BAB 2

TINJAUAN PUSTAKA

2.1 Tinjauan Perusahaan

2.1.1 Sejarah Perusahaan

Balantrax Art Shop adalah salah satu toko yang bergerak di industri kerajinan tangan dari ukiran kayu dan kesenian tradisional seperti alat musik. Balantrax Art Shop menjual berbagai ukiran kayu yang berasal dari kota Jatinagor-Bandung. Sejak pertama kali dibuka pada tahun 1990 di Jatinangor letaknya di Jln. Raya Cileunyi Jatinangor Cibeusi/Cikeuruh Sumedang, Balantrax Art Shop telah mempunyai banyak pelanggan baik yang dari kota bandung ataupun yang dari luar kota bandung.

2.1.2 Visi dan Misi

2.1.2.1 Visi

Menjadi salah satu toko kerajinan ukiran kayu yang dapat memberikan pelayanan terbaik terhadap seluruh konsumen di Indonesia dan Mancanegara.

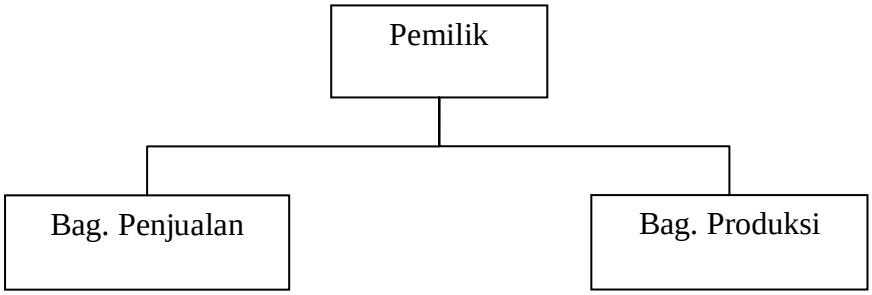
2.1.2.2 Misi

1. Menyediakan produk yang berkualitas dan terjangkau.
2. Memberikan pelayanan terbaik bagi seluruh 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.



Gambar 2.1 Struktur Organisasi Balantrax Art Shop

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 barang yang ada di toko.

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.

2.2 Konsep Dasar Sistem

Konsep dasar sistem ini meliputi pengertian dasar dari sistem dan karakteristik sistem.

2.2.1 Pengertian Sistem

Terdapat dua kelompok pendekatan dalam mendefinisikan *system*, yaitu yang menekankan pada prosedurnya dan yang menekankan pada komponen atau elemennya [2].

a. Pendekatan sistem yang lebih menekankan pada prosedur.

Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu.

b. Pendekatan sistem yang lebih menekankan pada elemen atau komponennya.

Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu.

2.2.2 Karateristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu diantaranya sebagai berikut [2].

a. Komponen-komponen (*Components*)

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerjasama untuk membentuk satu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.

b. Batas Sistem (*Boundary*)

   Batas sistem (*boundary*) merupakan daerah yang membatasi antara satu *system* yang lain atau dengan linkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan.

c. Lingkungan Luar Sistem (*Environment*)

   Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem.

d. Penghubung Sistem (*Interface*)

   Penghubung (*interface*) merupakan media penghubung antara satu sub sistem dengan subsistem lain.

e. Masukan Sistem (*Input*)

   Masukan adalah energi yang dimasukkan kedalam sistem.

f. Keluaran Sistem (*Output*)

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

g. Pengolahan Sistem (*Process*)

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

h. Sasaran (*Objective*)

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran-sasaran (*objectives*). Sasaran dari sistem sangat membutuhkan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.

2.2.3 Perancangan Sistem

2.2.3.1 *Flow map*

Flow map adalah bagan aliran yang menunjukan alir dalam program atau prosedur sistem secara manual. Digunakan terutama sebagai alat bantu komunikasi dan dokumentasi[3].

Hal-hal yang harus diperhatikan dalam menggambarkan *flow map* adalah sebagai berikut:

1. Penggambaran *flow map* dimulai dari atas halaman ke bagian bawah, kmudian dari kiri kebagian kanan
2. Penggambaran *flow map* dilakukan berdasarkan pembagian atau sub sistem
3. Dalam *flow map* harus jelas dimana awal suatu status informasi, kemudian dimana akhir silkus
4. Semua bagian siklus informasi harus jelas Menggunakan kertas kerja yang jelas sesuai dengan yang akan dilakukan dalam sistem

2.2.3.2 ERD *(Entity Relationship Diagram)*

ERD adalah sebuah diagram yang digunakan untuk menggambarkan digunakan untuk menggambarkan ERM (*Entity Relational Model*). ERM adalah model entity yang berisi komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut-atribut yang merepresentasikan seluruh fakta dari dunia nyata yang kita tinjau dapat digambarkan dengan lebih sistematis menggunakan Diagram Entity-Relationship[8]. Dapat disimpulkan bahwa ERD adalah model konseptual yang menjabarkan hubungan antar penyimpanan data dan hubungan 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

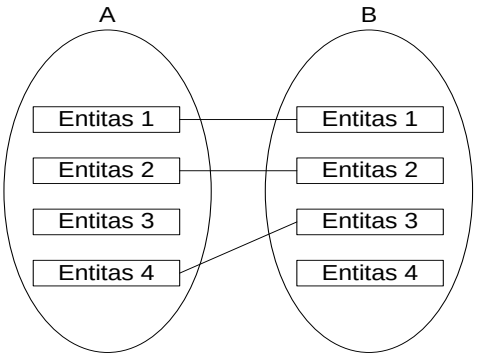
Atribut adalah sifat atau karakteristik dari tiap entitas atau tiap relationship. Maksudnya, atribut adalah sesuatu yang menjelaskan apa yang sebenarnya dimaksud entitas maupun *relationship* sehingga sering dikatakan bahwa atribut adalah elemen dari setiap entitas dan *relationship*.

2.2.3.2.1 Kardinalitas Relasi/Derajat 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[8]. Kardinalitas relasi yang terjadi diantara dua himpunan entitas (misalnya A dan B) dapat berupa :

a. Satu ke Satu (*One to One*)

   Setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas A berhubungan dengan paling banyak satu entitas pada himpunan entitas B, dan begitu juga sebaliknya setiap entitas pada himpunan B berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas A. Ilustrasi derajat relasi satu ke satu dapat dilihat pada gambar 2.2.

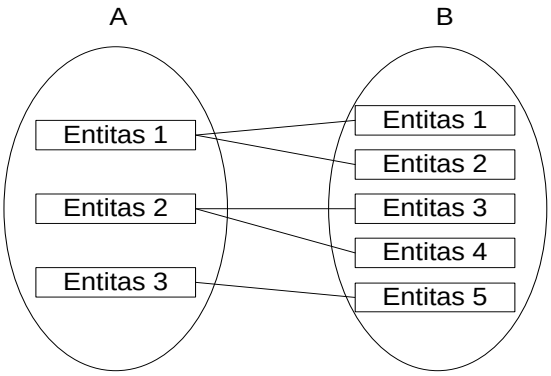


Gambar 2.2 Derajat Relasi Satu Ke Satu(*One to One*)

b. Satu ke Banyak (One to Many)

   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 paling banyak dengan

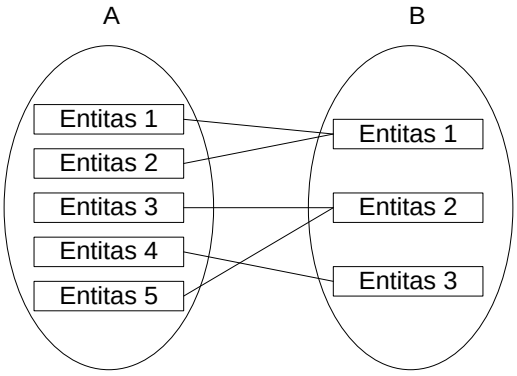
satu entitas pada himpunan entitas A. Ilustrasi derajat relasi satu ke banyak dapat dilihat pada gambar 2.3.



Gambar 2.3 Derajat Relasi Satu ke Banyak(*One to Many*)

c. Banyak ke Satu (Many to One)

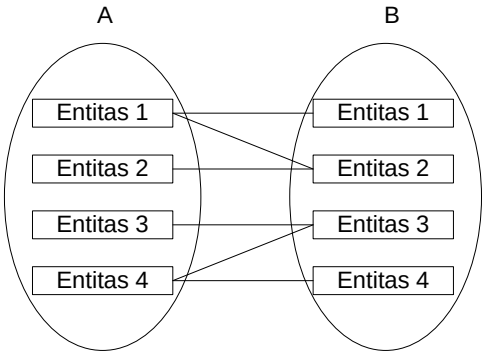
setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B, tetapi tidak sebaliknya, dimana setiap entitas pada himpunan entitas A berhubungan dengan paling banyak satu entitas pada himpunan entitas B. Ilustrasi derajat relasi banyak ke satu dapat dilihat pada gambar 2.4.



Gambar 2.4 Derajat Relasi Banyak ke Satu(*Many to One*)

d. Banyak ke Banyak (*Many to Many*)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, dan demikian juga sebaliknya, dimana setiap entitas pada himpunan entitas B dapat berhubungan dengan banyak entitas A. Ilustrasi derajat relasi banyak ke banyak dapat dilihat pada gambar 2.5.



Gambar 2.5 Derajat Relasi Banyak ke Banyak(*Many to Many*)

2.2.3.2.2 Tahap Pembuatan Diagram E-R(ERD)

Diagram ER dibuat secara bertahap, ada dua kelompok tahapan yang biasa ditempuh didalam pembuatan diagram ER, yaitu[8];

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).

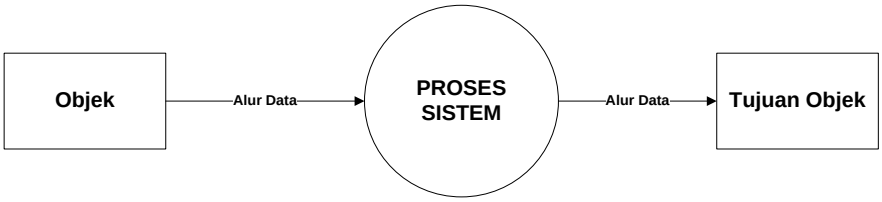
2.2.3.3 Diagram Konteks

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[3].

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.



Gambar 2.6 Proses Diagram Konteks

2.2.3.4 *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.

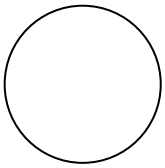
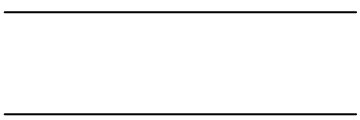
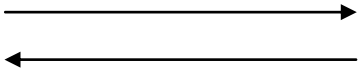
DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur. Kelebihan utama pendekatan aliran data, yaitu sebagai berikut.

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

Disamping itu terdapat kelebihan tambahan, yaitu.

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

Tabel 2.1 Simbol-simbol DFD

No.	Simbol	Fungsi
1.		Proses, digunakan untuk menggambarkan sebuah sistem atau proses.
2.		*Eksternal Entity*, digunakan untuk menggambarkan sebuah objek.
3.		*Storage*, digunakan untuk menggambarkan/ menunjukkan sebuah file data
4.		Arus data, digunakan untuk menggambarkan sebuah arus data.

2.2.3.5 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.

2.2.3.6 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.

2.3 Definisi Analisis Sistem

Kegiatan analisis sistem adalah kegiatan untuk melihat sistem yang sudah berjalan, melihat bagian mana yang bagus dan tidak bagus, dan kemudian mendokumentasikan kebutuhan yang akan dipenuhi dalam sistem yang baru. Hal tersebut terlihat sederhana, namun sebenarnya tidak. Banyak hambatan yang akan ditemui dalam proses tersebut.

Pada banyak proyek sistem informasi, proses analisis dan desain sering kali berjalan bersama-sama. Jadi selama kegiatan analisis, kegiatan desain juga dilakukan. Hal ini karena pada banyak kasus, *user* seringkesulitan untuk mendefinisikan kebutuhan mereka. Jadi mereka akan lebih mudah mendefinisikan kebutuhan, jika mereka telah melihat gambar rancangan antarmuka.

Oleh karena itu, sering kali batasan mengenai bagian mana yang dianggap sebagai analisis dan bagian mana yang dianggap sebagai desain banyak terjadi perbedaan.

Ada yang mendefinisikan analisis sistem adalah sebagai berikut :

a. Seorang yang menggunakan pengetahuan aplikasi komputer yang dimilikinya untuk memecahkan masalah-masalah bisnis, dibawah petunjuk Manajer Sistem.
b. Seorang yang bertanggung jawab menterjemahkan kebutuhan-kebutuhan si pemakai sistem (user) ke dalam spesifikasi teknik yang diperlukan oleh Programmer dan diawasi oleh Manajemen.

