BAB 2
TINJAUAN PUSTAKA

2.1 Tinjauan Perusahaan

Tinjauan perusahaan menjelaskan tentang profil perusahaan tempat penelitian yang terdiri dari sejarah perusahaan, struktur organisasi dan deskripsi yang ada di toko Pelangi Baby Shop

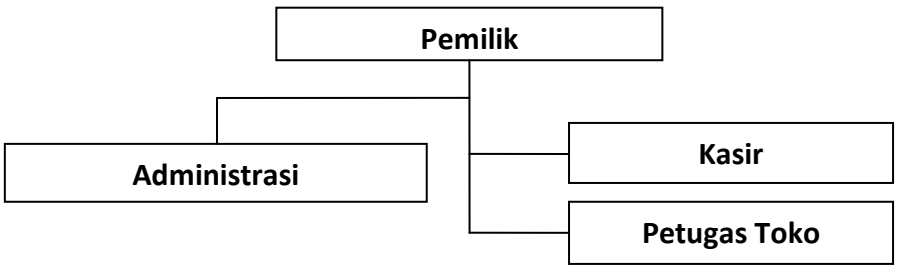
2.1.1 Sejarah Perusahaan

Pelangi Baby Shop merupakan sebuah perusahaan yang bergerak di bidang perlengkapan bayi. Pelangi Baby Shop didirikan pada tahun 2011 oleh Paul Christianto Susanto, ST. Pelangi Baby Shop didirikan untuk membantu keluarga baru yang ingin membeli atau melengkapi perlengkapan untuk bayi atau balita mereka. Pelangi Baby Shop hadir dengan konsepbutik dengan harga yang terjangkau namun dengan barangyang berkualitas, menarik dan beragam.Produk yang dijual Pelangi Baby Shop berupa perlengkapan mandi bayi, baju, topi, sepatu, kereta dorong sampai perlengkapan makan untuk bayi. Pelangi Baby Shop sendiri berlokasi di Jl. Candi Gebang Permai Blok 1. No. 6 Wedomartani, Sleman, Yogyakarta.

.

2.1.2 Struktur Organisasi

Struktur organisasi dari Pelangi Baby Shop dapat dilihat pada gambar 2.1.



Gambar 2. 1 Struktur Organisasi Pelangi Baby Shop

2.1.3 Deskripsi Tugas

Berikut ini adalah deskripsi jabatan dari struktur organisasi yang terdapat

pada Pelangi Baby Shop:

1. Pemilik
    a. Mengkoordinasi secara kesuluruhan kegiatan di toko.
    b. Bertindak sebagai pengambil keputusan.
    c. Menetapkan harga dan diskon untuk setiap item produk yang dijual.
    d. Mengarahkan pegawai.

2. Administrasi
    a. Melaksanakan kegiatan yang berhubungan dengan urusan keuangan toko.
    b. Pengadaan inventaris toko.

3. Kasir
    a. Menangani transaksi penjualan.
    b. Membuat laporan penjualan.

4. Petugas toko
    a. Melayani dan bertatap muka langsung dengan konsumen.
    b. Pemeliharaan barang terhadap barang yang ada di *counter*.
    c. Membuat nota transaksi.
    d. Mengecek stok barang.

2.2 Konsep Dasar Sistem

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

Pengertian sistem yang telah dijelaskan dapat diambil suatu kesimpulan bahwa suatu sistem merupakan elemen yang saling berkaitan dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu.

2.2.1 Karakteristik Sistem

Suatu sistem memiliki karakteristik atau sifat-sifat tertentu, yaitu memiliki komponen-komponen (*components*), batas sistem (*boundary*), lingkungan luar sistem (*system environment*), sistem penghubung (*interface*), masukan (*input*), keluaran (*output*), pengolah (*process*), sasaran (*objectives*) dan tujuan (*goal*) [2].

1. Komponen Sistem (*System Components*)

   Komponen sistem atau elemen-elemen sistem dapat berupa suatu sub sistem atau bagian-bagian dari sistem. Setiap sistem baik besar maupun kecil, selalu mengandung komponen-komponen atau subsistem-subsistem. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

2. Batas Sistem (*Boundary*)

   Batas sistem merupakan daerah-daerah yang membatasi antara satu sistem dengan sistem lainnya dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan. Batas suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem (*Environment*)

   Lingkungan luar sistem dari suatu sistem adalah apapun di luar batas dari sistem yang mempengaruhi operasi sistem yang dapat bersifat menguntungkan dan dapat pula merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem yang harus dijaga dan dipelihara. Sedangkan yang merugikan harus ditahan dan dikendalikan, karena akan mengganggu kelangsungan hidup sistem.

4. Penghubung Sistem (*Interface*)

   Penghubung sistem merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lainnya. Dengan penghubung akan terjadi interaksi antar subsistem, sehingga membentuk satu kesatuan.

5. Masukan Sistem (*Input*)

Masukan adalah suatu energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Contoh *maintenance input* di dalam sistem komputer adalah program, yang digunakan untuk mengoperasikan computer. Sedangkan *signal input* adalah energi yang diproses untuk mendapatkan keluaran. Contoh *signal input* di dalam sistem komputer adalah data, yang dapat diolah menjadi informasi.

6. Keluaran Sistem (*Output*)

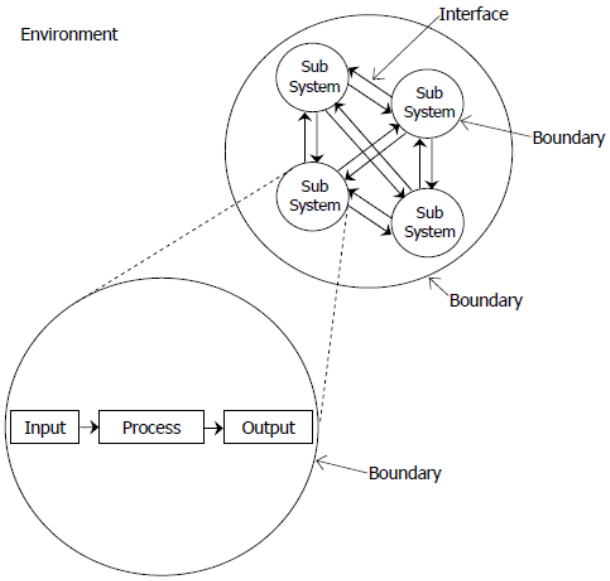
   Keluaran (*Output*) merupakan hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna.

7. Pengolahan Sistem (*Process*)

   Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya, yang bertugas untuk merubah masukan menjadi keluaran.

8. Sasaran Sistem (*System Objective*)

   Suatu sistem pasti memiliki tujuan (*goal*) atau sasaran (*objective*). Suatu operasi sistem akan berguna dan berhasil apabila mencapai sasaran atau tujuannya. Sasaran sistem sangat menentukan masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Gambaran dari karakteristik sistem dapat dilihat pada gambar 2.2.



Gambar 2.2 Karakteristik Sistem [4]

2.2.2 Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya adalah sebagai berikut :

1. Sistem diklasifikasikan sebagai sistem abstrak (*abstract system*) dan sistem fisik (*physical system*). Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Sistem fisik merupakan sistem yang ada secara fisik.
2. Sistem diklasifikasikan sebagai sistem alamiah (*natural system*) dan sistem buatan manusia (*human made system*). Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia. Sistem buatan manusia yang melibatkan interaksi antara manusia dengan mesin disebut dengan *human-machine-system* atau ada yang menyebut dengan *man-machine-system*.
3. Sistem diklasifikasikan sebagai sistem tertentu (*deterministic system*) dan sistem tak tentu (*probabilistic system*). Sistem tertentu beroperasi dengan tingkah laku yang sudah diprediksi. Interaksi diantara bagian-bagiannya dapat dideteksi dengan pasti, sehingga keluaran dari sistem dapat diramalkan. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

4. Sistem diklasifikasikan sebagai sistem tertutup (*closed system*) dan sistem terbuka (*open system*). Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak diluarnya. Secara teoritis sistem tertutup ini ada, tetapi kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah *relatively closed system* (secara relatif tertutup, tidak benar-benar tertutup). Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk lingkungan luar atau subsistem yang lainnya [2].

2.2.3 Sistem Rekomendasi

Smart Recommendation System : Fitur rekomendasi barang. Di web akan tampil promo-promo produk yang best seller atau sesuai dengan type pelanggannya.

