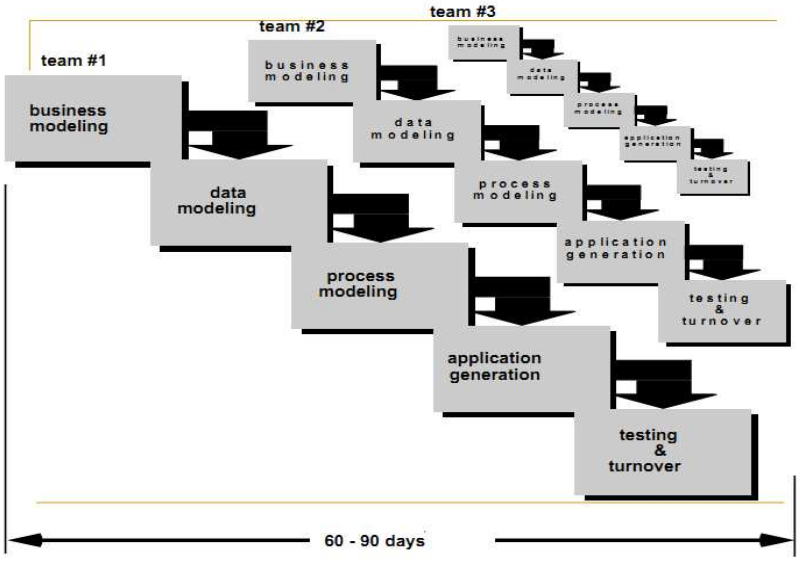
Kelebihan :

a. Dapat digunakan untuk sistem yang besar
b. Sangat cocok sebagai mekanisme mengurangi resiko

Kelemahan :

a. Terlalu banyak memikirkan resiko yang akan terjadi
b. Masih jarang digunakan

5. *Rapid Application Development* (RAD)



Gambar 2.7 *Rapid Application Development*

a. Mengumpulkan spesifikasi menggunakan workshop atau group khusus
b. Melakukan tes berulang-ulang oleh user terhadap desain yang diawali dengan prototipe
c. Menggunakan kembali komponen perangkat lunak yang ada
d. Jadwal yang ketat terhadap perbaikan desain produk versi selanjutnya
e. Komunikasi yang tidak terlalu formal antar anggota tim

Kelebihan :

a. Waktu pengembangan singkat

Kelemahan :

a. Untuk proyek besar memerlukan lebih banyak sumber daya
b. Sangat memerlukan kerjasama antara customer dan developer
c. Tidak cocok untuk kebutuhan yang tidak dapat dimodulkan
d. Tidak cocok untuk sistem yang memerlukan banyak perbaikan
e. Tidak sesuai untuk pengembangan sistem dengan resiko tinggi (aplikasi dengan teknologi baru)

2.2.1.5. Perancangan Sistem

2.2.1.5.1. Flowmap

Flow map adalah bagan aliran yang menunjukan alir dalam program atau prosedur sistem secara manual. Digunakan terutama sebagai alat bantu komunikasi dan dokumentasi.

2.2.1.5.2. Flowchart

Flowchart atau diagram alir adalah sekumpulan simbol-simbol atau skema yang menunjukkan atau menggambarkan rangkaian kegiatan-

kegiatan program dari mulai awal sampai akhir. Inti pembuatan flowchart ini adalah penggambaran urutan langkah-langkah pengerjaan dari suatu algoritma.

2.2.1.5.3. Diagram Konteks

Diagram konteks adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram konteks merupakan level tertinggi dari DFD yang menggambarkan seluruh input ke sistem atau output dari sistem. Diagram tersebut akan memberi gambaran tentang keseluruhan sistem. Sistem dibatasi oleh *boundary* (dapat digambarkan dengan garis putus). Dalam diagram konteks hanya ada satu proses. Tidak boleh ada store dalam diagram konteks.

Diagram konteks berisi gambaran umum (secara garis besar) sistem yang akan dibuat. Secara kalimat, dapat dikatakan bahwa diagram konteks ini berisi “siapa saja yang memberi data (dan data apa saja) ke sistem, serta kepada siapa saja informasi (dan informasi apa saja) yang harus dihasilkan sistem.”

Jadi, yang dibutuhkan dalam pembangunan diagram ini adalah Siapa saja pihak yang akan memberikan data ke sistem, Data apa saja yang diberikannya ke sistem, kepada siapa sistem harus memberi informasi atau laporan, dan apa saja isi/jenis laporan yang harus dihasilkan sistem.

Kata “Siapa” di atas dilambangkan dengan kotak persegi (disebut dengan *terminator*), dan kata “apa” di atas dilambangkan dengan aliran

data (disebut dengan *data flow*), dan kata “sistem” dilambangkan dengan lingkaran (disebut dengan *process*).

2.2.1.5.4. *Data Flow Diagram* (DFD)

DFD atau Diagram Arus Data adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut.

DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau dimana data tersebut akan disimpan.

DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur. Kelebihan utama pendekatan aliran data, yaitu :

1. Kebebasan dari menjalankan implementasi teknis sistem.
2. Pemahaman lebih jauh mengenai keterkaitan satu sama lain dalam sistem dan subsistem.
3. Mengkomunikasikan pengetahuan sistem yang ada dengan pengguna melalui diagram aliran data.
4. Menganalisis sistem yang diajukan untuk menentukan apakah data-data dan proses yang diperlukan sudah ditetapkan.

Disamping itu terdapat kelebihan tambahan, yaitu :

1. Dapat digunakan sebagai latihan yang bermanfaat bagi penganalisis, sehingga bisa memahami dengan lebih baik keterkaitan satu sama lain dalam sistem dan subsistem.
2. Membedakan sistem dari lingkungannya dengan menempatkan batas-batasnya.
3. Dapat digunakan sebagai suatu perangkat untuk berinteraksi dengan pengguna.
4. Memungkinkan penganalisis menggambarkan setiap komponen yang digunakan dalam diagram.

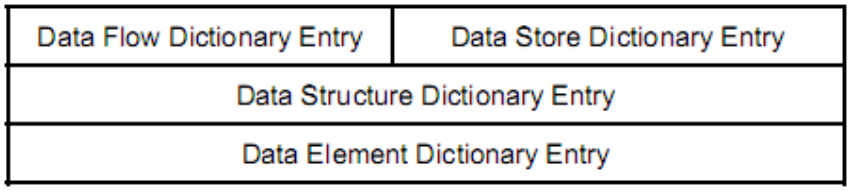
2.2.1.5.5. Kamus Data

Kamus data adalah suatu daftar data elemen yang terorganisir dengan definisi yang tetap dan sesuai dengan sistem, sehingga user dan analis sistem mempunyai pengertian yang sama tentang input, output, dan komponen *data store*. Kamus data ini sangat membantu analis sistem dalam mendefinisikan data yang mengalir di dalam sistem, sehingga pendefinisian data itu dapat dilakukan dengan lengkap dan terstruktur. Pembentukan kamus data dilaksanakan dalam tahap analisis dan perancangan suatu sistem.