Pada pembahasan ini yang dibahas pada bagian anlisis adalah bagaimana metode pengumpulan data dan bagai mana mendokumentasikannya. Sedangkan analisis sistem dianggap merupakan bagian dari desain sistem yang akan dibahas pada pembahasan yang terkait dengan *web system*.

2.3.1 Teknik Pengumpulan Data

Hal pertama yang dilakukan dalam analisis sistem adalah melakukan pengumpulan data. Ada beberapa teknik pengumpulan data yang sering dilakukan yaitu sebagai berikut[4]:

1. Teknik Wawancara
2. Teknik Observasi
3. Teknik Kuisioner

2.3.1.1 Teknik Wawancara

Pengumpulan data dengan menggunakan wawancara mempunyai beberapa keuntungan sebagai berikut:

1. Lebih mudah dalam menggali bagian sistem mana yang dianggap baik dan bagai mana yang dianggap kurang baik.
2. Jika ada bagian tertentu yang menurut Anda perlu untuk digali lebih dalam, Anda dapat langsung menayakan kepada narasumber.
3. Dapat menggali kebutuhan *user* secara lebih bebas.
4. *User* dapat mengungkapkan kebutuhan secara lebih bebbas.

Selain mempunyai beberapa kelebihan tersebut, teknik wawancara juga mempunyai beberapa kelemahan. Berikut ini adalah beberapa kelemahan dari teknik wawancara:

1. Wawancara akan sulit dilakukan jika narasumber kurang dapat mengungkapkan kebutuhannya.
2. Pertanyaan dapat menjadi tidak terarah, terlalu fokus pada hal-hal tertentu dan mengabaikan bagian lainnya.

Berikut ini adalah beberapa panduan dalam melakukan kegiatan wawancara agar memperoleh data yang diharapkan:

1. Buatlah jadwal wawancara dengan narasumber dan beritahukan maksud dan tujuan wawancara.

2. Buatlah panduan wawancara yang akan Anda jadikan arahan agar pertanyaan dapat fokus kepada hal-hal yang dibutuhkan.
3. Gunakan pertanyaan yang jelas dan mudah dipahami.
4. Cobalah untuk menggali mengenai kelebihan dan kekurangan sistem yang telah berjalan sebelumnya.
5. Anda boleh berimprovisasi dengan mencoba menggali bagian-bagian tertentu yang menurut Anda penting.
6. Catat hasil wawancara tersebut.

2.3.1.2 Teknik Observasi

Pengumpulan data dengan menggunakan observasi mempunyai keuntungan yaitu:

1. Analisis dapat melihat langsung bagaimana sistem lama berjalan.
2. Mampu menghasilkan gambaran lebih baik jika dibanding dengan teknik lainnya.

Sedangkan kelemahan dengan menggunakan observasi adalah:

1. Membutuhkan waktu cukup lama karena jika observasi waktunya sangat terbatas maka gambaran sistem secara keseluruhan akan sulit untuk diperoleh.
2. Orang-orang yang sedang diamati biasanya perilakunya akan berbeda dengan perilaku sehari-hari (cenderung berusaha terlihat baik). Hal ini akan menyebabkan gambaran yang diperoleh selama observasi akan berbeda dengan perilaku sehari-hari.

3. Dapat mengganggu pekerjaan orang-orang pada bagian yang sedang diamati.

Berikut ini adalah beberapa petunjuk untuk melakukan observasi:

1. Tentukan hal-hal apa saja yang akan diobservasi agar kegiatan observasi menghasilkan sesuai dengan yang diharapkan.
2. Mintalah izin kepada orang yang berwenang pada bagian yang akan diobservasi.
3. Berusaha sedikit mungkin agar tidak menganggu pekerjaan orang lain.
4. Jika ada yang tidak Anda mengerti, cobalah bertanya. Jangan membuat asumsi sendiri.

2.3.1.3 Teknik Kuisioner

Pengumpulan data dengan mengunakan kuisioner mempunyai keuntungan yaitu:

1. Hasilnya lebih objektif, karena kuisioner dapat dilakukan kepada banyak orang sekaligus.
2. Waktunya lebih singkat.

Sedangkan kelemahan pengumpulan data menggunakan kuisioner adalah sebagai berikut:

1. Responden cenderung malas untuk mengisi kuisioner.
2. Sulit untuk membuat pertanyaan yang singkat, jelas dan mudah dipahami.

Berikut ini adalah beberapa cara yang dapat dilakukan untuk membuat tenik kuisioner mengahsilkan data yang baik.

1. Hindari pertnyaan isian, lebih baik pilihan ganda, karena responden biasanya malas untuk menulis banyak, dan jika responden menuliskan sesuatu sering kali susah untuk dipahami. Dan juga dengan pertanyaan pilihan ganda, akan memudahkan Anda untuk melakukan rekapitulasi data hasil kuisioner.
2. Buatlah pertanyaan yang tidak terlalu banyak.
3. Buatlah pertanyaan yang singkat, padat dan jelas.

2.3.2 Jenis Kebutuhan

Kebutuhan (requirement) yang dikumpulkan dengan menggunakan wawancara, observasi, kuisioner atau gabungan dari ketiga hal tersebut dapat dikelompokan menjadi beberapa kategori sebagai berikut (tidak semua kebutuhan ini harus ada)[3].

1. *Functional requirement*

   Kebutuhan yang terkait dengan fungsi produk, misalnya sistem informasi harus mampu mencetak laporan, sistem informasi harus mampu menampilken grafik, dan lain-lain.

2. *Development requirement*

   Kebutuhan yang terkait *tools* untuk mengembangkan sistem informasi baik perangkat keras maupun perangkat lunak, misalnya sistem informasi dikembangkan dengan menggunakan alat bantu Eclipse untuk pengembangan dan StarUML untuk pemodelan.

3. *Depeloyment requirement*

Kebutuhan terkait dengan lingkungan di mana sistem informasi akan digunakan baik perangkat lunak maupun perangkat keras. Contoh kebutuhan ini misalnya sistem informasi harus mampu berjalan pada server dengan spesifikasi perangkat keras *memory* 4 GB DDR3, *processor* Intel Xeon Core, dan spesifikasi sistem operasi Ubuntu Server9.

4. *Performance requirement*

   Kebutuhan yang terkait dengan ukuran kualitas maupun kuantitas, khususnya terkait dengan kecepatan, skalabilitas, dan kapasitas. Misalnya sistem informasi tersebut harus mampu diakses oleh minimal 1000 orang pada waktu yang bersamaan.

5. *Documentation requirement*

   Kebutuhan ini terkait dengan dokumen apa saja yang disertakan pada produk akhir. Dokumen yang biasanya dihasilkan pada tahap akhir pengembangan sistem informasi antara lain dokumen teknis (mulai dari dokumen perencanaan proyek, analisis, desain sampai pengujian).

6. *Support requirement*

   Kebutuhan yang terkait dukungan yang diberikan setelah sistem informasi digunakan. Dukungan teknis tersebut misalnya adanya pelatihan bagi calon pengguna.

7. *Miscellaneous requirement*

   Kebutuhan ini adalah kebutuhan-kebutuhan tambahan lainnya yang belum tercakup pada beberapa kategori kebutuhan yang telah terdefinisi diatas.

2.3.3 Definisi Desain Sistem

Desain atau perancangan dalam pembangunan perangkat lunak merupkan upaya untuk mengonstruksi sebuah sistem yang memberikan kepuasan (mungkin informal) akan spesifikasi kebutuhan fungsional, memenuhi target, memenuhi kebutuhan secara implisit atau eksplisit dari segi performasi maupun penggunaan sumber daya, kepuasan batasan pada proses desain dari segi biaya, waktu, dan perangkat. Kualitas perangkat lunak biasanya dinilai dari segi kepuasan pengguna perangkat lunak terhadap perangkat lunak yang digunakan.

2.4 Konsep Dasar Informasi

Informasi menjadi suatu hal yang sangat berperan penting dan tidak dapat terpisahkan baik oleh individu maupun organisasi. Kualitas informasi yang baik dapat menjadi sebuah perekat dalam suatu organisasi serta dapat meningkatkan pula kualitas pemahaman para pengelola organisasi tersebut.

2.4.1 Pengertian Informasi

Informasi merupakan hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian (*event*) yang nyata (*fact*) yang digunakan untuk pengambilan keputusan.

Pendekatan dalam pendefinisian data yang diungkapkan Jogiyanto pada bukunya, sebagai berikut.

“Data merupakan bentuk yang masih mentah sehingga perlu diolah lebih lanjut” [9].

Data merupakan bentuk yang masih mentah belum diolah atau diproses yang berisi penggambaran tentang benda, kejadian, aktivitas, dan transaksi yang belum mempunyai makna.

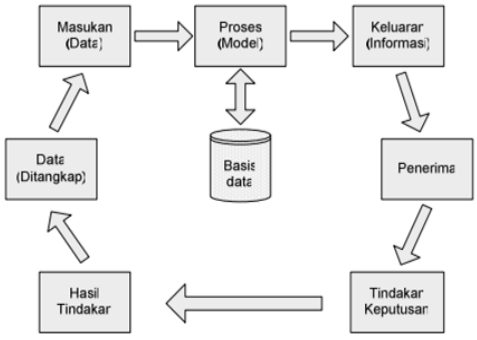
Sementara definisi informasi oleh Jogiyanto yang menjelaskan bahwa informasi merupakan sebuah bentuk yang lebih berguna dan lebih berarti bagi penerimanya adalah sebagai berikut.

"Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya" [9].

2.4.2 Siklus Informasi

Data yang diolah untuk menghasilkan informasi menggunakan suatu model proses tertentu. Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali [9].

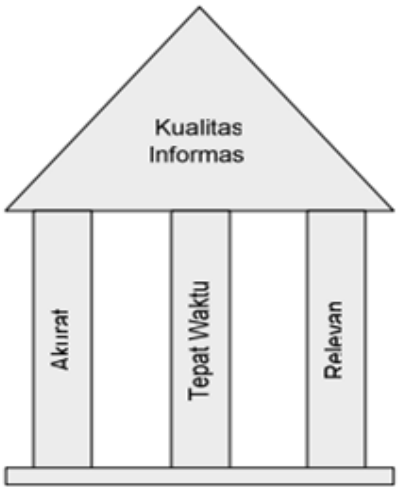
Data tersebut akan ditangkap sebagai *input*, diproses kembali lewat suatu model dan seterusnya membentuk suatu siklus. Siklus ini disebut juga dengan siklus pengolahan data.



Gambar 2.7 Siklus Informasi

2.4.3 Kualitas Informasi

Kualitas dari suatu informasi tergantung dari tiga hal, yaitu informasi yang akurat, tepat pada waktunya dan relevan [9].



Gambar 2.8 Pilar Kualitas Informasi

Ketiga pilar tersebut dapat dijelaskan seperti berikut.

a. Akurat (*accurate*) berarti informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan, akurat juga berarti harus jelas mencerminkan maksudnya.
b. Tepat waktu (*Timeliness*) berarti informasi yang datang pada penerima tidak boleh terlambat.
c. Relevan (*Relevance*) berarti informasi tersebut mempunyai manfaat untuk pemakainya, relevan untuk tiap-tiap orang berbeda.

2.4.4 Nilai Informasi

Nilai informasi ditentukan dari dua hal, yaitu manfaat dan biaya mendapatkannya. Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya. Informasi yang digunakan didalam suatu sistem inforamsi umumnya digunakan untuk beberapa kegunaan. Sehingga

tidak memungkinkan dan sulit untuk menghubungkan suatu bagian informasi pada suatu masalah tertentu dengan biaya untuk memperolehnya [9].

2.5 Konsep Dasar Sistem Informasi

Konsep dasar sistem informasi meliputi pengertian sistem informasi dan komponen sistem informasi.

2.5.1 Pengertian Sistem Informasi

Sistem Informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pegolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan [9].

2.5.2 Komponen Sistem Informasi

sistem informasi terdiri dari komponen-komponen yang disebutnya dengan istilah blok bangunan, yang terdiri dari beberapa blok diantaranya sebagai berikut [3].

a. Blok Masukan

   Masukan mewakili data yang masuk ke dalam sistem informasi. Masukan disini termasuk metode-metode dan media untuk menangkap data yang akan dimasukan berupa dokumen dasar.

b. Blok Model

   Blok model terdiri dari kombinasi prosedur, logika dan model matematika yang akan memanipulasi data *input* dan data tersimpan di dalam basis data untuk menghasilkan keluaran yang diinginkan.

c. Blok Keluaran

Merupakan produk dari sistem informasi yang berupa informasi yang berkualitas dan dokumen yang berguna untuk tingkatan serta semua pemakai sistem.

d. Blok Teknologi

Teknologi merupakan *tool box* di dalam sistem informasi, teknologi yang digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data juga menghasilkan dan mengirimkan keluaran serta membantu pengendalian dari sistem keseluruhan.