Metode Rekomendasi Collaborative Filtering

Rekomendasi collaborative filtering adalah teknik yang paling banyak digunakan pada sistem rekomendasi. Seperti namanya, collaborative filtering bekerja dengan cara menjumlahkan rating atau pilihan dari suatu produk, menemukan profil / pola pengguna dengan melihat history rating yang diberikan pengguna, dan menghasilkan suatu rekomendasi baru berdasarkan perbandingan antar pola pengguna. Biasanya nilai rating dapat berupa binary (suka/tidak suka) atau voting. (Xu dkk, 2010:169-170)

a. User Based Collaborative Filtering

   Prinsip dasar dari user-based collaborative filtering adalah memberikan rekomendasi berdasarkan opini pengguna lain yang memiliki kesukaan yang sama dengan pengguna.

b. Item Based Collaborative Filtering

   Prinsip dasar dari algoritma ini adalah menghitung tingkat kesamaan antara beberapa produk yang berbeda berdasarkan penilaian pengguna.

   Rumus :

Dis = (nilaiprodukperson ± nilaiprodukotherperson) ^{2}

$Sim_{otherperson}$ = 1 / (1 + jumlah Dis)

Rekomendasi = (Sim x NilaiProduk) / Sim

Keterangan :
Dis : distance
Sim : similarity atau tingkat kesamaan pelanggan
nilaiprodukperson : nilai dari produk orang yang akan diberikan rekomendasi
nilaiprodukotherperson : nilai dari produk orang yang akan dibandingkan

2.3 Konsep Dasar Data dan Informasi

Konsep dasar Data dan Informasi berisi pengertian dasar dari data dan informasi. Pada konsep data, akan dijelaskan pengertian dan jenis-jenis data. Sedangkan pada konsep informasi, akan dijelaskan pengertian dari informasi, kualitas informasi, nilai informasi serta siklus informasi.

2.3.1 Pengertian Data

Data adalah deskripsi dari sesuatu dan kejadian yang kita hadapi (*the description of things and events that we face*). Definisi data yang lain adalah data merupakan kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata[3]. Kesatuan nyata adalah berupa suatu objek nyata seperti tempat, benda, dan orang yang betul-betul ada dan terjadi.

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.

2.3.1.1 Model Data

Data yang disimpan menggambarkan beberapa aspek dari suatu organisasi. Model data adalah himpunan deskripsi 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.

2.3.1.2 Model Data Hierarki

Model hierarki 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. Simpula yang terhubung ke simpul pada level di bawahnya disebut orang tua. Setiap orang tua bias memiliki satu (hubungan 1:1) atau beberapa anak (hubungan 1:M), tetapi setiap anak hanya memiliki satu orang tua.

2.3.1.3 Model Data Jaringan

Model jaringan distandarisasi pada tahun 1971 oleh Data Base Task Group (DBTG). 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 hierarki, dengan perbedaan suatu simpul anak bias memiliki lebih dari satu orang tua. Oleh karena sifatnya demikian, model ini bias menyatakan hubungan 1:1 (satu orang tua punya satu anak), 1:M (satu orang tua punya banyak anak), maupun N:M (beberapa anak bias mempunyai beberapa orangtua). Pada model jaringan, orang tua disebut pemilik dan anak disebut anggota.

2.3.1.4 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*.

2.3.2 Pengertian Informasi

Menurut Gordon B. Navis dalam buku [3] "Informasi adalah data yang telah diproses/diolah ke dalam bentuk yang sangat berarti untuk penerimanya dan merupakan nilai yang sesungguhnya atau dipahami dalam tindakan atau keputusan yang sekarang atau nantinya".

Sumber dari informasi adalah data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata, tetapi data merupakan bentuk yang masih mentah dan belum dapat memberikan arti banyak bagi pemakai, sehingga perlu diolah lebih lanjut untuk menghasilkan informasi yang dibutuhkan. Dengan kata lain informasi adalah hasil dari pengolahan data.

2.3.2.1 Kualitas Informasi

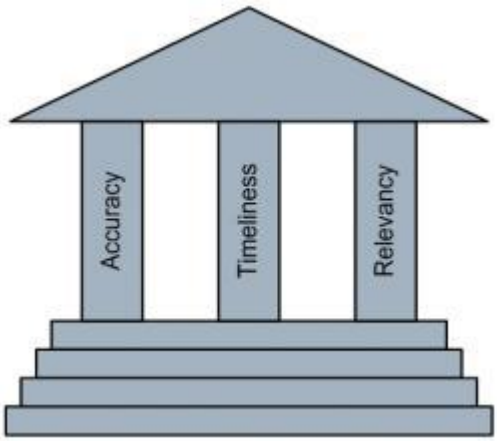
Menurut [4] Kualitas dari suatu informasi tergantung pada 3 hal pokok, diantaranya :

1. Akurat (*accurate*)

   Informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan, dalam hal ini informasi harus jelas mencerminkan maksudnya.
2. Tepat Waktu (*time lines*)

   Informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah using tidak akan mempunyai nilai lagi karena informasi merupakan suatu landasan dalam mengambil sebuah keputusan dimana bila pengambilan keputusan terlambat maka akan berakibat fatal untuk organisasi.
3. Relevan (*relevance*)

   Informasi harus mempunyai manfaat untuk pemakainya, dimana relevansi informasi untuk tiap-tiap individu berbeda tergantung pada yang menerima dan yang membutuhkan.



Gambar 2.3 Atribut Kualitas Informasi [4]

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 kemampuannta 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 diragukan 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.

2.3.2.2 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 perlakuan seperti ini mengakibatkan keputusan yang diambil tidak sesuai dengan yang diharapkan.

Menurut [4] mengemukakan bahwa nilai informasi ditentukan dari dua hal yaitu manfaat dan biaya. Suatu informasi dikatakan bernilai apabila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya. Sebagian besar informasi dinikmati oleh lebih dari satu pihak sehingga sulit untuk menghubungkan suatu informasi dengan biaya untuk memperolehnya dan sebagian besar informasi tidak dapat ditaksirkan keuntungannya dengan satuan uang tetapi dapat ditaksir nilai efektivitasnya.

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 niali 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/using, 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

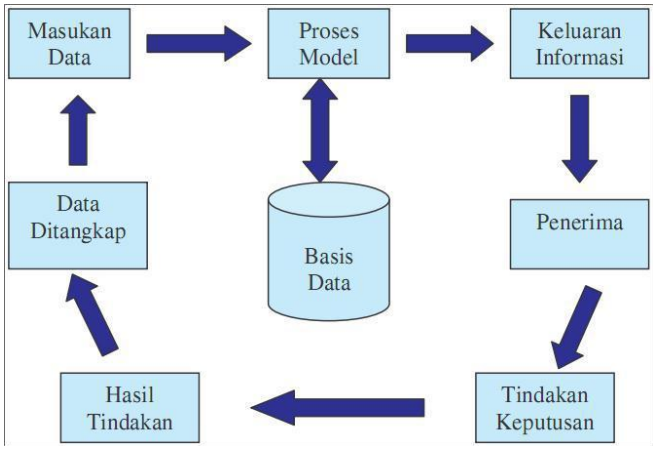
   Nilai informasi semakin sempurna apabila informasi tersebut tidak menimbulkan prasangka dan keraguan adanya kesalahan informasi.

10. Dapat diukur

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

2.3.2.3 Siklus Informasi

Data agar menjadi lebih berarti dan berguna dalam bentuk Informasi, maka perlu diolah melalui suatu model tertentu. Data yang telah diolah tersebut kemudian diterima oleh penerima, lalu penerima membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai *input*, dan diproses kembali lewat suatu model dan seterusnya sehingga membentuk suatu siklus. Siklus ini disebut dengan siklus Informasi (*information cycle*) atau disebut pula siklus pengolahan data (*processing cycles*).



Gambar 2.4 Siklus Informasi [5]

2.4 Konsep Dasar Sistem Informasi

Sistem informasi adalah suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan informasi yang diperlukan untuk pengambilan keputusan.

