BAB II

LANDASAN TEORI

2.1 Tinjauan Perusahaan

Tinjauan perusahaan menjelaskan tentang profil perusahaan tempat penelitian yang terdiri dari sejarah perusahaan, visi dan misi, struktur organisasi dan deskripsi yang ada di toko Rumah Buku Bandung.

2.2 Sejarah Rumah Buku

Rumah Buku didirikan pada tanggal 9 September 2009 yang berlokasi di jalan WR. Supratman Bandung, dengan misi menjadi sebuah toko buku diskon termurah dan menunjang buku-buku berkualitas dikalangan mahasiswa dan masyarakat sekitar. Dengan semangat berbagi, Rumah Buku didirikan sebagai alternatif pusat referensi buku yang cukup lengkap. Toko Rumah Buku sampai saat ini telah tumbuh dan berkembang menjadi salah satu toko buku yang cukup besar di kota bandung.

2.2.1 Visi dan Misi

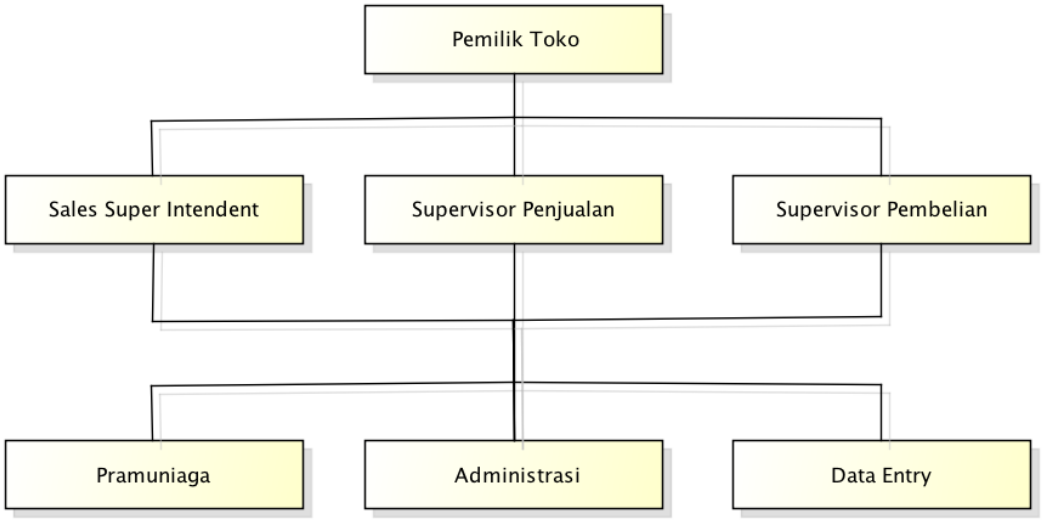
Visi dan misi dari toko Rumah Buku adalah menjadi toko buku diskon termudah di kalangan institusi perguruan tinggi yang berorientasikan kebutuhan mahasiswa dengan tetap mengedepankan pelayanan unggul yang bersifat interaktif demi kepuasan pelanggan dengan dasar "*win the heart*".

2.2.2 Struktur Organisasi

Struktur organisasi merupakan susunan yang terdiri dari fungsi-fungsi dan hubungan-hubungan yang menyatakan keseluruhan kegiatan untuk mencapai

suatu sasaran. Secara fisik struktur organisasi dapat dinyatakan dalam bentuk bagan yang memperlihatkan hubungan unit-unit organisasi dan garis-garis wewenang yang ada.

Beberapa keuntungan yang dapat diperoleh dari penggunaan bagan organisasi adalah dapat memperlihatkan karateristik utama dari suatu perusahaan tersebut, tentang gambaran pekerjaan dan hubungan-hubungan yang ada didalam perusahaan serta digunakan untuk merumuskan rencana kerja yang ideal sebagai pedoman untuk dapat mengetahui siapa bawahan dan atasannya. Struktur keorganisasian pada Toko Rumah Buku dapat dilihat pada bagan struktur organisasi pada gambar 2.1berikut ini :



Gambar 2.1 Struktur Organisasi pada toko Rumah Buku

2.2.3 Deskripsi Pekerjaan

Definisi tugas digunakan untuk mengetahui tugas, wewenang, tanggung jawab dari masing-masing bagian. Definisi tugas yang ada di Toko Rumah Buku adalah sebagai berikut :

1. Pemilik Toko
    a. Pengambil keputusan.
    b. Sebagai koordinator semua kegiatan yang telah dilaksanakan.
    c. Mengevaluasi semua kegiatan yang telah dilaksanakan.
    d. Menerapkan dan mengesahkan kebijakan yang menyangkut eksistensi toko.
    e. Melaksanakan pemeriksaan yang meliputi seluruh aspek kegiatan manajemen keuangan dan operasional agar pengelolaan dapat dilaksanakan secara efektif dan efisien.
2. *Sales Super Intendent*

    Sales super intendent adalah *supervisor* senior yang bertugas untuk mengawasi dan membantu semua pekerjaan dari *supervisor*.
3. *Supervisor* Penjualan
    a. Mengendalikan operasional penjualan toko buku seperti menyusun jadwal tugas pramuniaga.
    b. Menjalankan program kerja penjualan dari kepala toko.
    c. Menarik data dari *cash register* dan mesin EDC
    d. Memotivasi dan membina karyawan.
    e. Mengamati pasar dan pesaing diluar serta mengkasi ulang strategi penjualan.
4. *Supervisor* Pembelian
    a. Membuat rekomendasi *order* dan retur buku melalui RBS dan mengirimkan PO melalui email.

b. Mengupayakan buku-buku pesanan *customer* dan konfirmasi *supplier*.
c. Membuat surat pesanan buku baru untuk setiap *supplier* yang diterima sebagai pemasok.
d. Mengadakan produk dengan kondisi yang menguntungkan (tepat jumlah, tepat waktu, diskon optiman, syarat dan termin pembayaran yang menguntungkan) dan memperhatikan perputaran barang.
e. Membina hubungan baik dengan *supplier* dan menyeleksi *supplier* yang ingin menjadi pemasok.

