BAB 2

TINJAUAN PUSTAKA

2.1 Tinjauan Perusahaan

2.1.1 Profil Perusahaan

Toko Gideon Musik yang terletak di jalan gadjah mada no 238 jember merupakan usaha yang bergerak dalam bidang penjualan alat-alat musik.Toko Gideon Musik berdiri sejak tahun 2000 ,yang dimiliki oleh seorang yang bernama Gideon sebagai pemilik langsung toko gideon musik.

Pada awalnya toko ini tidak begitu berkembang Karena tidak dikenal toko Gideon musik di sekitar masyarakat jember tapi seiring berjalannya waktu dan pelayanan yang baik kepada konsumen sehingga toko ini dikenal oleh konsumen pencinta musik.

2.1.2 Visi Dan Misi

2.1.2.1 Visi

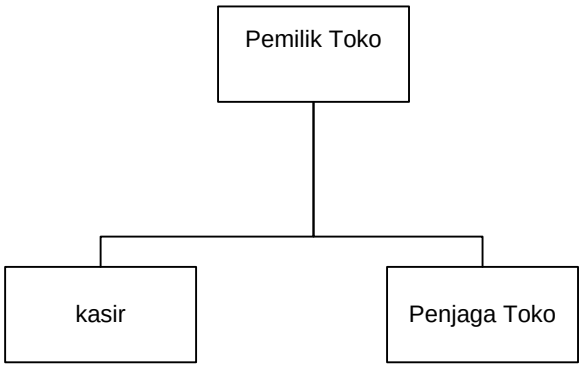
Menjadi toko yang profesional, berkarakter, inovatif, produktif yang dapat menghasilkan nilai tambah dan menjadi toko nomor satu di indonesia dalam penjualan alat-alat musik.

2.1.2.2 Misi

Misi dari toko gideon musik adalah mempromosikan produk-produk yang berkualitas tinggi, yang memiliki manfaat untuk kelangsungan permusikan khususnya di indonesia.

2.1.3 Struktur Organisasi

Ada 3 orang yang berperan penting dalam struktur organisasi di toko gideon musik. Untuk lebih jelasnya akan dijabarkan pada gambar 2.1 sebagai berikut:



Gambar 2.1 Struktur Organisasi

2.1.4 Deskripsi Struktur Organisasi

Berikut ini adalah deskripsi dari struktur organisasi di toko Gideon Musik yaitu:

1. Pemilik toko, mempunyai tugas sebagai pemeriksa laporan data pembelian barang dan laporan penjualan barang hingga pembuatan data laba tiap minggu,bulan dan tahun pada toko gideon musik.
2. Kasir toko, mempunyai tugas melayani konsumen secara langsung, pembuatan nota penjualan 2 rangkap untuk disimpan di kasir dan untuk konsumen, pembuatan buku laporan data barang, pembuatan buku laporan penjualan, pembuatan laporan pembelian barang dan pembuatan laporan stok barang.

3. Penjaga toko, mempunyai tugas melayani konsumen langsung, menjaga barang-barang yang akan di jual agar tetap rapi.

2.2 Tinjauan Pustaka

2.2.1 Konsep Dasar Sistem

2.2.1.1 Pengertian Sistem

Sistem adalah kumpulan dari komponen-komponen yang saling terkait dan bekerja bersama ke arah pencapaian tujuan dengan menerima *input* dan menghasilkan *output* di dalam sebuah proses transformasi atau perubahan yang terorganisir [7].

Sistem pada dasarnya adalah:

1. Sekelompok *user* yang erat berhubungan satu dengan yang lainnya, yang berfungsi bersama-sama untuk mencapai tujuan tertentu.
2. Merupakan sekumpulan sumber daya yang berhubungan untuk mencapai tujuan bersama.

Di lain pihak ada dua kelompok dalam mendefinisikan sistem, yaitu yang menekankan pada prosedurnya dan yang menekankan pada komponennya.

Pendekatan sistem yang lebih menekankan pada prosedurnya mendefinisikan sistem sebagai berikut, suatu sistem adalah jaringan kerja dari beberapa prosedur yang saling berhubungan, berkumpul bersama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu.

Pendekatan sistem yang lebih menekankan pada komponen atau elemennya mendefinisikan sistem sebagai berikut, sistem adalah sekelompok

elemen yang terintegrasi dan berinteraksi dengan maksud yang sama untuk mencapai suatu tujuan tertentu.

Adapun definisi sistem menurut Van de Poel yang diterjemahkan oleh Winardi adalah: "sebagai sekumpulan elemen dimana terdapat adanya hubungan-hubungan". Kerapkali dalam literatur dapat diketemukan kata-kata tambahan elemen-elemen mana ditunjukan ke arah pencapaian sasaran umum tertentu[12].

Dari pengertian diatas dapat diuraikan bahwa suatu sistem terdiri dari sekumpulan elemen yang saling berhubungan satu sama lain yang memiliki pencapaian sasaran yang telah ditentukan,

Sedangkan Shrode dan Voich dalam Lukman mengatakan bahwa "*system is whole compounded of several parts...*" yang dapat diartikan, sistem adalah suatu keseluruhan yang tersusun dari sekian banyak bagian. Senada dengan pendapat diatas maka Awad menyatakan bahwa "*system is an organized, funtioning, relationship among unit or components*". Yang dapat diartikan, sistem adalah hubungan yang berlangsung diantara satuan-satuan atau komponen secara teratur [7].

Riggs dalam Winardi mengemukan pengertian sistem sebagai suatu kelompok yang terdiri dari beberapa komponen atau bagian yang saling berinteraksi dalam hubungan yang dinamis.

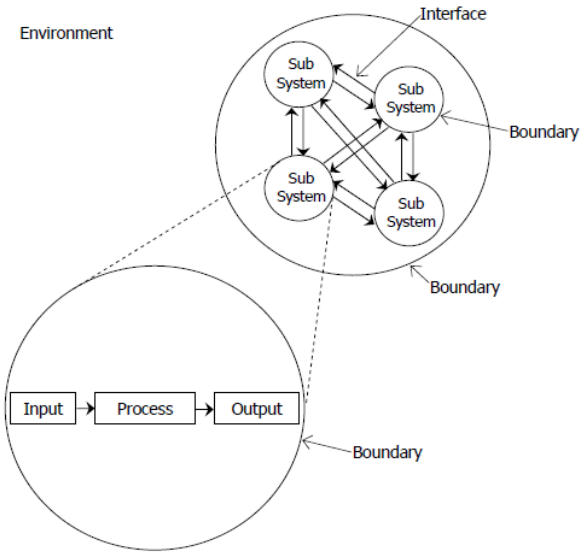
Kemudian Johnson, Katzs dan Rosenzwig dalam Lukman mendefinisikan "sistem sebagai suatu kebulatan atau keseluruhan yang kompleks atau terorganisir; suatu himpunan atau perpaduan hal-hal atau bagian-bagian yang membentuk suatu kebulatan atau keseluruhan yang kompleks atau utuh" [12].

Selanjutnya menurut Winardi menyatakan bahwa pada intinya “sistem menunjuk pada dua hal, yaitu pertama menunjukan suatu identitas, suatu wujud benda (abstrak maupun konkrit, termasuk yang konseptual) yang memiliki susunan struktural dari bagian-bagiannya ; kedua menunjukan suatu metode atau tata cara yang menunjuk pada suatu rencana metode, alat atau tata cara untuk mencapai sesuatu”.

Sejalan dengan pendapat di atas, Lukman mengartikan sistem sebagai wujud (entitas). Suatu sistem seringkali dianggap sebagai suatu himpunan bagian yang saling berhubungan sehingga merupakan suatu kesatuan. Contoh wujud (entitas) adalah lembaga pemerintah yang berkewajiban memberi pelayanan yang berkualitas (pelayanan prima) kepada pelanggannya.

2.2.1.2 Karakterisik Sistem

Sistem itu sendiri memiliki karakterisitik atau beberapa sifat tertentu, yaitu mempunyai komponen (*components*), batas sistem (*boundary*), lingkungan luar sistem (*environments*), penghubung (*interface*), masukan (*input*), keluaran (*output*), pengolah (*proses*), dan sasaran suatu tujuan (*goal*) [7]. Adapun penjelasan dari karateristik dari suatu sistem adalah sebagai berikut:



Gambar 2.2 Karakteristik Sistem

a. Komponen sistem (*Components*)

Bagian sistem yang saling berinteraksi dan membentuk satu kesatuan. Komponen sistem dapat berupa subsistem atau beberapa bagian sistem.

b. Batas sistem (*Boundary*)

Daerah yang membatasi antara suatu sistem dengan lingkungannya atau dengan sistem lainnya. Batas sistem inilah yang membuat sistem dipandang sebagai satu kesatuan.

c. Lingkungan Luar Sistem (*Environments*)

Segala sesuatu yang berada diluar sistem yang mempengaruhi sistem. Lingkungan luar sistem dapat bersifat menguntungkan sistem atau merugikan sistem.

d. Penghubung Sistem (*Interface*)

Merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Penghubung inilah yang menyebabkan beberapa subsistem berintegrasi dan membentuk satu kesatuan.

e. Masukan Sistem (*Input*)

Sesuatu yang dimasukkan ke dalam sistem yang berasal dari lingkungan.

f. Keluaran Sistem (*Output*)

Suatu hasil dari proses pengolahan sistem yang dikeluarkan ke lingkungan.

g. Pengolah Sistem (*Process*)

Bagian dari sistem yang mengubah masukan (*input*) menjadi keluaran (*output*).

h. Sasaran Sistem (*Objectives*) atau Tujuan (*Goal*)

Sasaran sistem adalah sesuatu yang menyebabkan mengapa sistem itu dibuat atau ada. Suatu sistem dikatakan berhasil bila tepat sasaran atau tujuannya.

2.2.2 Konsep Dasar Data

2.2.2.1 Pengertian Data

Data sering disebut sebagai bahan mentah informasi. Tapi menurut Murdick, dkk (1984) merumuskan bahwa "data adalah fakta yang tidak sedang digunakan pada proses keputusan, biasanya dicatat dan diarsipkan tanpa maksud untuk segera diambil kembali untuk pengambilan keputusan" [10].

Menurut Zulkifli Amsyah (1987) “data adalah fakta yang sudah ditulis dalam bentuk catatan atau direkam ke dalam berbagai bentuk media” [8].

Adapun definisi dari kata data adalah suatu istilah majemuk dari datum yang berarti fakta atau bagian dari kata yang mengandung arti, yang berhubungan dengan kenyataan, simbol-simbol, gambar-gambar, kata-kata angka-angka, huruf-huruf atau simbol-simbol yang menunjukkan ide, objek, kondisi atau situasi. Jelasnya data itu dapat berupa apa saja dan dapat ditemui dimana saja. Kegunaan data adalah sebagai bahan dasar yang objektif dalam proses penyusunan kebijakan dan keputusan. Dalam kaitannya dengan pengolahan data dengan komputer, pengertian data dapat dibatasi pada fakta-fakta yang dapat direkam. Dalam setiap pengolahan data, data merupakan sumber informasi yang dapat dihasilkan

2.2.2.2 Model Data

Data yang disimpan menggambarkan beberapa aspek dari suatu organisasi. Model data adalah himpunan deksripsi data level tinggi yang dikonstruksi untuk menyembunyikan beberapa detail dari penyimpanan level rendah. Beberapa manajemen basis data didasarkan pada model data relasional, model data hierarki, atau model data jaringan.

2.2.2.3 Model Data Hierarki

Model hirarki biasa disebut model pohon, karena menyerupai pohon yang dibalik. Model ini menggunakan pola hubungan orang tua-anak. Setiap simpul (biasa dinyatakan dengan lingkaran atau kotak) menyatakan sekumpulan medan. Simpul yang terhubung ke simpul pada level di bawahnya disebut orang

tua. Setiap orang tua bisa memiliki satu (hubungan 1:1) atau beberapa anak (hubungan 1:M), tetapi setiap anak hanya memiliki satu orang tua. Simpul-simpul yang dibawahi oleh simpul orang tua disebua anak. Simpul orang tua yang tidak memiliki orang tua disebut akar. Simpul yang tidak mempunyi anak disebut daun. Adapun hubungan antara anak dan orang tua disebut cabang.