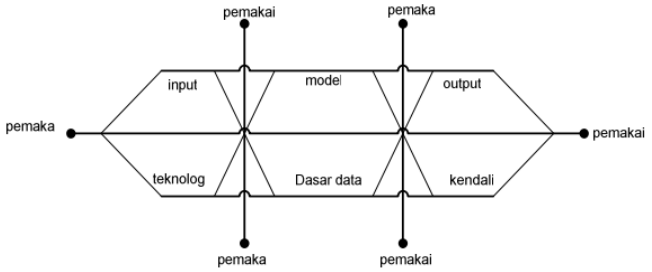
e. Blok Basis Data

Basis data merupakan kumpulan dari data yang saling berhubungan antara satu dengan yang lainnya.

f. Blok Kendali

Pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang merusak sistem dapat dicegah dan bila dalam sistem terdapat kesalahan dapat diatasi secara langsung.

Sebagai suatu sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lainnya membentuk suatu kesatuan untuk mencapai suatu sasaran. Berikut gambar tentang blok sistem informasi tersebut.



Gambar 2.9 Blok Sistem Informasi yang berinteraksi

2.6 Konsep Dasar *Web*

2.6.1 *Internet*

Internet (Interconnected Network) merupakan jaringan komputer yang terdiri dari ribuan jaringan komputer independen yang dihubungkan satu dengan lainnya dengan suatu standar set protokol [6]. Sedangkan dari segi ilmu pengetahuan, *internet* merupakan sebuah perpustakaan besar yang didalamnya teradapat jutaan (bahkan milyaran) informasi atau data yang dapat berupa teks, grafik, suaram animasi dan lain-lain dalam bentuk media elektronik.

2.6.2 *TCP/IP*

TCP/IP (singkatan dari *Transmission Control Protocol/ Internet Protocol*) adalah standar komunikasi data yang digunakan oleh komunitas *internet* dalam proses tukar-menukar data dari satu komputer ke komputer lain di dalam jaringan *Internet* [15]. Protokol ini tidaklah dapat berdiri sendiri, karena memang protokol ini berupa kumpulan protokol (*protocol suite*). Protokol ini juga merupakan protokol yang paling banyak digunakan saat ini. Data tersebut diimplementasikan dalam bentuk perangkat lunak (*software*) di sistem operasi. Istilah yang diberikan kepada perangkat lunak ini adalah *TCP/IP stack*.

Protokol *TCP/IP* dikembangkan pada akhir dekade 1970-an hingga awal 1980-an sebagai sebuah protokol standar untuk menghubungkan komputer-komputer dan jaringan untuk membentuk sebuah jaringan yang luas (WAN). *TCP/IP* merupakan sebuah standar jaringan terbuka yang bersifat independen terhadap mekanisme trans*port* jaringan fisik yang digunakan, sehingga dapat digunakan di mana saja. Protokol ini menggunakan skema pengalamatan yang

sederhana yang disebut sebagai alamat IP (IP *Address*) yang mengizinkan hingga beberapa ratus juta komputer untuk dapat saling berhubungan satu sama lainnya di *Internet*. Protokol ini juga bersifat *routable* yang berarti protokol ini cocok untuk menghubungkan sistem-sistem berbeda (seperti Microsoft Windows dan keluarga UNIX) untuk membentuk jaringan yang heterogen.

2.6.3 *HTTP*

Pengertian *HTTP* (*HyperText Transfer Protocol*) adalah sebuah protokol untuk meminta dan menjawab antara *client* dan *server* [6]. Sebuh *client HTTP* seperti *web browser*, biasanya memulai permintaan dengan membuat hubungan *TCP/IP* ke *port* tertentu di tempat yang jauh (biasanya *port* 80). Sebuah *server HTTP* yang mendengarkan di *port* tersebut menunggu *client* mengirim kode permintaan (*request*) yang akan meminta halaman yang sudah ditentukan, diikuti dengan pesan MIME yang memiliki beberapa informasi kode kepala yang menjelaskan aspek dari permintaan tersebut, diikut dengan badan dari data tertentu.

HTTP berkomunikasi melalui TCP/IP. *Client HTTP* terhubung ke *server HTTP* menggunakan TCP. Setelah membuat sambungan, *client* dapat mengirim pesan permintaan *HTTP* ke *server*. *HTTP* digunakan untuk mengirimkan permintaan dari *client web* (*browser*) ke *web server*, dikembalikan ke konten *web* (halaman *web*) dari *server* ke *client*.

HTTP tidaklah terbatas untuk penggunaan dengan *TCP/IP*, meskipun *HTTP* merupakan salah satu protokol aplikasi *TCP/IP* paling populer melalui *Internet*.

Memang *HTTP* dapat diimplementasikan di atas protokol yang lain di atas *Internet* atau di atas jaringan lainnya.

2.6.4 *HTML*

HyperText Markup Language (*HTML*) adalah sebuah bahasa *markup* yang digunakan untuk membuat sebuah halaman *web*, menampilkan berbagai informasi di dalam sebuah penjelajah *web Internet* dan *formating hypertext* sederhana yang ditulis kedalam berkas format *ASCII* agar dapat menghasilkan tampilan wujud yang terintegrasi [6]. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata dan disimpan kedalam format *ASCII* normal sehingga menjadi *home page* dengan perintah-perintah *HTML*. Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan *SGML* (*Standard Generalized Markup Language*), *HTML* adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman *web*. *HTML* saat ini merupakan standar *Internet* yang didefinisikan dan dikendalikan penggunaannya oleh *World Wide Web* Consortium (W3C). *HTML* dibuat oleh kolaborasi Caillau TIM dengan Berners-lee robert ketika mereka bekerja di CERN pada tahun 1989 (CERN adalah lembaga penelitian fisika energi tinggi di Jenewa).

a. Tahun 1980, IBM memikirkan pembuatan suatu dokumen yang akan mengenali setiap elemen dari dokumen dengan suatu tanda tertentu. IBM kemudian mengembangkan suatu jenis bahasa yang menggabungkan teks dengan perintah-perintah pemformatan dokumen. Bahasa ini dinamakan *Markup Language*, sebuah bahasa yang menggunakan tanda-tanda sebagai

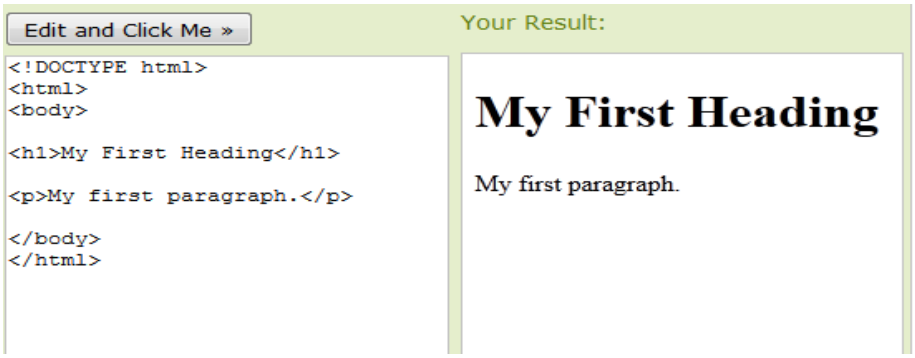
basisnya. IBM menamakan sistemnya ini sebagai *Generalized Markup Language* atau *GML*.

b. Tahun 1986, ISO menyatakan bahwa IBM memiliki suatu konsep tentang dokumen yang sangat baik, dan kemudian mengeluarkan suatu publikasi (ISO 8879) yang menyatakan markup language sebagai standar untuk pembuatan dokumen-dokumen. ISO membuat bahasa ini dari *GML* milik IBM, tetapi memberinya nama lain, yaitu *SGML* (*Standard Generalized Markup Language*).

HTML berupa kode-kode tag yang menginstruksikan penjelajah *web* untuk menghasilkan tampilan sesuai dengan yang diinginkan. Sebuah file yang merupakan file *HTML* dapat dibuka dengan menggunakan penjelajah *web* seperti Mozilla Firefox atau Microsoft Internet Explorer. *HTML* juga dapat dikenali oleh aplikasi pembuka surel ataupun dari PDA dan perangkat lunak lain yang memiliki kemampuan *browser*. Dengan menggunakan perintah-perintah *HTML* memungkinkan pengguna untuk melakukan tugas-tugas berikut.

a. Menentukan ukuran dan alur tulisan.
b. Mengintegerasikan gambar dengan tulisan.
c. Membuat Pranala.
d. Mengintegerasikan berkas suara dan rekaman gambar hidup.
e. Membuat form interaktif.

Contoh script html dapat dilihat pada gambar 2.10.



Gambar 2.20 Contoh *Script HTML*

2.6.5 *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 [14].

2.6.6 *World Wide Web*

WWW (*World Wide Web*), merupakan kumpulan *web server* dari seluruh dunia yang berfungsi menyediakan data dan informasi untuk dapat digunakan bersama. *WWW* atau biasa disebut *web* adalah bagian yang paling menarik dari *Internet*. Melalui *web*, dapat mengakses informasi-informasi yang tidak hanya berupa teks tetapi bisa juga berupa gambar, suara, video dan animasi [12].

Fasilitas ini tergolong masih baru dibandingkan surel (*email*), sebenarnya *WWW* merupakan kumpulan dokumen-dokumen yang sangat banyak yang berada pada komputer *server* (*web server*), di mana *server-server* ini tersebar di lima benua termasuk Indonesia, dan terhubung menjadi satu melalui jaringan *Internet*. Dokumen-dokumen informasi ini disimpan atau dibuat dengan format *HTML* (*Hypertext Markup Language*). Suatu halaman dokumen informasi dapat terdiri atas teks yang saling terkait dengan teks lainnya atau bahkan dengan dokumen lain. Keterkaitan halaman lewat teks ini disebut *hypertext*. Dokumen infomasi ini tidak hanya terdiri dari teks tetapi dapat juga berupa gambar, mengandung suara bahkan klip video. Kaitan antar-dokumen yang seperti itu biasa disebut *hypermedia*.

Jadi dapat disimpulkan bahwa *WWW* adalah sekelompok dokumen multimedia yang saling terkoneksi menggunakan *hyperteks link*. Dengan mengklik *hyperlink*, maka bisa berpindah dari satu dokumen ke dokumen lainnya.

2.6.7 Aplikasi *Web*

Aplikasi *Web* adalah sekumpulan halaman *web* yang mampu berinteraksi dengan pengunjung, dengan sesama halaman *web* dan dengan berbagai sumber data yang ada di *server 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.

2.6.8 *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* untuk berinteraksi dengan halaman situs. *HTTP* memperbolehkan *web browser* (*client HTTP*) [12].

2.6.9 *Webhosting*

Webhosting merupakan bagian yang sangat penting dari *E-Commerce*, karena di tempat inilah semua data dan *software* akan diletakkan. Ada beberapa syarat yang harus dipenuhi untuk menentukan webhosting yang baik[11].

a. Uptime yang bagus
b. Technical Support yang bagus
c. Koneksi di internet yang sangat cepat
d. Kompatibel dengan provider transaksi online
e. Memiliki reputasi yang sangat sangat baik

2.6.10 SEO (*Search Engine Optimization*)

SEO *(Search Engine Optimization)* adalah serangkaian proses yang dilakukan secara sistematis yang bertujuan untuk meningkatkan volume dan kualitas trafik dari mesin pencari menuju alamat situs web tertentu dengan memanfaatkan mekanisme kerja alami algoritma mesin pencari tersebut.

Menurut David Viney, SEO adalah "*Sebuah teknik pengoptimasian sebuah halaman yang mengandung kata kunci atau frase yang bersangkutan yang akan di index oleh mesin pencari sebagai kata kunci pecarian"*.

Tujuan dari SEO adalah menempatkan sebuah situs *web* pada posisi teratas, atau setidaknya halaman pertama hasil pencarian berdasarkan kata kunci tertentu yang ditargetkan. Secara logis, situs *web* yang menempati posisi teratas pada hasil pencarian memiliki peluang lebih besar untuk mendapatkan pengunjung.

Sejalan dengan makin berkembangnya pemanfaatan jaringan internet sebagai media bisnis, kebutuhan atas SEO juga semakin meningkat. Berada pada posisi teratas hasil pencarian akan meningkatkan peluang sebuah perusahaan pemasaran berbasis *web* untuk mendapatkan pelanggan baru. Peluang ini dimanfaatkan sejumlah pihak untuk menawarkan jasa optimisasi mesin pencari bagi perusahaan-perusahaan yang memiliki basis usaha di internet.

Menurut Danny Sullivan, istilah *search engine optimization* pertama kali digunakan pada 26 Juli tahun 1997 oleh sebuah pesan spam yang diposting di Usenet. Pada masa itu algoritma mesin pencari belum terlalu kompleks sehingga mudah dimanipulasi.