Sistem informasi dalam suatu organisasi dapat dikatakan sebagai suatu sistem yang menyediakan informasi bagi semua tingkatan dalam organisasi tersebut kapan saja diperlukan. Sistem ini menyimpan, mengambil, mengubah, mengolah dan mengkomunikasikan informasi yang diterima dengan menggunakan sistem informasi atau peralatan sistem lainnya [2].

2.4.1 Komponen Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang disebut blok bangunan (*building block*), yang terdiri dari blok masukan (*input block*), blok model (*model block*), blok keluaran (*block output*), blok teknologi (*technology block*), blok basis data (*database block*), dan blok kendali (*controls block*). Semua komponen tersebut saling berinteraksi satu dengan yang lain membentuk suatu kesatuan untuk mencapai sasaran [2].

1. Blok Masukan

   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. Blok 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 yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran

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

4. Blok Teknologi

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

5. Blok Basis Data

   Basis data (*database*) merupakan kumpulan data yang saling berkaitan dan berhubungan satu dengan yang lain, tersimpan diperangkat keras komputer dan menggunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan dalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data didalam 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*).

6. Blok Kendali

   Banyak hal yang dapat merusak sistem informasi, seperti bencana alam, api, temperature, 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.

2.4.2 Tujuan Sistem Informasi

Sistem Informasi memiliki beberapa tujuan, yaitu :

1. Integrasi Sistem
   a. Menghubungkan sistem individu/kelompok.
   b. Pengkolektifan data dan penyambungan secara otomatis.
   c. Peningkatan koordinasi dan pencapaian sinergi.
2. Efisiensi pengelolaan
   a. Penggunaan basis data dalam upaya kesamaan pengadministrasian data.
   b. Pengelolaan data berkaitan dengan karakteristik Informasi.
   c. Penggunaan dan pengambilan Informasi.
3. Dukungan keputusan untuk manajemen
   a. Melengkapi Informasi guna kebutuhan proses pengambilan keputusan.
   b. Akuisisi Informasi eksternal melalui jaringan komunikasi.
   c. Ekstraksi dari Informasi internal yang terpadu.

2.4.3 Manfaat Sistem Informasi

Sistem Informasi memiliki beberapa manfaat, yaitu :

1. Menghemat tenaga kerja
2. Peningkatan efisiensi

3. Mempercepat proses
4. Perbaikan dokumentasi
5. Pencapaian standar
6. Perbaikan keputusan

2.5 Perdagangan Elektronik (*E-Commerce*)

Perdagangan Elektronik, atau *Electronic Commerce* (EC) adalah sebuahsistem perniagaan dan jual beli yang banyak menggunakan kemudahan teknologimaklumat terutama telekomunikasi canggih sehingga dapat melindungi danmemuaskan penggunaan-penggunaan yaitu yang terdiri daripada penjual, pembelidan pihak-pihak yang ketiga seperti bank, syarikat keuangan, syarikat kartu kredit, pengesahan digital dan lain-lain. Kebelakangan ini, banyakmasyarakat menggunakan internet khususnya *web* untuk mengadakan saturangkaian yang menyatukan pelanggan, penjual dan pihak-pihak ketiga dalam satupersekitaran elektronik yang selamat untuk semua pihak dan boleh dipercayai.Media elektronik yang digunakan di sini adalah jaringan internet karenainternet sudah dikenal oleh banyak orang dan penggunaannya juga sangat mudah.Namun hal ini pada suatu saat nanti tidak menutup kemungkinan untukmenggunakan jaringan yang lain[6].

2.5.1 Sejarah Perkembangan *E-Commerce*

Istilah *e-commerce* telah berubah sejalan dengan waktu. Awalnya, perdagangan elektronik berarti pemanfaatan transaksi komersial, seperti penggunaan EDI untuk mengirim dokumen komersial seperti pesanan pembelian atau *invoice*secara elektronik. Kemudian dia berkembang menjadi suatu aktivitas yang mempunyai istilah yang lebih tepat “perdagangan web” – pembelian *produk* dan jasa melalui *World Wide Web* melalui server aman (HTTPS), protokol server khusus yang menggunakan enkripsi untuk merahasiakan data penting pelanggan.

E-commerce pertama kali diperkenalkan pada tahun 1994 pada saat pertama kali banner-elektronik dipakai untuk tujuan promosi dan periklanan di suatu halaman-website. Pada awalnya ketika web mulai terkenal di masyarakat

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

2.5.2 Klasifikasi *E-Commerce*

Penggolongan *E-Commerce* yang lazim dilakukan orang ialah berdasarkan sifat transaksinya, antara lain :

1. *Business to Business* (B2B)

   Jenis transaksi dimana pembeli biasanya membeli dalam jumlah besar karena akan dijual kembali. Contoh penjualan grosir.
2. *Business to Consumer* (B2C)

   Jenis transaksi dimana pembelinya perorangan dan tidak punya tujuan untuk menjualnya kembali biasanya semacam toko online yang menjual berbagai macam produk.
3. *Consumer to Consumer* (C2C)

   Jenis transaksi dimana pembelinya perorangan yang tidak mempunyai tujuan untuk dijual kembali dan penjualnya juga perorangan yang tidak menyediakan bermacam-macam produk melainkan hanya beberapa produk saja. Contoh : *online advertising*.
4. *Consumer to Business* (C2B)

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

2.5.3 Kelebihan *E-Commerce*

Secara sederhana, perbedaan antara proses perdagangan secara manual dengan menggunakan *e-commerce* terjadi efisiensi pada penggunaan fix, pencetakan dokumen, entry ulang dokumen, serta jasa kurir. Efisiensi tersebut akan menunjukkan pengurangan biaya dan waktu/kecepatan proses. Selain itu, *e-*

commerce juga sangat bermanfaat bagi *customer*/pelanggan dan masyarakat umum. Secara umum, ada berbagai manfaat lain yang didapat perusahaan saat melakukan *e-commerce*. Beberapa manfaat lain itu adalah sebagai berikut :

1. Keuntungan Bagi Perusahaan
    a. Memperpendek jarak

       Perusahaan dapat lebih mendekatkan diri dengan *customer*. Dengan hanya mengklik *link-link* yang ada pada situs-situs, *customer* dapat menuju ke perusahaan dimanapun saat itu mereka berada.
    b. Perluasan pasar

       Jangkauan pemasaran semakin menjadi luas dan tidak terbatas oleh area geografis dimana perusahaan berada.
    c. Perluasan jaringan mitra bisnis

       Pada perdagangan tradisional, sangat sulit bagi suatu perusahaan untuk mengetahui posisi geografis mitra kerjanya yang berada di negara lain atau benua lain. Bagaimana pun juga, mitra kerja sangat penting untuk konsultasi dan kerjasama baik teknis maupun non teknis. Dengan adanya *e-commerce* lewat jaringan internet, hal-hal tersebut bukan menjadi masalah yang besar lagi.
    d. Efisien

       Perusahaan yang berdagang secara elektronik tidak membutuhkan kantor dan toko yang besar, menghemat kertas-kertas yang digunakan untuk transaksi-transaksi, periklanan dan pencatatan-pencatatan. Selain itu *e-commerce* juga sangat efisien dari sudut waktu yang digunakan. Pencarian informasi-informasi produk/jasa dan transaksi-transaksi bisa dilakukan lebih cepat serta lebih akurat.
2. Keuntungan Bagi *Customer*
    a. Efektif

       Customer dapat memperoleh informasi tentang produk/jasa yang dibutuhkan dan bertransaksi dengan cara yang cepat dan mudah.
    b. Aman secara fisik

Customer tidak perlu mendatangi toko tempat perusahaan menjajakan produknya dan ini memungkinkan *customer* dapat bertransaksi dengan aman sebab di daerah-daerah tertentu sangan berbahaya jika berkendara dan membawa uang tunai dalam jumlah yang besar.

c. Fleksibel

Customer dapat melakukan transaksi dari berbagai lokasi, baik dari rumah, kantor, warnet, atau tempat-tempat lainnya. *Customer* juga tidak perlu berdandan rapi seperti perdagangan tradisional pada umumnya.