5. Pramuniaga

   Pramuniaga bertugas untuk melayani konsumen secara langsung dan menjaga kebersihan toko.

6. Administrasi
   a. Menyiapkan modal kasir dan menghitung setoran kasir.
   b. Mengerjakan data setoran kasir ke program adm kas dan mencatat omset harian pada buku kas harian, buku mutasi dan membuat jurnal memorial.
   c. Memasukan data-data faktur pajak kedalam program pajak dan memasukan pajak masuk dan keluar.

7. *Data Entry*

   Data Entry bertugas untuk memasukan semua data buku, mencetak *barcode, input* data atau cari data penerimaan dan merubah harga buku apabila ada perubahan harga.

2.3 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.

2.4 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*) dan sasaran (*objectives*) dan tujuan (*goal*) **JOG[4]**.

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 menujukkan ruang lingkup (*scope*) dari sistem tersebut

3. Lingkungan Luar Sistem (*Environments*)

   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 komputer. 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*)

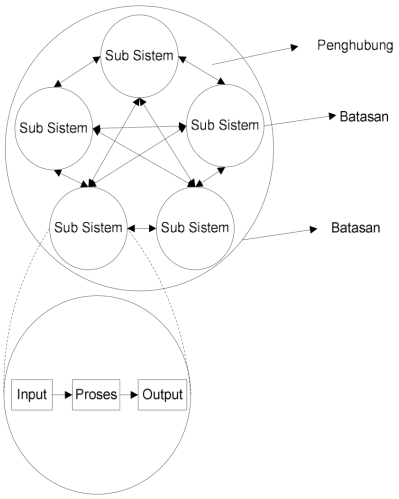
   Keluaran (*Output*) merupakan hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna.

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

2.5 Klasifikasi Sistem

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

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 diperdiksi. 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 **JOG[4]**.

2.6 Konsep Dasar Informasi

Informasi dalam sebuah organisasi sangat penting peranannya. Suatu sistem yang kurang mendapatkan informasi akan menjadi lemah dan akhirnya berakhir. 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 **JOG[4]**.

2.7 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 **JOG[4]**.

2.7.1 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 **JOG[4]**.

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, 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.

2.8 *Internet*

Internet merupakan suatu jaringan komputer global yang menghubungkan jaringan *private* dan *public* untuk berbagi informasi (antar lembaga pendidikan, penelitian, pemerintahan, bisnis, masyarakat umum) dan standar komunikasi yang berbasis protokol TCP/IP**.** Saat ini orientasi *web* mulai digunakan untuk komersial sejak tahun1995**.** Dikenal sebagai basis (*foundation*) dari "*New Digital Networked Economy*" Karena sangat esensial, di Amerika Serikat banyak inisiatif yang dimotori oleh pemerintah untuk menjamin dominasi Amerika di bidang ekonomi baru ini Bandung *High Tech Valley* (BHTV), salah satu inisiatif Indonesia di negara lain yang perkembangannya pesat.

2.9 *E-Commerce*

Electronic Commerce (Perniagaan elektronik), sebagai bagian dari *electronic business* (bisnis yang dilakukan dengan menggunakan *electronic transmission*), secara umum *e-commerce* dapat didefinisikan sebagai segala bentuk transaksi perdagangan atau perniagaan barang atau jasa (*trade of goods and service*) dengan mengunakan media elektronik. Onno W.Purbo dan Aang Wahyudi yang mengutip pendapatnya David Baum menyebutkan bahwa *e-commerce* merupakan suatu set dinamis teknologi, aplikasi dan proses bisnis yang menghubungkan perusahaan, konsumen dan komunitas melalui transaksi elektronik dan perdagangan barang, pelayanan dan informasi yang dilakukan secara elektronik. Jika di klasifikasikan *e-commerce* terbagi menjadi tiga tipe aplikasi, yaitu:

1. *Electronic Market* (EMs)

   EMs adalah sebuah sarana yang menggunakan informasi dan komunikasi untuk melakukan atau menyajikan penawaran dalam sebuah segmen pasar, sehingga pembeli dapat membandingkan berbagai macam harga yang ditawarkan. Dalam pengertian lain, EMs adalah sebuah sistem informasi antara organisasi yang menyediakan fasilitas-fasilitas bagi para penjual dan pembeli untuk bertukar informasi tentang harga dan produk yang ditawarkan. Keuntungan fasilitas EMs bagi pelanggan adalah terlihat lebih nyata dan produk yang ditawarkan. Sedangkan bagi penjual dapat mendistribusikan informasi mengenai produk dan *service* yang ditawarkan lebih cepat sehingga dapat menarik pelanggan lebih banyak.