Versi awal algoritma pencarian didasarkan sepenuhnya pada informasi yang disediakan oleh *webmaster* melalui *meta tag* pada kode html situs *web* mereka. *Meta tag* menyediakan informasi tentang konten yang terkandung pada suatu halaman *web* dengan serangkaian katakunci (*keyword*). Sebagian *webmaster* melakukan manipulasi dengan cara menuliskan katakunci yang tidak sesuai dengan konten situs yang sesungguhnya, sehingga mesin pencari salah menempatkan dan memeringkat situs tersebut. Hal ini menyebabkan hasil pencarian menjadi tidak akurat dan menimbulkan kerugian baik bagi mesin pencari maupun bagi pengguna internet yang mengharapkan informasi yang relevan dan berkualitas.

Larry Page dan Sergey Brin, dua mahasiswa doktoral ilmu komputer Universitas Stanford, berusaha mengatasi permasalahan tersebut dengan membangun *Backrub*, sebuah mesin pencari sederhana yang mengandalkan perhitungan matematika untuk memeringkat halaman *web*. Algoritma tersebut, yang dinamakan *PageRank*, merupakan fungsi matematika yang kompleks berupa kombinasi antara perhitungan jumlah *link* yang mengarah pada suatu halaman *web* dengan analisis atas kualitas masing-masing *link* tersebut.

Berdasarkan prinsip kerja *PageRank*, secara umum bisa dikatakan bahwa halaman *web* yang memperoleh peringkat tinggi adalah halaman *web* yang banyak di-*link* oleh halaman *web* lain. Nilai *PageRank* juga akan semakin tinggi apabila halaman *web* yang mengarah kepadanya juga memiliki kualitas yang tinggi. Nilai sebuah *link* dari situs berkualitas tinggi seperti Yahoo! atau DMOZ dapat bernilai

lebih tinggi daripada kombinasi nilai *link* dari seratus situs *web* berkualitas rendah.

Backrub hanyalah sebuah permulaan. Pada tahun 1998 Page dan Brin mendirikan Google yang merupakan versi tingkat lanjut dari *Backrub*. Dalam waktu singkat Google memperoleh reputasi dan kepercayaan dari publik pengguna internet karena berhasil menyajikan hasil pencarian yang berkualitas (tidak dimanipulasi), cepat, dan relevan. *PageRank* lantas menjadi standar baik bagi mesin pencari lain maupun bagi *webmaster* yang berusaha agar situs *web*nya memperoleh nilai *PageRank* setinggi mungkin sehingga menempati posisi tertinggi pada hasil pencarian.

Dapat disimpulkan bahwa fungsi dan tujuan utama dari SEO adalah sebagai berikut:

a. Menaikkan *pageranking* sebuah *web*site agar selalu terindex pada *search engine* sehingga dapat ditampilkan dihalaman utama.
b. Mendatangkan trafik / pengunjung ke *web*site melalui *search engine*.
c. Membantu meningkatkan pencapaian target penjualan melalui rekomendasi *web*.
d. Meminimalkan biaya pemasaran *online*.

Mesin pencari besar seperti Google dan Yahoo! menyediakan program dan panduan yang memungkinkan webmaster mengoptimalkan situsnya agar terindeks dengan baik. Google menyediakan aplikasi *Webmaster Tool* (anda harus mempunyai akun di Google guna melihat *tool* ini) dan memperkenalkan *sitemap* berbasis XML standar mereka, sedangkan Yahoo!

menyediakan program *Site Explorer* (juga harus login dengan akun pengguna Yahoo! anda) yang memungkinkan webmaster mendaftarkan URL situs, mengecek jumlah halaman web mereka yang telah terindeks di data-base Yahoo!, dan melihat informasi link masuk. Namun demikian mesin pencari tidak mentolerir metode SEO yang manipulatif dan menghalalkan segala cara.

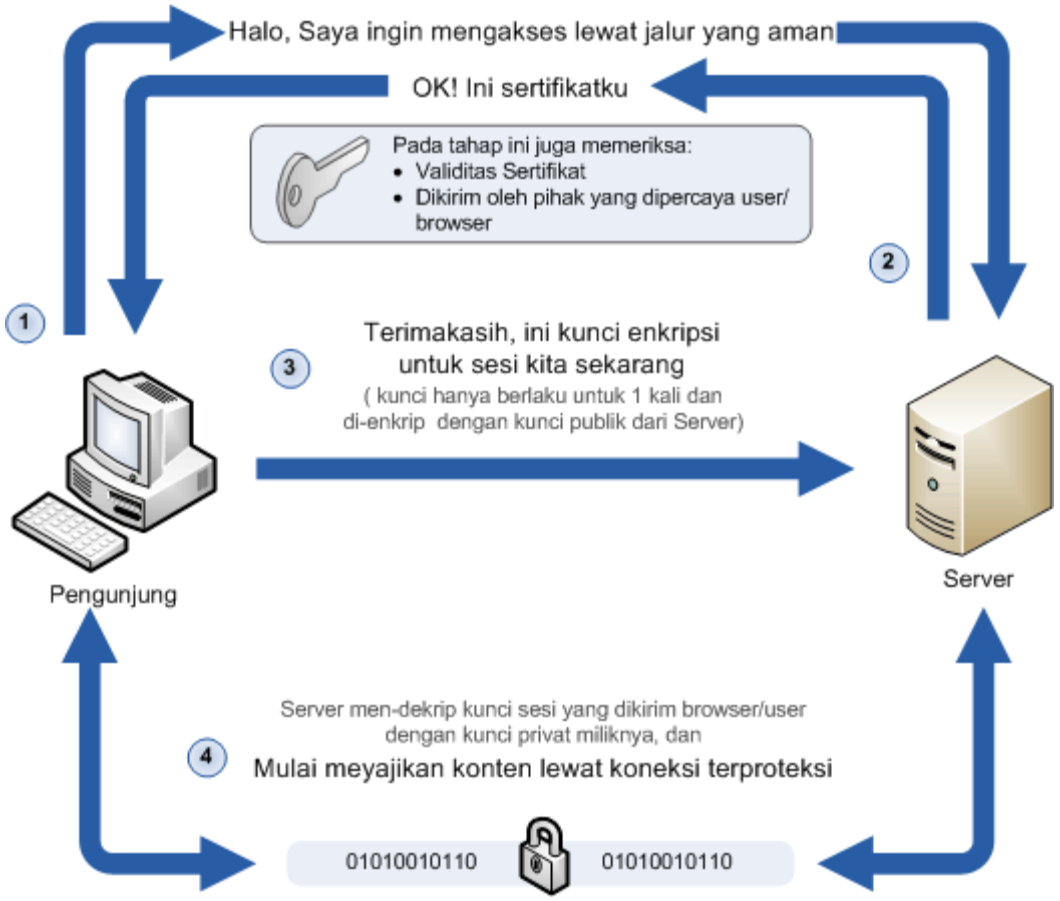
2.6.11 *SSL(Secure Socket Layer)*

Secure Socket Layer adalah suatu protokol yang diciptakan oleh Netscape Communication pada tahun 1994 untuk memastikan keamanan dalam bertransaksi di internet antara *webserver* dan *browser* dari klien. Protokol ini menggunakan sebuah badan yang biasa disebut CA*(Certificate Authority)* untuk mengidentifikasi memverifikasi pihak-pihak yang bertransaksi. Teknologi *SSL* menggunakan konsep teknologi kriptografi untuk bisa mencapai komunikasi yang aman ini antara server dan pengunjungnya. Kedua pihak yang berkomunikasi ini (server dan pengunjungnya) saling mengirimkan data yang disamarkan dan untuk membacanya digunakan sandi dan kunci yang hanya dimiliki kedua pihak yang berkomunikasi tersebut, sehingga pihak lain yang mencoba menyadap data yang dikirim tersebut tidak akan bisa membacanya karena sandi dan kunci yang dibutuhkan tersebut hanya dimiliki oleh kedua pihak yang berkomunikasi tadi [11].

Secara sederhana, komunikasi internet dengan *SSL* ini bisa dilihat dengan cara mengakses alamat URL-nya yang diawali dengan "**https://**" (misalnya https://www.namadomainanda.com), sedangkan yang tanpa SSL alamatnya hanya http biasa (misalnya http://www.nama_domain.com).

2.6.11.1 Cara Kerja *Secure Socket Layer (SSL)*

SSL atau *Secure Sockets Layer* adalah sebuah protokol keamanan data yang digunakan untuk menjaga pengiriman data web server dan pengguna situs web tersebut. Sebagai ilustrasi, mari kita lihat gambar berikut:



Gambar 2.31 Ilustrasi *SSL*

1. Seorang pengunjung mengakses ke situs yang sudah terpasang sertifikat *SSL* lewat saluran (port) yang aman dengan browser-nya (Internet Explorer, Mozilla Firefox, Opera, Safari, dsb), biasanya port 443. Koneksi aman ini

ditandai dengan cara mengakses alamat **https://** (misalnya https://www.namadomainanda.com)

2. Server/situs akan mengirim kunci publik sertifikat *SSL* yang dimilikinya ke pengunjung tadi. Saat browser pengunjung menerimanya, browser akan memeriksa validitas kunci publik tersebut, apakah:
    a. kunci publik sertifikat itu masih berlaku atau belum kadaluarsa,
    b. kunci publik sertifikat itu benar-benar berlaku untuk situs atau domain yang diakses (misalnya www.namadomainanda.com tadi),
    c. kunci publik sertifikat itu juga ada dalam database sertifikat yang tersimpan dalam browser.
3. Jika kunci publik sertifikat tadi bisa terverifikasi oleh browser pengunjung, maka browser akan mengirimkan kunci publik miliknya ke server atau situs yang diakses.
4. Dengan kunci publik yang dikirim browser pengunjung tadi dan kunci privat yang dimiliki server akan digunakan server untuk membuat sandi untuk mengirim data ke pengunjung. Browser pengunjung akan membuka sandi tersebut dan membaca data yang dikirim. Sandi yang dikirim tadi merupakan sandi yang unik dan hanya berlaku satu kali dan hanya bisa dibaca oleh browser yang tadi memintanya, sehingga pihak lain tidak bisa membacanya. Dan selanjutnya setiap data yang diminta atau dikirim antara pengunjung dan server akan melalui proses ini.

2.6.11.2 Keuntungan *Secure Socket Layer (SSL)*

Transaksi bisnis ke bisnis atau bisnis ke pelanggan yang tidak terbatas dan menambah tingkat kepercayaan pelanggan untuk melakukan transaksi online dari situs anda. Untuk lebih jelasnya keuntungan *Secure Socket Layer (SSL)* akan ada pada tabel 2.1 sebagai berikut :

Tabel 2.2 Keuntungan *SSL*

Kriteria	***SSL***
Enkripsi	1. Kuat 2. Berbasis browser
Dukungan otentikasi	1. Otentikasi satu arah 2. Sertifikat digital 3. Otentikasi dua arah
Security	1. End-to-end security 2. Client to resource encrypted
Metode pengaksesan	Kapanpun dan di manapun user berada
Instalasi	1. plug and play. 2. Tidak memerlukan perangkat lunak atau keras tertentu.
Biaya	Rendah dan murah
Kenyamanan User	1. Friendly

	2. Tidak membutuhkan training
Perangkat lunak yang dibutuhkan client	web browser standar
Skalabilitas	scalable dan mudah dideploy
User	User-customer, partner, pegawai, vendor, dsb

2.6.11.3 Kerugian *Secure Socket Layer (SSL)*

Sebagian besar penyelenggara Internet banking di Indonesia mengklaim menggunakan teknologi *Secure Socket Layer (SSL)* untuk menjamin keamanan layanan mereka. Jaminan *SSL* 128 bit inilah yang sering digunakan dalam iklan dan dalam meyakinkan kustomer. Kata-kata lainnya yang sering digunakan dalam menjamin keamanan para pengguna adalah penggunaan firewall, *Public Key Infrastructure* dan *Encryption Accelerator Card*. Pendekatan keterbukaan belum menjadi suatu tradisi pada Internet Banking di Indonesia. Sehingga penjelasan sekuriti relatif masih berfungsi sebagai PR belaka.

Sayangnya seringkali informasi yang diterima pengguna kuranglah lengkap mengenai apa yang diamankan oleh *SSL* ini. Begitu juga dengan firewall kurang dijelaskan apa yang diamankan oleh firewall ini. Hal ini mengakibatkan munculnya, pemahaman akan adanya jaminan keamanan semu dalam benak pengguna. Pengguna sering memiliki anggapan karena sudah memakai *SSL* maka pasti koneksi yang dilakukannya aman, tak ada masalah keamanan yang bisa timbul. Hal ini juga dididorong oleh informasi yang kurang lengkap dari penyedia jasa Internet Banking.