2.2.2.4 Model Data Jaringan

Model jaringan distandarisasi pda tahun 1971 oleh DBTG (*DataBase Task Group*). Itulah sebabnya disebut model DBTG. Model ini juga disebut model CODASYL (*Conference on Data System Languages*), karena DBTG adalah bagian dari CODASYL.

Model ini menyerupai model hirarki, dengan perbedaan suatu simpul anak bisa memilki lebih dari satu orang tua. Oleh karena sifatnya demikian, model ini bisa menyatakan hubungan 1:1 (satu arang tua punya satu anak), 1:M (satu orang tua punya banyak anak), maupun N:M (beberapa anak bisa mempunyai beberapa orangtua). Pada model jaringan, orang tua diseut pemilik dan anak disebut anggota.

2.2.2.5 Model Data Relasional

Model relasional adalah model data yang paling banyak digunakan saat ini. Pembahasan pokok pada model ini adalah relasi, yang dimisalkan sebagai himpunan dari *record*. Deskripsi data dalam istilah model data disebut skema. Pada model relasional, skema untuk relasi ditentukan oleh nama, nama dari tiap *field* (atau atribut atau kolom), dan tipe dari tiap *field.*

2.2.3 Konsep Dasar Informasi

Informasi dalam sebuah organisasi sangat penting peranannya. Suatu sistem yang kurang mendapatkan informasi akan menjadi lemah dan akhirnya berakhir. Informasi itu sendiri dapat didefinisikan sebagai berikut, informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya.

Sumber dari informasi adalah data. Data merupakan bentuk jamak dari bentuk tunggal data atau data item. Data adalah kenyataan yang menggambarkan sesuatu yang terjadi pada saat tertentu.

2.2.3.1 Kualitas Informasi

Kualitas informasi diantaranya ditentukan oleh beberapa hal, yaitu :

1. Relevan

   Dalam hal ini informasi yang diterima harus memberikan manfaat bagi pemakainya. Kadar *relevancy* informasi antara orang satu dengan yang lainnya berbeda-beda tergantung kepada kebutuhan masing-masing pengguna informasi tersebut. *How is the message used for problem solving* (*decision masking*)?

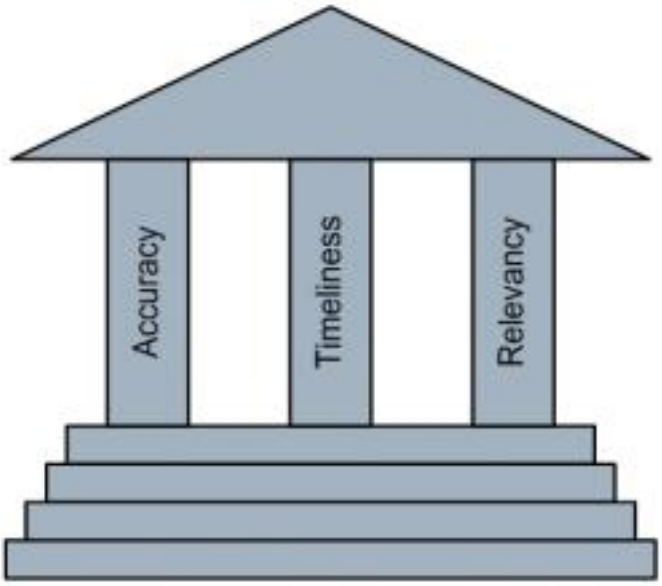
2. Akurat

   Informasi harus bebas dari kesalahan-kesalahan. Selain itu informasi yang didapatkan tidak boleh bias atau menyesatkan bagi penggunanya, serta harus dapat mencerminkan dengan jelas maksud dari informasi tersebut. Ketidak akuratan data terjadi karena sumber dari informasi tersebut mengalami gangguan dalam penyampaiannya baik hal itu dilakukan secara

sengaja maupun tidak sehingga menyebabkan data asli tersebut berubah atau rusak.

3. Tepat waktu

   Informasi yang dibutuhkan oleh si pemakai tidak dalam hal penyampaiannya tidak boleh terlambat (usang) karena informasi yang usang maka informasi tersebut tidak mempunyai nilai yang baik dan kualitasnya pun menjadi buruk sehingga tidak berguna lagi. Jika informasi tersebut digunakan sebagai dasar pengambilan keputusan maka akan berakibat fatal sehingga salah dalam pengambilan keputusan tersebut. Kondisi tersebut mengakibatkan mahalnya nilai suatu informasi, sehingga kecepatan untuk mendapatkan, mengolah serta mengirimnya memerlukan teknologi terbaru.



Gambar 2.3 Atribut Kualitas Informasi

Komponen keakuratan suatu informasi diantaranya:

1. *Completeness*

   Are necessary message items present? Hal ini dapat berarti bahwa informasi yang dihasilkan atau dibutuhkan harus memiliki kelengkapan yang baik, karena bila informasi yang dihasilkan sebagian-sebagian tentunya akan mempengaruhi dalam pengambilan keputusan atau menentukan tindakan secara keseluruhan, sehingga akan berpengaruh terhadap kemampuannya untuk mengontrol atau memecahkan suatu masalah yang terjadi dalam suatu organisasi tersebut.

2. *Correctness*

   Are message items correct? maksudnya bahwa informasi yang diterima kebenarannya tidak perlu diragukan lagi. Kebenaran dari informasi tersebut harus dapat dipertanggung jawabkan.

3. *Security*

   Did the message reach all or only the intended systems users? Informasi yang diterima harus terjamin keamanan datanya.

4. *Economy* (Ekonomis)

   What level of resources is needed to move information through the problem-solving cycle?. Kualitas dari Informasi yang digunakan dalam pengambilan keputusan juga bergantung pada nilai ekonomi yang terdapat didalamnya.

5. *Efficiency* (Efisien)

   What level of resources is required for each unit of information output? Informasi harus bisa dikonsumsi oleh semua kalangan.

6. *Reliability* (Dapat dipercaya)

   Informasi yang didapatkan oleh pemakai harus dapat dipercaya, hal ini menentukan terhadap kualitas informasi serta dalam hal pengambilan keputusan setiap tingkatan manajemen.

2.2.3.2 Nilai Informasi

Fungsi informasi yang dibutuhkan untuk pengambilan keputusan terkadang diperlukan dengan proses yang cepat dan tidak terduga. Hal itu mengakibatkan penggunaan informasi hanya berdasarkan perkiraan-perkiraan serta informasi yang apa adanya. Dengan perlakukan seperti ini mengakibatkan keputusan yang diambil tidak sesuai dengan yang diharapkan. Oleh karena itu untuk memperbaiki keputusan yang telah diambil maka pencarian informasi yang lebih tepat perlu dilakukan. Suatu Informasi memiliki nilai karena informasi tersebut dapat menjadikan keputusan yang baik serta menguntungkan (memiliki nilai informasi yang tepat). Besarnya nilai informasi yang tepat dapat didapatkan dari perbedaan hasil yang didapat dari keputusan yang baru dengan hasil keputusan yang lama dikurangi dengan biaya untuk mendapatkan informasi tersebut. Penghitungan atas informasi yang tepat memberikan banyak manfaat diantaranya untuk menghilangkan pemborosan biaya yang dilakukan untuk mendapatkan informasi yang dibutuhkan dalam pengambilan keputusan tersebut.

Menurut Gordon B. Davis nilai informasi dikatakan sempurna apabila perbedaan antara kebijakan optimal, tanpa informasi yang sempurna dan kebijakan optimal menggunakan informasi yang sempurna dapat dinyatakan dengan jelas [7].

Nilai suatu informasi dapat ditentukan berdasarkan sifatnya. Ada 10 sifat yang dapat menentukan nilai informasi yaitu sebagai berikut :

1. Kemudahan dalam memperoleh

   Informasi memperoleh nilai yang lebih sempurna apabila dapat diperoleh secara mudah. Informasi yang penting dan sangat dibutuhkan menjadi tidak bernilai jika sulit diperoleh.

2. Sifat luas dan kelengkapannya

   Informasi mempunyai nilai yang lebih sempurna apabila mempunyai lingkup/ cakupan yang luas dan lengkap. Informasi sepotong dan tidak lengkap menjadi tidak bernilai, karena tidak dapat digunakan secara baik.

3. Ketelitian (*accuracy*)

   Informasi mempunyai nilai yang lebih sempurna apabila mempunyai ketelitian yang tinggi/akurat. Informasi menjadi tidak bernilai jika tidak akurat, karena akan mengakibatkan kesalahan pengambilan keputusan.

4. Kecocokan dengan pengguna (*relevance*)

   Informasi mempunyai nilai yang lebih sempurna apabila sesuai dengan kebutuhan penggunanya. Informasi berharga dan penting menjadi tidak bernilai jika tidak sesuai dengan kebutuhan penggunanya, karena tidak dapat dimanfaatkan untuk pengambilan keputusan.

5. Ketepatan waktu

   Informasi mempunyai nilai yang lebih sempurna apabila dapat diterima oleh pengguna pada saat yang tepat. Informasi berharga dan penting menjadi tidak bernilai jika terlambat diterima/ usang, karena tidak dapat dimanfaatkan pada saat pengambilan keputusan.

6. Kejelasan (*clarity*)

   Informasi yang jelas akan meningkatkan kesempurnaan nilai informasi. Kejelasan informasi dipengaruhi oleh bentuk dan format informasi.

7. Fleksibilitas/ keluwesannya

   Nilai informasi semakin sempurna apabila memiliki fleksibilitas tinggi. Fleksibilitas informasi diperlukan oleh para manajer/pimpinan pada saat pengambilan keputusan.

8. Dapat dibuktikan

   Nilai informasi semakin sempurna apabila informasi tersebut dapat dibuktikan kebenarannya. Kebenaran informasi bergantung pada validitas data sumber yang diolah.

9. Tidak ada prasangka

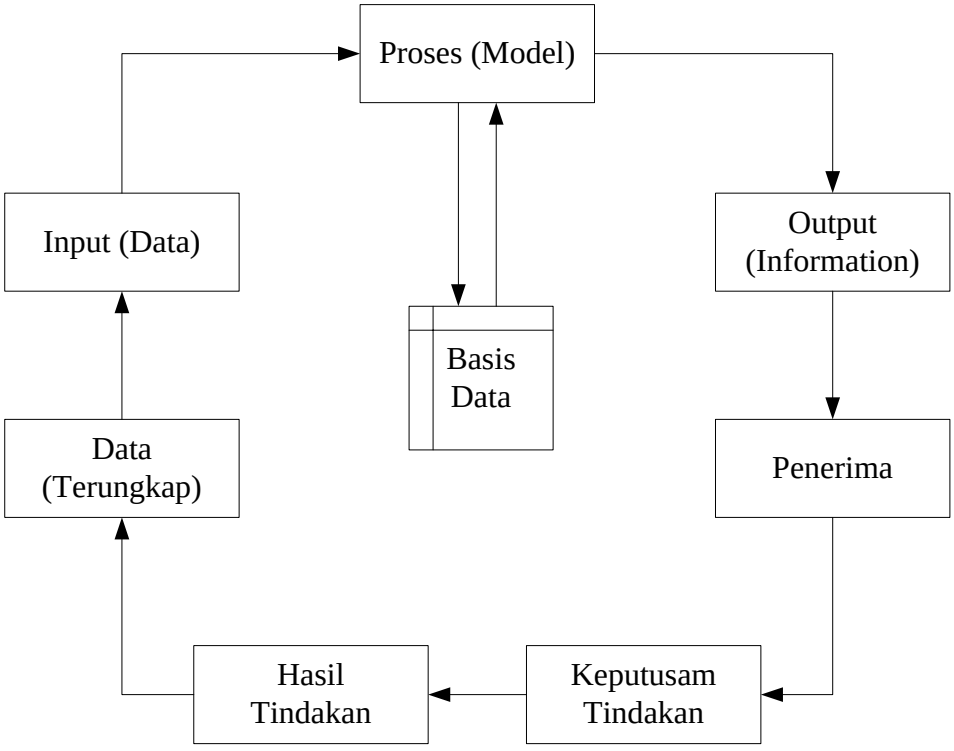
   Nilai informasi semakin sempurna apabila informasi tersebut tidak menimbulkan prasangka dan keraguan adanya kesalahan informasi.