2. *Elektronik Data Interchange* (EDI)

   EDI adalah sarana untuk mengefisiensikan pertukaran data transaksi-transaksi regular yang berulang dalam jumlah besar antara organisasi-organisasi komersial. Secara formal Edi didefinisikan oleh *International Data Exchange Associaton* (IDEA) sebagai *transfer* data terstruktur dengan format standard yang telah disetujui yang dilakukan dari satu sistem komputer ke sistem komputer yang lain dengan menggunakan media elektronik. EDI sangat luas penggunaanya, biasanya digunakan oleh kelompok *retail* yang besar ketika melakukan bisnis dagang dengan para *supplier*. EDI memiliki standarisasi pengkodean transaksi perdagangan, sehingga organisasi komersial tersebut dapat berkomunikasi secara langsung dari satu sistem komputer yang satu ke sistem komputer yang lain tanpa memerlukan

hardcopy, faktur, serta terhindar dari penundaan, kesalahan yang tidak disengaja dalam penanganan berkas dan dapat mengurangi biaya operasional, mengurangi kesalahan, memperoleh respon yang cepat, pengiriman faktur yang cepat dan akurat serta pembayaran dapat dilakukan secara elektronik dengan keamanan yang terjamin.

3. *Internet Commerce*

   Internet commerce adalah penggunaan *internet* yang berbasis teknologi informasi dan komunikasi untuk perdagangan. Kegiatan komersial ini seperti iklan dalam penjualan produk dan jasa. Transaksi yang dapat dilakukan di internet antara lain pemesanan/pembelian barang dimana barang akan kirim melalui pos atau saran lain setelah uang ditransfer ke rekening penjual. Penggunaan internet sebagai media pemasaran dan saluran penjualan terbukti mempunyai keuntungan antara lain untuk beberapa produk tertentu lebih sesuai ditawarkan melalui internet harga lebih murah mengingat membuat situs di internet lebih murah biayanya dibandingkan dengan membuka outlet retail diberbagai tempat dengan harga yang relatif lebih murah serta pembelian melalui *internet* akan diikuti dengan layanan pengantaran barang sampai ditempat pemesan.

2.9.1 Keuntungan *E-Commerce*

Keuntungan yang diperoleh dengan menggunakan transaksi melalui *e-commerce* bagi suatu perusahaan adalah sebagai berikut :

1. Meningkatkan pendapatan dengan menggunakan *online channel* dengan biaya yang lebih murah.

2. Mengurangi biaya-biaya yang berhubungan dengan kertas, seperti biaya pos surat, pencetakan, *report*, dan sebagainya.
3. Mengurangi keterlambatan dengan mengunakan *transfer* elektronik atau pembayaran yang tepat waktu dan dapat langsung dicek.
4. Mempercepat pelayanan ke pelanggan, dan pelayanan lebih responsif.

2.9.2 Mekanisme *E-Commerce*

Transaksi elektronik (pihak yang menawarkan barang atau jasa melalui internet) dengan *e-customer*, (pihak yang membeli barang atau jasa melalui internet) yang terjadi di dunia maya atau di internet pada umumnya berlangsung secara *paperless transaction*, sedangkan dokumen yang digunakan dalam transaksi tersebut bukanlah *paper document*, melainkan dokumen elektronik (*digital document*).

Kontak *online* dalam *e-commerce* menurut Santiago Cavanilas dan A.Martines Nadal, seperti yang dikutip oleh Arsyad Sanusi memiliki banyak tipe dan variasi yaitu:

1. Kontak melalui *chatting* dan *video conference*

   Chatting dan *video conference* adalah fasilitas yang disediakan oleh internet yang biasa digunakan untuk dialog interaktif secara langsung. *Chatting* memungkinkan seseorang dapat berkomunikasi secara langsung dengan orang lain persis seperti telpon, hanya saja komunikasi lewat *chatting* ini adalah tulisan atau pernyataan yang terbaca pada komputer masing-masing. Sesuai dengan namanya, *video conference* adalah alat untuk berbicara dengan

beberapa pihak dengan memiliki gambar dan mendengar suara secara langsung pihak yang dihubungi dengan alat ini.

2. Kontak melalui *e-mail*

   Kontak melalui *email* adalah salah satu kontak *online* yang populer karena pengguna *e-mail* saat ini sangat banyak dan mendunia dengan biaya yang sangat murah dan waktu yang efisien.

3. Kontak melalui *web* atau situs

   Kontak melalui *web* dapat dilakukan dengan menggunakan *web* seorang *supplier* (baik yang berlokasi di *server supplier* maupun diletakkan pada *server* pihak ketiga) memiliki diskripsi produk atau jasa dan satu seri halaman yang bersifat *self-contractio,* yaitu dapat digunakan untuk membuat kontrak sendiri, yang memungkinkan pengunjung web untuk memesan produk atau jasa tersebut. Para konsumen harus menyediakan informasi *personal* dan harus menyertakan nomor kartu kredit.

2.9.3 Karakteristik *E-Commerce*

Berbeda dengan transaksi perdagangan biasa, transaksi *e-commerce* memiliki beberapa karakteristik yang sangat khusus, yaitu:

1. Transaksi tanpa batas

   Sebelum *internet*, batas-batas geografi menjadi penghalang suatu perusahaan atau individu yang ingin *go-internasional*. Sehingga, hanya perusahaan atau individu dengan modal besar yang dapat memasarkan produknya secara internasional.

2. Transaksi Anonim

   Para penjual dan pembeli dalam transaksi melalui *internet* tidak harus bertemu muka satu sama lainnya. Penjual tidak memerlukan nama dari pembeli sepanjang pembayaran telah diotorisasi oleh penyedia sistem pembayaran yang ditentukan biasanya dengan kartu kredit ataupun dengan fasilitas pembayaran *online* seperti *paypal*.