Pada tahap analisis, kamus data merupakan alat komunikasi antara user dan analis sistem tentang data yang mengalir di dalam sistem, yaitu tentang data yang masuk ke sistem dan tentang informasi yang dibutuhkan oleh user. Sementara itu, pada tahap perancangan sistem kamus data digunakan untuk merancang input, laporan dan database.

Pembentukan kamus data didasarkan atas alur data yang terdapat pada DFD. Alur data pada DFD ini bersifat global, dalam arti hanya menunjukkan nama alur datanya tanpa menunjukkan struktur dari alur data itu. Untuk menunjukkan struktur dari alur data secara terinci maka dibentuklah kamus data yang didasarkan pada alur data di dalam DFD.

Suatu sistem dapat diuraikan ke dalam 4 form kamus data yang menerangkan isi database sistem dalam bentuk hirarki seperti yang digambarkan pada gambar 2.8 sebagai berikut :



Gambar 2.8 *Hierarki Kamus Data*

Dari gambar di atas tampak bahwa *data flow* dan *data store* ada pada level tertinggi. Di sini lebih baik menganggap *data flow* dan *data store* sebagai *file* dari data. Selanjutnya struktur data yang ada pada *data flow* dan data store terletak pada level kedua atau *middle level*. Di sini struktur data dianggap sebagai *record* data. Yang terakhir adalah data element yang terletak pada level terendah, karena data element merupakan bagian dari struktur data. Di sini data elemen dianggap sebagai *field*.

2.2.1.5.5.1. Data Flow Dictionary Entry

Data flow dictionary entry ini menerangkan setiap *data flow* pada DFD. *Data flow* ini dapat berupa :

a. Satu struktur yang terdiri dari satu elemen data tunggal.

b. Satu struktur yang terdiri dari satu paket elemen data.

c. Multiple struktur.

Berdasarkan uraian di atas, maka hubungan antara alur data pada DFD dan alur data pada elemen kamus data adalah *one to one relationship* (relasi satu-satu). Jika DFD berisi 40 alur data, maka kamus data harus mempunyai 40 elemen alur data.

Data flow dictionary entry berisi hanya *summary* data atau data ringkasan, dan menerangkan alur yang mengidentifikasikan dari mana alur itu berasal dan kemana alur itu menuju.

2.2.1.5.5.2. Data Store Dictionary Entry

Data store dictionary entry menerangkan setiap data store yang unik dalam DFD. Jika data store yang sama muncul lebih dari satu, maka hanya satu bentuk tunggal yang akan digunakan. Seperti halnya data *flow dictionary entry*, *data store dictionary entry* hanya berisi *summary* data.

2.2.1.5.5.3. Data Structure Dictionary Entry

Data structure dictionary entry ini dilengkapi dengan setiap struktur yang ada pada bentuk *data store* dan *data flow*. Tujuan dari *data structure dictionary entry* adalah untuk menghubungkan *summary description* (deskripsi ringkasan) dari *data flow* dan *data strore dictionary entry* ke deskripsi detail dari *data element dictionary entry*.

2.2.1.5.5.4. Data Element Dictionary Entry

Data element dictionary entry menyediakan dasar untuk skema database. Bentuk ini menyediakan *data element dictionary* (DED) dari kamus data yang berdasarkan komputer. Bentuk elemen data digunakan oleh setiap elemen data, termasuk semua struktur, baik yang ada pada *data flow* maupun *data store*. Hanya bentuk tunggal yang digunakan untuk masing-masing elemen data, walaupun elemen data itu muncul beberapa kali di dalam sistem.

Tujuan dari *data element dictionary entry* adalah untuk menstandarkan deskripsi dari suatu elemen sehingga elemen itu direferensikan dengan cara yang sama setiap kali digunakan. Hal ini sangat penting, khususnya jika suatu sistem dikembangkan dan dimaintain oleh sekelompok user dan *information specialists.* Jadi mereka dapat menggunakan istilah yang sama untuk satu elemen yang sama pula, dan tidak akan ada penggunaan istilah yang berbeda untuk elemen yang sama.

2.2.1.5.6. Struktur Menu

Struktur menu merupakan gambaran mengenai struktur menu program yang akan dibuat yang digambarkan dalam bentuk diagram. Setiap menu dalam program dijelaskan satu persatu sesuai dengan penempatan dan pengelompokkannya.

2.2.2. Konsep Web

2.2.2.1. Internet

Internet berasal dari kata *interconnected network* yang mempunyai arti hubungan berbagai komputer dan berbagai tipe komputer yang membentuk sistem jaringan yang mencakup seluruh dunia. Dimana antara satu komputer dengan komputer lain di dunia (*world wide*) dapat saling berhubungan atau berkomunikasi. Menurut Mico Pardosi dalam bukunya *Pengenalan Internet Burst of Energy*, Internet adalah suatu alat komunikasi, sumber informasi, serta alat hiburan. Internet adalah jaringan luas dari computer, yang lazim disebut *worldwide network*[1].

Asal usul Internet dari jaringan komputer yang disusun oleh APRA (*Advance Research Project Agency*) yang bernaung dibawah departement pertahanan Amerika Serikat atau DoD (*Departemen of Defense*) sehingga terbentuk ARPAnet, jaringan komputer bentukan departemen pertahanan Amerika Serikat, tahun 70-an.

Pada awalnya, ARPAnet hanya menghubungkan empat buah situs aja, yaitu:

1. SRI (Stanford Research Institute).
2. UCSB (University of California at Santa Barbara).
3. UCLA (University of California at Los Angles).
4. University of Utah at Utah.

Jaringan tersebut selanjutnya diperbaharui dan dikembangkan, dan saat ini digunakan sebagai tulang punggung jaringan informasi yang disebut dengan Internet. Penelitian di Stanford menghasilkan sebuah protokol yang

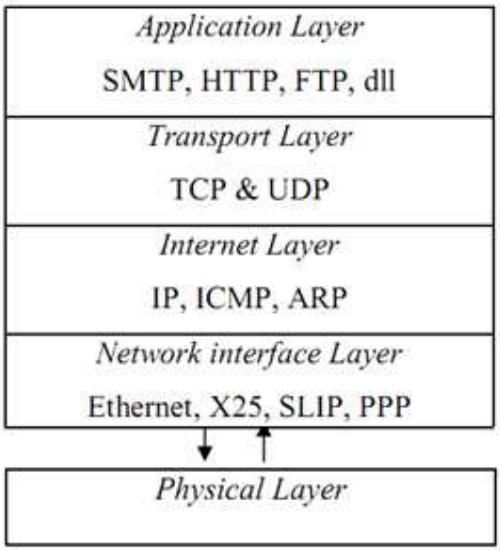
disebut TCP/IP, inilah yang berkembang terus hingga sekarang dan menjadi protocol standar dalam Internet.

2.2.2.2. Protokol TCP/IP

Protokol yang digunakan sebagai format transmisi data pada jaringan Internet adalah TCP/IP. TCP/IP berasal dari dua protokol yang sangat penting yaitu TCP (*Transmission Control Protokol*) dan IP(*Internet Protokol*). Sebagai gambaran , bila anda mengirim e-mail ke seorang teman di negara lain, TCP akan membagi pesan tersebut ke dalam paket data yang kecil-kecil. Setiap paket data akan ditandai dengan nomor urut dan alamat tujuan. Selain itu, TCP akan menyertakan informasi untuk mengontrol jika terjadi kesalahan transmisi.