10. Dapat diukur

    Informasi untuk pengambilan keputusan seharusnya dapat diukur agar dapat mencapai nilai yang sempurna.

2.2.3.3 Siklus Informasi

Untuk memperoleh informasi yang bermanfaat bagi penerimanya, perlu untuk dijelaskan bagaimana siklus yang terjadi atau dibutuhkan dalam menghasilkan informasi. Pertama-tama data dimasukkan ke dalam model yang umumnya memiliki urutan proses tertentu dan pasti, setelah diproses akan dihasilkan informasi tertentu yang bermanfaat bagi penerima (*level management*) sebagai dasar dalam membuat suatu keputusan atau melakukan tindakan tertentu, Dari keputusan atau tindakan tersebut akan menghasilkan atau diperoleh kejadian-kejadian tertentu yang akan digunakan kembali sebagai data yang nantinya akan dimasukkan ke dalam model (proses), begitu seterusnya. Dengan demikian akan membentuk suatu siklus informasi atau siklus pengolahan data, seperti gambar berikut :



Gambar 2.4 Siklus Informasi

2.2.4 Konsep Dasar Sistem Informasi

Informasi dapat diperoleh dari sistem informasi (*Information system*) atau disebut juga dengan *processing system* atau *information processing system* atau *information genereting system*. Sistem informasi didefinisikan oleh Robert A. Leitch dan K. Roscoe Davis sebagai berikut, "Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan yang diperlukan" [7].

Sistem informasi sendiri memiliki sejumlah komponen tertentu. Seperti yang dikemukakan oleh Robert dan Donald Symanzky, bahwa sistem informasi terdiri dari beberapa komponen yang berbeda yaitu, manusia, data, hardware, dan software. Sebagai suatu sistem, setiap komponen tersebut berinteraksi satu dengan lainnya membentuk satu kesatuan untuk mencapai sasarannya.

2.2.4.1 Komponen Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang disebut blok bangunan (*building block*), yang terdiri dari komponen *input*, komponen model, komponen *output*, komponen teknologi, komponen *hardware*, komponen *software*, komponen basis data, dan komponen kontrol. Semua komponen tersebut saling berinteraksi satu dengan yang lain membentuk suatu kesatuan untuk mencapai sasaran.

1. Komponen Input

   Input mewakili data yang masuk kedalam sistem informasi. Input disini termasuk metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2. Komponen Model

   Komponen ini terdiri dari kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yag sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

3. Komponen *Output*

   Hasil dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua pemakai sistem.

4. Komponen Teknologi

   Teknologi merupakan "*tool box*" dalam sistem informasi, Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, neghasilkan dan mengirimkan keluaran, dan membantu pengendalian dari sistem secara keseluruhan.

5. Komponen *Hardware*

   Hardware berperan penting sebagai suatu media penyimpanan vital bagi sistem informasi.Yang berfungsi sebagai tempat untuk menampung database atau lebih mudah dikatakan sebagai sumber data dan informasi untuk memperlancar dan mempermudah kerja dari sistem informasi.

6. Komponen *Software*

   Software berfungsi sebagai tempat untuk mengolah,menghitung dan memanipulasi data yang diambil dari hardware untuk menciptakan suatu informasi.

7. Komponen Basis Data

   Basis data (*database*) merupakan kumpulan data yang saling berkaitan dan berhubungan satu dengan yang lain, tersimpan di perangkat keras komputer dan menggunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan dalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data di dalam basis data perlu diorganisasikan sedemikian rupa supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyimpanannya. Basis data diakses atau dimanipulasi menggunakan perangkat lunak paket yang disebut DBMS (*Database Management System*)

8. Komponen Kontrol

   Banyak hal yang dapat merusak sistem informasi, seperti bencana alam, api, temperatur, air, debu, kecurangan-kecurangan, kegagalan-kegagalan sistem itu sendiri, ketidak efisienan, sabotase dan lain sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi.

2.2.5 Konsep Dasar Analisis Sistem

2.2.5.1 Unified Modeling Language (UML)

UML (unified Modelling Language) adalah salah satu alat bantu yang sangat handal didunia pengembangan sistem yang berorientasi obyek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membentuk cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (sharing) dan mengkomunikasikan rancangan mereka dengan yang lain [3].

UML merupakan kesatuan dari bahasa pemodelan yang dikembangkan oleh Booch, OMT (*Object Modeling Technique*) dan OOSE (*Object Oriented Sofware Enginering*). Metode Booch dari Grady Booch sangat terkenal dengan nama metode *Design Object Oriented*. Metode ini menjadikan proses ananlisis dan design kedalam empat tahapan iterative, yaitu: identifikasi kelas-kelas dan obyek-obyek, identifikasi semantik dari hubungann obyek dan kelas tersebut, perincian interface dan implementasi. Keungulan metode Booch adalah pada detil dan kayanya dengan notasi dan elemen [3]. Pemodelan OMT yang dikembangkan oleh Rumbaugh didasarkan pada analisis terstruktur dan pemodelan entity-relationship. Metode OOSE dari lacobson lebih member penekanan pada *use case* [13].

Dengan UML, metode Booch,OMT dan OOSE digabungkan dengan membuang elemen-elemen yang tidak praktis ditambah dengan elemen-

elemen dari metode lain yang lebih efektif dan elemen-elemen baru yang belum ada pada metode terdahulu sehingga UML lebih ekspresif dan seragam dari pada metode lainnya.[3]

UML menyediakan kemampuan untuk menangkap karakteristik sebuah sistem dengan menggunakan notasi-notasi tertentu. UML menyediakan sederetan notasi sederhana yang mudah dipahami untuk medokumentasikan sistem berdasarkan prinsip-prinsip perancangan berorientasi objek. UML sudah menjadi *standardized Modelling Language* yang terdiri dari kumpulan-kumpulan diagram, dikembangkan untuk membantu para pengembang sistem dan *software* agar bisa menyelesaikan tugas-tugas seperti:

1. Spesifikasi
2. Visualisasi
3. Desain arsitektur
4. Konstruksi
5. Simulasi dan testing
6. Dokumentasi

2.2.5.2 Model *Diagram* UML

UML terdiri atas banyak elemen-elemen grafis yang digabungkan membentuk diagram. Tujuan representasi elemen-elemen grafis ke dalam diagram adalah untuk menyajikan beragam sudut pandang dari sebuah sistem berdasarkan fungsi masing-masing diagram tersebut. Kumpulan dari beragam sudut pandang inilah yang disebut sebuah model [3].

2.2.5.2.1 *Class Diagram*

Class Diagram adalah sebuah spesifikasi yang jika diinstansikan akan menghasilkan sebuah obyek dan merupakan inti dari pengembangan dan desain berorientasi objek.[3]

Class menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, package dan object beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain. Sebuah *class* memiliki tiga area pokok:

1. Nama , merupakan nama dari sebuah kelas.
2. Atribut, merupakan properti dari sebuah kelas. Atribut melambangkan batas nilai yang mungkin ada pada obyek dari *class*.
3. Operasi, adalah sesuatu yang bisa dilakukan oleh sebuah atau yang dapat dilakukan oleh *class* lain terhadap sebuah *class*. Atribut dan metoda dapat memiliki salah sifat berikut:
    a. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan
    b. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya.
    c. *Public*, dapat dipanggil oleh siapa saja.
    d. *Package*, hanya dapat dipanggil oleh *instance* sebuah *class* pada paket yang sama.

Class juga dapat merupakan implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metode. *Interface* tidak dapat langsung

diintansikan, tetapi harus diimplementasikan dahulu menjadi sebuah class. Dengan demikian *interface* mendukung resolusi metoda pada saat *runtime.*

Relasi atau relationship merupakan keterhubungan antar kelas yang muncul pada saat sebuah kelas berinteraksi dengan kelas-kelas lainnya. Didalam *class diagram,* setiap kelas pasti akan berinteraksi dengan baik satu maupun lebih kelas. Relasi yang muncul pada setiap keterhubungan antar kelas juga memiliki atribut-atribut yang akan lebih menjelaskan defenisi dari sebuah relasi yang terjadi.

1. Asosiasi (*Association*)

   Para pemodel menggunakan pemahaman asosiasi adalah pada saat beberapa kelas saling terhubung satu sama lain secara konseptual. Asosiasi juga dapat menjadi lebih kompleks pada saat beberapa kelas terhubung ke satu kelas

2. *Constraint* pada Asosiasi

   Kadangkala sebuah asosiasi diantara dua class harus mengikuti sebuah aturan dan aturan ini bias diletakkan dalam sebuah *constraint* pada garis asosiasi dan diletakkan dalam kurung kurawal. Bentuk lain dari tipe *constraint* adalah relasi OR yang ditulis dengan {or} dalam garis putus-putus yang menghubungkan dua garis asosiasi. Kondisi OR ini menghadapkan kepada keadaan bahwa sebuah kelas terhubung dengan dua kelas tetapi pada saat mendefenisikan relasinya kelas harus memilih salah satu dari kedua kelas tersebut, sedangkan kondisi untuk memilih keduanya adalah *invalid.*

3. Kelas Asosiasi (*Association Class*)

   Sebuah asosiasi dapat memiliki atribut dan operasi sepertinya halnya sebuah class. Sebuah a*ssociation class* sebenarnya diperlukan apabila salah satu dari kelas yang terhubung mempunya sebuah atau beberapa atribut yang tidak layak dimiliki kelas tersebut, karena secara logis atribut tersebut lebih layak dimiliki oleh asosiasi yang menghubungkan kedua kelas tersebut.

4. Multiplisitas (Multiplicity)

   Multiplisitas atau kardinalitas adalah jumlah banyaknya objek sebuah *class* yang berelasi dengan sebuah objek lain pada class lain yang berasosiasi pada *class* tersebut. Untuk menyatakan multiplisitas anda dapat meletakkannya diatas garis asosiasi berdekatan dengan *class* yang sesuai. Ada banyak multiplisitas yang mingkin untuk dipakai. Tabel berikut menjabarkan multiplisitas yang dapat digunakan.

Tabel 2.1 Nilai Multiplisitas potensial

No	**Indicator**	**Meaning**
1.	0..1	Zero or one
2.	1	One only
3.	0..*	Zero or more
4.	1..*	One to more
5.	N	Only n (where n >1)
6.	0..n	Zero to n (where n > 1)
7.	1..n	One to n(where n > 1)

Sebuah class (*child class* atau *subclass*) dapat mewarisi atribut-atribut dan operasi-operasi dari class lainnya (*parent class atau super class)* dimana *parent class* bersifat lebih umum dari pada *child class.*

Generalisasi pada konsep *Object Oriented* digunakan untuk menjelaskan hubungan kesamaan diantara class. Dengan menggunakan generalisasi bisa dibangun struktur logis yang bisa menampilkan derajat kesamaan atau perbedaan diantara clas-class. Manfaat lain dari struktur hirarkis juga memungkinkan untuk penambahan *subclass (child class)* baru tanpa harus merubah struktur yang sudah ada.

Inheritance adalah sebuah mekanisme pengimplementasian *generalisasi* dan spesialisasi. Aturan *inheritance* dapat diklasifikasikan sebagai berikut

1. *Subclass* dapat mewarisi semua sifat dari *superclass.*
2. Defenisi *subclass* selalu mencakup paling tidak satu detil yang tidak diturunkan dari *supeclassnya.*

Inheritance sangat dekat dengan asosiasinya dengan *generalisasi*. *Generalisasi* menjelaskan hubungan logis antar elemen-elemen yang mempunyai karakteristik yang sama. Sedangkan *inheritance* menerangkan mekanisme agar bagi pakai (*sharing)* bisa terjadi.