3. Transaksi *digital* dan non *digital*

   Produk-produk seperti *software* komputer, musik dan produk lain yang bersifat digital dapat dipasarkan melalui *internet* dengan cara mengunduh secara elektronik. Dalam perkembangannya objek yang ditawarkan melalui *internet* juga meliputi barang-barang kebutuhan hidup lainnya.

4. Produk barang tak berwujud

   Banyak perusahaan yang bergerak dibidang *e-commerce* dengan menawarkan barang tak berwujud seperti data, *software* dan ide-ide yang dijual melalui *internet*.

2.10 *Paypal*

Paypal adalah salah satu alat pembayaran dengan menggunakan *internet* yang terbanyak digunakan orang di dunia karena merupakan alat pembayaran teraman. *Paypal* mengatasi kekurangan dalam pengiriman uang tradisional seperti cek atau *money order* yang prosesnya dapat memakan waktu. Layaknya rekening bank, pertama *member* diwajibkan membuat *account*, lalu mengisi *account* tersebut dengan dana dari kartu kredit yang dapat diterima *Paypal* setelah itu *paypal* sudah dapat digunakan untuk bertransaksi.

Paypal juga melindungi penjual dari pengembalian uang atau komplain tergantung situasi dan pembuktian. Kebijakan perlindungan tertulis untuk penjual dirancang untuk melindungi penjual dari klaim pembeli yang mengaku telah mengirim uang yang tidak ada catatan bukti pembayarannya, setiap pembeli yang menggunakan *Paypal* selalu ada catatan bukti pembayarannya di *account Paypal* pengirim dan penerima uang, sebagai bukti jika benar telah terjadi pengiriman uang.

2.11 SSL (*Secure Socket Layer*)

Secure Sockets Layer atau yang disingkat SSL adalah sebuah protokol keamanan data yang digunakan untuk menjaga pengiriman data antara *web server* dan pengguna situs *web* tersebut. SSL umumnya sudah terpasang didalam mayoritas *browser web* yang ada (*IE*, *Netscape*, *Firefox*, dll), sehingga pengguna situs web dapat mengidentifikasi tingkat keamanan situs *web* tersebut yang menggunakan protokol keamanan SSL ini.

Browser web secara otomatis akan mengecek apakah sertifikat SSL dan identitas situs *web valid* dan situs tersebut terdaftar pada otoritas sertifikasi (CA) SSL (cth. Verisign). Dengan demikian, SSL ini menjadi sangat penting terutama untuk situs web yang menjalankan transaksi *online*.

Koneksi SSL akan memproteksi informasi vital dengan mengenkripsi informasi yang dikirim dan diterima antara *pc* pengguna situs dan *web server*, sehingga informasi yang berjalan tidak mungkin dapat diambil ditengah jalan dan dibaca isinya. Hal ini berarti pengguna tidak perlu ragu untuk mengirim informasi

vital seperti nomor kartu kredit kepada situs web yang telah memasang SSL tersertifikat ini.

2.12 *Model–View–Controller* (MVC)

Arsitektur *Model-View-Controller* adalah sebuah pola yang terbukti membangun proyek secara lebih efektif. Hal itu dilakukan dengan memilah komponen antara *Model*, *View* dan *Controller* pada bagian–bagian dalam proyek.

Aplikasi apapun, bagian dalam kode yang sering mengalami perubahan adalah bagian *user interface*. *User interface* adalah bagian yang paling terlihat oleh *user* dan bagaimana aplikasi tersebut berinteraksi dengan aplikasi, membuatnya menjadi titik fokus pengubahan berdasar kemudahan penggunaan. *Business logic* yang rumit pada *user interface* membuat pengubahan pada *user* interface menjadi lebih kompleks dan mudah terjadi kesalahan. Perubahan pada satu bagian memiliki potensi keterkaitan dengan keseluruhan aplikasi.

Pola MVC menyediakan sebuah solosi terhadap permasalahan tersebut dengan membagi aplikasi menjadi bagian–bagian tersendiri, *Model*, *View* dan *Controller*, memisahkan antar bagian tersebut dan membuat tata interaksi diantaranya.

2.12.1 Model

Pola MVC memiliki *layer* yang disebut dengan *model* yang merepresentasikan data yang digunakan oleh aplikasi sebagaimana proses bisnis yang diasosiasikan terhadapnya. Dengan memilahnya sebagai bagian terpisah, seperti penampungan data, *persistence*, serta proses manipulasi, terpisah dari bagian lain aplikasi.

Terdapat beberapa kelebihan dalam pendekatan ini. Pertama, membuat detail dari data dan operasinya dapat ditempatkan pada area yang ditentukan (*model*) dibanding tersebar dalam keseluruhan lingkup aplikasi. Hal ini memberikan keuntungan dalam proses *maintenance* aplikasi.

Kedua, dengan pemisahan total antara data dengan implementasi *interface*, komponen model dapat digunakan kembali oleh aplikasi lain yang memiliki kegunaan yang hampir sama.

2.12.2 *View*

Layer ini mengandung keseluruhan detail dari implementasi *user interface*. Disini, komponen grafis menyediakan representasi proses *internal* aplikasi dan menuntun alur interaksi *user* terhadap aplikasi. Tidak ada *layer* lain yang berinteraksi dengan *user*, hanya *View*.

Penggunaan *layer View* memiliki beberapa kelebihan yaitu memudahkan pengabungan divisi desain dalam *development team.* Divisi desain dapat berkonsentrasi pada *style*, *look* & *feel*, dan sebagainya, dalam aplikasi tanpa harus memperhatikan lebih pada detail yang lain.