3. Keuntungan Bagi Masyarakat Umum

a. Mengurangi polusi dan pencemaran lingkungan

Dengan adanya *e-commerce* yang dapat dilakukan dimana saja, *customer* tidak perlu melakukan perjalanan ke toko-toko, diaman hal ini pada gilirannya akan mengurangi jumlah kendaraan yang berlalu lalang dijalanan.

b. Membuka peluang kerja baru

Era *e-commerce* akan membuka peluang-peluang kerja baru bagi mereka yang tidak ‘buta’ teknologi. Muncul pekerjaan baru seperti pemrogram komputer, perancang web, ahli dibidang basis data, analisis sistem, ahli dibidang jaringan komputer, dan sebagainya.

c. Menguntungkan dunia akademis

Berubahnya pola hidup masyarakat dengan hadirnya *e-commerce*, kalangan akademisi akan semakin diperkaya dengan kajian-kajian psikologis, antropologis, social-budaya, dan sebagainya yang berkaitan dengan cara dan pola hidup yang berkaitan dengan dunia maya.

d. Meningkatkan kualitas sumber daya manusia

E-commerce, seperti juga teknologi komputer pada umumnya, hanya bisa dilakukan oleh orang-orang yang tidak gagap teknologi, sehingga pada gilirannya akan merangsang orang-orang untuk mempelajari teknologi computer demi kepentingan mereka sendiri.

2.5.4 Kekurangan *E-Commerce*

Walaupun adanya *e-commerce* memberi banyak keuntungan, masih terdapat berbagai kekurangan dari *e-commerce* antara lain :

1. Bagi organisasi/perusahaan
   a. Keamanan sistem rentan diserang

      Terdapat sejumlah laporan mengenai website dan basis data yang dihack, dan berbagai lubang kelemahan keamanan dalam *software*. Hal ini dialami oleh sejumlah perusahaan besar seperti Microsoft dan lembaga perbankan. Masalah keamanan ini menjadi sangat penting karena bila pihak lain yang tidak berwenang bisa menembus sistem maka dapat menghancurkan bisnis yang telah berjalan.

   b. Persaingan tidak sehat

      Di bawah tekanan untuk berinovasi dan membangun bisnis untuk memanfaatkan kesempatan yang ada dapat memicu terjadinya tindakan ilegal yaitu penjiplakan ide dan perang harga.

   c. Masalah kompabilitas teknologi lama dengan yang lebih baru

      Dengan perkembangan dan inovasi yang melahirkan teknologi baru, sering muncul masalah yaitu sistem bisnis yang lama tidak dapat berkomunikasi dengan infrastruktur berbasis web dan internet. Hal ini memaksa perusahaan untuk menjalankan dua sistem independen yang tidak dapat saling barbagi, hal ini dapat mengakibatkan pembengkakan biaya.

2. Bagi Konsumen
   a. Perlunya keahlian komputer

      Tanpa menguasai keahlian komputer, mustahil konsumen dapat berpartisipasi dalam *e-commerce*. Pengetahuan dasar komputer diperlukan, antara lain pengetahuan mengenai internet dan web.

   b. Biaya tambahan untuk mengakses internet

      Untuk ikut serta dalam *e-commerce* dibutuhkan koneksi internet yang tentu saja menambah pos pengeluaran bagi konsumen.

   c. Biaya peralatan komputer

Komputer diperlukan untuk mengakses internet, tentu saja dibutuhkan biaya untuk mendapatkannya. Perkembangan komputer yang sangat pesat menyarankan konsumen untuk juga mengupdate peralatannya apabila tidak ingin ketinggalan teknologi.

d. Resiko bocornya privasi dan data pribadi

   Segala hal mungkin terjadi saat konsumen mengakses internet untuk menjalankan *e-commerce*, termasuk resiko bocornya data pribadi karena ulah orang lain yang ingin membobol sistem.

e. Berkurangnya waktu untuk berinteraksi secara langsung dengan orang lain. Transaksi *e-commerce* yang berlangsung secara *on-line* telah mengurangi waktu konsumen untuk dapat melakukan proses social dengan orang lain. Hal ini tidak baik karena dikhawatirkan akan dapat mengurangi rasa kepedulian terhadap lingkungan sekitarnya.

f. Berkurangnya rasa kepercayaan karena konsumen berinteraksi hanya dengan komputer.

3. Bagi Masyarakat

   a. Berkurangnya interaksi antar manusia

      Karena masyarakat lebih sering berinteraksi secara elektronik, dimungkinkan terjadi berkurangnya kemampuan sosial dan personal manusia untuk bersosialisasi dengan orang lain secara langsung.

   b. Kesenjangan social

      Terdapat bahaya potensial karena dapat terjadi kesenjangan sosial antara orang-orang yang memiliki kemampuan teknis dalam *e-commerce* dengan yang tidak, yang memiliki keahlian digaji lebih tinggi dari pada yang tidak.

   c. Adanya sumber daya yang terbuang

      Munculnya teknologi baru akan membuat teknologi lama tidak dimanfaatkan lagi. Misalnya dengan komputer model lama atau *software* model lama yang sudah tidak *relevan* untuk digunakan.

   d. Sulitnya mengatur internet

Sejumlah kriminalitas telah terjadi di internet dan banyak yang tidak terdeteksi. Karena jumlah jaringan yang terus berkembang semakin luas dan jumlah pengguna yang semakin banyak, seringkali membuat pihak berwenang kesulitan dalam membuat peraturan untuk internet.

2.5.5 Manfaat *E-Commerce*

Manfaat yang diperoleh bagi organisasi pemilik *E-Commerce* :

1. Memperluas *market place* hingga kepasar nasional dan internasional.
2. Dengan *capital outplay* yang minim, sebuah perusahaan dapat dengan mudah menemukan lebih banyak pelanggan, *supplier* yang lebih baik dan *partner* bisnis yang paling cocok dari seluruh dunia.
3. *E-Commerce* menurunkan biaya pembuatan, pemrosesan, pendistribusian, penyimpanan, dan pencarian informasi yang menggunakan kertas.
4. *E-Commerce* memungkinkan pengurangan *inventory* dan *overhead* dengan menyederhanakan *supply chain management* tipe "pull". Dalam *supply chain management* tipe "pull", proses dimulai dari pesanan pelanggan serta digunakan *manufacturing just-in-time.*
5. *E-Commerce* mengurangi waktu antara *outlay modal* dan penerimaan produk dan jasa.
6. *E-Commerce* mendukung upaya-upaya *business process reengineering*. Dengan mengubah prosesnya, maka produktivitas *sales-people,* pegawai yang berpengetahuan, dan administrator bisa meningkat 100% atau lebih.
7. *E-Commerce* memperkecil biaya telekomunikasi. Karena internet lebih murah dibandingan VAN.
8. Akses informasi menjadi lebih cepat.

Manfaat bagi konsumen lain :

1. *E-Commerce* memungkinkan pelanggan untuk berbelanja atau melakukan transaksi selama 24 jam sehari sepanjang tahun dari hamper setiap lokasi.
2. *E-Commerce* memberikan lebih banyak pilihan kepada pelanggan, mereka bisa memilih berbagai produk dari banyak vendor.

3. *E-Commerce* menyediakan produk dan jasa yang tidak mahal kepada pelanggan dengan cara mengunjungi banyak tempat dan melakukan perbandingan secara cepat.
4. Pelanggan bisa menerima informasi yang *relevan* secara detail dalam hitungan detik, bukan lagi hari atau minggu.
5. *E-Commerce* member tempat kepada para pelanggan untuk berinteraksi dengan pelanggan lain di *electronic community* dan bertukar pikiran serta pengalaman.

Manfaat bagi masyarakat lain :

1. *E-Commerce* memungkinkan orang untuk bekerja didalam rumah dan tidak harus keluar rumah untuk berbelanja. Ini berakibat menurunkan arus kepadatan lalu lintas dijalan serta mengurangi polusi udara.
2. *E-Commerce* memungkinkan orang dinegara-negara dunia ketiga dan wilayah pedesaan untuk menikmati aneka produk dan jasa yang akan susah mereka dapatkan tanpa *E-Commerce.*

2.5.6 Komponen Utama *E-Commerce*

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

1. *Electronic Data Interchange* (EDI)

   Electronic Data Interchange (EDI) didefinisikan sebagai pertukaran data komputer antar berbagai bidang organisasi atas suatu informasi terstruktur dalam format yang standard an 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 produk,

manager kantor dan lain-lainnya. Hal ini akan mengurangi tenaga entry data. Pada dasarnya, data bisa dikirimkan dengan lebih efisien dengan menggunakan EDI.