Didalam UML, ada perlakuan khusus yang disebut dengan "bagian dai *(part of)*" yang menangani antar objek-objek dimana salah satunya adalah bagian dari yang lain. Dengan kata lain sebuah objek terdiri atas objek-objek yang lain. Hal sepeti ini di dalam UML disebut *agregasi.* Sebuah *agregasi* adalah kasus khusus dari *asosiasi. Agregasi* disimbolkan

dengan jajaran genjang yang diletakkan pada class yang mengandung objek. Kadangkala relasi OR harus digunakan pada *agregasi.* Untuk memodelkan hal tersebut, sebuah *constraint* dengan kalimat atau dalam tanda kurung kurawal pada garis putus-putus harus dibuat untuk menghubungkan dua objek.

Komposit *(composite)* adalah sebuah tipe *agregasi* yang kuat dimana bagian dari obyek tergantung penuh secara keseluruhan terhadap obyeknya sehingga bila sebuah obyek komposit dibuat maka bagian yang tergantung pada komponen tersebut akan terbuang juga pada saat yang bersamaan. Notasi komposit sama seperti agregasi hanya saja jajaran genjangnya terisi *(solid).*

Interface adalah satu set *operation* yang memberikan spesifikasi beberapa aspek dari perilaku dan operasi disuatu *class* ke *class* yang lain. Contohnya, keyboard pada komputer sebenarnya merupakan interface yang bisa dipakai ulang karena tombol-tombol *keyboard* sebenarnya berasal dari mesin ketik, hanya saja mungkin ada beberapa *operasionalisasi* tombol-tombol yang berbeda yang sudah ditransfer ke sistem yang lain. Pemodelan *interface* sama dengan pemodelan *class* hanya saja pada *interface* tidak mempunyai atribut dan pada penamaannya perlu ditambahkan *stereotype* <<*interface*>> atau ditambahkan huruf "I" di depan nama *interface.* Relasi antar *class* dan *interface* disebut *realization.* Realisasi dituliskan dengan garis putus-putus dengan segitiga yang mengarah ke *interface.*

2.2.5.2.2 *Use case Diagram*

Use case adalah deskripsi fungsi dari sebuah sistem dari *perspektif* / sudut pandang para pengguna sistem.*use case* mendefenisikan "apa" yang dilakukan oleh sistem dan elemen-elemennya saling berinteraksi [3]. *Use case bekerja dengan menggunakan "scenario",* yaitu deskripsi urutan-urutan langkah yang menerangkan apa yang dilakukan pengguna terhadap sistem maupun sebaliknya. *Use case* diagram mengindentifikasin fungsionalitas yang dipinyai oleh sistem *(use case),* user yang berinteraksi dengan sistem dan asosiasi/ keterhubungan antara user dengan fungsionalitas sistem.

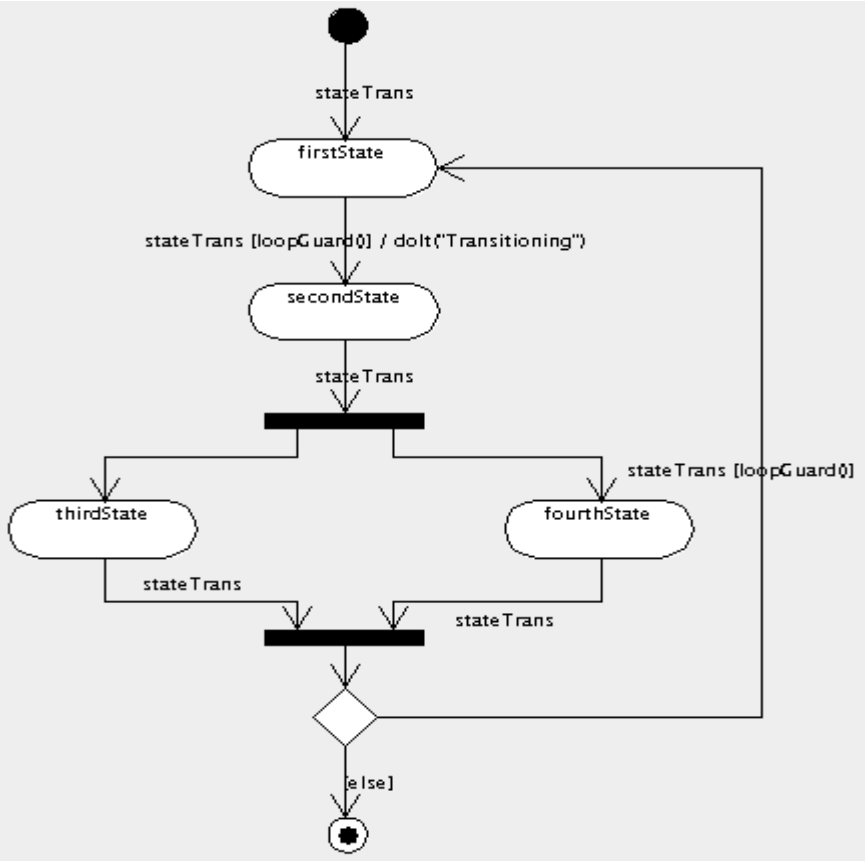
Fungsi dari diagram *use case* adalah menggambarkan fungsionalitas yang diharapkan dari sebuah sistem, dengan menekankan aspek-aspek apa yang dilakukan sistem bukan bagaimana sistem melakukannya. Sebuah *use case* merepresentasikan sebuah interaksi antara *actor* dengan sistem.

Berbicara mengenai *use case* diagram tidak akan terlepas dengan hal yang disebut *stereotype. Stereotype* adalah sebuah model khusus yang terbatas untuk kondisi tertentu. Untuk menunjukkan *stereotype* digunakan symbol "<<" diawalnya dan ">>" diakhirnya. Terdapat 2 *stereotype* paling sering digunakan dalam *use case* yaitu << *include* >> dan << *extend* >>. << *extend* >> digunakan untuk menunjukkan bahwa satu *use case* merupakan tambahan fungsional dari *use case* yang lain jika kondisi atau syarat tertentu dipenuhi. Sedangkan << *include* >> digunakan untuk

menggambarkan bahwa suatu *use case* seluruhnya merupakan *fungsionalitas* dari *use case* lainnya.

2.2.5.2.3 *Activity Diagram*

Activity diagram digunakan untuk mendokumentasikan alur kerja pada sebuah sistem, yang dimulai dari pandangan *business* level hingga ke *operational level* [3]. Pada dasarnya, *activity* diagram merupakan variasi dari statechart diagram. *Activity* diagram mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity* diagram bisa mendukung perilaku parallel sedangkan *flowchart* tidak bisa. Berikut adalah *activity diagram* :



Gambar 2.5 *Activity Diagram*

2.2.5.2.4 *Sequence Diagram*

Sequence diagram menurut Munawar adalah grafik dua dimensi dimana obyek ditunjukkan dalam dimensi horizontal, sedangkan *lifeline* ditunjukkan dalam dimensi vertikal[3].

2.2.5.2.5 *Component Diagram*

Komponen perangkat lunak adalah bagian fisik dari sebuah sistem yang menetap di komputer. komponen merupakan implementasi *software* dari sebuah *class*. Komponen bisa berupa tabel, file data, file exe, file DLL, dokumen dan lain-lain.

Component diagram mengandung komponen, *interface* dan *relationship*. Komponen diagram ini digunakan pada saat anda ingin memecah sistem menjadi komponen-komponen dan ingin menampilkan hubungan-hubungan mereka dengan antarmuka atau pemecahan komponen menjadi struktur yang lebih rendah. Secara umum dapat kita katakan bahwa *component* diagram kita gunakan untuk menjelaskan kebergantungan antar beragam komponen-komponen *software* seperti misalnya kebergantungan antara file-file *executable* dengan file-file sumbernya *(source file*) dll. Berikut adalah notasi dari *component diagram* :



Gambar 2.6 Contoh Notasi Component

2.2.5.2.6 *Deployment Diagram*

Deployment diagram menunjukkan tata letak sebuah sistem secara fisik, menampakkan bagian-bagian *software* yang berjalan pada bagian-bagian hardware yang digunakan untuk mengimplementasikan sebuah sistem dan keterhubungan antara komponen-komponen *hardware* tersebut [3]. *Deployment* diagram dapat digunakan pada bagian-bagian awal proses perancangan sistem untuk mendokumentasikan arsitektur fisik sebuah sistem. Berikut adalah notasi-notasi yang digunakan pada *deployment diagram* :

2.2.5.2.7 Collaboration Diagram

Collaboration diagram menurut Munawar adalah perluasan dari objek *diagram*. Objek diagram menunjukkan objek-objek yang hubungannya satu dengan yang lain. *Collaboration diagram* menunnjukkan *mesage-message* objek yang dikirim satu sama lain [3].

2.2.6 Framework

Framework adalah kumpulan perintah atau fungsi dasar yang membentuk aturan-aturan tertentu dan saling berinteraksi satu sama lain sehingga dalam pembuatan aplikasi *website,* kita harus mengikuti aturan dai *framework* tersebut [1]. Dengan *framework* (dalam hal ini *framework* PHP), kita tidak perlu memikirkan kode perintah/fungsi dasar dari aplikasi website kita. Seperti bagaimana mengambil data dari *database* untuk ditampilkan. Kita hanya memikirkan apa kode sqlnya dan ditampilkan kemana. Hal-hal penunjang lainnya seperti koneksi database, validasi

form, GUI, dan keamanan telah disediakan oleh *framework* sehingga jumlah baris kode yang kita buat jauh lebih sedikit dibandingkan jika semua kode dari kita. Jadi, keuntungan yang diperoleh dari pengguna *framework* adalah:

1. Waktu pembuatan aplikasi *website* kita jauh lebih singkat.
2. Kode aplikasi *website* lebih mudah dibaca, karena sedikit dan sifatnya pokok. Detailnya adalah kode dari *framework* dan ini terjamin.
3. *Website* kita menjadi lebih mudah diperbaiki.
4. Kita tidak perlu lagi membuat kode penunjang website seperti koneksi *database, validation form*, GUI, dan keamanan.

2.2.6.1 Manfaat Framework

1. Manfaatnya dapat membantu kerja *developer* dalam membangun aplikasi sehingga aplikasi bisa selesai dalam waktu yang singkat.
2. Penerapan *Design Pattern* memudahkan dalam rancangan, pengembangan dan pemeliharaan sistem.
3. *Stability* dan *Reliability* ,aplikasi yang kita bangun lebih stabil dan handal karena berbasis pada *framework* yang sudah teruji stabilitas dan kehandalannya.
4. *Coding Style* konsisten, memudahkan dalam membaca kode dan dalam menemukan *bugs*.
5. *Security Concern*, *framework* mengantisipasi dan memasang perisai terhadap adanya berbagaimasalah keamanan yang mungkin timbul.

6. Dokumnetasi, *framework* dapat mendisiplinkan kita untuk menulis dokumentasi untuk apa yang kita tulis.

2.2.6.2 Kelebihan *Framewok*

Pertama kelebihan dengan adanya *framework* akan lebih mempermudah memahami mekanisme kerja dari sebuah aplikasi. Ini tentunya akan sangat membantu proses pengembangan sistem yang dilakukan secara team. Semua anggota di wajibkan untuk memahami dari pola kerja *framework* tersebut selebihnya anggota team hanya mempelajari proses bisnis yang di kehendaki oleh system untuk kemudian di tuangkan kedalam *framework* tersebut. Dalam artian setiap orang harus mempunyai metode yang sama dalam menyelesaikan aplikasi tersebut. Kedua dengan memakai *framework* akan menghemat waktu pengerjaan suatu aplikasi, karena setiap anggota sudah memiliki sebuah acuan dalam menyelesaikan modul. Dalam hal ini misalnya semakin banyak *library* yang ada semakin mempercepat anggota untuk menemukan solusi karena tidak setiap anggota harus membuat *class* atau fungsi untuk kasus yang relatif sama [1].