Dan juga, memiliki *layer View* yang terpisah memungkinkan ketersediaan *multiple interface* dalam aplikasi. Jika inti dari aplikasi terletak pada bagian lain (dalam *Model*), *multiple interfaces* dapat dibuat (*Swing*, *Web*, *Console*), secara keseluruhan memiliki tampilan yang berbeda namun mengeksekusi komponen *Model* sesuai fungsionalitas yang diharapkan.

2.12.3 *Controller*

Arsitektur MVC memiliki *layer Controller*. *Layer* ini menyediakan *detail* alur program dan transisi *layer*, dan juga bertanggungjawab akan penampungan *events* yang dibuat oleh *user* dari *View* dan melakukan *update* terhadap komponen *Model* menggunakan data yang dimasukkan oleh *user*.

Kelebihan dalam penggunaan *layer Controller* secara terpisah yaitu pertama, dengan menggunakan komponen terpisah untuk menampung detail dari transisi *layer*, komponen *view* dapat didesain tanpa harus memperhatikan bagian lain secara berlebih. Hal ini memudahkan *team* pengembang *multiple interface* bekerja secara terpisah dari yang lain secara simultan. Interaksi antar komponen *View* terabstraksi dalam *Controller*.

Kedua, dengan menggunakan *layer* terpisah yang melakukan *update* terhadap komponen *Model*, *detail* tersebut dihapus dari *layer* presentasi. *Layer* presentasi kembali pada fungsi utamanya untuk menampilkan data kepada *user*. *Detail* tentang bagaimana data dari *user* mengubah ketetapan aplikasi disembunyikan oleh *Controller*. Hal ini memisahkan dengan jelas antara *presentation logic* dengan *business logic*.

Tidak dapat disimpulkan bahwa pola MVC hadir dengan kelebihan–kelebihan tanpa ada efek samping. Pembagian aplikasi dalam 3 bagian terpisah meningkatkan kompleksivitas. Pada aplikasi kecil yang tidak membutuhkan *loose coupling* pada *Model*, hal ini dapat menjadi blok penghalang dalam penggunaan pola ini. Bagaimanapun, yang terbaik adalah untuk meyakini bahwa sebuah aplikasi umumnya dimulai dari aplikasi sederhana, dan berkembang menjadi

sistem yang kompleks, sehingga *factor loose coupling* harus selalu diutamakan dan diperhatikan.

2.13 *Tools* Pendukung

Tools pendukung yang digunakan terdiri dari metode pemodelan, bahasa pemrograman, *software* pembangun aplikasi dan *sever* basis data yang dipakai.

2.13.1 *Unified Modeling Language* (UML)

Unified Modeling Language (UML) ialah bahasa pemodelan umum yang digunakan untuk melakukan spesifikasi, visualisasi, konstruksi dan dokumentasi artifak dari *software* sistem.

UML terdiri dari diagram, notasi, konsep dan aturan yang dipakai dalam memodelkan sistem. Diagram UML terdiri dari 9 jenis diagram yang memiliki fungsi dan notasi masing-masing. Diagram UML dibagi dalam 2 kategori yaitu :

1. Diagram yang menggambarkan unsur statis dari sistem
   a. *Class Diagram* dan *Object Diagram*

      Menggambarkan struktur hubungan statis dari elemen-elemen sebuah model, diantaranya *class*, *package*, dan *relationship*. Pada *Object diagram* menggambarkan hubungan antar elemen dalam model, tapi dengan memakai objek, bukan *class*. *Class* ialah kumpulan dari objek yang memiliki *attribute*, *behaviour* atau *operation* yang sama.

   b. *Component Diagram*

      Merupakan gambaran aspek fisik sistem berbasis objek dengan menunjukkan hubungan dan ketergantungan dalam serangkaian komponen fisik *software* termasuk *source code*, *run time* (*binary*) *code*, *file*, *table*,

library, dan dokumen. Terdapat pula *interface*, *dependency*, *generalization*, *association*, *realization*, *notes*, *constraints*, *packages*, *subsystem* dari sebuah model. Diagram ini digunakan untuk memodelkan implementasi sistem yang statis.

c. *Deployment Diagram*

Menggambarkan sumber fisik dalam sistem, termasuk *node*, komponen dan koneksi (model implementasi sistem yang statistik). Dalam hal ini meliputi topologi *hardware* yang dipakai sistem.

2. Diagram untuk menggambarkan unsur dinamis dari sistem

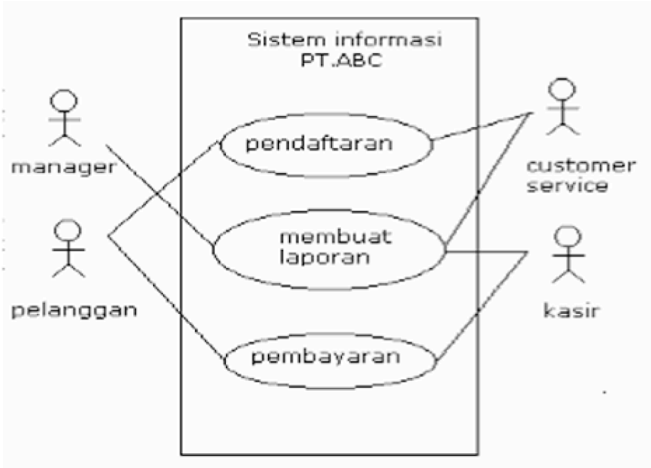
a. *Use Case Diagram*

Use case ialah salah satu model dalam analisis dan desain sistem berorientasi objek. *Use case modeling* digunakan untuk mendokumentasikan sistem *behaviour* dan subsistem pada saat pengembangan sistem, termasuk di dalamnya fungsi internal suatu sistem (*use case*), pengguna sistem (*user*) dan hubungan interaksi antara keduanya (*use case diagram*).