Komponen utama dari EDI standar adalah sebagai berikut :

a. *Data Element*

   Merupakan potongan data seperti tanggal, harga atau nama organisasi. Setiap data elemen diidentifikasikan dengan nomor referensi tertentu yang berisi judul, keterangan, jenis, nomor dan panjang minimum/maximum.

b. *Data Segment*

   Dalam suatu bisnis data disebut dengan segmen dan setiap item di dalam segmen mewakili satu elemen. Misalnya segmen baris pesanan pembelian terdiri atas nomor produk, keterangan, jumlah, unit pengukuran, dan harga produk. Setiap segmen memiliki satu identifier, satu data *element delimiter*, *element diagrams*, data segmen terminator dan *notes.*

c. *Transaction Set*

   Suatu *transaction set* merupakan dokumen khusus seperti dokumen pesanan pembelian. Di dalam *transaction set*, ada 3 area utama yaitu *area header, area detail* dan *area summary.*

d. *Functional Group*

   Merupakan sekelompok *transaction set* yang sejenis. *Transaction set* di dalam *functional group* dikelompokkan berdasarkan *functional identifier* yang sama. Mengirimkan transaksi EDI pada konsumen, diperlukan 4 fungsi dasar yaitu *Mapping* elemen dalam suatu database, *Extraction* atas data yang belum diidentifikasi dari database, *Translation* atas data yang sudah diekstrak ke format EDI, dan Transmisi pesan dalam format EDI melalui media komunikasi.

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 produk dan jasa, mata uang dan koin serta token lainnya.
c. Bisa disimpan dan diambil lagi.
d. Sulit diduplikasi atau dipalsukan.

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 penawaran produk dan jasa. *E-catalogs* umumnya mendukung *on-line shopping* dan kemampuan pemesanan dan pembayaran produk. Aplikasi *e-catalog* sebaiknya memiliki karakteristik seperti bersifat interaktif, mampu diperbaharui secara dinamis, *hypertextuality* dan *global presence.*

4. *Intranet dan Extranet*

   Umumnya *intranet* 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 *intranet* yang bisa diakses oleh kelompok di luar anggota kelompok *intranet*, tapi dengan otorisasi tertentu. Fitur *intranet* standar dalam suatu organisasi memiliki 4 kemampuan dasar yaitu *e-mail, on line publishing, on line searches* dan *application distribution*. Sedangkan *extranet* memperluas fitur ini ke partner bisnis jika dimungkinkan.

   Keuntungan menggunakan intranet di dalam suatu organisasi :

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

2.5.7 Transaksi Keuangan *Online*

Transaksi *online* merupakan transaksi yang dilakukan dengan menggunakan sarana internet. Transaksi ini tidak melibatkan manusia dalam prosesnya melainkan menggunakan mesin atau perangkat lunak dan perangkat keras computer. Karena menggunakan mesin maka transaksi ini dapat berjalan selama 24 jam tanpa berhenti.

Keuntungan menggunakan sistem pembayaran *online* bagi perusahaan dan *customer* yaitu sebagai berikut :

a. Keuntungan bagi perusahaan adalah transaksi dapat berjalan selama 24 jam penuh. *Customer* dapat berasal dari negara mana saja diseluruh dunia selama *customer* tersebut terhubung dengan internet. Transaksi berjalan dengan cepat.
b. Keuntungan bagi *customer* adalah transaksi dapat dilakukan kapan saja dan dari mana saja.

Kerugian menggunakan sistem pembayaran *online* bagi perusahaan dan *customer* yaitu sebagai berikut :

a. Kerugian bagi perusahaan adalah jika terjadi *fraund* atau penyalahgunaan data *customer* untuk transaksi dengan pihak *merchant.*
b. Kerugian bagi *customer* adalah rata-rata pihak *merchant* membebaskan biaya transaksi ke *customer.*

2.5.8 Keamanan *E-Commerce*

Masalah keamanan merupakan salah satu aspek penting dari sebuah sistem *E-Commerce*, beberapa solusi agar sebuah sistem tidak diakses oleh pihak yang tidak bertanggung jawab, diantaranya :

1. Teknologi Kriptografi

Teknologi kriptografi menjelaskan bagaimana mengamankan data dengan menggunakan enkripsi. Berbagai sistem sudah dikembangkan seperti sistem *private key* dan *public key*. Salah satu masalah dalam mengamankan enkripsi adalah bagaimana memastikan bahwa hanya sang penerima yang dapat mengakses data.

2. Konsultan Kemanan

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

2.5.8.1 *Secure Socket Layer* (SSL)

*Secure Socket Layer*atau SSL adalah metode standar yang digunakan untuk mengamankan proses komunikasi di internet, khususnya pada aktifitas penjelajahan (*browsing*) situs web. Dengan menggunakan SSL, suatu data/pesan yang dikirimkan antara *server* dengan *browser* computer *client* di enkripsi dengan menggunakan PKI (*public key infrastructure*). Dengan demikian apabila ada pihak yang tidak bertanggung jawab menyusup untuk mencuri data/pesan yang dikirim, penyusup tersebut tidak akan dapat menggunakan pesan yang telah terenkripsi tersebut.

2.6 Konsep Dasar Analisis Sistem

Konsep dasar analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang untuk kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi segala permasalahan yang terjadi.

2.6.1 Metode Pendekatan Sistem Berorientasi Objek

Metode berorientasi objek mulai berkembang ketika Grady Booch pada tahun 80-an mempublikasikan suatu paper bagaimana melakukan perancangan untuk bahasa ADA namun memberi judul *paper* tersebut dengan : *Object-*

Oriented Design. Selanjutnya ide tersebut terus ia kembangkan sampai tahun 90-an.

Pada tahun 1991 Peter Coad dan Yourdon memperkenalkan metode berorientasi objek yang lebih sederhana dibandingkan Booch. Metode ini menjadi cepat popular karena mendukung layanan-layanan yang terdapat pada C++. Pada waktu itu C++ merupakan bahasa pemrograman berorientasi objek yang paling popular.

Pada tahun 1991 juga Rumbaugh memperkenalkan *Object Modelling Technique*(OMT). Pendekatan yang digunakan tidak jauh berbeda dengan pendekatan yang digunakan Coad Yourdon namun dengan notasi yang berbeda. OMT tidak hanya sepenuhnya berbasis pada *data driven* tapi juga memisahkan proses dari data dengan penggunaan *data flow diagram* yang terpisah dengan diagram kelas. OMT juga menggunakan notasi *state transition diagram* untuk memodelkan aspek dinamis sistem.

Pada tahun 1994 Ivan Jacobson memperkenalkan konsep *use case* dan *object oriented software engireering*. Pada tahun 1994 itu juga yaitu bulan Oktober 1994 Booch, Rumbaugh dan Jacobson, mempelopori usaha untuk penyatuan notasi pendekatan berorientasi objek. Pada tahun 1995 dihasilkan *draft* pertama dari UML (versi 0.8). sejak tahun 1996 pengembangan tersebut dikoordinasikan oleh Object Management Group (OMG – htpp://www.omg.org).

Tahun 1997 UML versi 1.1 muncul, dan saat ini versi terbaru adalah versi 1.5 yang dirilis bulan Maret 2003. Booch, Rumbaugh dan Jacobson menyusun tiga buku serial tentang UML pada tahun 1999. Sejak saat itulah UML telah menjelma menjadi standar bahasa pemodelan untuk aplikasi berorientasi objek.

2.6.2 Keunggulan Pendekatan Sistem Berorientasi Objek

Pendekatan berorientasi objek dalam pengembangan sistem memiliki beberapa keunggulan dalam hal transisi dari dunia eksternal menjadi objek-objek perangkat lunak.

2.6.2.1 Menaikkan tingkat keterpakaian kembali (*reusability*)

Perangkat lunak bersifat dinamis. Hal ini disebabkan kebutuhan pengguna berubah dengan cepat. Perkembangan teknologi informasi dan kebutuhan akan pengolahan informasi itu memaksa setiap organisasi memperbarui sistemnya. Dengan demikian perangkat lunak harus dibangun dengan *reusability* tinggi. Metode yang mendukung *reusability* tersebut adalah metode berorientasi objek.