Berikutnya Team tidak akan di susahkan dengan adanya perputaran anggota dalam artian jika ada anggota yang tidak bisa melanjutkan lagi pekerjaannya anggota yang lain bisa meng-cover kekosongan tersebut. Bayangkan jika setiap modul yang dikembangkan mempunyai logika yang berbeda tentunya akan memakan waktu yang banyak untuk proses pemahan akan sistem tersebut.

Kesekian kalinya dengan adanya *framework* akan menjaga integritas dari modul-modul yang dikembangkan. Tentunya hal ini juga tergantung dengan metode yang dikembangkan sendiri. *Framework* hanya membantu dan memungkinkan/mempermudah proses integrasi,tidak berarti dengan adanya *framework* sistem otomatis akan terintegrasi.

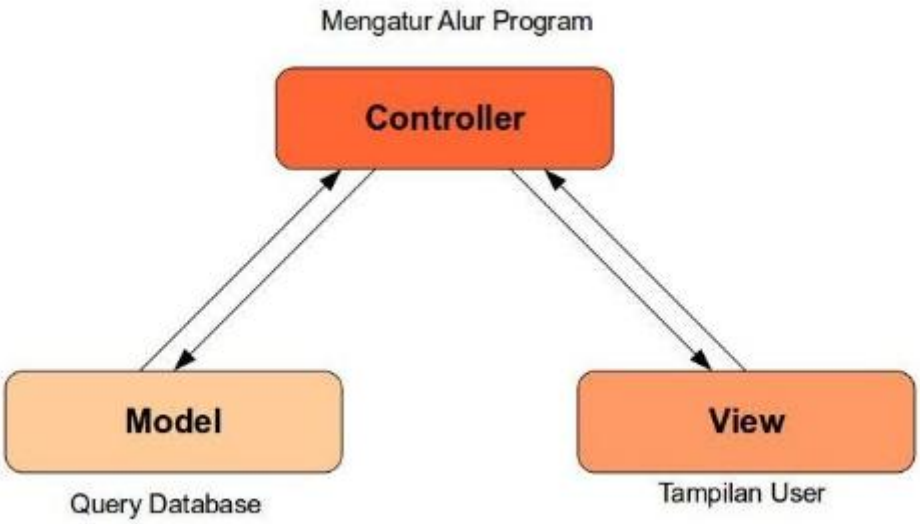
2.2.6.3 Kekurangan dari *Framework*

a. Para *programmer* mungkin akan menemukan batasan-batasan ketika merancang aplikasi menggunakan *framework*.
b. Kemungkinan akan menambah biaya *development* apabila *framework* yg digunakan kurang terdokumentasi dan kurang di *support*.
c. Performa dan kecepatan eksekusi.

2.2.6.4 *Framework* mendukung konsep MVC

MVC merupakan implementasi dari teknik pemrograman OOP (*Object Oriented Programming)*. Sedangkan MVC itu sendiri adalah konsep pemrograman, dimana logika pemrograman , *query database* dan tampilannya dipisahkan [1]. Sehingga kode-kode penyusun sebuah aplikasi web terlihat lebih sederhana dan terstruktur. Jadi, tidak akan ada *query* database di dalam sebuah file yang memang ditujukan untuk tampilan.

Gambaran tentang konsep MVC, kira-kira seperti gambar di bawah ini :



Gambar 2.7 Gambaran Konsep MVC

MVC (*Model, View, Controller*) yaitu:

a. ***Model*** adalah sebuah layer pada MVC yang merepresentasikan data yang digunakan oleh aplikasi sesuai dengan proses yang terjadi pada data tersebut, dengan memilahnya menjadi beberapa bagian terpisah kembali, seperti penampungan data, *persistence*, proses manipulasi, dan lain-lain. Secara singkat, model bertugas untuk menangani *query-query* database.

b. ***View*** adalah sebuah layer pada MVC yang mengandung keseluruhan detai dari implementasi *user interface* dengan melibatkan komponen grafis yang menyediakan representasi proses internal aplikasi dan meuntun alur interaksi user terhadap aplikasi.

c. ***Controller*** adalah sebuah layer pada MVC yang menyediakan detai alur program dan transisi layer, dan bertanggung jawab akan penampungan

event yang dibuat oleh *user* dari *view* dan melakukan update terhadap komponen model menggunakan data dari *user*.

2.2.6.5 *Framework Codeigniter*

CodeIgniter (CI) sebuah *Framework* untuk *programmer* yang ingin membangun *website* dengan PHP. CI merupakan langkah cepat untuk membuat sebuah *website* daripada membangun sebuah *website* dengan alur *coding* yang biasa. Dengan mempaketkan *library-library* ,alur pemrograman akan terstruktur dan logika bisnis/proses akan lebih mudah dimengerti oleh *developer*/*programmer* lain. CI telah mempaketkan itu semua sehingga mampu meminimalisir jumlah *coding* yang dibangun[1].

2.2.6.6 Kelebihan *Codeigniter*

Adapun kelebihan-kelebihan pada *codeigniter* adalah sebagai berikut:

a. *CodeIgniter* merupakan *framework* yang gratis, dimana kita bisa mengunduh dan menginstalnya tanpa harus membayar.
b. *CodeIgniter* sangat ringan,dimana sistem utamanya hanya memerlukan sedikit sekali *library*. Berbeda dengan kebanyakan *framework* yang membutuhkan banyak *resources*.
c. *Framework* yang cepat dibandingkan framework PHP lainnya.
d. Menggunakan pendekatan konsep MVC (*Model-Visual-Controller*),sehingga pengembangan aplikasi lebih teratur dan terstruktur.
e. *CodeIgniter* dapat dengan mudah diperluas kemampuannya sesuai dengan yang kita inginkan melalui penggunaan *plugin* dan *helper*.

f. Konfigurasi yang sangat minim (*nearly zero configuration*) : tentu saja untuk menyesuaikan dengan database dan keleluasaan routing tetap diizinkan melakukan konfigurasi dengan mengubah beberapa file konfigurasi seperti *database.php* atau *autoload.php*, namun untuk menggunakan *codeigniter* dengan *setting standard*, anda hanya perlu merubah sedikit saja file pada *folder config*.

g. Banyak komunitas: dengan banyaknya komunitas CI ini, memudahkan kita untuk berinteraksi dengan yang lain, baik itu bertanya atau teknologi terbaru.

h. Dokumentasi yang sangat lengkap : Setiap paket instalasi *codeigniter* sudah disertai *user guide* yang sangat bagus dan lengkap untuk dijadikan permulaan, bahasanya pun mudah dipahami.

2.2.6.7 Kekurangan *Codeigniter*

a. Meskipun *Open Source CodeIgniter* tidak dikembangkan oleh komunitas tapi oleh EllisLab (pengembang *Expression Engine*) sehingga membuat update dari *core engine* tidak secepat *Framework* yang dikembangkan oleh komunitas.

b. Jika terjadi *error* yang berhubungan dengan *Framework*, mau tidak mau kita harus *searching* dan *browsing* masalah tersebut.

c. Bukan untuk kelas *enterprise*, meskipun bisa di *extend* dengan tersedianya banyak *library* dari komunitas.

d. Tidak ada *support*, tentu saja tidak ada support karena *open source*. Sebagai gantinya anda harus sering-sering mampir di forum-forum *codeigniter*.

2.2.7 Electronic Commerce (E-Commerce)

2.2.7.1 Sejarah Perkembangan E-Commerce

E-commerce merupakan suatu cara berbelanja atau berdagang secara *online* atau *direct selling* yang memanfaatkan fasilitas Internet dimana terdapat *website* yang dapat menyediakan layanan "*get and deliver*"[8].

Pada awalnya ketika web mulai terkenal di masyarakat pada 1994, banyak jurnalis memperkirakan bahwa *e-commerce* akan menjadi sebuah sektor ekonomi baru. Namun, baru sekitar empat tahun kemudian *protokol* aman seperti HTTPS memasuki tahap matang dan banyak digunakan. Antara 1998 dan 2000 banyak bisnis di AS dan Eropa mengembangkan situs web perdagangan ini.

2.2.7.2 Kelebihan E-Commerce

Adapun kelebihan dari *E-Commerce* antara lain adalah :

1. Jangkauan lebih luas (dunia). Tanpa batas-batas wilayah dan waktu.
2. Penghematan sumber daya
3. Ruang untuk toko (fisik) dan SDM
4. Availabilitas, buka 24 jam sehari, 7 hari seminggu. Tidak mengenal hari libur, dan hari besar

5. *Skalabilitas*, dapat diperluas atau diperbanyak item barang tanpa batasan.
6. *No Tax,* belum jelas regulasi mengenai pajak
7. Konsumen memperoleh informasi yang beragam dan mendetail.
8. Melalui internet konsumen dapat memperoleh aneka informasi barang dan jasa dari berbagai toko dalam berbagai variasi merek lengkap dengan spesifikasi harga, cara pembayaran, cara pengiriman
9. *Disintermediation* adalah proses meniadakan calo dan pedagang perantara.

2.2.7.3 Kekurangan E-Commerce

Selain kelebihannya, *e-commerce* pun memiliki beberapa kekurangan seperti :

1. Isu *security*
2. Pembajakan kartu kredit, *stock exchange fraud*, *banking fraud*, hak atas kekayaan intelektual, akses ilegal ke sistem informasi (*hacking*) perusakan *website* sampai dengan pencurian data.
3. Ketidak sesuaian jenis dan kualitas barang yang dijanjikan,
4. Ketidak tepatan waktu pengiriman barang
5. *No cash payment*.
6. Indonesia belum memiliki perangkat hukum yang mengakomodasi perkembangan *e-commerce.*
7. Masalah kultur, yaitu sebagian masyarakat kurang merasa puas bila tidak melihat langsung barang yang akan dibelinya.

2.2.7.4 Manfaat E-Commerce

Manfaat dari digunakannya *E-Commerce* ini adalah dapat menekan biaya barang dan jasa, serta dapat meningkatkan kepuasan konsumen sepanjang yang menyangkut kecepatan untuk mendapatkan barang yang dibutuhkan dengan kualitas yang terbaik sesuai dengan harganya.

2.2.7.5 Klasifikasi E-Commerce

Kita mengenal adanya *pure e-commerce* dan *partial e-commerce*. Suatu *e-commerce* dikategorikan *pure* atau *partial* berdasarkan pada tingkat digitasi dari suatu produk yang diperdagangkan, proses, dan agen pengirimannya. Apabila segala aspek dalam sistem *e-commerce* itu digital maka dapat dikategorikan sebagai *pure e-commerce*. Selain itu, ciri lain dari *pure e-commerce* adalah organisasi penyelenggara benar-benar organisasi *online*, menggunakan model bisnis *new-economy organization*, dan menjual produk atau jasanya hanya secara *online*. Sedangkan, *partial e-commerce* dicirikan dengan penggabungan antara aspek digital dan tradisional/fisik, penggunaan model bisnis *click-and-mortar organization* (penggabungan antara *offline* dan *online*), serta melakukan kegiatan-kegiatan bisnis utamanya di dunia nyata.

Selain itu, klasifikasi *e-commerce* berdasarkan kecenderungan bertransaksi yakni *Business to Business* (B2B), *Business to Consumer* (B2C), *Business to Business to Consumer* (B2B2C). Penggiat dari B2B adalah antar perusahaan. Sedangkan, pada B2C, suatu perusahaan menjual produk/jasa mereka kepada pembeli individual. Contoh dari B2C adalah

penjualan mobil bermerek ABC kepada konsumennya. B2B2C adalah gabungan dari keduanya, di mana terdapat perusahaan yang menyediakan produk/jasa mereka kepada klien/agen dan di lain sisi klien/agen tersebut juga mempunyai *customer* tempat mereka menjual produk/jasa tersebut.