Selanjutnya, paket-paket data dikirim melalui jaringan komputer, dimana dalam tahap ini IP membawa paket-paket data ke alamat tujuan. Pada alamat tujuan TCP menerima paket-paket data, memeriksa kesalahan, serta menyusun paket-paket data tersebut menjadi pesan asli yang dikirimkan.

Arsitektur protokol TCP/IP ini dapat dimodelkan dengan empat layer sebagaimana yang dapat dilihat pada gambar 2.9 dibawah ini:



Gambar 2.9 *Arsitektur Protokol TCP/IP*

TCP/IP terdiri atas empat lapis kumpulan protokol yang bertingkat. Empat tingkat itu adalah :

1. *Aplication layer*, yaitu layer yang bertanggung jawab atas aplikasi aplikasi yang menggunakan TCP/IP.
2. *Transport layer*, yaitu layer yang bertanggung jawab memberikan layanan pengiriman data dari satu node ke node yang lain.
3. *Internet layer*, yaitu layer yang menentukan jenis data gram dan menangani perutean data yang dikirim.
4. *Network interface layer*, yaitu layer yang bertanggung jawab untuk mengirim dan menerima data pada media fisik transmisi yang digunakan.

2.2.2.3. *World Wide Web* (WWW)

WWW atau yang lebih dikenal dengan nama *World Wide Web* atau disingkat web merupakan salah satu layanan yang didapat oleh pemakai

komputer yang terhubung ke Internet. Web pada awalnya adalah ruang informasi dalam Internet dengan menggunakan teknologi hypertext. Pemakai dituntun untuk menemukan informasi dengan mengikuti link yang disediakan dalam dokumen web yang ditampilkan dalam web browser[2].

Sejarah lahirnya Web dimulai ketika tahun 1993, Tim Berners-Lee dan peneliti lain di European Particle Physics Lab (CERN) di Geneva Swiss mengembangkan suatu cara untuk men-*share* data antar koleganya dengan menggunakan sesuatu yang disebut dengan *hypertext*. Pemakai di CERN dapat menyajikan dokumen pada layar komputer dengan software terbaru. Kode-kode khusus disisipkan ke dalam dokumen elektronik tersebut, memungkinkan pemakai untuk meloncat dari satu dokumen ke dokumen yang lain pada layar hanya dengan memilih sebuah *hyperlink*. Kemampuan Internet dimasukkan ke dalam browser ini sehingga proses tersebut tidak hanya terjadi pada satu komputer saja, tapi yang letaknya pada komputer remote.

Web bekerja sesuai dengan proses berikut:

1. Informasi web disimpan dalam dokumen yang disebut dengan halaman-halaman web (*Web Pages*).
2. *Web Pages* adalah file-file yang disimpan dalam komputer yang disebut dengan server-server web (*Web Server*).
3. Komputer-komputer tersebut membaca web pages disebut komputer client.

4. Komputer-komputer tersebut menampilkan web pages menggunakan browser.

2.2.2.4. Aplikasi Web

Aplikasi Web adalah sekumpulan halaman web yang mampu berinteraksi dengan pengunjung, dengan sesama halaman web dan dengan berbagai sumber data yang ada diserver web. Isi dari suatu halaman aplikasi web ini berbeda dari satu pengunjung yang lain. Ini terjadi karena sisi halaman aplikasi web ditentukan oleh permintaan dari pengunjung yang tidak sama. Permintaan ini diproses oleh server dan kemudian dikirimkan ke browser. Oleh karena halaman-halaman aplikasi web lebih banyak dikerjakan di sisi server, maka teknologi ini sering juga disebut dengan web dinamis berbasiskan server. Sebagai contoh dari aplikasi web adalah PHP-Nuke, Mambo/Joomla, Wordpress, phpMyAdmin dan lain-lain.

2.2.2.5. Web Server

Web server adalah sebuah perangkat lunak server yang berfungsi menerima permintaan HTTP atau HTTPS dari client yang dikenal dengan browser web dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman web yang umumnya berbentuk dokumen HTML. Server web yang terkenal diantaranya adalah Apache dan *Microsoft Internet Information Service* (*IIS*). Apache merupakan server web antarplatform (perangkat lunak yang dapat digunakan di beberapa sistem operasi yang berbeda), sedangkan IIS hanya dapat beroperasi di sistem operasi Windows. Server web juga dapat berarti komputer yang berfungsi seperti definisi di atas.

2.2.2.6. Apache

Server HTTP Apache atau Server Web/WWW Apache adalah server web yang dapat dijalankan di banyak sistem operasi (Unix, BSD, Linux, Microsoft Windows dan Novell Netware serta platform lainnya) yang berguna untuk melayani dan memfungsikan situs web. Protokol yang digunakan untuk melayani fasilitas web/www ini mengunakan HTTP.

Apache memiliki fitur-fitur canggih seperti pesan kesalahan yang dapat dikonfigurasi, autentikasi berbasis basis data dan lain-lain. Apache juga didukung oleh sejumlah antarmuka pengguna berbasis grafik (GUI, *Grafik User Interface*) yang memungkinkan penanganan server menjadi mudah. Apache merupakan perangkat lunak kode sumber program terbuka dikembangkan oleh komunitas terbuka yang terdiri dari pengembang-pengembang dibawah naungan *Apache Software Foundation*.

Apache adalah komponen server web dari paket perangkat lunak LAMP (Linux, Apache, MySQL, PHP/Perl/bahasa pemrograman Python). Tetapi pada pengembangannya sekarang ini Apache dapat di install dan dijalankan di berbagai macam platform, seperti Unix, BSD, Microsoft Windows dan Novell Netware serta platform lainnya.

Pada awal mulanya, Apache merupakan perangkat lunak sumber terbuka yang menjadi alternatif dari server web Netscape (sekarang dikenal sebagai *Sun Java System Web Server*).

Asal mula nama Apache berasal ketika sebuah server web populer yang dikembangkan pada awal 1995 yang bernama NCSA HTTPd 1.3 memiliki

sejumlah perubahan besar terhadap kode sumbernya (*patch*). Dikarenakan banyaknya *patch* pada perangkat lunak tersebut sehingga disebut sebuah server yang memiliki banyak *patch* ("*a patchy*" server). Tetapi pada halaman FAQ situs web resminya, disebutkan bahwa "*Apache*" dipilih untuk menghormati suku asli Indian Amerika Apache (Indé), yang dikenal karena keahlian dan strategi perangnya. Versi 2 dari Apache ditulis dari awal tanpa mengandung kode sumber dari NCSA.

2.2.2.7. Web Browser

Web browser adalah suatu aplikasi yang dapat menampilkan dan berinteraksi dengan teks, gambar, atau yang lainnya dimana teks atau gambar tadi yang terletak pada suatu halaman situs yang di tempatkan di web server dan terhubung ke Internet atau suatu jaringan lokal (LAN, *Local Area Network*). Halaman-halaman situs inipun akan terhubung dengan halaman-halaman lainnya dari server yang lain juga melalui *hyperlink*.