2.6.2.2 Menghilangkan kompleksitas transisi antar tahap pada pengembangan perangkat lunak

Pada pendekatan konvensional (terstruktur), notasi yang digunakan pada tahap analisis, perancangan dan tahap lainnya berbeda-beda. Hal ini menyebabkan transisi antar tahap pengembangan menjadi kompleks. Pada pendekatan berorientasi objek notasi yang digunakan pada tahap analisis, perancangan dan implementasi relatif sama.

2.6.3 Konsep dalam Metode Berorientasi Objek

Paradigma pemrograman berorientasi objek yakni semua data dan fungsi dibungkus dalam kelas-kelas atau objek-objek. Model data berorientasi objek dikatakan dapat member fleksibilatas yang lebih, kemudahan mengubah program dan digunakan luas dalam teknik piranti lunak skala besar.

2.6.3.1 Objek

Objek adalah suatu kombinasi antara data dan aspek logik yang merepresentasikan suatu entitas dunia nyata. Contoh : mobil Toyota kijang adalah suatu objek, maka yang menjadi atribut adalah warna, pembuat, harga, konsumsi, bahan bakar, mekanisme pengereman. Sedangkan yang menjadi aspek logiknya adalah menampilkan jarak tempuh, berhenti, berjalan, belok.

2.6.3.2 Kelas

Pada sistem berorientasi objek, kelas didefinisikan sebagai himpunan objek yang memiliki struktur umum dan perilaku umum yang sama. Sedangkan

objek yang dapat dikatakan sebagai instansiasi suatu kelas. Sebagai contoh mobil adalah suatu kelas, maka mobil Toyota kihang, mobil Proton adalah objek.

2.6.3.3 Atribut

Atribut merepresentasikan karakteristik atau keadaan objek. Pada contoh kasus di atas, sebuah mobil dapat memiliki atribut warna, harga, dan pembuat. Pada tataran implementasi, warna dapat direpresentasikan sebagai suatu *string* (domain nilainya misalnya : merah, biru, kuning). Harga dapat berupa bilangan *floating point* atau bilangan integer. Sedangkan pembuat dapat bertipe struktur yang terdiri dari nama, identitas korporat.

2.6.3.4 Metode

Metode adalah suatu fungsi atau prosedur yang didefinisikan untuk dapat mengakses keadaan internal suatu objek dari suatu kelas. Tiap fungsi atau prosedur mendefinisikan dan mendeskripsikan perilaku khusus suatu objek. Sebagai contoh : kelas Pegawai memiliki metode Hitung Gaji. Metode sebenarnya merupakan antarmuka yang disediakan untuk dapat memanfaatkan perilaku objek tersebut. Sebagai contoh : jika diinginkan dilakukannya perhitungan gaji, maka *message* "Hitung Gaji" harus dikirimkan ke objek Pegawai.

2.6.3.5 Message

Message pada dasarnya adalah pemanggilan fungsi. Namun *message* berbeda dari pemanggilan subrutin. Dengan *message* yang sama dua objek berbeda dapat melakukan operasi yang berbeda pula. Konsep ini dikenal sebagai *Polymorphism.*

2.6.3.6 Enkapsulasi

Enkapsulasi memadukan karakteristik unit di dalam suatu objek (data dan metode). Konsep ini bertujuan untuk menyembunyikan informasi dan karakteristik objek. Objek dapat dimanfaatkan hanya dengan cara memanggil metode yang dimiliki objek tersebut.

2.7 *Unified Modeling Language* (UML)

UML mengintegrasikan konsep dari Booch, OMT, OOSE, dan juga *Class-Relation* dengan menggabungkan mereka menjadi suatu kesatuan bahasa modeling yang bisa berguna bagi siapa saja. UML bertujuan menjadi standar bahasa modeling yang mampu memodelkan sistem yang konkuren dan juga terdistribusi. UML sudah menjadi standar industri yang dibuat dibawah pengawasan *Object Management Group* (OMG). OMG ditugaskan untuk memberi informasi mengenai *Object-Oriented Methodology* yang mampu untuk membuat bahasa untuk modeling software. Dengan menggunakan UML para pengembang dapat membuat model untuk semua jenis aplikasi perangkat lunak, dimana aplikasi tersebut dapat berjalan dalam perangkat keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa pemrograman berorientasi objek seperti Java, C++, C#, VB.NET, PHP Framework dan lain-lain. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi procedural dalam Visual Basic atau C.

UML bukan merupakan metode dan tidak juga berusaha untuk menjadi suatu metode, namun demikian ia merupakan kebutuhan *Rational Unified Process* yang telah dibuat oleh *Rational Software Corporation*. Metode depelopment lainnya seperti *Dynamic Systems Depelopment Method* juga menggunakan teknik dari UML. UML menyediakan kemampuan untuk menangkap karakteristik sebuah sistem dengan menggunakan notasi-notasi tertentu. UML menyediakan sederetan notasi sederhana yang mudah dipahami untuk mendokumentasikan sistem berdasarkan prinsip-prinsip perancangan berorientasi objek. UML sudah menjadi standarisasi *Modeling 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

2.7.1 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.

Adapun konsep dasar dalam UML untuk mendeskripsikan OOP (*Object Oriented Programming*) dengan menggunakan beberapa diagram menurut [8], antara lain :

2.7.1.1 *Use Case Diagram*

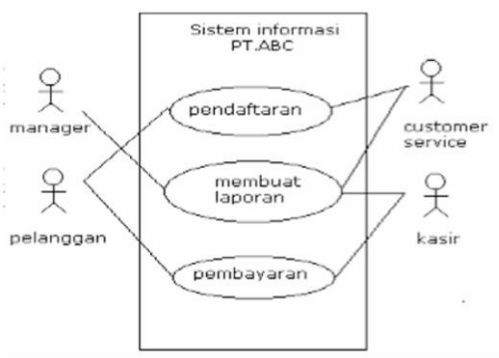
Use Case Diagram adalah gambaran fungsionalitas dari suatu sistem, sehingga pelanggan atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun. *Use case modeling* digunakan untukmendokumentasikan sistem *behaviour* dan subsistem pada saatpengembangan sistem, termasuk di dalamnya fungsi internal suatu system (*use case*), pengguna sistem (*user*) dan hubungan interaksi antarakeduanya (*use case diagram*).

Bagian utama dalam *use case modeling,* yaitu :

1. *Actor* adalah perwujudan dari pengguna sistem, proses dan segalasesuatu yang berinteraksi dalam sistem tersebut. *Actor* tidak termasukdalam sistem, tapi dapat menggambarkan interaksi dari *external actor*dengan sistem tersebut. Setiap *actor* berinteraksi dengan satu ataulebih *use case* dengan pertukaran pesan.
2. *Use case* merupakan bagian dari suatu sistem yang menyediakan sebuah fungsi atau tugas tertentu dan terdiri dari urutan aksi. *Use case* memperlihatkan *external behaviour* dari sebuah sistem yang dilihat dari

segi pengguna eksternal. *Use case* tidak seperti *operation* karena sebuah *use case* dapat terus menerima *input* dari *actor* pada saat dijalankan.

3. *System Boundary* menjelaskan batasan suatu sistem dengan lingkungannya, sehingga memberi batasan yang jelas sampai dimanasuatu sistem bekerja, termasuk membatasi sistem dengan *actor* yangberada di luar dari sistem. Gambar mengenai contoh *use case diagram* dapat dilihat pada gambar 2.5.



Gambar 2.5 *Use Case Diagram*

2.7.1.2 *Class Diagram*

Class adalah sebuah spesifikasi yang jika diinstanisasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/property) 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 objek 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 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 metoda. 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 definisi 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 bisa 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 mendefinisikan relasinya kelas harus memilih salah satu dari kedua kelas tersebut, sedangkan kondisi untuk memilih keduanya adalah *invalid*.

3. Kelas Asosiasi (*Association Class*)

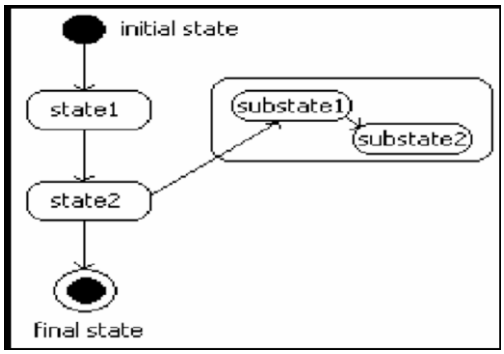
Sebuah asosiasi dapat memiliki atribut dan operasi seperti halnya sebuah *class*. Sebuah *association class* sebenarnya diperlukan apabila salah satu dari kelas yang terhubung mempunyai sebuah atau beberapa atribut yang tidak layak dimiliki kelas tersebut, karena secara logis atribut tersebut lebih layak 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.