SSL (Secure Socket Layer) pada dasarnya merupakan suatu mekanisme yang melindungi koneksi dari usaha penyadapan. Hal ini karena komunikasi yang terjadi antara client-server melalui suatu jalur yang di enkripsi. Tetapi sistem ini tidak melindungi dari salah masuknya pengguna ke host yang berbahaya, ataupun tak melindungi apakah suatu kode yang di download dari suatu situs bisa dipercaya, atau apakah suatu situs itu bisa dipercaya. Abadi (1996) telah menunjukkan kelemahan protokol SSL versi awal secara teoritis. Jadi jelas *SSL* ini tidak melindungi dari beberapa hal misal (detail dari tiap ancaman ini tidak dibahas pada tulisan ini) :

1. Denial of Services
2. Buffer overflow
3. Man-in-the-middle attack
4. Cross scripting attack

Pada model *SSL,* user-lah yang harus bertanggung jawab untuk memastikan apakah server di ujung sana yang ingin diajak berkomunikasi benar-benar merupakan server yang ingin dituju. Pada dunia nyata untuk meyakinkan bahwa orang yang dihubungi adalah orang sesungguhnya, dapat dilakukan dengan mudah karena orang saling mengenal. Dengan melihat muka, suara, bau dan sebagainya kita bisa mendeteksi bahwa dia orang yang sesungguhnya.

Pada dunia internet hal seperti itu sulit dilakukan, oleh karenanya digunakan sertifikat digital untuk melakukan hal ini. Sertifikat ini mengikat antara suatu public key dengan suatu identitas. Sertifikat ini dikeluarkan oleh sebuah pihak yang disebut CA *(Certificate Authority)* misal dalam hal ini *Verisign* atau *Thawte.*

CA sendiri memperoleh sertifikat dari CA lainnya. CA yang tertinggi disebut root dan tidak memerlukan sertifikat dari CA lainnya. Penanganan sertifikat ini dilakukan secara hierarki dan terdistribusi.

Sayangnya sertifikat digital saja, bukanlah obat mujarab yang bisa mengobati semua jenis permasalahan sekuriti. Agar SSL dapat bekerja dengan semestinya (melakukan koneksi terenkripsi dengan pihak yang semestinya), maka penggunalah yang harus memverifikasi apakah sertifikat yang dimiliki oleh server yang ditujunya adalah benar. Berikut ini adalah beberapa hal minimal harus diperhatikan :

1. Apakah sertifikat tersebut dikeluarkan oleh CA yang dipercaya.
2. Apakah sertifikat tersebut dikeluarkan untuk pihak yang semestinya (perusahaan yang situsnya dituju).
3. Apakah sertifikat itu masih berlaku.

Sayangnya banyak orang tak peduli terhadap permasalahan di atas. Sebetulnya ketika melakukan koneksi ke sebuah situs yang mendukung SSL, hal tersebut ditanyakan oleh browser, tetapi sebagian besar pengguna selalu menekan Yes ketika ditanya untuk verifikasi sertifikat ini. Untuk melihat ketiga hal tersebut, dapat dilakukan dengan double-click pada tombol kunci yang ada di bagian kiri bawah browser.

Begitu juga dengan keterangan 128-bit *SSL*. Seringkali tanpa dilengkapi dengan penjelasan semestinya apa maksud 128-bit ini, dan apa kaitannya dengan PIN pengguna, ataupun hal lainnya. Masih banyak perusahaan yang mengambil mentah-mentah keyakinan akan keamanan *SSL* ini tanpa mencoba memahami atau

menerangkan keterbatasan SSL dalam melakukan perlindungan. Sebagai dampaknya pengguna menjadi tak peduli terhadap ditail mekanisme transaksi yang dilakukannya.

Dengan memanfaatkan kekurang-waspadaan pengguna dapat timbul beberapa masalah sekuriti. Berikut ini adalah celah sekuriti dalam penggunaan SSL yang diakibatkan oleh server si penyerang di luar server asli. Celah seperti ini relatif sulit dideteksi dan dijejaki tanpa adanya tindakan aktif, karena terjadi di server lain. Celah ini pada dasarnya dilakukan dengan cara mengalihkan akses user dari situs aslinya ke situs palsu lainnya, sehingga dikenal dengan istilah page hijacking.

Beberapa kemungkinan teknik yang digunakan untuk melakukan hal ini adalah :

1. Ticker symbol smashing. Biasanya digunakan pada pengumuman press release, dengan memanfaatkan simbol dari perusahaan besar lainnya. Sehingga secara tersamar pengguna akan belok ke situs ini. Misal Perusahaan KUMBAYO baru saja meluncurkan produknya. Perusahaan ini tak ada hubungan dengan Bank Ha Ha. Misal Bank Ha-Ha adalah suatu bank besar. Dengan cara ini orang akan terdorong ke situs perusahaan KUMBAYO, yang semula akan ke Bank Ha-Ha.
2. *Web Spoofing*. Memanipulasi alamat *URL* pada sisi client, sehingga akan memaksa si korban melakukan browsing dengan melalui situs tertentu terlebih dahulu. Dengan cara ini dapat menyadap segala tindakan si korban, ketika melakukan akses ke situs-situs. Sehingga si penyerang dapat memperoleh PIN ataupun password. Cara ini biasanya memanfaatkan trick

URL Rewrite. Umumnya pengguna awam tak memperhatikan apakah akses dia ke suatu situs melalui www.yahoo.com atau melalui www.perusak.org/www.yahoo.com. Karena yang tampil di browsernya adalah tetap halaman dari www.yahoo.com.

3. DNS *Spoofing* (Bellovin, 1995). Teknik ini digunakan untuk memanfaatkan DNS server untuk membangkitkan celah sekuriti. Dengan cara ini penyerang mampu membelokkan seorang pengguna ke server DNS lain yang bukan server semestinya, ketika ia memasukkan nama situs. Dengan cara ini maka penipuan dapat dilanjutkan misalnya dengan mengumpulkan PIN atau password.
4. Typo Pirates. Dengan cara mendaftar nama domain yang hampir mirip, dan membuat situs yang mirip. Pengguna yang tak waspada akan masuk ke situs ini dan memberikan PIN dan password. Cara inilah yang terjadi pada kasus KlikBCA palsu. Hal ini disebabkan sebagian besar pengguna tak waspada, apakah alamat URL (Universal Resource Locator) yang dimasukkannya benar pada saat ia mengakses suatu situs web, dan apakah sertifikat yang diterima sama dengan sertifikat seharusnya pada saat ia mengakses situs web yang mendukung *SSL*.
5. Cybersquating. Membeli nama domain yang mungkin akan digunakan orang. Tujuan penggunaan cara ini adalah lebih kepada mengambil keuntungan keuangan dengan menjual kembali domain tersebut pada harga yang jauh lebih tinggi daripada harga sebenarnya.

6. *Man-in-the-middle-attack*. Cara ini dilakukan dengan memaksa orang percaya bahwa situs yang dituju sama halnya dengan situs asli. Hal itu dilakukan dengan mencegat akses pengguna ketika hendak melakukan koneksi ke situs asli, teknik seperti TCP *Hijack* sering digunakan, lalu meneruskan akses pengguna ke web situs sebenarnya. Sepintas lalu hal ini tidak terlihat oleh pengguna. Serangan ini lebih berbahaya daripada sekedar typo pirates. Resiko ini bisa timbul ketika jalur penyerang berada di antara pengguna dan situs penyedia layanan.

Trik-trik di atas sebagian besar memanfaatkan kelengahan pengguna, atau keawaman pengguna. Dalam mendisain sistem maka perlu diperhatikan kelengahan pengguna ini. Baik kesalahan dia mengetik nama situs, dan lain-lainnya. Untuk itu sudah sepantasnya pemahaman tentang user Indonesia perlu dilakukan lebih dalam sebelum dilakukan perancangan sistem ini.

2.6.12 *URL (Uniform Resource Locators)*

URL adalah suatu sarana yang digunakan untuk menentukan lokasi informasi pada suatu web server *URL* dapat diibaratkan seperti suatu alamat dimana alamat tersebut terdiri dari [8]:

a. Protocol yang digunakan oleh web server
b. Nama komputer(server) dimana informasi tersebut berada.
c. Jalur/Path serta nama file dari suatu informasi

   Sebagai contoh : http:// www.amazon.com/books/index.html

   http merupakan protocol yang digunakan.

   www.amazon.com merupkan host/server dimana informasi tersebut berada

books/ merupakan jalur atau path dan index.html merupakan nama file dari suatu informasi.

2.6.13 *FTP (File Transfer Protocol)*

FTP merupakan *protocol* yang dirancang untuk memungkinkan pengguna mentransfer file ke suatu *server* di *internet*. Dua hal yang penting dalam *FTP* adalah *FTP Server* dan *FTP Client. FTP server* adalah suatu server yang menjalankan software yang berfungsi untuk memberikan layanan tukar menukar file dimana server tersebut selalu siap memberikan layanan *FTP* apabila mendapat permintaan (*request*) dari *FTP client*[8].

FTP client adalah komputer yang merequest koneksi ke *FTP server* untuk tujuan tukar menukar file. Setelah terhubung dengan *FTP server*, maka client dapat melakukan *download*, *upload*, *rename* dan *delete* sesuai dengan permission yang diberikan oleh *FTP server*. Tujuan dari *FTP server* adalah sebagai berikut :

1. Untuk tujuan sharing data
2. Untuk menyediakan indirect atau implicit remote computer
3. Untuk menyediakan tempat penyimpanan bagi Member
4. Untuk menyediakan transfer data yang reliable dan efisien.

2.6.14 *DNS (Domain Name System)*

DNS merupakan suatu sistem penamaan standar kompter-komputer di internet dengan tujuan untuk mempermudah pengelolaan server komputer di internet dan digunakan untuk pencarian nama komputer (*name resolution*) di jaringan yang mengunakan *TCP/IP (Transmission Control Protocol/Internet Protocol)*. *DNS* biasa digunakan pada aplikasi yang terhubung ke Internet seperti

web browser atau e-mail, dimana *DNS* membantu memetakan host name sebuah komputer ke *IP address*. Selain digunakan di Internet, *DNS* juga dapat di implementasikan ke *private network* atau *intranet* dimana *DNS* memiliki keunggulan seperti[8]:

a. Mudah, *DNS* sangat mudah karena user tidak lagi direpotkan untuk mengingat *IP* address sebuah komputer cukup host name (nama komputer).
b. Konsisten, *IP address* sebuah komputer bisa berubah tapi host name tidak berubah.
c. Simple, user hanya menggunakan satu nama domain untuk mencari baik di Internet maupun di Intranet.

2.7 *E-Commerce (Electronic Commerce)*

2.7.1 Sejarah *E-Commerce*

E-commerce pertama kali diperkenalkan pada tahun 1994 pada saat pertama kali *banner-elektronik* dipakai untuk tujuan promosi dan periklanan di suatu halaman-*web* (*website*). Menurut Riset Forrester, perdagangan elektronik menghasilkan penjualan seharga AS$12,2 milyar pada 2003. Menurut laporan yang lain pada bulan oktober 2006 yang lalu, pendapatan ritel *online* yang bersifat non-travel di Amerika Serikat diramalkan akan mencapai seperempat trilyun dolar US pada tahun 2011[15].

2.7.2 Pengertian *E-Commerce*

E-commerce merupakan suatu sistem atau paradigma baru dalam dunia bisnis, yang menggeser paradigma tradisional *commerce* menjadi *electronic commerce* yaitu dengan memanfaatkan teknologi *ICT* (*Information and Communication Technology*), atau dengan kata lain teknologi *Internet*.

Menurut Stefan Probst (Opticom), definisi *e-commerce* adalah "*business* yang dilakukan secara *electronic* yang melibatkan aktivitas-aktivitas bisnis berupa *business to business* ataupun *business to consumen* melalui teknologi *Internet*".

Definisi *e-commerce* secara umum dapat diartikan: "Proses membeli, menjual, baik dalam bentuk barang, jasa ataupun informasi, yang dilakukan melalui media *Internet*" (disarikan dari berbagai sumber).

2.7.3 Model *E-Commerce*

Pada tradisional bisnis konsumen dapat membedakan model bisnis dari pelaku bisnis berdasarkan jenis layanan yang ditawarkan. Misalnya, layanan pada model bisnis Bank akan berbeda dengan layanan pada model bisnis *stock exchange* (bursa saham).

Secara umum, interaksi dan transaksi antara pelaku bisnis yang akan menggunakan teknologi *e-commerce* dapat dikategorikan dalam jenis-jenis berikut.

a. B2B (*business to business*)

   Adalah model *e-commerce* dimana pelaku bisnisnya adalah perusahaan, sehingga proses transaksi dan interaksinya adalah antara satu perusahaan dengan perusahaan lainnya.

b. B2C (*business to consumen*)

Adalah model *e-commerce* dimana pelaku bisnisnya melibatkan langsung antara penjual (penyedia jasa *e-commerce*) dengan *individual buyers* atau pembeli.

c. C2B (*consumen to business*)

Adalah model *e-commerce* dimana pelaku bisnis perorangan atau individual melakukan transaksi atau interaksi dengan suatu atau beberapa perusahaan. Jenis *e-commerce* seperti ini sangat jarang dilakukan di Indonesia.

d. C2C (*consumen to consumen*)

Adalah model *e-commerce* dimana perorangan atau individu sebagai penjual berinteraksi dan bertransaksi langsung dengan individu lain sebagai pembeli. Konsep *e-commerce* jenis ini banyak digunakan dalam situs *online auction* atau lelang secara *online*.