Web browser memperbolehkan pengguna dengan cepat mengakses informasi antara satu server dengan server lainnya melalui *hyperlink* ini. Web browser berkomunikasi dengan web server dengan menggunakan protokol HTTP. Contoh web browser yang popular antara lain Mozilla Firefox, Intenet Explorer(IE), Opera dan sebagainya.

2.2.2.8. *Hypertext Transfer Protokol* (HTTP)

HTTP (*Hypertext Transfer Protokol*) adalah suatu protocol yang menentukan aturan yang perlu diikuti oleh web browser dalam meminta atau mengambil suatu dokumen dan oleh web server dalam menyediakan

dokumen yang diminta oleh web server. Protokol ini merupakan protokol standar yang digunakan dalam mengakses dokumen HTML[4].

2.2.2.9. *Uniform Resource Locator* (URL)

URL (*Uniform Resource Locator*) adalah suatu sarana yang digunakan untuk menentukan lokasi informasi pada web server[4]. URL dapat diibaratkan sebuah alamat, dimana alamat tersebut terdiri dari[4]:

1. Protokol yang digunakan oleh suatu browser untuk mengambil informasi.
2. Nama komputer (Server) dimana informasi tersebut berasal.
3. Jalur/path serta nama file dari suatu informasi.

Contoh URL adalah http://www.amazon.com/buku/index.php[4].

2.2.2.10. Hypertext Markup Language (HTML)

HTML (*Hypertext Markup Language*) adalah sebuah bahasa yang digunakan untuk menulis halaman web. HTML dirancang untuk digunakan tanpa tergantung pada suatu platform tertentu (*Platform Independent*). Dokumen HTML adalah suatu dokumen teks biasa, dan disebut sebagai *Markup Language* karena mengandung tanda-tanda (tag) tertentu yang digunakan untuk menentukan tampilan suatu teks dan tingkat kepentingan dari teks tersebut dalam suatu dokumen.

2.2.2.11. Cascading Style Sheet (CSS)

CSS (*Cascading Style Sheet*) merupakan salah satu bahasa pemrograman web untuk mengendalikan beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam.Sama halnya *styles* dalam

aplikasi pengolahan kata seperti Microsoft Word yang dapat mengatur beberapa *style*, misalnya *heading*, *subbab*, *bodytext*, *footer*, *images*, dan *style* lainnya untuk dapat digunakan bersama-sama dalam beberapa berkas (*file*). Pada umumnya CSS dipakai untuk memformat tampilan halaman web yang dibuat dengan bahasa HTML dan XHTML.

CSS dapat mengendalikan ukuran gambar, warna bagian tubuh pada teks, warna tabel, ukuran border, warna border, warna hyperlink, warna *mouse over*, spasi antar paragraf, spasi antar teks, margin dan parameter lainnya. CSS adalah bahasa *style sheet* yang digunakan untuk mengatur tampilan dokumen. Dengan adanya CSS memungkinkan kita untuk menampilkan halaman yang sama dengan format yang berbeda.

CSS dirancang terutama untuk memungkinkan pemisahan konten dokumen (yang ditulis dalam HTML atau sebuah bahasa markup yang sama) dari presentasi dokumen Pemisahan ini dapat meningkatkan konten aksesibilitas, menyediakan lebih fleksibilitas dan kontrol dalam spesifikasi karakteristik presentasi, memungkinkan beberapa halaman untuk berbagi format, dan mengurangi kompleksitas dan pengulangan isi struktural (seperti dengan memungkinkan untuk desain *web Tableless*).

2.2.2.12. Javascript

JavaScript adalah prototipe berbasis *object-oriented* bahasa *scripting* yang digunakan untuk mengaktifkan program akses ke objek-objek komputasi dalam lingkungan host. Meskipun juga digunakan di lain aplikasi, ini terutama digunakan dalam bentuk JavaScript sisi klien, yang

dilaksanakan sebagai bagian dari web browser, memberikan peningkatan *user interface* dan situs web dinamis.

JavaScript mendukung semua sintaks pemrograman terstruktur di bahasa C (misalnya, *if else, while loop, switch,* dll). Javascript juga hampir seluruhnya berbasis objek. Penggunaan utama dari JavaScript untuk menulis fungsi yang tertanam di dalam atau termasuk dari HTML halaman dan berinteraksi dengan *Document Object Model* (DOM) halaman.

Karena kode JavaScript dapat berjalan secara lokal dalam browser pengguna (bukan di server jauh) dapat menanggapi tindakan pengguna dengan cepat, membuat aplikasi merasa lebih responsif.

2.2.2.13. Jquery

JQuery adalah librari JavaScript yang memungkinkan kita untuk membuat program web pada suatu halaman web, tanpa harus secara eksplisit kita menambahkan event atau pun properti pada halaman web tersebut.

Dengan JQuery, suatu halaman web yang menjadi aplikasi web, jika dilihat sourcenya, akan terlihat seperti dokumen HTML biasa; tidak ada kode JavaScript yang terlihat langsung. Teknik pemrograman web seperti ini disebut sebagai *unobstrusive JavaScript programming.*

JQuery merupakan salah satu librari yang membuat program web di sisi klien, tidak terlihat sebagai program JavaScript biasa, yang harus secara eksplisit disisipkan pada dokumen web. Pada teknik pemrograman sisi klien dengan menggunakan JavaScript biasa, setiap elemen yang akan memiliki

event, akan secara eksplisit terlihat ada event yang dilekatkan pada elemen tersebut.

JQuery dikembangkan pertama kali oleh John Resig, yang dibuat lebih ramping dari librari Prototype yang menjadi inspirasi dari librari JQuery ini. Secara pemrograman, JQuery memiliki kemiripan seperti Prototype.

2.2.2.14. Ajax

Asynchronous JavaScript and XMLHTTP, atau disingkat AJaX, adalah suatu teknik pemrograman berbasis web untuk menciptakan aplikasi web interaktif. Tujuannya adalah untuk memindahkan sebagian besar interaksi pada komputer web surfer, melakukan pertukaran data dengan server di belakang layar, sehingga halaman web tidak harus dibaca ulang secara keseluruhan setiap kali seorang pengguna melakukan perubahan. Hal ini akan meningkatkan interaktivitas, kecepatan, dan usability.

Ajax merupakan kombinasi dari:

1. DOM yang diakses dengan client side scripting language, seperti VBScript dan implementasi ECMAScript seperti JavaScript dan JScript, untuk menampilkan secara dinamis dan berinteraksi dengan informasi yang ditampilkan.
2. Objek XMLHTTP dari Microsoft atau XMLHttpRequest yang lebih umum di implementasikan pada beberapa browser. Objek ini berguna sebagai kendaraan pertukaran data asinkronus dengan web server.
3. XML umumnya digunakan sebagai dokumen transfer, walaupun format lain juga memungkinkan, seperti HTML, plain text. XML dianjurkan

dalam pemakaian teknik AJaX karena kemudahan akses penanganannya dengan memakai DOM.

4. JSON dapat menjadi pilihan alternatif sebagai dokumen transfer, mengingat JSON adalah JavaScript itu sendiri sehingga penanganannya lebih mudah.