2.7.1.3 *Activity Diagram*

Activity Diagram menggambarkan aliran aktivitas dalam sistem yang dirancang, yang setiap aliran berawal dari sebuah *decision* yang mungkin terjadi dan bagaimana mereka berakhir. *Activity diagram* merupakan *state diagram* khusus, diaman sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya. *Activity diagram* tidak menggambarkan *behavior internal* sebuah sistem dan interaksi antar subsistem secara fisik, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Sebuah aktivitas dapat direalisasikan dari sebuah *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem dalam melakukan aktivitas. Diagram *activity* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab dalam melakukan aktivitas. Notasi dari *activity diagram* dapat dilihat pada gambar 2.6.

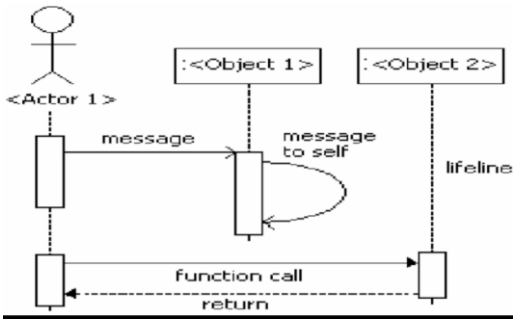


Gambar 2.6 ***Activity Diagram***

2.7.1.4 *Sequence Diagram*

Sequence diagram menggambarkan perilaku pada sebuah *scenario*. Diagram ini menunjukkan sejumlah contoh objek dan *message* (pesan) yang diletakkan diantara objek-objek didalam *use case. Sequence diagram* terdiri antar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). Perilaku yang digambarkan adalah rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan sebuah *output.* Diawali dari apa yang meng-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

Masing-masing objek, termasuk aktor memiliki *lifeline* vertical. *Message* yang digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class. Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message.*Notasi dari *sequence diagram* dapat dilihat pada gambar 2.7.



Gambar 2.7 ***Sequence Diagram***

2.7.1.5 *Collaboration Diagram*

Collaboration diagram adalah bentuk lain dari *sequence diagram*. Bila *sequence diagram* diorganisir menurut waktu, maka *collaboration diagram* diorganisir menurut ruang/*space. Collaboration diagram* merupakan asosiasi diantara objek panah didekat garis asosiasi menunjukkan *message*, sedangkan *content message* ditujukan dengan label. Dengan *collaboration diagram* memungkinkan untuk memodelkan pengiriman sebuah *message* kebanyak objek pada *class* yang sama, demikian juga halnya dengan menunjukkan adanya objek aktif yang mengendalikan aliran *message.*

2.7.1.6 *Component Diagram*

Component diagram menggambarkan organisasi serta kebergantungan pada komponen-komponen yang telah ada sebelumnya. *Component diagram* ini berhubungan dengan *class diagram* dimana komponen secara tipikal dipetakan kedalam satu atau lebih *class-class, interface-interface* serta kolaborasi-kolaborasi.

2.7.1.7 *Deployment Diagram*

Deployment diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, dimana komponen akan terletak (pada mesin, server atau perangkat keras lainnya), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server dan hal-hal lainnya yang bersifat fisikal sebuah *node* adalah server *workstation*, atau perangkat keras lain yang digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya.

2.8 Framework

Framework adalah kerangka kerja. Framework juga dapat diartikan sebagai kumpulan script (terutama *class* dan *function*) yang dapat membantu developer/programmer dalam menangani berbagai masalah-masalah dalam pemrograman seperti koneksi ke database, pemanggilan variable, file dll.

Sehingga developer lebih focus dan lebih cepat membangun aplikasi. Bisa juga dikatakan Framework adalah komponen pemrograman yang siap *re-use* kapan saja, sehingga programmer tidak harus membuat script yang sama untuk tugas yang sama. Misalkan saat anda membuat aplikasi web berbasis ajax yang setiap kali harus melakukan XMLHttpRequest, maka Xajax telah mempermudah untuk anda dengan menciptakan sebuah objek khusus yang siap digunakan untuk operasi Ajax berbasis PHP.

2.8.1 Jenis-jenis Framework

Dalam bahasa pemrograman PHP, terdapat beberapa jenis framework yang dapat menjadi pilihan para programmer dalam membangun sebuah website, diantaranya :

a. Yii
b. CodeIgniter
c. CakePHP
d. Symphony
e. Zend
f. Kohana

2.8.2 Manfaat Framework

Beberapa manfaat yang dapat diperoleh dengan menggunakan framework dalam pengembangan aplikasi, diantaranya :

a. Membantu kerja developer dalam membangun aplikasi sehingga aplikasi bisa selesai dalam waktu yang singkat.
b. Penerapan *Design Pattern* memudahkan dalam rancangan, pengembangan dan pemeliharaan sistem.
c. *Stability* dan *Reliability*, aplikasi yang kita bangunlebih stabil dan handal karena berbasis pada framework yang sudah teruji stabilitas dan kehandalannya.
d. *Coding Style* konsisten, memudahkan dalam membaca kode dan dalam menemukan bugs.

e. *Security Concern*, framework mengantisipasi dan memasang perisai terhadap adanya berbagai masalah keamanan yang mungkin timbul.
f. Dokumentasi, framework dapat mendisiplinkan kita untuk menulis dokumentasi untuk apa yang kita tulis.

2.8.3 Kelebihan Framework

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 diwajibkan untuk memahami dari pola kerja framework tersebut selebihnya anggota team hanya mempelajari proses bisnis yang dikehendaki oleh sistem untuk kemudian dituangkan 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.

2.8.4 Kekurangan Framework

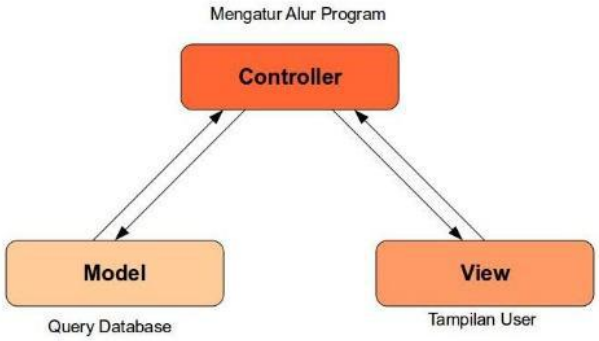
Ada beberapa kekurangan jika harus menggunakan framework, diantaranya :

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

2.8.5 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. Sehingga kode-kode penyusun sebuah aplikasi web terlihat lebih sederhana dan terstruktur. Jadi, tidak akan ada *query database* didalam sebuah file yang memang ditujukan untuk tampilan.



Gambar 2.8 Gambaran Konsep MVC [4]

MVC (Model, View, Controller) :

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 detail dari implementasi *user interface* dengan melibatkan komponen grafis yang menyediakan representasi proses internal aplikasi dan menuntun alur interaksi *user* terhadap aplikasi.
c. **Controller** adalah sebuah layer pada MVC yang menyediakan detail 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*.

2.8.6 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.

2.8.6.1 Kelebihan CodeIgniter

Framework CodeIgniter memiliki beberapa kelebihan, diantaranya :

a. CodeIgniter merupakan framework yang gratis, dimana kita bisa mengunduh dan menginstallnya tanpa harus membayar.
b. CodeIgniter sangat ringan, dimana sistem utama hanya memerlukan sedikit sekali library. Berbeda dengan kebanyakan framework yang membutuhkan banyak resources.
c. Framework yang cepat dibandingkan framework PHP lainnya.
d. CodeIgniter dapat dengan mudah diperluas kemampuannya sesuai dengan yang kita inginkan melalui penggunaan *plugin* dan *helper*.
e. 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.
f. Dokumentasi yang sangat lengkap : Setiap paket instalasi codeigniter sudah disertai *user guide* yang sangat bagus dan lengkap untuk dijadikan permulaan, bahasanya pun mudah dipahami.

2.8.6.2 Kekurangan CodeIgniter

Framework CodeIgniter juga tidak terlepas dari beberapa kekurangan, diantaranya :

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.