	Business (Organization)	**Customer (individual)**
Business (Organization)	**B2B (e.g CIMB Group)**	**B2C (e.g Amazon)**
Customer (individual)	**C2B (e.g Priceline)**	**C2C (e.g eBay)**

Gambar 2.12 Jenis-jenis *E-commerce*

2.7.4 Keuntungan *E-Commerce*

E-commerce menjanjikan dampak yang sangat besar untuk setiap industri. Kemampuan untuk membandingkan dan membeli produk pada *internet* mengubah praktek bisnis banyak perusahaan karena memiliki beberapa keuntungan.

Menurut Adi Nugroho (2006), ada beberapa keuntungan *e-commerce* antara lain adalah sebagai berikut [7].

a. Keuntungan bagi Perusahaan
    1. Memperpendek Jarak. Dengan hanya mengklik *link-link* yang ada pada situs-situs, konsumen dapat menuju ke perusahaan dimanapun saat itu mereka berada.
    2. Perluasan Pasar. Jangkauan pemasaran menjadi sangat luas dan tidak terbatas oleh area geografis dimana perusahaan berada.
    3. Perluasan Jaringan Mitra Bisnis. Pada perdagangan konvensional, sangat sulit bagi suatu perusahaan untuk mengetahui posisi geografis mitra kerjanya yang berada di negara-negara lain atau benua lain.
    4. Efisien. *E-Commerce* akan sangat memangkas biaya-biaya operasional perusahaan.
b. Keuntungan bagi Konsumen.
    1. Efektif. Konsumen dapat memperoleh informasi tentang produk/jasa yang dibutuhkan dan bertransaksi dengan cara yang cepat dan murah.

2. Aman Secara Fisik. Karena konsumen tidak perlu mendatangi took-toko untuk mendapatkan produk dan ini memungkinkan konsumen aman secara fisik.
3. Fleksibel. Konsumen juga dapat bertransaksi dimanapun dia berada.

c. Keuntungan bagi Masyarakat Umum

1. Mengurangi Pencemaran dan Polusi Lingkungan. Konsumen tidak perlu mendatangi took-toko, dimana hal ini akan mengurangi jumlah kendaraan yang berlalu-lalang, berarti menghemat bahan bakar dan tingkat polusi.
2. Membuka Peluang Kerja Baru. Era perdagangan elektronik akan membuka peluang-peluang kerja baru bagi orang-orang yang tidak 'buta' akan teknologi.
3. Menguntungkan Dunia Akademis. Berubahnya pola hidup masyarakat dengan hadirnya perdagangan elektronik, kalangan akademis akan semakin diperkaya dengan kajian-kajian psikologis, antropologis, sosial budaya, dan sebagainya.
4. Meningkatkan Kualitas Sumberdaya Manusia. Perdagangan elektronik ini yang nantinya akan merangsang dan memacu orang-orang untuk mempelajari teknologi komputer demi kepentingan mereka sendiri.

2.7.5 Kerugian *E-commerce*

Di samping segala hal yang menguntungkan di atas, *E-Commerce* juga memiliki sedikit kerugian, Adi Nugroho (2006) menyebutkan [7].

a. Meningkatkan Individualisme. Pada *E-Commerce* seseorang dapat bertransaksi tanpa bertemu dengan penjual produk/jasa tersebut, ini

menimbulkan beberapa orang menjadi berpusat pada diri sendiri (egois) dan tidak terlalu membutuhkan kehadirian orang lain.

b. Terkadang Menimbulkan Kekecewaan. Apa yang dilihat dari layar komputer kadang berbeda dengan apa yang dilihat secara kasat mata.
c. Tidak Manusiawi. Di *internet*, meski kita dapat mengobrol (*chatting*) dengan orang lain, kita mungkin tidak dapat merasakan jabat tangannya, senyuman ramahnya atau candanya.

2.8 Konsep Dasar Basis Data

Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya[2].

Basis data terdiri dari 2 kata yaitu basis dan data. Basis kurang lebih berarti sebagai markas/gudang/tempat bersarang/tempat berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia, hewan, pristiwa, keadaan dan sebagainya yang direkam dalam bentuk angka, huruf, symbol, teks, bunyi atau kombinasinya[2].

Basis data sendiri dapat didefinisikan dari sejumlah sudut pandang seperti :

a. Himpunan kelompok data(arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
b. Kumpulan data yang saling berhubungan yang saling disimpan secara bersama sedemikian rupa tanpa pengulangan (redudansi) yang tidsk perlu untuk memenuhi berbagai kebutuhan.

c. Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

Basis data dan lemari arsip sesungguhnya memiliki prinsip kerja dan tujuan yang sama. Prinsip utamanya adalah pengaturan data dan tujuan utamanya adalah kemudahan dan kecepatan dalam pengembalian data. Perbedaannya hanya terletak pada media penyimpanan yang digunakan. Maka basis data menggunakan media penyimpanan elektronik seperti disk (disket atau *hardisk*). Hal ini karena lemari arsip langsung dikelola oleh manusia sementara basis data dikelola melalui perantara alat atau mesin pintar elektronis (yang kita kenal komputer).

Satu hal yang harus diperhatikan, bahwa basis data bukan hanya sekedar penyimpanan data secara elektronis (dengan bantuan komputer), artinya tidak semua bentuk penyimpanan data secara elektronis bisa disebut basis data. Kita dapat menyimpan dokumen berisi data dalam file teks (dengan program pengolah data), file *spread sheet* dan lain-lain. Tetapi tidak bisa disebut basis data karena didalamnya tidak ada pemilahan dan pengelompokan data sesuai jenis dan fungsi data. Yang sangat ditonjolkan basis data adalah pengaturan, pemilahan, pengelompokan, pengorganisasian data yang akan kita simpan sesuai dengan fungsinya.

Jenjang data dari basis data menurut modul pengajaran basis data sebagai berikut [4].

a. *Database* : Kumpulan dari *file*/tabel membentuk suatu *database*.

b. *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.
c. *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.
d. *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*.
e. *Characters* merupakan bagian data yang terkecil, dapat berupa karakter numerik, huruf ataupun karakter-karakter khusus (*special character*) yang membentuk suatu *item* data/*field*.

2.8.1 Operasi Dasar Basis Data

Di dalam sebuah disk, basis data dapat diciptakan dan dapat pula ditiadakan. Sementara dalam sebuah basis data, kiata dapat menempatkan satu ataulebih file ataupun tabe. Setiap basis data pada umumnya dibuat untuk mewakili sebuah basis data sebuah semesta data yang spesifik[4].

Operasi-operasi dasar yang dapat kita lakukan berkenaan dengan basis data dapat meliputi:

1. Pembuatan basis data baru (*create database*), yang identik dengan pembuatan lemari arsip yang baru.

2. Penghapusan basis data (*drop database*), yang identik dengan perusakan lemari arsip (sekaligus dengan isinya, jika ada).
3. Pembuatan file atau tabel dari suatu basis data (*create table*), yang identik dengan penambahan map arsip baru ke sebuah lemari arsip yang telah ada.
4. Penghapusan File atau tabel dari suatu basis data (*drop table*), yang identik dengan perusakan map arsip lama yang ada disebuah lemari arsip .
5. Penambahan atau pengisian data baru kesebuah file atau tabel disebuah basis data (*insert*) yang identik dengan penambahan lembaran arsip kesebuah map arsip.
6. Pengambilan data dari sebuah file atau tabel (*retrieve* atau *search*) yang identik dengan pencarian lembaran arsip dari sebuah map arsip.
7. Pengubahan data dari sebuah file atau tabel (*update*), yang identik dengan perbaikan isi lembaran arsip yang ada disebuah map arsip.
8. Penghapusan data dari sebuah file atau tabel (*delete*) yang identik dengan penghapusan sebuah lembaran arsip yang ada disebuah map arsip.

2.8.2 Sistem Pengelolaan Basis Data *(DBMS)*

Pengelolaan basis data secara fisik tidak dilakukan oleh pemakai secara langsung, tetapi ditangani oleh sebuah perangkat lunak (sistem) yang khusus. Perangkat lunak ini disebut (DBMS) yang akan menentukan bagaimana data diorganisasikan, disimpan, diubah, dan diambil kembali, ia juga menerapakan mekanisme penagamanan data, pemakaian data secara bersama, pemaksaan keakuratan atau konsistensi data dan sebagainya DBMS merupakan perantara bagi pemakai dengan basis data dalam *disk*[9].

2.8.3 Bahasa Basis Data

Cara beritnteraksi/berkomunikasi antara pemakai dengan basis data tersebut diatur dalam suatu bahasa khusus yang ditetapkan oleh perusahaan pembuat DBMS. Bahasa itu dapat kita sebut sebagai bahasa basis data yang terdiri atas sejumlah perintah *(statement)* yang diformulasikan dan dapat diberikan user dan dikenali/diproses oleh DBMS untuk melakukan suatu aksi/pekerjaan tertentu. Contoh-contoh bahasa basis data adalah SQL, dBase, QUEL dan sebagainya[2].

Sebuah bahasa basis data biasanya dapat dipilah ke dalam 2 bentuk yitu :

a. *Data Definition Language(DDL)* dan
b. *Data Manipulation Language(DML).*

2.8.3.1 *Data Definition Language (DDL)*

Struktur atau skema basis data secara keseluruhan dispesifikasikan dengan bahasa khusus yang disebut DDL*(Data Definition Language).* Dengan bahasa inilah kita dapat membuat table baru, indeks, mengubah table menentukan struktur penyimpanan table dsb. Hasil dari kompilasi perintah DDL adalah kumpulan table yang disimpan dalam file khusus yang disebut kamus data*(data dictionary)* [2]..

2.8.3.2 *Data Manipulation Language* **(*DML)***

Merupakan bentuk bahasa basis data yang berguna untuk melakukan manipulasi dan pengambilan data pada suatu basis data yang dapat berupa[2]. :

1. Penyisipan atau penambahan data barui ke suatu basis data.
2. Penghapusan data dari suatu basi data.
3. Pengubahan data dari suatu basis data.

2.9 Paypal

2.9.1 Pengertian paypal

PayPal adalah alat pembayaran elektronik global dan bebasis di Amerika, yang memungkinkan para pelaku kegiatan bisnis melakukan pembayaran dan transfer uang dilakukan melalui *internet*. Selain sebagai alat pembayaran, PayPal juga menyediakan jasa untuk situs *e-commerce*, situs lelangan maupun jenis usaha lainnya yang membutuhkan metode kirim dan terima uang yang mudah, cepat, dan aman. Perusahaannya sendiri bernama PayPal Inc. yang berpusat di San Jose, California, USA.

2.9.2 Sejarah Paypal

PayPal sebenarnya adalah sebuah perusahaan hasil merger Confinity dengan X.com pada tahun 2000. Confinity didirikan oleh Peter Thiel dan Max Levchin pada Desember 1998 dengan tujuan sebagai perantara pembayaran Palm Pilot dan juga penyedia layanan kriptografi. X.com didirikan oleh Elon Musk pada Maret 1999 sebagai penyedia jasa perencanaan keuangan. Kantor pusat kedua perusahaan ini awalnya terletak di 165 University Avenue, Palo Alto, California, USA.

Masa depan PayPal yang cukup menjanjikan, membuat eBay pada 2002 menjatuhkan pilihan kepada PayPal. PayPal diakusisi eBay sekitar USD 1.5 milliar, dan mulai sepenuhnya menggunakan PayPal. Efek dari berkembang pesatnya PayPal, membuat banyak pesaing menutup operasionalnya seperti Citibank C2IT (tutup tahun 2003), Yahoo! PayDirect (tutup tahun 2004), dan layanan BidPay (tutup tahun 2005). Sedangkan pesaing PayPal sekarang ini

adalah Liberty Reserve, AlertPay, Moneybookers, dan Kagi. Saat ini (2011), PayPal beroperasi di 190 negara, dan mengelola kurang lebih 232 juta rekening, dan sekitar 100 juta account PayPal adalah rekening aktif.

2.9.3 Jenis paypal

Jenis-jenis akun pada *Paypal* yang tersedia saat ini ada 3 jenis akun, diantaranya sebagai berikut:

a. 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 kartu kredit atau debit. Tidak ada biaya untuk setiap transaksi yang Anda lakukan pada tipe ini. Ada batasan 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.

b. 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*.

c. 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*.