2.2.2.15. Hypertext Preprosessor (PHP)

PHP adalah suatu bahasa Server-side yang didesain khusus untuk aplikasi web. PHP dapat disisipkan diantara bahasa HTML dan karena bahasa *server-side*, maka bahasa PHP akan dieksekusi di server, sehingga yang dikirimkan ke browser adalah "hasil jadi" dalam bentuk HTML, dan kode PHP-nya tidak akan terlihat.

Kelebihan menggunakan PHP antara lain:

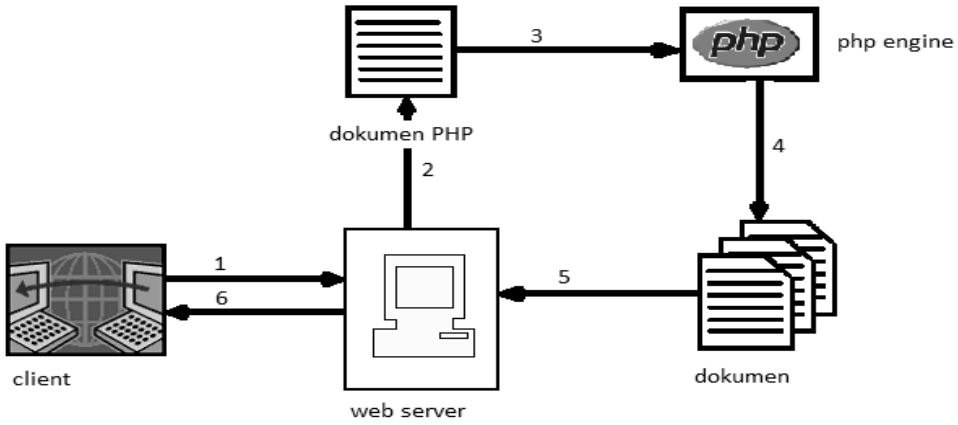
1. PHP mudah dibuat dan memiliki kecepatan akses yang tinggi.
2. PHP dapat berjalan dalam web server yang berbeda dan dalam sistem operasi yang berbeda pula.
3. PHP diterbitkan secara gratis.
4. PHP dapat berjalan pada Microsoft Personal Web Server, Apache, IIS, Xitami dsb.
5. PHP termasuk bahasa server-side programming.

Sedangkan sistem Database yang didukung oleh PHP antara lain:

1. Oracle
2. Sybase
3. mSQL

4. MySQL
5. Solid
6. Generic ODBC
7. Postgres SQL

Gambaran mengenai bagaimana skrip PHP dieksekusi oleh web server sehingga menjadi sebuah teks HTML dapat dilihat pada gambar 2.7 sebagai berikut:



Gambar 2.10 *Eksekusi PHP*

Keterangan :

1. User meminta request via web browser ke web server.
2. Web server akan mengenalinya sebagai file PHP (.PHP, .PHP3, .phtml).
3. File PHP dikirim ke *engine* PHP.
4. Output (hasil) dari proses di dalam *engine* PHP dalam bentuk plain HTML (HTML biasa).
5. Dokumen hasil eksekusi dikirim balik ke web server.

6. Web server meneruskan ke web browser sebagai respon dari request untuk ditampilkan.

2.2.2.16. Macromedia Dreamweaver

Macromedia dreamweaver adalah sebuah HTML editor profesional untuk mendesain secara visual dan mengelola situs web maupun halaman web. Bilamana kita menyukai untuk berurusan dengan kode-kode HTML secara manual atau lebih menyukai bekerja dengan lingkungan secara visual dalam melakukan editing, dreamweaver mambuatnya menjadi lebih mudah dengan menyediakan tool-tool yang sangat berguna dalam peningkatan kemampuan dan pengalaman kita dalam mendesain web.

Dreamweaver dalam hal ini digunakan untuk web desain. Dreamweaver mengikutsertakan banyak tool untuk kode-kode dalam halaman web beserta fasilitas-fasilitasnya, antara lain : Referensi HTML, CSS dan Javascript, Javascript debugger, dan editor kode (tampilan kode dan *code inspector*) yang mengizinkan kita mengedit kode Javascript, XML, dan dokumen teks lain secara langsung dalam dreamweaver. Teknologi dreamweaver *roundtrip* HTML mampu mengimpor dokumen HTML tanpa perlu memformat ulang kode tersebut dan kita dapat menggunakan dreamweaver pula untuk membersihkan dan memformat ulang HTML bila kita menginginkannya.

Selain itu dreamweaver juga dilengkapi kemampuan manajemen situs, yang memudahkan kita mengelola keseluruhan elemen yang ada dalam situs. Kita juga dapat melakukan evaluasi situs dengan melakukan

pengecekan broken link, kompatibilitas browser, maupun perkiraan waktu download halaman web.

2.2.3. *E-Commerce*

E-Commerce dapat berarti pemasangan iklan, penjualan, dan dukungan dan pelayanan terbaik menggunakan sebuah web shop 24 jam sehari bagi seluruh pelanggan.

Kalakota dan Whinston (1997) mendefinisikan *E-Commerce* dari beberapa perspektif berikut :

1. Dari perspektif komunikasi, *E-Commerce* merupakan pengiriman informasi, produk/layanan, atau pembayaran melalui lini telepon, jaringan komputer atau sarana elektronik lainnya.
2. Dari perspektif proses bisnis, *E-Commerce* merupakan aplikasi teknologi menuju otomatisasi transaksi dan aliran kerja perusahaan.
3. Dari perspektif layanan, *E-Commerce* merupakan satu alat yang memenuhi keinginan perusahaan, konsumen, dan manajemen dalam memangkas service cost ketika meningkatkan mutu barang dan kecepatan pelayanan.
4. Dari perspektif online, *E-Commerce* kapasitas jual beli produk dan informasi di Internet dan jasa online lainnya.

2.2.3.1. Jenis *E-Commerce*

1. ***Business-to-business*** **(B2B**).

Kebanyakan *E-Commerce* yang diterapkan saat ini merupakan tipe B2B. *E-Commerce* tipe ini meliputi transaksi IOS yang digambarkan tadi

serta transaksi antar organisasi yang dilakukan di *electronic market*. Contohnya Wal-Mart dengan Warner-Lambert.

2. *Business-to-consumer* (B2C).

Ini merupakan transaksi eceran dengan pembeli perorangan. Pembeli khas di Amazon.com adalah seorang konsumen, atau seorang pelanggan. Contoh yang lain, misalnya Barnes & Nobles, Cisco, Dell, Compaq dan sebagainya.

3. *Consumer-to-business* (C2B).

Termasuk ke dalam kategori ini adalah perseorangan yang menjual produk-produk atau layanan ke organisasi, dan perseorangan yang mencari penjual, berinteraksi dengan mereka, dan menyepakati suatu transaksi.

4. *Consumer-to-consumer* (C2C).