2.9 PHP (*PHP Hypertext Preprosessor*)

PHP singkatan dari *PHP Hypertext Preprosessor.* Ia merupakan bahasaberbentuk skrip yang ditempatkan dalam server dan diproses di server. Hasilnyayang dikirimkan ke klien, tempat pemakai menggunakan *browser.*

Secara khusus, PHP dirancang untuk membentuk *web* dinamis. Artinya, iadapat membentuk suatu tampilan berdasarkan permintaan terkini. Misalnya, andabisa menampilkan isi *database* ke halaman *Web*. Pada prinsipnya, PHPmempunyai fungsi yang sama dengan skrip-skrip seperti ASP (*Active ServerPage*), *Cold Fusion*, ataupun *Perl.*

Kelahiran PHP bermula saat Rasmus lerdorf membuat sejumlah *skrip Perl*yang dapat mengamati siapa saja yang melihat-lihat daftar riwayat hidupnya,yakni pada tahun 1994. Skrip-skrip ini selanjutnya dikemas menjadi *tool* yangdisebut "*Personal Home Page*". Paket inilah yang menjadi cikal-bakal PHP. Padatahun 1995, Rasmus menciptakan PHP/F1 Versi 2. Pada versi inilah pemrogramandapat menempelkan kode terstruktur di dalam tag HTML. Yang menarik, kodePHP juga bisa berkomunikasi dengan *database* dan melakukan perhitunganperhitunganyang kompleks sambil jalan.

PHP cukup populer sebagai peranti pemrograman *web*,terutama di lingkungan Linux. Walaupun demikian, PHP sebenarnya juga dapatberfungsi pada server-server yang berbasis UNIX, Windows NT, dan Macintosh.Bahkan versi untuk Windows 95/98 pun tersedia.

Pada awalnya, PHP dirancang untuk diintegrasikan dengan *web serverApache*. Namun, belakangan PHP juga dapat bekerja dengan *web server* sepertiPWS (*Personal Web Server*), IIS (*Internet Information Server*), dan Xitami.

Untuk mencoba PHP, anda tidak perlu menggunakan komputer berkelasserver. Dengan hanya sebuah komputer biasa, Anda bisa mempelajari danmempraktekan PHP.

2.10 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis dataSQL atau DBMS yang *multithread* dan*multi-user*.

MySQL adalah Relational Database Management System (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat closed source atau komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (Structured Query Language). SQL adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu sistem database (DBMS) dapat diketahui dari cara kerja optimizer-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh user maupun program-program aplikasinya.

Sebagai database server, MySQL dapat dikatakan lebih unggul dibandingkan database server lainnya dalam query data. Hal ini terbukti untuk query yang dilakukan oleh single user, kecepatan query MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan Interbase.

Selain itu MySQL juga memiliki beberapa keistimewaan, antara lain :

a. Portability

   MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.

b. Open Source

MySQL didistribusikan secara open source (gratis), dibawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.

c. Multiuser

MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

d. Perfomance Tuning

MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

e. Column Types

MySQL memiliki tipe kolom yang sangat kompleks, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.

f. Command and Functions

MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam query.

g. Security

MySQL memiliki beberapa lapisan sekuritas seperti level subnetmask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta password terenkripsi.

h. Scalability dan limits

MySQL mampu menangani database dalam skala besar, dengan jumlah records lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

i. Connectivity

MySQL dapat melakukan koneksi dengan client menggunakan protokol TCP/IP, Unix soket (UNIX), atau Named Pipes (NT).

j. Localisation

MySQL dapat mendeteksi pesan kesalahan pada client dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa Indonesia belum termasuk didalamnya.

k. Interface

MySQL memiliki interface (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).

l. Clients dan tools

MySQL dilengkapi dengan berbagai tool yang dapat digunakan untuk administrasi database, dan pada setiap tool yang ada disertakan petunjuk online.

m. Struktur tabel

MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan database lainnya semacam PostgreSQL ataupun Oracle.

Perintah-perintah MySQL itu antara lain adalah sebagai berikut :

1. *Create database*, digunakan untuk membuat *database* pada *database server*. Sintaksnya adalah :

   Create database database_name

   Database_name adalah nama *database* yang akan dibuat.

2. *Use database*, digunakan untuk menunjuk *database* yang akan digunakan. Sintaksnya adalah :

   Use database_name

   Database_name adalah nama *database* yang akan digunakan.

3. *Create table*, digunakan untuk membuat tabel pada *database* yang digunakan. Sintaksnya adalah :

   Create table table_name
   (
   Column1 column_type column_attributes,
   Column2 column_type column_attributes,
   Primary_key (column_name)
   );

   Table_name adalah nama tabel yang akan dibuat. *Column1* adalah nama kolom yang akan dibuat pada tabel. *Column_type* adalah tipe dari kolom tersebut , dapat berupa *char, varchar, tinytext, mediumtext, longtext, enum,*

int, tinyint, mediumint, bigint, float, decimal, time, date, datetime,timestamp, year.

2.11 PSPad editor

Universal freeware Editor, bermanfaat bagi orang yang:

a. Bekerja dengan teks biasa - editor memiliki kekayaan fungsi format, termasuk spell checker
b. Membuat halaman web - sebagai editor web authoring, PSPad berisi banyak alat unik yang menghemat waktu Anda
c. Menggunakan IDE yang baik untuk compiler - PSPad menangkap dan mem-parsing compiler output, mengintegrasikan file bantuan eksternal, membandingkan versi.

Tidak ada proses instalasi yang kompleks, PSPad siap bekerja segera tanpa memerlukan kustomisasi. Editor mendukung banyak jenis dan bahasa berkas, dengan sintaks. Ada macro, file klip dan template untuk mengotomatisasi tugas-tugas repetative. Terpadu HEX Editor, dukungan Proyek, FTP Client, Perekam Makro, File Search / Ganti, Kode Explorer, Kode halaman konversi ini hanya beberapa dari banyak fitur yang menawarkan PSPad.

2.12 Paypal

Paypal adalah salah satu alat pembayaran (*Payment procesors*)menggunakan internet yang terbanyak digunakan didunia dan teraman. Penggunainternet dapat membeli barang di ebay, lisensi software original, keanggotaansitus, urusan bisnis, mengirim uang ke pengguna Paypal lain di seluruh dunia danbanyak fungsi lainnya dengan mudah dan otomatis menggunakan internet, Paypalmengatasi kekurangan dalam pengiriman uang tradisional seperti Cek atau Moneyorder yang prosesnya dapat memakan waktu. Pelanggan anda bisa membayardengan saldo PayPal mereka, dengan cek, atau bahkan dengan kartu kredit dankartu debit mereka. PayPal lebih aman dari alat pembayaran online lain.

Kebijaksanaan perlindungan tertulis untuk pembeli yang menggunakanPayPal menyatakan pembeli yang menggunakan PayPal dapat melakukankomplain dalam waktu 45 hari jika pembeli belum mendapatkan barang yangdipesan atau jika barang yang dipesan tidak sesuai deskripsi yang diberitahukanpenjual. Jika pembeli menggunakan kartu kredit akan mendapatkan pengembalianuang *charge back* dari perusahaan kartu kreditnya.

PayPal juga melindungi penjual dari pengembalian uang atau komplaintergantung situasi dan pembuktian. Kebijakan perlindungan tertulis untuk penjualdirancang untuk melindungi penjual dari klaim pembeli yang mengaku telahmengirim uang yang tidak ada catatan bukti pembayarannya. Setiap pembelianmenggunakan PayPay selalu ada catatan bukti pembayarannya di akun PayPaypengirim dan penerima uang, sebagai bukti jika benar telah terjadi pengirimanuang. Dari sini bisa diambil kesimpulan menggunakan PayPal lebih aman dari alatpembayaran *online* lain bagi pengirim dan penerima uang karena ada buktipembayaran dan dapat melakukan komplain jika terjadi sesuatu.

2.13 *Hypertext Markup Language* (HTML)

HyperText Markup Language (HTML) adalah sebuah bahasa *markup* yang digunakan untuk membuat sebuah halaman *web*, menampilkan berbagai informasi di dalam sebuah web *browser* internet dan *formating hypertext* sederhana yang ditulis kedalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegerasi. 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)*.