Paypal sangatlah bermanfaat dalam hal perekonomian/perbisnisan di dunia internet, namun paypal juga tidak terlepas dari berbagai keuntungan dan kerugian dalam menggunakan paypal, berikut adalah kelebihan dan kekurangan dari paypal:

Keuntungan:

1. Paypal salah satu alat pembayaran secara online yang paling banyak digunakan di seluruh dunia
2. Paypal Bisa menjadi media yang nyaman dalam pengiriman dan penerimaan uang.
3. Dapat digunakan untuk membeli barang di eBay
4. Membeli barang atau jasa secara online, keanggotaan situs, urusan bisnis.
5. Mengirim uang/transaksi ke pengguna PayPal lain
6. Paypal bekerja secara mudah dan otomatis tanpa campur tangan manusia secara langsung.

7. PayPal tidak terbatas hanya untuk membayar namun juga bisa digunakan untuk menerima uang, memproses pembayaran dengan kartu kredit, dan keperluan bisnis lainnya.

Adapun kerugian dari paypal adalah sebagai berikut:

1. Penyalahgunaan PayPal ada dimana-mana (menjadi sasaran utama bagi para cracker)
2. Limited Access atau pembatasan Akun PayPal (pembatasan nilai transaksi)
3. Fee Termasuk Besar (Potongan setiap menerima transaksi 2,4% + $0,30 USD hingga 3,4% + $0,30 USD)

2.10 *Software* Pendukung

2.10.1 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, otentikasi 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

Pada awal mulanya, Apache merupakan perangkat lunak sumber terbuka yang menjadi alternatif dari *server web* Netscape (sekarang dikenal sebagai Sun Java *System Web Server*). Sejak April 1996 Apache menjadi *server web* terpopuler di *internet*. Pada Mei 1999, Apache digunakan di 57% dari semua *web server* di dunia. Pada November 2005 persentase ini naik menjadi 71%. (sumber: *Netcraft Web Server Survey*, November 2005).

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*). Saking 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.

2.10.2 *PHP* (*Hypertext Processor*)

PHP (*Hypertext Preprocessor)* adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam *HTML*. *PHP* banyak dipakai untuk memprogram situs *web* dinamis.

Pada awalnya *PHP* merupakan kependekan dari *Personal Home Page* (Situs personal). *PHP* pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu *PHP* masih bernama Form Interpreted (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari *web*.

Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya *PHP*/FI. Dengan perilisan kode sumber ini menjadi sumber terbuka, maka banyak pemrogram yang tertarik untuk ikut mengembangkan *PHP*.

Pada November 1997, dirilis *PHP*/FI 2.0. Pada rilis ini, interpreter *PHP* sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan *PHP*/FI secara signifikan.

Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang interpreter *PHP* menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis interpreter baru untuk *PHP* dan meresmikan rilis tersebut sebagai *PHP* 3.0 dan singkatan *PHP* diubah menjadi akronim berulang *PHP*: Hypertext Preprocessing.

Pada pertengahan tahun 1999, Zend merilis interpreter *PHP* baru dan rilis tersebut dikenal dengan *PHP* 4.0. *PHP* 4.0 adalah versi *PHP* yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi *web* kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

Pada Juni 2004, Zend merilis *PHP* 5.0. Dalam versi ini, inti dari interpreter *PHP* mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam *PHP* untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek.

Beberapa kelebihan *PHP* dari bahasa pemrograman *web*, adalah sebagai berikut:

a. Bahasa pemrograman *PHP* adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaanya.
b. *Web Server* yang mendukung *PHP* dapat ditemukan dimana - mana dari mulai apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah.
c. Dalam sisi pengembangan lebih mudah, karena banyaknya milis - milis dan *developer* yang siap membantu dalam pengembangan.
d. Dalam sisi pemahamanan, *PHP* adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.
e. *PHP* adalah bahasa open source yang dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara runtime melalui *console* serta juga dapat menjalankan perintah-perintah sistem.

2.10.2.1 Prinsip Kerja PHP

PHP dikatakan sebagai sebuah *server-side embedded script language* artinya sintaks-sintaks dan perintah yang kita berikan akan sepenuhnya dijalankan oleh server tetapi disertakan pada halaman HTML biasa. Aplikasi-aplikasi yang dibangun oleh PHP pada umumnya akan memberikan hasil pada web browser, tetapi prosesnya secara keseluruhan dijalankan di server.

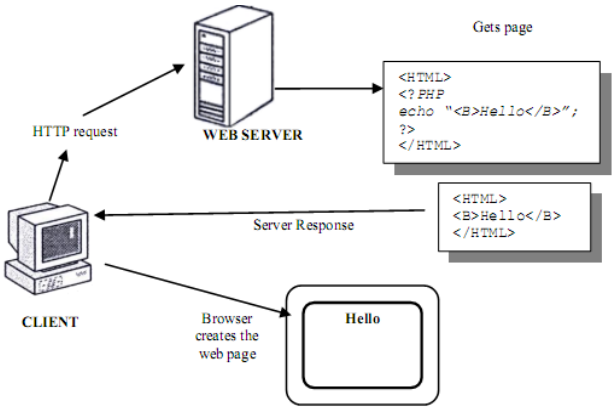
Pada prinsipnya server akan bekerja apabila ada permintaan dari client. Dalam hal ini client menggunakan kode-kode PHP untuk mengirimkan permintaan ke server (dapat dilihat pada gambar 2.8). Ketika menggunakan PHP sebagai *server-side embedded script language* maka server akan melakukan hal-hal sebagai berikut :

a. Membaca permintaan dari *client/browser*

b. Mencari halaman/*page* di *server*

c. Melakukan instruksi yang diberikan oleh PHP untuk melakukan modifikasi pada halaman/page.

d. Mengirim kembali halaman tersebut kepada *client* melalui internet atau intranet seperti terlihat pada gambar 2.13.



Gambar 2.13 Prinsip Kerja PHP

2.10.3 My SQL

Mysql adalah sebuah program database server yang mampu menerima dan mengirimkan data dengan sangat cepat multiuser serta menggunakan perintah standar *SQL(Structured Query Language)*. Sebagai database *server* yang memiliki konsep database modern, MySQL memiliki banyak kelebihan, diantaranya[8] :

1. *Protability*, MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FeeBSD, dan lain-lain.
2. *Open Source*, MySQL di distribusikan secara open source, sehingga dapat digunakan secara bebas.

3. *Multi User,* MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah. Hal ini memungkinkan sebuah database *server* MySQL dapat diakses *client* secara bersamaan.
4. *Performance Tuning,* MySQL memiliki kecepatan yang tinggi dalam menangani query. Dengan kata lain dapat memproses lebih banyak SQL persatuan waktu.
5. *Column Type*, MySQL memiliki tipe kolom yang sangat kompleks seperti signed, unsigned, integer, float, double, char, varchar, text, date, time, timestamp, year, sert dan enum.
6. *Commad dan Function*, MySQL, memilki operator dan fungsi secara penuh yang mendukung perintah select dan where dalam query.
7. *Security,* MySQL memiliki lapisan-lapisan sekuritas seperti level subnet mask, nama host dan izin akses *user* dengan sistem perizinan yang mendetail serta password terenkripsi.
8. *Scalability dan Limits*, MySQL mampu menangani database dengan skala besar dengan jumlah record lebih dari 50 juta dan 60 juta tabel serta 5 miliar baris. Selain itu, batas index yang dpat ditampung mencapai 32 indeks pada setiap tabelnya.
9. *Connectivity,* MySQL dapat melakukan koneksi dengan *client* menggunakan protokol TCP/IP, UNIX, SOCET atau name pipes (NT).
10. *Locallisation,* deteksi pesan kesalahan pada *client* dengan menggunakan lebih dari 20 bahasa.

11. *Interface*, terhadap berbagai aplikasi dan bahasa pemograman dengan menggunakan fungsi API (Application Programming Interface).
12. *Client dan Tools*, dilengkapi dengan berbagai tool yang dapat digunakan untuk administrasi database dan pada setiap tool yang ada disertakan petunjuk online.
13. Struktur Tabel, yang lebih fleksibel menangani ALTER TABLE, dibandingkan database lainnya semacam prostigre SQL maupun Oracel.

Tabel 2.3 Fungsi SQL

SQL Statement	**Syntax**
AND / OR	SELECT column_name(s) FROM table_name WHERE condition AND|OR condition
ALTER TABLE	ALTER TABLE table_name ADD column_name datatypeor ALTER TABLE table_name DROP COLUMN column_name
AS (alias)	SELECT column_name AS column_alias FROM table_nameor SELECT column_name FROM table_name AS table_alias
BETWEEN	SELECT column_name(s) FROM table_name WHERE column_name BETWEEN value1 AND value2
CREATE DATABASE	CREATE DATABASE database_name
CREATE TABLE	CREATE TABLE table_name (column_name1 data_type, column_name2 data_type, column_name2 data_type,)
CREATE INDEX	CREATE INDEX index_name

	ON table_name (column_name)or CREATE UNIQUE INDEX index_name ON table_name (column_name)
CREATE VIEW	CREATE VIEW view_name AS SELECT column_name(s) FROM table_name WHERE condition
DELETE	DELETE FROM table_name WHERE some_column=some_valueor DELETE FROM table_name (**Note:** Deletes the entire table!!) DELETE * FROM table_name (**Note:** Deletes the entire table!!)
DROP DATABASE	DROP DATABASE database_name
DROP INDEX	DROP INDEX table_name.index_name (SQL Server) DROP INDEX index_name ON table_name (MS Access) DROP INDEX index_name (DB2/Oracle) ALTER TABLE table_name DROP INDEX index_name (MySQL)
DROP TABLE	DROP TABLE table_name
GROUP BY	SELECT column_name, aggregate_function(column_name) FROM table_name WHERE column_name operator value GROUP BY column_name
HAVING	SELECT column_name, aggregate_function(column_name) FROM table_name WHERE column_name operator value GROUP BY column_name HAVING aggregate_function(column_name) operator value
IN	SELECT column_name(s) FROM table_name WHERE column_name IN (value1,value2,..)
INSERT INTO	INSERT INTO table_name VALUES (value1, value2, value3,....)*or* INSERT INTO table_name (column1, column2, column3,...)

	VALUES (value1, value2, value3,....)
INNER JOIN	SELECT column_name(s) FROM table_name1 INNER JOIN table_name2 ON table_name1.column_name=table_name2.column_name
LEFT JOIN	SELECT column_name(s) FROM table_name1 LEFT JOIN table_name2 ON table_name1.column_name=table_name2.column_name
RIGHT JOIN	SELECT column_name(s) FROM table_name1 RIGHT JOIN table_name2 ON table_name1.column_name=table_name2.column_name
FULL JOIN	SELECT column_name(s) FROM table_name1 FULL JOIN table_name2 ON table_name1.column_name=table_name2.column_name
LIKE	SELECT column_name(s) FROM table_name WHERE column_name LIKE pattern
ORDER BY	SELECT column_name(s) FROM table_name ORDER BY column_name [ASC|DESC]
SELECT	SELECT column_name(s) FROM table_name
SELECT *	SELECT * FROM table_name
SELECT DISTINCT	SELECT DISTINCT column_name(s) FROM table_name
SELECT INTO	SELECT * INTO new_table_name [IN externaldatabase] FROM old_table_name*or* SELECT column_name(s) INTO new_table_name [IN externaldatabase] FROM old_table_name

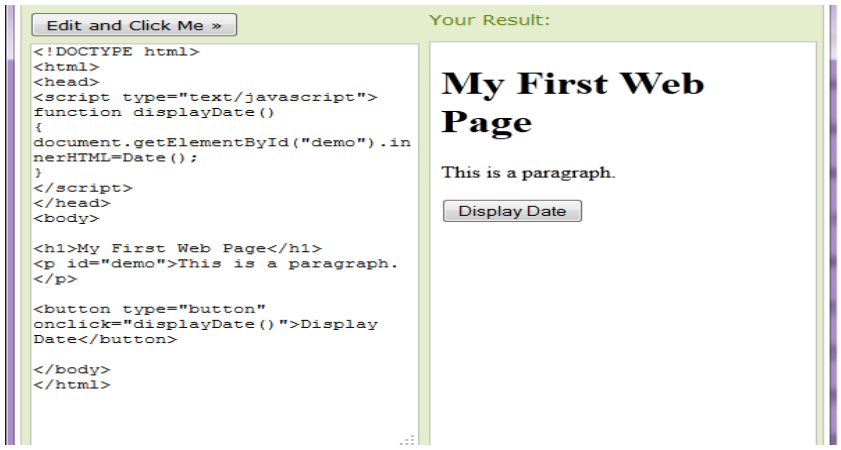
SELECT TOP	SELECT TOP number|percent column_name(s) FROM table_name
TRUNCATE TABLE	TRUNCATE TABLE table_name
UNION	SELECT column_name(s) FROM table_name1 UNION SELECT column_name(s) FROM table_name2
UNION ALL	SELECT column_name(s) FROM table_name1 UNION ALL SELECT column_name(s) FROM table_name2
UPDATE	UPDATE table_name SET column1=value, column2=value,... WHERE some_column=some_value
WHERE	SELECT column_name(s) FROM table_name WHERE column_name operator value

2.10.4 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[17].