Hal yang menarik dari B2B dan B2C adalah bahwa B2B ternyata memiliki prospek yang lebih menguntungkan dilihat dari peta *e-commerce* dunia. Data *Forrester Research Inc*. menunjukan transaksi B2B di tahun 1998 US$43 milyard dengan tingkat pertumbuhan 99% yang mencapai US$1.3 trilyun di tahun 2003. Sedangkan B2C bisa dilihat dari pembelanjaan iklan di Internet yang hanya US$ 2.8 Milyard di tahun 1998 dan meningkat ke US$ 22 Milyard di tahun 2004. Statistik *Forrester Research Inc*. Ini menunjukan kesimpulan yang sederhana yaitu bahwa B2B akan jauh lebih menguntungkan dibandingkan B2C.

Selain B2B, B2C, dan B2B2C, klasifikasi *e-commerce* berdasarkan kecenderungan bertransaksi lainnya adalah *Consumer to Business* (C2B) dan *Consumer to Consumer* (C2C). C2B terjadi ketika seorang individu menjual produk/jasa mereka melalui internet kepada sebuah perusahaan. Sedangkan, C2C adalah ketika seorang individu melakukan penjualan produk/jasa langsung kepada individu lainnya. Menurut saya, C2C adalah model *e-commerce* yang menjamur di Indonesia saat ini. Contoh dari C2C adalah iklan baris dan toko-toko buku *online* dadakan (dimiliki oleh individu yang umumnya memanfaatkan layanan blog gratis seperti multiply).

Selain itu, ada pula yang namanya *Business to Employee* (B2E) dan *Government to Citizens* (*G2C/E-Government*). Pada B2E, suatu perusahaan yang menyediakan informasi mengenai suatu produk/jasa yang dijual kepada karyawannya. Sedangkan pada G2C, pemerintahlah yang menyediakan informasi, produk/jasa kepada masyarakatnya atau kepada perusahaan-perusahaan yang ada dalam pemerintahan negara tersebut.

Sebenarnya, masih banyak lagi klasifikasi *e-commerce* berdasarkan kecenderungan bertransaksi, antara lain *Mobile Commerce* (*m-commerce*), *Location Commerce* (*l-commerce*), *Collaborative Commerce* (*c-commerce*), dan *Exchange to Exchange* (E2E)[8].

2.2.7.6 Komponen Utama *E-Commerce*

Berikut ini bebereapa komponen utama pada *e-commerce* :

1. *Electronic Data Intechange* (EDI)

   Electronic Data Interchange (EDI) didefinisikan sebagai pertukaran data komputer antar berbagai bidang organisasi atas suatu informasi terstruktur dalam format yang standar dan bisa diolah oleh komputer. EDI merupakan bentuk *e-commerce* sesuai definisinya, dan telah ada bentuk yang sama selama lebih dari 20 tahun. Saat ini teknologi dan implementasi EDI sudah sangat berkembang. Tujuan EDI adalah untuk memfasilitasi perdagangan dengan cara mengikat bisnis antar *partner* dagang, EDI meningkatkan proses manual untuk mempertukarkan informasi dengan bidang bisnis lainnya dalam berbagai cara, misalnya data hanya perlu untuk dimasukkan satu kali saja, kemudian data tersebut bisa

digunakan oleh pihak pengirim barang, *manager* kantor, dan lain-lainnya. Hal ini akan menurangi tenaga *entry data*. Pada dasarnya, data bisa dikirimkan dengan lebih efisien dengan menggunakan EDI[8].

Komponen utama dari EDI standar adalah sebagai berikut :

a. *Data Element*, merupakan potongan data seperti tanggal, harga atau nama organisasi. Setiap data element diidentifikasikan dengan nomor referensi tertentu yang berisi judul, keterangan, jenis, nomor, dan panjang minimum/maximum.
b. *Data Segment*, dalam suatu baris data disebut dengan segment dan setiap item di dalam segmen mewakili satu elemen. Misalnya segmen baris pesanan pembelian terdiri atas nomor barang, keterangan, jumlah, unit pengukuran, dan harga barang. Setiap segmen memiliki satu *identifier*, satu data *element deimiter*, *element diagrams*, data segmen *terminator* dan *notes*.
c. *Transaction Set*, suatu *transation set* merupakan dokumen khusus seperti dokumen pesanan pembelian. Di dalam *transaction set*, ada 3 area utama: area *header*, area *detail* dan area *summary*.
d. *Functional Group*, merupakan sekelompok *transaction set* yang sejenis. *Transation set* di dalam *functional group* dikelompokkan berdasarkan *functional identifier* yang sama. Untuk mengirimkan transaksi EDI pada konsumen, diperlukan 4 fungsi dasar: *Mapping elemen* dalam suatu *database*, *Extraction* atas data yang belum diidentifikasi dari *database*, *Transalation* atas data yang sudah

diekstrak ke format EDI, dan Transmisi pesan dalam format EDI melalui media komunikasi. Berikut ini adalah penjelasan singkat dari masing-masing fungsi tersebut. *Mapping*, Merupakan proses identifikasi elemen di dalam *database* yang diperlukan untuk membuat pesan dalam format EDI. *Mapping* adalah pekerjaan yang hanya satu kali dilakukan pada saat diperlukan transaksi EDI baru. *Software* EDI tidak bisa melaksanakan pekerjaan ini. *Extraction*, merupakan proses pengumpulan data yang belum diidentifikasi dan menempatkannya ke dalam format tertentu. Secara umum, data *extract* dari *database* dan dijadikan dalam bentuk *flat file*. Struktur dari *flat file* biasanya ditentukan oleh pembuat *translation software*. *Translation*, untuk mengirimkan pesan keluar, ketika data yang diperlukan masih dalam bentuk *flat file*, pembentukan pesan EDI bisa dilakukan menggunakan *software* translasi atau *formatting*. *Software* translasi akan mengatur data menjadi struktur tertentu yang sesuai dengan kebutuhan transaksi EDI. *Communication*, pengiriman/transmisi atas pesan EDI dikendalikan oleh s*oftware* komunikasi, yang akan mengatur dan memelihara: nomor telepon partner dagang, menjalankan *automatic dialing* dan *up/downloading*, juga membuat *activity log*. Setiap pesan EDI dibungkus dengan amplop khusus yang bertuliskan alamat tujuan, serta jenis transaksi EDI sebagai header dan *error checking codes* sebagai tambahan di

bawahnya. Untuk keperluan penerimaan pesan EDI, proses tersebut tinggal dibalik.

2. *Digital Currency*

Digital currency dimaksudkan untuk memungkinkan *user* untuk memindahkan dananya secara elektronik dalam lingkungan kerja tertentu. Saat ini, *digital currency* dirancang untuk versi elektronik dari uang kertas, dimana memiliki atribut yang sama dengan media fisik sebenarnya baik secara anatomis maupun dari segi likuiditasnya.

Karakteristik *digital currency* adalah sebagai berikut:

a. Mewakili suatu nilai moneter tertentu
b. Bisa ditukarkan sebagai alat pembayaran untuk barang dan jasa, mata uang dan koin serta token lainnya
c. Bisa disimpan dan diambil lagi.
d. Sulit diduplikasi atau dipalsukan.

Jenis-jenis *digital currency* antara lain:

a. Electronic Cash

Sistem *electronic cash* telah terintegrasi sepenuhnya dengan *software web browser* untuk memudahkan pembelian barang melalui internet. Sistem *electronic cash* bisa menunjukkan saldo terakhir pada user tertentu sesuai permintaan. *Electronic cash* pada umumnya memerlukan infrastruktur *public key* dan mekanisme enkripsi tertentu. Saat ini *electronic cash* belum sepopuler pengunaan *smart card* atau model pembayaran lainnya.

b. Micropayment

Micropayments adalah pembayaran untuk item dengan nilai relative rendah, misalnya informasi atau hiburan *on-line* yang biayanya bervariasi antara 1 *cent* sampai 10 *cent*. Sedangkan *Minipayment* adalah pembayaran untuk item dengan nilai antara $ 0,25 sampai $ 10. Ada beberapa skema yang mampu menangani *micropayments* yaitu: *Milicent, eCash, CyberCoin, Mondex, VisaCash* dan *NetBill*.

3. Electronic Catalogs

Electronic Catalogs (e-catalogs) telah berada pada aplikasi komersil yang dirancang untuk internet dan merupakan komponen utama dari sistem *e-commerce*. *E-catalogs* merupakan antar muka grafis (*Graphical User Interface*) yang umumnya berbentuk halaman WWW dimana menyediakan informasi tentang penwaran produk dan jasa. *e-catalog* umumnya mendukung *online shopping* dan kemampuan pemesanan dan pembayaran barang. Suatu *website* bias juga merupakan suatu koleksi katalog, misalnya *electronik mall* merupakan suatu katalog atas katalog. Aplikasi *e-catalog* sebaiknya memiliki karakteristik seperti: bersifat interaktif, mampu diperbaharui secara dinamis, *hypertextuality* dan *global presence*.

4. *Intranet* Dan *Extranet*

Umumnya *intrane*t digambarkan hanya sebagai *web server* di dalam perusahaan (internal), padahal sebenarnya *intranet* hanyalah kumpulan *web site* yang dimiliki oleh suatu kelompok (biasanya perusahaan) yang

bisa diakses hanya oleh anggota kelompok tersebut. Sedangkan *extranet* merupakan area tertentu dari *intarnet* yang bias diakes oleh kelompok di luar anggota kelompok *intranet*, tapi dengan otorisasi tertentu. Fitur *intranet* standar dalam suatu organisasi memiliki 4 kemamapuan dasar: *e-mail*, *on line publishing*, *on line searches* dan *application distribution*. Sedangkan *extranet* memperluas fitur ini ke partner bisnis jika dimungkinkan. Keuntungan menggunalkan *intranet* di dalam suatu organisasi:

a. Mempercepat proses bisnis
b. Memfasilitasi pertukaran informasi
c. Meningkatkan komunikasi dan kolaborasi.

2.2.7.7 Security E-Commerce

Secara umum, keamanan merupakan salah satu komponen atau servis yang dibutuhkan untuk menjalankan *e-commerce*. Beberapa bagian dari keamanan ini sudah dibahas di atas dalam bagian tersendiri, seperti Infrastruktur Kunci Publik (IKP), dan *privacy*. Untuk menjamin keamanan, perlu adanya kemampuan dalam bidang ini yang dapat diperoleh melalui penelitian dan pemahaman. Beberapa topik (*issues*) yang harus dikuasai antara lain akan didaftar di bawah ini.

Teknologi Kriptografi. Teknologi kriptografi menjelaskan bagaimana mengamankan data dengan menggunakan enkripsi. Berbagai sistem sudah dikembangkan seperti sistem *private key* dan *public key*. Penguasaan algoritma-algoritma populer digunakan untuk mengamankan

data juga sangat penting. Contoh algoritma ini antara lain DES, IDEA, RC5, RSA dan ECC (*Ellliptic Curve Cryptography*). Penelitian dalam bidang ini di perguruan tinggi merupakan suatu hal yang penting. Salah satu masalah dalam mengamankan enkripsi adalah bagaimana memastikan bahwa hanya sang penerima yang dapat mengakses data. Anda dapat menggembok data dan mengirimkannya bersama kuncinya ke alamat tujuan, tetapi bagaimana memastikan kunci itu tidak dicuri orang di tengah jalan? Salah satu cara untuk memecahkannya adalah bahwa si penerima yang mengirimkan gemboknya, tetapi tidak mengirimkan kuncinya. Anda menggembok data dengan gembok yang dikirim olehnya dan mengirimkannya. Si penerima kemudian akan membukanya dengan kunci miliknya yang tidak pernah dikirimkannya ke siapa-siapa. Kini masalahnya bila data yang digembok itu dicuri orang, tetapi dengan enkripsi yang kompleks akan sangat sulit bagi orang itu untuk mengakses data yanmg sudah digembok itu.