Bagian utama dalam *use case modelling*, yaitu :

1. *Actor* adalah perwujudan dari pengguna sistem, proses dan segala sesuatu yang berinteraksi dalam sistem tersebut. *Actor* tidak termasuk dalam sistem, tapi dapat menggambarkan interaksi dari *external actor* dengan sistem tersebut. Setiap *actor* berinteraksi dengan satu atau lebih *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 dimana suatu sistem bekerja, termasuk membatasi sistem dengan *actor* yang berada di luar dari sistem. Gambar mengenai contoh *use case diagram* dapat dilihat pada gambar 2.3.

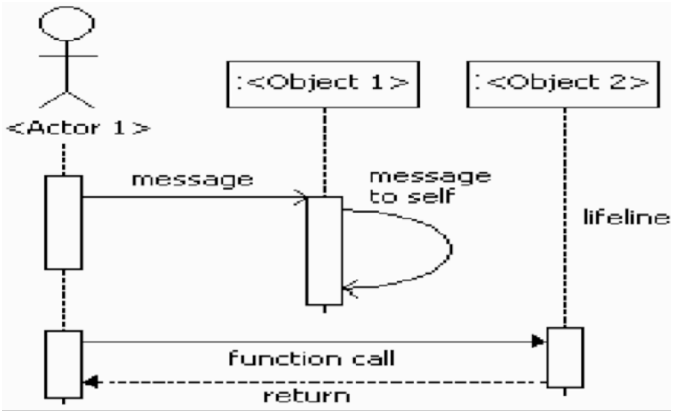


Gambar 2.3 *Use Case Diagram*

b. *Sequence Diagram*

Merupakan diagram yang menggambarkan pola hubungan diantara sekumpulan objek yang saling mempengaruhi menurut urutan waktu. Sebuah objek berinteraksi dengan objek lain melalui pengiriman *messages*.

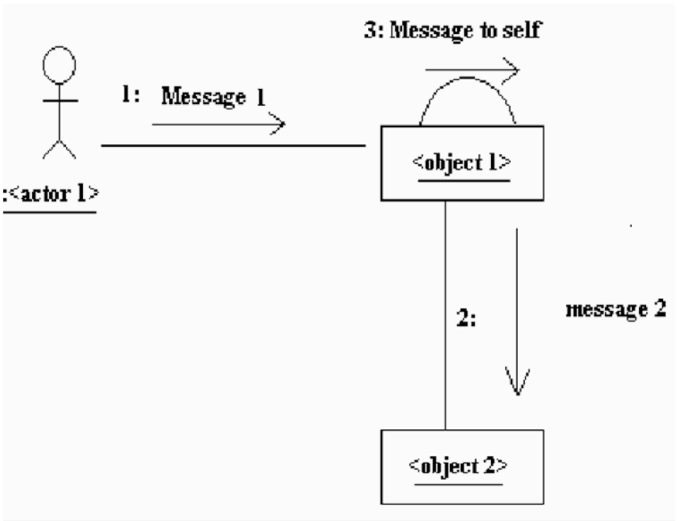
Sequence diagram biasanya digunakan untuk mengilustrasikan sebuah *use case*. Notasi lebih jelas untuk *sequence diagram* dapat dilihat pada gambar 2.4



Gambar 2.4 *Squence Diagram*

c. *Collaboration Diagram*

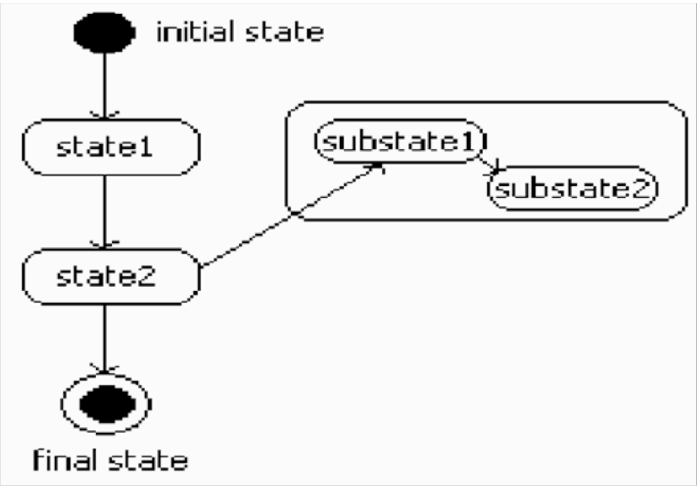
Collaboration diagram menggunakan interaksi antar objek dengan serangkaian *message* dan juga menggambarkan struktur yang statis dan *behaviour* yang dinamis dari sistem. Notasi dalam *collaboration diagram* dapat dilihat pada gambar 2.5.