JavaScript mendukung semua sintaks pemrograman terstruktur di bahasa C (misalnya, if laporan, while loop, switch laporan, 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. Contoh script CSS dapat dilihat pada gambar 2.14.



Gambar 2.14 Contoh *Script* Javascript

2.10.5 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 *source*-nya, 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.

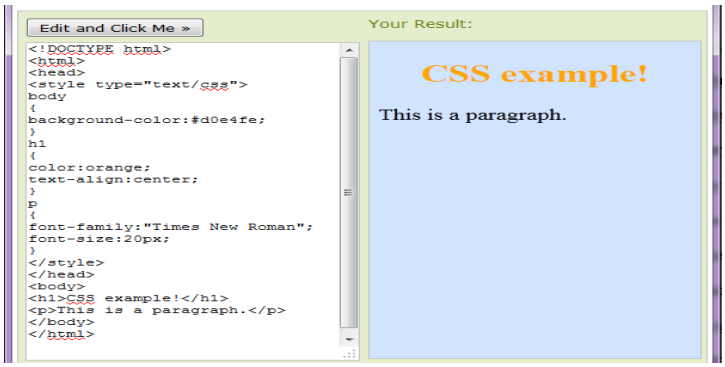
2.10.6 *CSS* (*Cascading Style Sheet*)

CSS 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 X*HTML*

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*). Contoh script css dapat dilihat pada gambar 2.15.



Gambar 2.15 Contoh *Script* CSS

2.10.7 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.

2.11 Browser Yang Digunakan

2.11.1 Mozila Firefox

Mozilla Firefox (aslinya bernama Phoenix dan kemudian untuk sesaat dikenal sebagai Mozilla Firebird) adalah peramban web lintas platform gratis yang dikembangkan oleh Yayasan Mozilla dan ratusan sukarelawan.

Sebelum rilis versi 1.0-nya pada 9 November 2004, Firefox telah mendapatkan sambutan yang sangat bagus dari pihak media, termasuk dari *Forbes* dan *Wall Street Journal*. Dengan lebih dari 5 juta *download* dalam 12 hari pertama rilisnya dan 6 juta hingga 24 November 2004, Firefox 1.0 adalah salah satu perangkat lunak gratis, sumber terbuka (*open-source*) yang paling banyak digunakan di antara pengguna rumahan.

Melalui Firefox, Yayasan Mozilla betujuan untuk mengembangkan sebuah peramban web yang kecil, cepat, sederhana, dan sangat bisa dikembangkan (terpisah dari Mozilla Suite yang lebih besar). Sejak 3 April 2003, Firefox dan

klien surel Thunderbird telah menjadi fokus utama pengembang Yayasan Mozilla untuk menggantikan Mozilla Suite.

Di antara fitur populer Firefox adalah pemblokir pop-up yang sudah terpasang di dalamnya, dan sebuah mekanisme pengembangan (*extension*) untuk menambah fungsionalitas tambahan. Meskipun fitur-fitur ini sudah tersedia untuk beberapa lamanya di peramban-peramban web lainnya seperti Mozilla Suite dan Opera, Firefox merupakan peramban web pertama yang mendapatkan penerimaan dalam skala sebesar ini. Firefox ditargetkan untuk mendapat sekitar 10% pangsa pasar Internet Explorer keluaran Microsoft (peramban web terpopuler dengan *margin* yang besar (per 2004) hingga tahun 2005, yang telah disebut oleh banyak orang sebagai tahun kembalinya perang peramban web.

Firefox telah mendapatkan perhatian sebagai alternatif kepada Internet Explorer sejak Explorer dikecam karena tuduhan ketidakamanannya pihak yang setuju terhadap anggapan ini mengatakan explorer tidak mengikuti standar *web*, menggunakan komponen ActiveX yang sering membahayakan, dan kelemahannya terhadap pemasangan *spyware* dan *malware* dan kurangnya fitur-fitur yang dianggap pemakai Firefox penting. Microsoft sendiri telah merespons bahwa mereka tidak menganggap jika isu-isu mengenai keamanan dan fitur Explorer perlu dikhawatirkan.

Firefox dapat dijalankan pada berbagai macam sistem operasi seperti Microsoft Windows, Linux, Mac OS X, dan FreeBSD. Versi stabil terbaru Firefox saat ini adalah 13.0.1, yang dirilis pada 15 Juni 2012

2.11.2 Google Chrome

Chrome adalah sebuah peramban web sumber terbuka yang dikembangkan oleh Google dengan menggunakan mesin *rendering* WebKit. Proyek sumber terbukanya sendiri dinamakan *Chromium.*

Versi beta untuk Microsoft Windows diluncurkan pada 2 september 2008 dalam 43 bahasa. Versi Mac OS X dan Linux sudah dirilis.

Tabel 2.4 Versi Google Chrome

Warna	Versi
Merah	Rilis Versi Lama
Hijau	Rilis Versi Terbaru
Biru	Rilis Versi Beta
Ungu	Rilis Versi dev

2.11.3 Internet Explorer

Windows Internet Explorer (dulu dikenal sebagai Microsoft Internet Explorer, disingkat IE atau MSIE) adalah sebuah peramban web dan perangkat lunak tak bebas yang gratis dari Microsoft, dan disertakan dalam setiap rilis sistem operasi Microsoft Windows sejak 1995. Pada mulanya, Internet Explorer dirilis sebagai bagian dari paket Plus for Windows 95 (Inggris) pada saat itu.Internet Explorer digunakan secara luas sejak tahun 1999, meliputi 95% dari keseluruhan penggunaan peramban web selama tahun 2002 hingga tahun 2003 dengan Internet Explorer 5 dan Internet Explorer 6. Lama kelamaan, ia kehilangan pangsa pasar sejak terlibat dengan kompetisi peramban web, dan kini meliputi 55,92%. Microsoft telah menghabiskan dana hingga 100 juta dolar per tahun untuk mengembangkan Internet Explorer pada tahun 1990an, dengan 1.000 orang yang bekerja di dalamnya.

Sejak rilis pertamanya, Microsoft telah menambahkan fitur dan teknologi seperti penampilan tabel dasar (di versi 1.5), XMLHttpRequest (Inggris) (di versi 5) yang membantu pembuatan halaman web, dan Nama Domain yang Terinternasionalisasi (di versi 7) yang memungkinkan situs web yang menggunakan alamat berkarakter non-Latin (seperti karakter Kanji) diketikkan secara normal pada address bar Internet Explorer.

Rilis stabil Internet Explorer paling baru adalah Internet Explorer 9, yang tersedia sebagai pembaharuan gratis bagi Windows 7, Windows Vista, dan Windows Server 2008.

2.12 Teori Kuisioner (Skala Likert)

Skala *Likert* digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomena sosial. Dalam penelitian, fenomena sosial ini telah ditetapkan secara spesifik oleh peneliti, yang selanjutnya disebut sebagai variabcl penelitian. Dengan Skala Likert, variabel yang akan diukur dijabarkan menjadi indikator variabel. Kemudian indikator tersebut dijadikan sebagai titik tolak untuk menyusun item-item instrumen yang dapat berupa pertanyaan atau pernyataan. Jawaban setiap item instrumcn yang ntcnggunakan skala Likert mempunyai gradasi dari sangat positif sampai sangat negatif, yang dapat berupa kata-kata antara lain: Sangat setuju, setuju, ragu-ragu, tidak setuju, sangat tidak setuju, sangat positif, positif, negative, sangat negative, selalu, sering, kadang-kadang, tidak pernah, sangat baik, baik, tidak baik, sangat tidak baik.

Untuk keperluan analisis kuantitatif, maka jawaban itu dapat diberi skor, misalnya:

1. Setuju/selalu/sangat positif diberi skor 5
2. Setuju/sering/positif diberi skor 4
3. Ragu-ragu/kadang-kadang/netral diberi skor 3
4. Tidak setuju/hampir tidak pernah/negatif diberi skor 2
5. Sangat tidak setuju/tidak pemah diberi skor 1

Instrumen penelitian yang menggunakan skala Likert dapat dibuat dalam bentuk checklist ataupun pilihan ganda.

a. Contoh Bentuk Checklist:

Berilah jawaban pertanyaan berikut sesuai dengan pendapat anda, dengan cara memberi tanda (V) pada kolom yang tersedia seperti pada tabel 2.5.

Tabel 2.5 Contoh pilihan ganda pada kuisioner

No	Pertanyaan	Jawaban				
		SS	ST	RG	TS	STS
1.2.	Prosedur kerja yang baru itu akan segera diterapkan perusahaan anda.		V			

SS = Sangat Setuju Diberi skor 5
ST = Setuju Diberi skor 4
RG = Ragu-ragu Diberi skor 3
TS = Tidak setuju Diberi skor 2
STS = Sangat Tidak Setuju Diberi skor 1

Kemudian dengan teknik pengumpulan data angket, maka instrumen tersebut misalnya diberikan kepada 100 orang karyawan yang diambil secara random. Dari 100 orang pegawai setelah dilakukan analisis misalnya:

25 Orang menjawab SS

40	Orang menjawab	ST
5	Orang menjawab	RG
20	Orang menjawab	TS
10	Orang menjawab	STS

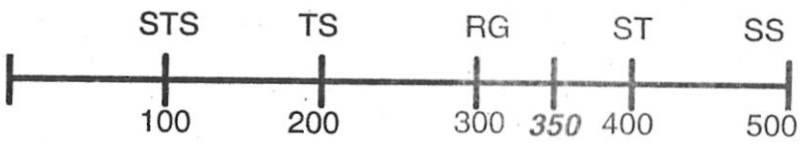
Berdasarkan data tersebut 65 orang (40 + 25) atau 65% karyawan menjawab setuju dan sangat setuju. Jadi kesimpulannya mayoritas karyawan setuju dengan adanya metode kerja baru.

Data interval tersebut juga dapat dianalisis dengan menghitung rata-rata jawaban berdasarkan skoring setiap jawaban dari responden. Berdasarkan skor yang telah ditetapkan dapat dihitung sebagai berikut:

Jumlah skor untuk 25 orang menjawab SS	= 25 x 5 = 125
Jumlah skor untuk 40 orang menjawab ST	= 40 x 4 = 160
Jumlah skor untuk 5 orang menjawab RG	= 5 x 3 = 15
Jumlah skor untuk 20 orang menjawab TS	= 20 x 2 = 40
Jumlah skor untuk 10 orang menjawab STS	= 10 x 1 = 10
Jumlah	= 350

Jumlah skor ideal (kriterium) untuk seluruh item = 5 x 100 = 500 (seandainya semua menjawab SS). Jumlah skor yang diperoleh dari penelitian = 350. Jadi berdasarkan data itu maka tingkat persetujuan terhadap metode kerja baru itu = (350 : 500) x 100% = 70% dari yang diharapkan (100%)

Secara kontinum dapat digambarkan seperti pada gamabar 2.16 sebagai berikut:



Gambar 2.16 Skor Hasil Kuisioner

Jadi berdasarkan data yang diperoleh dari 100 responden maka rata-rata 350 terletak pada daerah setuju.

b. Contoh bentuk pilihan ganda

Berilah salah satu jawaban terhadap pertanyaan berikut sesuai dengan pendapat anda, dengan cara memberi tanda lingkaran pada nomor jawaban yang tersedia.

Prosedur kerja yang baru itu akan segera diterapkan di lembaga anda?

a. Sangat tidak setuju
b. Tidak setuju
c. Ragu-ragu/netral
d. Setuju
e. Sangat setuju

Dengan bentuk pilihan ganda itu, maka jawaban dapat diletakkan pada tempat yang berbeda-beda. Untuk jawaban di atas "sangat tidak setuju" diletakkan pada jawaban nomor pertama. Untuk item selanjutnya jawaban "sangat tidak setuju" dapat diletakkan pada jawaban nomor terakhir.

Dalam penyusunan instrumen untuk variabel tertentu, sebaiknya butir-butir pertanyaan dibuat dalam bentuk kalimat positif, netral atau negatif, sehingga responden dapat menjawab dengan serius dan konsisten.

Contoh:

1. Saya mencintai mobil Diesel karena hemat bahan bakar (positif)-
2. Mobil Diesel banyak diproduksi di Jepang (netral).
3. Mobil Diesel sulit dihidupkan di tempat dingin (negatif).

Dengan cara demikian maka kecenderungan responden untuk menjawab pada kolom tertentu dari bentuk *checklist* dapat dikurangi. Dengan model ini juga responden akan selalu membaca pertanyaan setiap item instrumen dan juga jawabannya. Pada bentuk checklist, sering jawaban tidak dibaca, karena letak jawaban sudah menentu. Tetapi dengan bentuk checklist, maka akan didapat keuntungan dalam hal ini singkat dalam pembuatannya, hemat kertas, mudah mentabulasikan data, dan secara visual lebih menarik. Data yang diperoleh dari skala tersebut adalah berupa data interval.[18]