One Time Pasword. Penggunaan *password* yang hanya dapat dipakai sebanyak satu kali. Biasanya *password* angka digital yang merandom angka setiap kali transaksi.

Konsultan keamanan. Konsultan, organisasi, dan institusi yang bergerak di bidang keamanan dapat membantu meningkatkan dan menjaga keamanan. Contoh organisasi yang bergerak di bidang ini adalah IDCERT[8].

2.2.8 Internet

2.2.8.1 Sejarah *Internet*

Sejarah *internet* dimulai pada 1969 ketika Departemen Pertahanan Amerika, U.S. *Defense Advanced Research Projects Agency* (DARPA) memutuskan untuk mengadakan riset tentang bagaimana caranya menghubungkan sejumlah komputer sehingga membentuk jaringan organik. Program riset ini dikenal dengan nama ARPANET. Pada 1970, sudah lebih dari 10 komputer yang berhasil dihubungkan satu sama lain sehingga mereka bisa saling berkomunikasi dan membentuk sebuah jaringan[5].

Tahun 1972, Roy Tomlinson berhasil menyempurnakan program *e-mail* yang ia ciptakan setahun yang lalu untuk ARPANET. Program *e-mail* ini begitu mudah sehingga langsung menjadi populer. Pada tahun yang sama, *icon* "@" juga diperkenalkan sebagai lambang penting yang menunjukkan "*at*" atau "pada". Tahun 1973, jaringan komputer ARPANET mulai dikembangkan ke luar Amerika Serikat. Komputer University College di London merupakan komputer pertama yang ada di luar Amerika yang menjadi anggota jaringan Arpanet. Pada tahun yang sama, dua orang ahli komputer yakni Vinton Cerf dan Bob Kahn mempresentasikan sebuah gagasan yang lebih besar, yang menjadi cikal bakal pemikiran internet. Ide ini dipresentasikan untuk pertama kalinya di Universitas Sussex.

Hari bersejarah berikutnya adalah tanggal 26 Maret 1976, ketika Ratu Inggris berhasil mengirimkan *e-mail* dari *Royal Signals and Radar Establishment* di Malvern. Setahun kemudian, sudah lebih dari 100 komputer yang bergabung di ARPANET membentuk sebuah jaringan atau network. Pada 1979, Tom Truscott, Jim Ellis dan Steve Bellovin, menciptakan newsgroups pertama yang diberi nama USENET. Tahun 1981 France Telecom menciptakan gebrakan dengan meluncurkan telpon televisi pertama, dimana orang bisa saling menelpon sambil berhubungan dengan *video link*.

Karena komputer yang membentuk jaringan semakin hari semakin banyak, maka dibutuhkan sebuah protokol resmi yang diakui oleh semua jaringan. Pada tahun 1982 dibentuk *Transmission Control Protocol* atau TCP dan Internet Protokol atau IP yang kita kenal semua. Sementara itu di Eropa muncul jaringan komputer tandingan yang dikenal dengan Eunet, yang menyediakan jasa jaringan komputer di negara-negara Belanda, Inggris, Denmark dan Swedia. Jaringan Eunet menyediakan jasa *e-mail* dan *newsgroup* USENET.

Untuk menyeragamkan alamat di jaringan komputer yang ada, maka pada tahun 1984 diperkenalkan sistem nama domain, yang kini kita kenal dengan DNS atau *Domain Name System*. Komputer yang tersambung dengan jaringan yang ada sudah melebihi 1000 komputer lebih. Pada 1987 jumlah komputer yang tersambung ke jaringan melonjak 10 kali lipat menjadi 10.000 lebih.

Tahun 1988, Jarko Oikarinen dari Finland menemukan dan sekaligus memperkenalkan IRC atau Internet Relay Chat. Setahun kemudian, jumlah komputer yang saling berhubungan kembali melonjak 10 kali lipat dalam setahun. Tak kurang dari 100.000 komputer kini membentuk sebuah jaringan. Tahun 1990 adalah tahun yang paling bersejarah, ketika Tim Berners Lee menemukan program editor dan browser yang bisa menjelajah antara satu komputer dengan komputer yang lainnya, yang membentuk jaringan itu. Program inilah yang disebut www, atau *Worl Wide Web*.

Tahun 1992, komputer yang saling tersambung membentuk jaringan sudah melampaui sejuta komputer, dan di tahun yang sama muncul istilah surfing the internet. Tahun 1994, situs internet telah tumbuh menjadi 3000 alamat halaman, dan untuk pertama kalinya *virtual-shopping* atau *e-retail* muncul di internet. Dunia langsung berubah. Di tahun yang sama *Yahoo!* didirikan, yang juga sekaligus kelahiran *Netscape Navigator* 1.0.

2.2.7.2 Kegunaan Internet

Secara umum ada banyak manfaat yang dapat diperoleh apabila seseorang mempunyai akses ke internet seperti misalnya informasi untuk kehidupan pribadi (kesehatan, rekreasi, hobby, pengembangan pribadi, rohani, sosial) serta informasi untuk kehidupan profesional/pekerja (sains, teknologi, perdagangan, saham, komoditas, berita bisnis, asosiasi profesi, asosiasi bisnis, berbagai forum komunikasi).

Satu hal yang paling menarik ialah keanggotaan internet tidak mengenal batas negara, ras, kelas ekonomi, ideologi atau faktor faktor lain yang biasanya dapat menghambat pertukaran pikiran. Internet adalah suatu komunitas dunia yang sifatnya sangat demokratis serta memiliki kode etik yang dihormati segenap anggotanya. Manfaat internet terutama diperoleh melalui kerjasama antar pribadi atau kelompok tanpa mengenal batas jarak dan waktu. Untuk lebih meningkatkan kualitas sumber daya manusia di Indonesia, sudah waktunya para profesional Indonesia memanfaatkan jaringan internet dan menjadi bagian dari masyarakat informasi dunia.

2.2.7.3 Perkembangan Internet

Perkembangan besar Internet pertama adalah penemuan terpenting ARPA yaitu *packet switching* pada tahun 1960. *Packet switching* adalah pengiriman pesan yang dapat dipecah dalam paket-paket kecil yang masing-masing paketnya dapat melalui berbagai alternatif jalur jika salahsatu jalur rusak untuk mencapai tujuan yang telah ditentukan. *Packet switching* juga memungkinkan jaringan dapat digunakan secara bersamaan untuk melakukan banyak koneksi, berbeda dengan jalur telepon yang memerlukan jalur khusus untuk melakukan koneksi. Maka ketika ARPANET menjadi jaringan komputer nasional di Amerika Serikat pada 1969, *packet switching* digunakan secara menyeluruh sebagai metode komunikasinya menggantikan *circuit switching* yang digunakan pada sambungan telepon publik.

Perkembangan besar Internet kedua yang dicatat pada sejarah internet adalah pengembangan lapisan protokol jaringan yang terkenal karena paling banyak digunakan sekarang yaitu TCP/IP (*Transmission Control Protocol/ Internet Protocol*). Protokol adalah suatu kumpulan aturan untuk berhubungan antar jaringan. Protokol ini dikembangkan oleh Robert Kahn dan Vinton Cerf pada tahun 1974. Dengan protokol yang standar dan disepakati secara luas, maka jaringan lokal yang tersebar di berbagai tempat dapat saling terhubung membentuk jaringan raksasa bahkan sekarang ini menjangkau seluruh dunia. Jaringan dengan menggunakan protokol internet inilah yang sering disebut sebagai jaringan internet.

Jaringan ARPANET menjadi semakin besar sejak saat itu dan mulai dikelola oleh pihak swasta pada tahun 1984, maka semakin banyak universitas tergabung dan mulailah perusahaan komersial masuk. Protokol TCP/IP menjadi protokol umum yang disepakati sehingga dapat saling berkomunikasi pada jaringan internet ini.

Perkembangan besar Internet ketiga adalah terbangunnya aplikasi *World Wide Web* pada tahun 1990 oleh Tim Berners-Lee. Aplikasi *World Wide Web* (WWW) ini menjadi konten yang dinanti semua pengguna *internet*. WWW membuat semua pengguna dapat saling berbagi bermacam-macam aplikasi dan konten, serta saling mengaitkan materi-materi yang tersebar di internet. Sejak saat itu pertumbuhan pengguna *internet* meroket.

2.2.7.4 Web Server

Web server adalah *software* yang menjadi tulang belakang dari *world wide web* (www). *Web server* menunggu permintaan dari *client* yang menggunakan *browser* seperti *Netscape Navigator, Internet Explorer, Mozilla*, dan program *browser* lainnya. Jika ada permintaan dari *browser*, maka *web server* akan memproses permintaan itu kemudian memberikan hasil prosesnya berupa data yang diinginkan kembali ke *browser* [4].

2.2.7.5 Web Browser

Web browser dikenal juga dengan istilah *browser*, atau peselancar, atau internet *browser*. Adalah suatu program komputer yang menyediakan fasilitas untuk membaca halaman *web* di suatu komputer. Dua program *web browser* yang cukup populer saat ini adalah *Microsoft Internet Explorer* dan *Netscape Navigator*. Program browser pertama adalah *Mosaic*, yang merupakan suatu *text browser*, yang sekarang web browser telah berkembang ke dalam bentuk multimedia.

2.2.7.6 Website

Sebuah situs web (sering pula disingkat menjadi situs saja; *website*, *site*) adalah sebutan bagi sekelompok halaman web (*web page*), yang umumnya merupakan bagian dari suatu nama domain (*domain name*) atau subdomain di *World Wide Web* (WWW) di Internet. Halaman-halaman sebuah situs web diakses dari sebuah URL yang menjadi "akar" (*root*),

yang disebut *homepage* (halaman induk; sering diterjemahkan menjadi "beranda", "halaman muka" atau laman web), dan biasanya disimpan dalam *server* yang sama. Tidak semua situs web dapat diakses dengan gratis. Beberapa situs web memerlukan pembayaran agar dapat menjadi pelanggan, misalnya situs-situs yang menampilkan pornografi, situs-situs berita, layanan surat elektronik (*e-mail*), dan lain-lain.

2.2.7.7 *World* Wide Web (WWW)

World Wide Web (disingkat sebagai WWW atau Web) adalah suatu ruang informasi yang yang dipakai oleh pengenal global yang disebut *Uniform Resource Identifier* (URI) untuk mengidentifikasi sumber-sumber daya yang berguna. WWW sering dianggap sama dengan Internet secara keseluruhan, walaupun sebenarnya ia hanyalah bagian daripadanya. 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.

Fasilitas ini tergolong masih baru dibandingkan *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.

2.2.9 Secure Socket Layer (SSL)

2.2.9.1 Definisi Secure Socket Layer (SSL)

SSL adalah suatu protokol komunikasi pada *Internet* yang menyediakan fasilitas keamanan seperti kerahasiaan, keutuhan dan keabsahan [8]. Protokol ini bebas dipergunakan siapa saja, bahkan didukung oleh dua *browser* utama, yaitu *Netscape* Navigator dan *Microsoft Internet* Explorer. SSL juga tidak mengkhususkan diri untuk hanya mendukung protokol tertentu seperti HTTP misalnya, karenanya SSL menggunakan *port* 443 untuk berhubungan dengan pelayan internet yang juga memiliki fasilitas SSL. Lapisan aplikasi diatasnya dapat memanfaatkan kunci yang telah dinegosiasikan oleh SSL. SSL dirancang agar fasilitas keamanan pada aplikasi yang memanfaatkan SSL tidak

merepotkan pemakainya. Dengan memanfaatkan SSL, aplikasi internet dapat melakukan komunikasi yang aman melalui fasilitas yang disediakan oleh SSL :

1. Kerahasiaan pesan, sehingga tidak bisa dibaca oleh pihak yang tidak diinginkan
2. Keutuhan pesan, sehingga tidak bisa diubah-ubah di tengah jalan
3. Keabsahan, sehingga meyakinkan pihak-pihak yang berkomunikasi mengenai keabsahan pesan dan keabsahan jati diri lawan bicaranya.