Gambar 2.5 *Collaboration Diagram*

d. *Statechart Diagram*

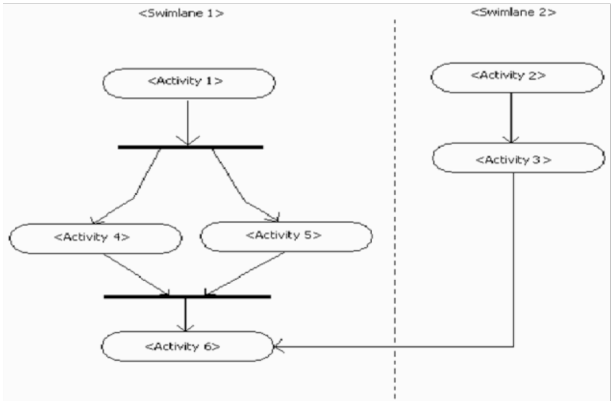
State Diagram merupakan diagram yang menggambarkan serangkaian *state* yang dapat terjadi pada sebuah objek. *State* dapat berubah karena adanya suatu *event* atau kondisi tertentu. *State* merupakan kondisi dimana objek melakukan sebuah kegiatan atau sedang menunggu *event*. *Statechart* diagram digunakan untuk menggambarkan *behaviour* yang penting dari subsistem atau sebuah objek. Notasi dari *statecart diagram* dapat dilihat pada gambar 2.6.



Gambar 2.6 *Statecart Diagram*

e. *Activity Diagram*

Activity diagram ialah diagram yang menggambarkan *flow of events* dari sebuah *use case* yang menjelaskan apa yang harus dilakukan oleh sistem agar dapat memberi nilai guna bagi aktor. Notasi dari *activity diagram* dapat dilihat pada gambar 2.7



Gambar 2.7 ***Activity Diagram***

2.13.2 *Framework Codeigniter*

CodeIgniter adalah suatu *Application Development Framework* untuk pembangunan suatu website berbasis PHP. Tujuannya adalah agar pembangunan projek bisa lebih cepat dibandingkan dengan membangunnya dari awal, dengan menyediakan *library* yang kaya yang umum digunakan untuk pembangunan suatu website, dengan *interface* dan struktur logika yang sederhana untuk mengaksesnya. *CodeIgniter* akan membuat *developer* semakin kreatif.

Tujuan dari pembuatan *framework CodeIgniter* adalah untuk menghasilkan *framework* yang akan dapat digunakan untuk pengembangan proyek pembuatan *website* secara lebih cepat dibandingkan dengan pembuatan *website* dengan cara koding secara manual, dengan menyediakan banyak sekali pustaka yang dibutuhkan dalam pembuatan *website*, dengan antarmuka yang sederhana dan struktur logika untuk mengakses pustaka yang dibutuhkan. *CodeIgniter* membiarkan *user* untuk memfokuskan diri pada pembuatan website dengan meminimalkan pembuatan kode untuk berbagai tujuan pembuatan *website*.

2.13.3 *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).*

2.13.4 PHP

PHP (*Hypertext prepocessor*) merupakan bahasa *scripting* yang menyatu dengan HTML dan berada pada *server* (*server-sideembeded scripting*). Program dari *server* akan dijalankan dari jarak jauh (*remote*) melalui *web browser* meskipun *platform* (Sistem Operasi, tipe mesin, struktur jaringan, dsb) *browser* dan *server* berbeda.

PHP berbeda dengan HTML karena dengan PHP, *programmer* tidak langsung menuliskan yang ingin dimuculkan di *browser*, tetapi *programmer* menuliskan kode yang memerintahkan server untuk memproses dan menghasilkan

halaman *web* yang diinginkan. *Server* memberikan memberikan hasilnya berupa halaman *web*.

PHP Pertama kali ditemukan pada 1995 oleh seorang *Software Developer* bernama Rasmus Lerdrof. Ide awal PHP adalah ketika itu Radmus ingin mengetahui jumlah pengunjung yang membaca *resume online* yang telah dibuat. *script* yang dikembangkan baru dapat melakukan dua pekerjaan, yakni merekam informasi *visitor*, dan menampilkan jumlah pengunjung dari suatu *website*. Sampai sekarang kedua tugas tersebut masih tetap populer digunakan oleh dunia *web* saat ini. Kemudian, dari situ banyak orang di milis mendiskusikan *script* buatan Rasmus Lerdrof, hingga akhirnya rasmus mulai membuat sebuah *tool*/*script*, bernama *Personal Home Page* (PHP).

Kebutuhan PHP sebagai *tool* yang serba guna membuat Lerdorf melanjutkan untuk mengembangkan PHP hingga menjadi suatu bahasa tersendiri yang mungkin dapat mengkonversikan data yang di dimasukan melalui *form* HTML menjadi suatu variabel, yang dapat dimanfaatkan oleh sistem lainnya. Untuk merealisasikannya, akhirnya Lerdrof mencoba mengembangkan PHP menggunakan bahasa C ketimbang menggunakan *Perl*. Tahun 1997, PHP versi 2.0 di rilis, dengan nama *Personal Home Page Form Interpreter* (PHP-FI). PHP semakin populer, dan semakin diminati oleh *programmer web* dunia.

Rasmus Lerdrof benar-benar menjadikan PHP sangat populer, dan banyak sekali *team developer* yang ikut bergabung dengan Lerdrof untuk mengembangkan PHP hingga menjadi seperti sekarang, hingga akhirnya dirilis

versi ke 3-nya, pada Juni 1998, dan tercatat lebih dari 50.000 *programmer* menggunakan PHP dalam membuat *website* dinamis.

Pengembangan demi pengembangan terus berlanjut, ratusan fungsi ditambahkan sebagai fitur dari bahasa PHP, dan di awal tahun 1999, netcraft mencatat, ditemukan 1.000.000 situs di dunia telah menggunakan PHP. Ini membuktikan bahwa PHP merupakan bahasa yang paling populer digunakan oleh dunia *web development*. Hal ini mengagetkan para *developer* termasuk Rasmus sendiri, dan tentunya sangat diluar dugaan sang pembuatnya. Kemudian Zeev Suraski dan Andi Gutsman selaku *core developer* (*programmer* inti) mencoba untuk menulis ulang PHP *parser*, dan diintegrasikan dengan menggunakan Zend *scripting engine*, dan mengubah jalan alur operasi PHP. Dan semua fitur baru tersebut di rilis dalam PHP 4 **loca[2]**.