Dalam kategori ini, seorang konsumen menjual secara langsung ke konsumen lainnya. Contohnya adalah ketika ada perorangan yang melakukan penjualan di *classified ads* (misalnya, www.classified2000.com) dan menjual properti rumah hunian, mobil, dan sebagainya. Mengiklankan jasa pribadi di Internet serta menjual pengetahuan dan keahlian merupakan contoh lain C2C. sejumlah situs pelelangan memungkinkan perorangan untuk memasukkan item-item agar disertakan dalam pelelangan. Akhirnya, banyak perseorangan yang menggunakan intranet dan jaringan organisasi untuk mengiklankan item-item yang akan dijual atau juga menawarkan aneka jasa. Contoh lain yang terkenal adalah eBay.com, yaitu perusahaan lelang.

2.2.3.2. Keuntungan *E-Commerce*

Keuntungan dari *e-commerce*:

1. Aliran pendapatan (*Revenue stream*) yang lebih menjanjikan.
2. Dapat meningkatkan pangsa pasar (*market exposure*).
3. Menurunkan biaya operasional (*operating cost*).
4. Melebarkan jangkauan (*global reach*).
5. Meningkatkan *customer loyality*.
6. Meningkatkan supplier management.
7. Memperpendek waktu produksi.
8. Meningkatkan mata rantai pendapatan (*value chain*).

2.2.3.3. Resiko dan Kerugian *E-Commerce*

Resiko dan kerugian dari *E-Commerce*:

1. Kehilangan segi financial secara langsung karena kecurangan.
2. Pencurian informasi rahasia yang berharga.
3. Kehilangan kesempatan bisnis karena gangguan pelayanan.
4. Penggunaan akses ke sumber oleh pihak yang tidak berhak.
5. Kehilangan kepercayaan dari para konsumen.
6. Kerugian-kerugian yang tidak terduga.

2.2.3.4. *Secure Socket Layer* (SSL)

Seperti dikemukakan di awal, *E-Commerce* banyak menggunakan teknologi Internet. Salah satu teknologi yang digunakan adalah standar TCP/IP dengan menggunakan socket. Untuk meningkatkan keamanan informasi keamanan layer socket perlu ditingkatkan dengan

menggunakan teknologi kriptografi. Netscape mengusulkan pengamanan dengan menggunakan *Secure Socket Layer* (SSL) ini.

SSL adalah Protokol berlapis. Dalam tiap lapisannya, sebuah data terdiri dari panjang, deskripsi dan isi. SSL mengambil data untuk dikirimkan, dipecahkan kedalam blok-blok yang teratur, kemudian dikompres jika perlu, menerapkan MAC, dienkripsi, dan hasilnya dikirimkan. Di tempat tujuan, data didekripsi, verifikasi, dekompres, dan disusun kembali. Hasilnya dikirimkan ke klien di atasnya.

SSL hanya mengenkripsikan data yang dikirim lewat http. Cara kerja dari SSL adalah sebgai berikut:

1. Pada saat koneksi mulai berjalan, klien dan server membuat dan mempertukarkan kunci rahasia, yang dipergunakan untuk mengenkripsi data yang akan dikomunikasikan. Meskipun sesi antara klien dan server diintip pihak lain, namun data yang terlihat sulit untuk dibaca karena sudah dienkripsi.
2. SSL mendukung kriptografi public key, sehingga server dapat melakukan autentikasi dengan metode yang sudah dikenal umum seperti RSA dan Digital Signature Standard (DSS).
3. SSL dapat melakukan verifikasi integritas sesi yang sedang berjalan dengan menggunakan algoritma digest seperti MD5 dan SHA. Hal ini menghindarkan pembajakan suatu sesi.

2.2.3.5. Paypal

Paypal adalah salah satu alat pembayaran (*Payment procesors*) menggunakan Internet yang terbanyak digunakan didunia dan teraman. Pengguna Internet dapat memberli barang die bay, lisensi software original, keanggotaan situs, urusan bisnis, mengirim uang ke pengguna Paypal lain diseluruh dunia dan banyak fungsi lainnya dengan mudah dan otomatis menggunakan Internet, Paypal mengatasi kekurangan dalam pengiriman uang tradisional seperti cek atau Money order yang prosesnya dapat memakan waktu.

Februari 2010 PayPal sudah menerima 190 negara dan 23 mata uang, namun pengguna PayPal di Indonesia masih harus menggunakan hitungan US Dollar karena Rupiah belum ada di PayPal. Dana yang ditarik langsung dari kartu kredit akan dikonversi ke US dollar otomatis.

Jenis-jenis akun pada PayPal terdapat 3 jenis, yakni:

1. Akun Personal

   Dengan akun tipe ini Anda sudah dapat mengirim dan menerima uang dan melakukan penjualan dengan eBay. Pada tipe ini, Anda bisa menerima pembayaran dari akun Paypal lain, namun tidak bisa menerima pembayaran dari credit atau debit card. Tidak ada biaya untuk setiap transaksi yang Anda lakukan pada tipe ini. Ada limit berapa banyak uang yang dapat Anda terima per bulannya. Jika Anda merencanakan ingin berjualan produk dalam jumlah yang besar, tipe personal tidak cocok buat Anda.

2. Akun Premier

Tipe ini hampir sama dengan akun Personal, bedanya pada tipe ini Anda bisa menerima pembayaran dari credit card, debit card dan rekening bank. Anda juga bisa menggunakan fasilitas shopping cart dan tool laporan pembayaran (payment reporting tool) Akun Premier cocok digunakan untuk penjual amatiran yang ingin menjual produknya secara reguler. Jika saat ini Anda memiliki akun personal, Anda dapat melakukan upgrade ke akun Premier.

3. Akun Business

Tipe Akun Business cocok digunakan untuk bisnis yang berskala besar atau online store. Pada tipe ini Anda diperbolehkan menggunakannya pada nama bisnis Anda, dan menggunakan laporan dan tool eBay tanpa adanya persyaratan mengenai jenis transaksi. Anda akan dikenakan biaya dalam menggunakan akun tipe ini. Jika saat ini Anda memiliki akun Personal atau Premier, Anda dapat melakukan upgrade ke akun Business.

2.2.4. Konsep Basis Data

Basis data terdiri dari 2 kata, yaitu basis dan data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang/berkumpul[3]. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek sperti manusia (pegawai,siswa,pembeli,pelanggan), barang, hewan, peristiwa,konsep, keadaan dan sebagainya, yang direkam dalam bentuk angka, huruf, symbol, teks, gambar, bunyi, atau kombinasinya[3].

Basis data sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti[3]:

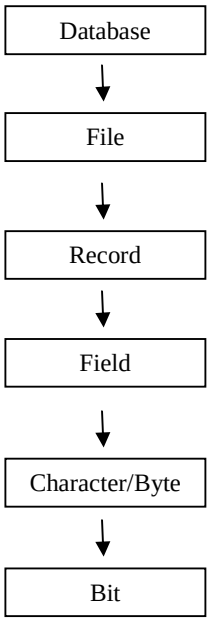
1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redundansasi*) yang tidak perlu,untuk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

Kumpulan dari item data yang saling berhubungan satu dengan yang lainnya yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, tersimpan di hardware komputer dan dengan software untuk melakukan manipulasi untuk kegunaan tertentu.