2.2.9.2 Cara Kerja *Secure Socket Layer* (SSL)

Teknologi SSL menggunakan konsep teknologi *kriptografi* kunci publik 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.

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

2.2.10 Pembayaran

2.2.10.1 Off Line

2.2.10.1.1 Tunai

Pembayaran tunai adalah pembayaran yang dilakukan langsung di tempat transaksi dengan melunasi pembayaran secara langsung menggunakan uang kontan sesuai dengan jumlah pembayaran yang sudah disepakati.

2.2.10.1.2 Bayar di Bank

Sama seperti pembayaran tunai, pembayaran di Bank adalah pembayaran menggunakan uang kontan sesuai jumlah pembayaran yang sudah disepakati hanya saja tempat pembayarannya dilakukan di Bank. Hal ini bertujuan untuk menghemat waktu, konsumen cukup datang ke Bank yang seudah ditentukan dengan mentransferkan sejumlah uang yang sudah disepakati ke rekening yang dituju.

2.2.10.2 On Line

2.2.11.2 E-Banking

Bank menyediakan layanan *Electronic Banking* atau *E-Banking* untuk memenuhi kebutuhan Anda akan alternative media untuk melakukan transaksi perbankan, selain yang tersedia di kantor cabang dan ATM. Dengan *Electronic Banking*, Anda tidak perlu lagi membuang waktu untuk antri di kantor-kantor bank atau ATM, karena saat ini banyak transaksi pebankan dapat dilakukan dimanapun, dan kapanpun dengan midah dan

praktis melalui jaringan *elektronik*, seperti *internet*, *handphone,* dan telepon. Contohnya adalah transfer dana antar rekening maupun antar bank, pembayaran tagihan, pembelian pulsa isi ulang, ataupun pengecekan mutasi dan saldo rekening.

2.2.10.2.2 Via ATM

Pembayaran via ATM dilakukan dengan cara mentransferkan sejumlah uang yang telah disepakati dari rekening konsumen di mesin ATM, tentu saja melakukan pembayaran dengan cara ini harus mempunyai ATM terlebih dahulu. Pembayaran melalui ATM relative praktis karena konsumen tidak perlu membawa uang tunai.

2.2.10.2.3 *Paypal*

PayPal adalah salah satu alat pembayaran (*Payment procesors*) menggunakan internet yang terbanyak digunakan didunia dan teraman. Pengguna *internet* dapat membeli barang di *ebay*, lisensi *software* original, keanggotaan situs, urusan bisnis, mengirim dan menerima donasi/sumbangan, mengirim uang ke pengguna *PayPal* lain di seluruh dunia dan banyak fungsi lainnya dengan mudah dan otomatis menggunakan internet atau mobile, *PayPal* mengatasi kekurangan dalam pengiriman uang tradisional seperti Cek atau *Money order* yang prosesnya dapat memakan waktu *PayPal* seperti rekening bank, pertama anda membuat *account*, lalu dana akan ditarik langsung dari kartu kredit anda ke *account* tersebut atau dengan dana dari transferan *account PayPal*

orang lain ke Saldo / *Balance PayPal* anda, dan anda sudah dapat menggunakan *account PayPal* untuk bertransaksi. Februari 2010 *PayPal* sudah menerima 190 negara dan 23 mata uang, pengguna *PayPal* di Indonesia masih harus menggunakan hitungan US dollar karena rupiah belum ada di *PayPal*, dana yang ditarik langsung dari kartu kredit akan dikonversi ke US dollar otomatis.

Paypal lebih aman dari alat pembayaran online lain karena Pembeli/*Buyer* yang merasa dirugikan atau tidak sesuai deskripsi yang diberitahukan bisa meminta uang kembali kepada Penjual / *Seller* dalam waktu maksimal 45 hari setelah terjadi pembayaran. Jika pembeli menggunakan kartu kredit akan mendapatkan pengembalian uang *charge back* dari perusahaan kartu kreditnya.

PayPal juga melindungi penjual dari pengembalian uang atau komplain yang tidak benar dari pembeli sesuai pembuktian. Perlindungan untuk penjual dirancang untuk melindungi penjual dari klaim pembeli yang mengaku telah mengirim uang yang tidak mempunyai catatan bukti pembayaran dan deskripsi transaksi, setiap pembelian dan pembayaran menggunakan *PayPal* selalu ada catatan bukti pembayarannya di *account PayPal* pengirim dan penerima uang, sebagai bukti jika benar telah terjadi pengiriman uang antara keduanya.

2.2.11 Perangkat Lunak Penunjang

2.2.11.1 HTML

HTML dewasa ini dikenal sebagai standar untuk membuat dokumen web. Sesungguhnya HTML justru tidak dibuat untuk mempublikasikan informasi di web, namun oleh karena kesederhanaan serta kemudahan penggunaannya, HTML kemudian dipilih untuk mendistribusikan informasi di web [11].

Perintah-perintah HTML diletakkan dalam file berekstensi *.html dan ditandai dengan mempergunakan *tag* (tanda) berupa karakter "< dan >". Kode-kode HTML dibaca oleh *browser* dari atas ke bawah tanpa adanya lompatan-lompatan.

Struktur sebuah dokumen HTML pada dasarnya dibagi menjadi dua bagian besar, yaitu *header* dan *body*. Masing-masing ditandai oleh pasangan *container* tag <*head*> dan <*body*>. Bagian head berisikan judul dokumen dan informasi-informasi dasar lainnya sedangkan bagian *body* berisi data dokumennya. Pengaturan format teks dan pembentukan *link* dilakukan terhadap objeknya langsung dengan ditandai oleh *tag-tag* HTML.

2.2.11.2 CSS

Cascading Style Sheet (CSS) merupakan salah satu bahasa pemrograman web untuk mengendalikan beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam [6].

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

CSS dapat mengendalikan ukuran gambar, warna bagian tubuh pada teks, warna tabel, ukuran *border*, warna *border*, warna *hyperlink*, warna *mouse over*, spasi antar paragraf, spasi antar teks, *margin* kiri, kanan, atas, bawah, 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.

2.2.11.3 PHP

PHP (PHP : *Hypertext Pree Processor*) merupakan bahasa pemograman scripting yang bersifat *open source*. PHP adalah salah satu bahasa *Server-side* yang didesain khusus untuk aplikasi *web*. PHP dapat disisipkan diantara bahasa HTML dan karena bahasa *Server side*, maka bahasa PHP akan dieksekusi di *server*, sehingga yang dikirimkan ke *browser* adalah “hasil jadi” dalam bentuk HTML, dan kode PHP anda tidak akan terlihat[4].

PHP merupakan bahasa *script* yang digunakan untuk membuat halaman web yang dinamis. Dinamis berarti halaman yang akan

ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru. Semua *script* PHP dieksekusi pada *server* dimana *script* tersebut dijalankan. Oleh karena itu, spesifikasi *server* lebih berpengaruh pada eksekusi dari *script* php daripada spesifikasi *client*. Namun tetap diperhatikan bahwa halaman *web* yang dihasilkan tentunya harus dapat dibuka oleh browser pada *client*. Dalam hal ini versi dari html yang digunakan harus didukung oleh *browser client*.

Sejak Januari 2001 PHP3 dan PHP4 disertakan pada sejumlah produk *server web* komersial seperti *Strong Holld Red Hat*. Banyak produk aplikasi yang telah sukses diimplementasi dengan PHP, seperti PHPNuke yang merupakan produk portal untuk berita dan forum diskusi dengan menggunkan *database* MySQL. PHPNuke dapat didownload dari situs http://www.phpnuke.org.

2.2.11.4 Java Script

JavaScript adalah bahasa pemrograman berbasis prototipe yang berjalan disisi klien. Jika kita berbicara dalam konteks web, sederhananya, kita dapat memahami *JavaScript* sebagai bahasa pemrograman yang berjalan khusus untuk di *browser* atau halaman web agar halaman web menjadi lebih hidup. Kalau dilihat dari suku katanya terdiri dari dua suku kata, yaitu *Java* dan *Script*. *Java* adalah Bahasa pemrograman berorientasi objek, sedangkan *Script* adalah serangkaian instruksi program[9].

Secara fungsional, *JavaScript* digunakan untuk menyediakan akses script pada objek yang dibenamkan (*embedded*). Contoh sederhana dari penggunaan *JavaScript* adalah membuka halaman *pop up*, fungsi validasi pada *form* sebelum data dikirimkan ke *server*, merubah image kursor ketika melewati objek tertentu, dan lain lain[9].

2.2.11.5 MYSQL

MySQL adalah sebuah perangkat lunak *database* (basis data) sistem tebuka yang sangat terkenal dikalangan pengembang sistem *database* dunia yang di gunakan untuk berbagai aplikasi terutama untuk aplikasi berbasis web. MySQL mempunyai fungsi sebagai SQl (*Structured Query Language*) yang di miliki sendiri dan telah di perluas. MySQL umumnya digunakan bersamaan dengan PHP untuk membuat aplikasi yang dinamis dan *powerful*[2].

Database adalah sebuah struktur yang umumnya dikategorikan dalam 2 hal sebuah *database* flat dan sebuah *database* relasional. *Database* relasional lebih disukai karena lebih masuk akal dibandingkan *database* flat. MySQL adalah sebuah *database* relasional.

Pada *database* yang memiliki struktur relasional meimiliki tabel-tabel yang menyimpan data. Setiap tabel terdiri dari kolom dan baris. Sebuah kolom mendefinisikan jenis informasi apa yang akan disimpan. Diperlukan kolom khusus untuk setiap jenis informasi yang ingin di simpan (misalnya umur, tinggi, berat, alamat). Kalau kolom

mendefinisikan jenis informasi apa yang akan disimpan, maka sebuah baris adalah data aktual yang disimpan.

Setiap baris dari tabel adalah masukan dari tabel tersebut dan berisi nilai-nilai untuk setiap kolom tabel tersebut. (contoh masukannya adalah Fadli: umur 55 tahun, 65 kg dan 165cm). Lalu mengapa menggunakan *database*? *Database* merupakan representasi kumpulan fakta yang saling berhubungan disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan. Data perlu disimpan dalam *database* untuk keperluan penyediaan informasi lebih lanjut. Data di dalam *database* perlu diorganisasikan sedemikian rupa, supaya informasi yang dihasilkan berkualitas. Organisasi *database* yang baik juga berguna untuk efisiens kapasitas penyimpanannya.

2.2.11.6 *Adobe Dreamweaver* CS3

Adobe Dreamweaver CS3 merupakan *software* yang cukup populer, yang digunakan untuk mendesain halaman web. Dengan kemampuan desain dan pemograman, baik dengan bahasa html maupun bahasa pemograman lain, menjadikan *Dreamwweaver* CS3 merupakan pilihan paling tepat bagi Anda yang ingin mendesain web [11].

2.2.11.7 XAMPP

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program[6].

Fungsinya adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program Apache HTTP *Server*, MySQL *database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU *General Public License* dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis.

2.2.11.8 Adobe Photoshop CS3

Adobe Photoshop, atau biasa disebut *Photoshop*, adalah perangkat lunak editor citra buatan *Adobe System* yang dikhususkan untuk pengeditan foto/gambar dan pembuatan efek. Perangkat lunak ini banyak digunakan oleh fotografer *digital* dan perusahaan iklan sehingga dianggap sebagai pemimpin pasar (*market leader*) untuk perangkat lunak pengolah gambar, dan, bersama *Adobe Acrobat*, dianggap sebagai produk terbaik yang pernah diproduksi oleh *Adobe System*. Versi kedelapan aplikasi ini disebut dengan nama *Photoshop CS* (*Creative Suite*), versi sembilan disebut *Photoshop CS2*, versi sepuluh disebut *Adobe Photoshop CS3* , dan versi yang terakhir (kesebelas) adalah *Adobe Photoshop CS4*.