2.13.5 *Apache Webserver*

Web server merupakan *server internet* yang mampu melayani koneksi transfer data dalam protokol HTTP. *Web server* merupakan hal yang terpenting dari server di internet dibandingkan server lainnya seperti *e-mail server*, *ftp server* ataupun *news server*. Hal ini disebabkan *web server* telah dirancang untuk dapat melayani beragam jenis data, dari *text* sampai grafis 3 dimensi. Kemampuan ini telah menyebabkan berbagai institusi seperti universitas maupun perusahaan dapat menerima kehadirannya dan juga sekaligus menggunakannya sebagai sarana di internet.

Salah satu *software* yang biasa digunakan oleh banyak *web master* di dunia adalah *apache*. *Software* tersebut dapat diunduh secara gratis dari web

resmi apache, yaitu http://www.apache.org. Dalam penggunaannya, *Apache* merupakan *software open source* yang sekarang ini sudah merebut pasar dunia lebih dari 50%. *Web server* ini *fleksibel* terhadap berbagai sistem operasi seperti windows9x/NT ataupun unix/linux. *Apache* merupakan turunan dari *webserver* yang dikeluarkan oleh NCSA yaitu NCSA HTTPd pada sekitar tahun 1995.

2.13.6 *MySql*

MySQL adalah salah satu jenis *database server* yang sangat terkenal. Mysql populer disebabkan bersifat *free* (tidak perlu membayar untuk menggunakannya) pada berbagai *platform* (kecuali pada windows, yang bersifat software atau perlu membayar setelah melakukan evaluasi dan memutuskan digunakan untuk keperluan produksi). Perangkat lunak *MySQL* sendiri bisa di download dari http: // www.mysql.org atau http://www.mysql.com.

MySQL termasuk jenis RDMS (*Relational Database Management System*). Itulah sebabnya istilah seperti tabel, baris, dan kolom digunakan pada MySQL. Pada MySQL, sebuah *database* mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom. Menurut Luke Welling dan Laura Thomson, *MySQL* adalah sebuah *relational database management systems* yang sangat cepat dan kuat.

MySQL adalah sebuah perangkat lunak sistem *management* basis data SQL atau DBMS yang *multithread*, *multiuser*, dengan sekitar 6 juta instalasi diseluruh dunia. *MySQL* dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia *MySQL* AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan

MySQL AB yaitu David Axmark, Allan Larsson, dan Michael "Monty" Widenius. *MySQL* tersedia sebagai perangkat lunak gratis dibawah lisensi *General Public License* (GPL), tetapi *MySQL* juga dijual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. Untuk melakukan administrasi dalam basis data MySQL, dapat menggunakan modul yang sudah termasuk yaitu *command-line* (perintah *mysql* dan *mysqladmin*) juga dapat diunduh dari situs *MySQL* yaitu sebuah modul berbasis grafik (GUI). *MySQL Administrator* dan *MySQL Query Browser*. Selain itu terdapat juga sebuah perangkat lunak gratis untuk administrasi basis data *MySQL* berbasis *web* yang sangat populer yaitu *phpMyAdmin*.

2.13.7 *Netbeans*

NetBeans mengacu pada dua hal, yakni *platform* untuk pengembangan aplikasi desktop java, dan sebuah *Integrated Development Environment* (IDE) yang dibangun menggunakan *platform NetBeans*.

Platform NetBeans memungkinkan aplikasi dibangun dari sekumpulan komponen perangkat lunak moduler yang disebut 'modul'. Sebuah modul adalah suatu arsip Java (*Java archive*) yang memuat kelas-kelas Java untuk berinteraksi dengan *NetBeans Open* API dan *file* manifestasi yang mengidentifikasinya sebagai modul. Aplikasi yang dibangun dengan modul-modul dapat dikembangkan dengan menambahkan modul-modul baru. Karena modul dapat dikembangkan secara independen, aplikasi berbasis platform NetBeans dapat dengan mudah dikembangkan oleh pihak ketiga secara mudah dan *powerful*.

2.13.8 *Microsoft Visio*

Microsoft Visio adalah sebuah program aplikasi komputer yang sering digunakan untuk membuat diagram, diagram alir (*flowchart*), *brainstorm*, dan skema jaringan yang dirilis oleh *Microsoft Corporation*. Aplikasi ini menggunakan grafik vektor untuk membuat diagram-diagramnya.

Visio sebenarnya bukanlah buatan *Microsoft Corporation*, melainkan buatan *Visio Corporation,* yang diakusisisi oleh *Microsoft* pada tahun 2000. Versi yang telah menggunakan nama *Microsoft Visio* adalah *Visio* 2002, *Visio* 2003, dan *Visio* 2007 yang merupakan versi terbaru. *Visio* 2007 *Standard* dan *Professional* menawarkan antarmuka pengguna yang sama, tapi seri *Professional* menawarkan lebih banyak pilihan template untuk pembuatan diagram yang lebih lanjut dan juga penataan letak (*layout*). Selain itu, edisi *Professional* juga memudahkan pengguna untuk mengoneksikan diagram-diagram terhadap beberapa sumber data dan juga menampilkan informasi secara *visual* dengan menggunakan grafik.