Jenjang Data dari basis data sebagai berikut :

1. *Database* : Kumpulan dari file / tabel membentuk suatu *database*.
2. *File*: *File* terdiri dari *record-record* yang menggambarkan satu kesatuan data yang sejenis. Misalnya file mata pelajaran berisi data tentang semua mata pelajaran yang ada.
3. *Record*: Kumpulan dari *field* membentuk suatu *record*. *Record* menggambarkan suatu unit data individu yang tertentu. Kumpulan dari *record* membentuk suatu *file*. Misalnya *file* personalia, tiap-tiap *record* dapat mewakili data tiap-tiap karyawan.

4. *Field* merepresentasikan suatu atribut dari *record* yang menunjukkan suatu item dari data, seperti misalnya nama, alamat dan lain sebagainya. Kumpulan dari field membentuk suatu *record*.
    a. *field name*: harus diberi nama untuk membedakan *field* yang satu dengan lainnya
    b. *field representation*: tipe *field* (karakter, teks, tanggal, angka, dsb), lebar *field* (ruang maksimum yang dapat diisi dengan karakter-karakter data).
    c. *field value* : isi dari *field* untuk masing-masing *record*.
5. *Characters* merupakan bagian data yang terkecil, dapat berupa karakter numerik, huruf ataupun karakter-karakter khusus (spesial karakter) yang membentuk suatu item data / *field*.



Gambar 2.11 *Hirarki Basis Data*

2.2.4.1. Sistem Basis Data

Sistem basis data adalah suatu sistem penyusunan dan pengelolaan *record-record* dengan menggunakan komputer, dengan tujuan untuk menyimpan atau merekam serta memelihara data operasional lengkap sebuah organisasi/perusahaan, sehingga mampu menyediakan informasi yang optimal yang diperlukan pemakai untuk kepentingan proses pengambilan keputusan.

2.2.4.2. *Entity Relationship Diagram* (ERD)

ERD atau *Entity Relationship Diagram* merupakan sebuah diagram yang digunakan untuk menggambarkan ERM (*Entity Relational Model*). ERM merupakan suatu model data yang digunakan untuk menjelaskan hubungan antar data dalam basis data kepada pemakai secara *logic*. Dengan kata lain, ERD adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem.

Bagi perancang, ERD berguna untuk memodelkan sistem yang nantinya akan dikembangkan basis datanya. Model ini juga membantu perancang basis data pada saat melakukan analisis dan perancangan basis data karena model ini dapat menunjukan macam-macam data yang dibutuhkan dan kerelasian antar data di dalamnya. Bagi pemakai, model ini sangat membantu dalam hal pemahaman model sistem dan rancangan basis data yang akan dikembangkan oleh perancang basis data.

Elemen-elemen yang ada dalam ERD ini diantaranya sebagai berikut:

1. *Entity*

Pada ERD, *entity* digambarkan dengan bentuk persegi panjang. *Entity* adalah sesuatu apa saja yang ada dalam sistem, nyata maupun abstrak dimana data tersimpan atau dimana terdapat data. *Entity* diberi nama dengan kata benda dan dapat dikelompokan dalam empat jenis nama yaitu, orang, benda, lokasi, dan kejadian.

2. Relationship

Pada ERD, *relationship* digambarkan dengan sebuah belah ketupat. *Relationship* adalah hubungan alamiah yang terjadi antara *entity*. Pada umumnya, relationship diberi nama dengan kata kerja.

3. Relationship Degree

Relationship degree adalah jumlah entitas yang berpartisipasi dalam suatu *relationship*. *Relationship* yang sering digunakan di ERD adalah sebagai berikut:

a. *Unary Relationship*

Unary relationship adalah model *relationship* yang terjadi antara *entity* yang berasal dari *entity* set yang sama. Model ini biasanya disebut *Recursive Relationship* atau *Reflective Relationship*.

b. *Binary Relationship*

Binary Relationship adalah model *relationship* antara *instance-instance* dari suatu tipe *entity* (dua *entity* yang berasal dari *entity* yang sama).

c. *Ternary Relationship*

Ternary Relationship adalah *relationship* antara *instance-instance* dari tiga tipe entitas secara serentak.

4. Atribut

Atribut adalah sifat atau karakteristik dari tiap entitas atau tiap relationship. Maksudnya, atribut adalah sesuatu yang menjelaskan apa yang sebenarnya dimaksud entitas maupun *relationship* sehingga sering dikatakan bahwa atribut adalah elemen dari setiap entitas dan *relationship*.

2.2.4.2.1. Kardinalitas Relasi

Dalam ERD, hubungan (relasi) dapat terdiri dari sejumlah entitas yang disebut derajat relasi. Derajat relasi maksimum disebut dengan kardinalitas, sedangkan derajat minimum disebut dengan modalitas. Jadi kardinalitas relasi menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas lain. Kardinalitas relasi yang terjadi diantara dua himpunan entitas (misalnya A dan B) dapat berupa;

1. Satu ke satu (*One to one* / 1-1)

Setiap entitas pada himpunan entitas A dapat berelasi dengan paling banyak 1 (satu) entitas pada himpunan entitas B, demikian juga sebaliknya. Contoh pria menikahi wanita (asumsi tidak ada poligami).

2. Satu ke banyak (*One to many* / 1-N)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi tidak sebaliknya, di mana

setiap entitas pada himpunan entitas B berhubungan dengan 30 paling banyak dengan satu entitas pada himpunan entitas A. Contoh ibu mempunyai anak.

3. Banyak ke banyak (*Many to many* / N-N)

Setiap entitas pada himpunan entitas A dapat berelasi dengan banyak entitas pada himpunan entitas B, demikian juga sebaliknya.

2.2.4.2.2. Tahap Pembuatan ERD

Diagram ER dibuat secara bertahap, ada dua kelompok pentahapan yang biasa ditempuh didalam pembuatan diagram ER, yaitu;

1. Tahap pembuatan diagram ER awal (*Preliminary design*).
2. Tahap optimasi digram ER (*Final design*).

Tujuan dari tahap pertama adalah untuk mendapatkan sebuah rancangan basis data minimal yang dapat mengakomodasi kebutuhan penyimpanan data terhadap sistem yang sedang ditinjau. Tahap awal ini umumnya mengabaikan anomali-anomali (proses pada basis data yang memberikan efek samping yang tidak diharapkan) yang memang ada sebagai suatu fakta. Anomali-anomali tersebut biasanya baru dipertimbangkan pada tahap kedua.

Tahap kedua mempertimbangkan anomali-anomali dan juga memperhatikan aspek-aspek efisiensi, performasi dan fleksibilitas. Tiga hal tersebut seringkali dapat saling bertolak belakang. Karena itu, tahap kedua ini ditempuh dengan melakukan koreksi terhadap tahap pertama. Bentuk koreksi yang terjadi dapat berupa pendekomposisian himpunan

entitas, penggabungan himpunan entitas, pengubahan derajat relasi, penambahan relasi baru atau perubahan (penambahan dan pengurangan) atribut-atribut untuk masing-masing entitas dan relasi.

Langkah-langkah teknis yang dapat dilakukan untuk mendapatkan ERD awal adalah:

1. Mengidentifikasi dan menetapkan seluruh himpunan entitas yang akan terlibat.
2. Menentukan atribut-atribut kunci (*key*) dari masing-masing himpunan entitas.
3. Mengidentifikasi dan menetapkan seluruh himpunan relasi diantara himpunan entitas-himpunan entitas yang ada beserta kunci tamu (*foreign key* / kunci asing).
4. Menentukan derajat / kardinalitas relasi untuk setiap himpunan relasi.
5. Melengkapi himpunan entitas dan himpunan relasi dengan atribut dekriptif (atribut yang bukan kunci).

2.2.4.3. *Structured Query Language* (SQL)

Antar muka standar untuk sistem manajemen basis data relasional, termasuk sistem yang beroperasi pada komputer pribadi. SQL memungkinkan seorang pengguna untuk mengakses informasi tanpa mengetahui dimana lokasinya atau bagaimana informasi tersebut disusun. SQL lebih mudah untuk digunakan dibandingkan dengan bahasa pemrograman tetapi lebih rumit dibanding software lembar kerja dan pengolah kata.

Sebuah pernyataan SQL yang sederhana dapat menghasilkan set permintaan untuk informasi yang tersimpan pada komputer yang berbeda di berbagai lokasi yang tersebar, sehingga membutuhkan waktu dan sumber daya komputasi yang banyak. SQL dapat digunakan untuk investigasi interaktif atau pembuatan lampiran ad hoc atau disisipkan dalam program aplikasi.

Bahasa pemrograman yang dirancang khusus untuk mengirimkan suatu perintah query (pengaksesan data berdasarkan pengalamatan tertentu) terhadap sebuah database. Kebanyakan software database yang ada saat ini dapat diakses melalui SQL. Setiap aplikasi yang spesifik dapat mengimplementasikan SQL secara sedikit berbeda, tapi seluruh database SQL mendukung subset standar yang ada.

SQL secara garis besar terdiri dari 4 macam, yaitu :

a. *Data Definition Language* (DDL)

DDL merupakan kelompok perintah yang berfungsi untuk mendefinisikan atribut-atribut database, tabel, atribut (kolom), batasan-batasan terhadap suatu atribut serta hubungan antar tabel. Yang termasuk kelompok DDL ini adalah :

i. CREATE untuk menciptakan table ataupun indeks
ii. ALTER untuk mengubah struktur table
iii. DROP untuk menghapus table ataupun indeks

b. *Data Manipulation Language* (DML)

DML adalah kelompok perintah yang berfungsi untuk memanipulasi data, misalnya untuk pengambilan, penyisipan pengubahan dan penghapusan data. Yang termasuk DML adalah :

i. SELECT memilih data
ii. INSERT menambah data
iii. DELETE menghapus data
iv. UPDATE mengubah data

c. *Data Control Language* (DCL)

Berisi perintah-perintah untuk mngendalikan pengaksesan data. Yang termasuk DCL adalah :

i. GRANT memberikan kendali pada pengaksesan data.
ii. REVOKE mencabut kemampuan pengaksesan data
iii. LOCK TABLE mengunci table

d. *Transaction Control Language* (TCL)

TCL adalah perintah-perintah yang berfungsi untuk mengendalikan pengeksekusian transaksi. Yang termasuk kelompok ini adalah :

i. COMMIT menyetujui rangkaian perintah yang berhubungan erat yang telah berhasil dilakukan
ii. ROLLBACK membatalkan transaksi yang dilakukan karena adanya kesalahan atau kegagalan pada salah satu rangkaian perintah.

2.2.4.4. Normalisasi

Proses normalisasi adalah proses untuk memperoleh properti-properti skema relasi yang bagus menjadi bentuk normal lebih tinggi sehingga dapat mengoptimalisasi *redudansi* (pengulangan data yang tidak perlu) dan menghilangkan anomali. Tujuan proses normalisasi adalah mengkonversi relasi menjadi bentuk normal lebih tinggi.

Terdapat beragam tingkat bentuk normal, yaitu :

1. Bentuk normal pertama (1NF)

Bentuk normal pertama adalah ekivalen dengan definisi model relasional. Relasi adalah bentuk normal pertama (1NF) jika semua nilai atributnya adalah sederhana (bukan komposit).

2. Bentuk normal kedua (2NF)

Bentuk normal kedua memiliki ketentuan harus telah berbentuk normal pertama (1NF), dan semua atribut bukan utama harus bergantung fungsional penuh pada kunci relasi. Relasi pada bentuk normal kedua harus tidak menyimpan fakta-fakta mengenai bagian kunci relasi. Bentuk normal kedua menghilangkan kebergantungan parsial dan masih memiliki anomali-anomali yang secara praktis tidak dapat diterima.

3. Bentuk normal ketiga (3NF)

Bentuk normal ketiga memiliki ketentuan harus telah berbentuk normal kedua (2NF) dan relasi tidak boleh memuat kebergantungan fungsional di antara atribut-atribut bukan utama. Bentuk normal ketiga menghilangkan kebergantungan transitif, awalnya bentuk normal ketiga dipikir sebagai

bentuk normal puncak/paling akhir. Namun kemudian dapat ditemukan bentuk normal lebih kuat yaitu Bentuk Normal Boyce-Codd. BCNF memiliki ketentuan yaitu masing-masing atribut utama bergantung fungsional penuh pada masing-masing kunci dimana kunci tersebut bukan bagiannya. Relasi adalah BCNF jika setiap determinan atribut-atribut relasi adalah kunci relasi. Relasi adalah BCNF jika kapanpun fakta-fakta disimpan mengenai beberapa atribut, maka atribut-atribut ini merupakan satu kunci relasi. BCNF dapat memiliki lebih dari satu kunci. Properti penting BCNF adalah relasi tidak memiliki informasi yang redundan.

4. Bentuk normal keempat (4NF)

Relasi dalam bentuk normal keempat (4NF) jika relasi dalam BCNF dan tidak berisi kebergantungan banyak nilai. Untuk menghilangkan kebergantungan banyak nilai dari satu relasi, kita membagi relasi menjadi dua relasi baru. Masing-masing relasi berisi dua atribut yang mempunyai hubungan banyak nilai.

5. Bentuk normal kelima (5NF)

Bentuk normal kelima (5NF) berurusan dengan properti yang disebut join tanpa adanya kehilangan informasi (lossless join). Bentuk normal kelima (5NF) juga disebut PJNF (projection-join normal form). Kasus-kasus ini sangat jarang muncul dan sulit untuk dideteksi secara praktis.

2.2.4.5. MySQL

MySQL adalah sebuah program database server yang mampu menerima dan mengirimkan datanya dengan sangat cepat, *multi-user* serta

menggunakan perintah dasar SQL (*Structured Query Language*). MySQL memiliki 2 lisensi, yakni yang berbentuk *freeware* dan *shareware*.

Kelebihan dari MySQL diantaranya:

1. MySQL sebagai *DataBase Management System* (DBMS) dan RDBMS.
2. MySQL bersifat *Open Source*.
3. MySQL merupakan database server dan client.
4. MySQL mampu menerima query bertumpuk dalam satu permintaan atau yang disebut multi-threading.
5. Kapasitas penyimpanan yang besar.
6. MySQL didukung oleh driver ODBC.
7. MySQL merupakan server database yang multi user.
8. MySQL dapat menciptakan lebih dari 16 kunci per tabel dan dalam satu kunci memungkinkan belasan *field* (kolom).