BAB 2
TINJAUAN PUSTAKA

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 Gareu Shoes.

2.1.1 Sejarah Perusahaan

Toko Gareu bergerak dibidang perdagangan dan industri sepatu, berdiri pada tahun 2006. Toko Gareu sendiri merupakan bagian dari Mabarroh Cahaya Megah yang didirikan oleh Bapak H.Moch. Andi Sutadiwangsa. Beberapa *brand* sudah didirikan oleh perusahaan ini diantaranya Garsel dan Garucci yang sudah lama beredar di pasaran. Nama *brand* Gareu sendiri berawal dari kunjungan Bapak Fahmi Idris yang pada saat itu menjabat sebagai Menteri perindustrian beliau menyarankan untuk membuat *brand* baru dikarenakan Bapak Andy Sutadiwangsa berasal dari Garut maka diberikan nama *brand* tersebut Gareu yang artinya "Garut Euy". Atas dasar rekomendasi tersebut maka Bapak Moch. Andy Sutadiwangsa menggunakan nama tersebut untuk *brand* barunya yang diresmikan pada tanggal 22 Januari 2006 dan dibuka secara langsung oleh menteri perindustrian yaitu Bapak Fahmi Idris, dikarenakan nama *brand* tersebut unik dan pemberian dari seorang menteri maka produk ini dibuat secara *ekslusive* dengan kualitas terbaik yang diharapkan dapat menembus pasar untuk kalangan menengah ke atas.

Berbagai macam *inovasi* dan terobosan-terobosan telah banyak dilakukan oleh Toko Gareu Shoes untuk dapat meningkatkan kualitas produk terbaik. Mulai dari *inovasi* dari segi desain, bahan dasar, *grafis*, bahkan pola pemasarannya. Pada tahun 2009 untuk penerbitan katalog yang ketiga diresmikan secara langsung oleh Bapak Yusuf Kala yang pada saat itu menjabat sebagai wakil presiden. Sehingga hal tersebut menjadi modal selanjutnya untuk dapat meningkatkan produk yang berkualitas.

Distribusi produk sepatu Gareu sendiri sudah menyebar ke berbagai daerah di Indonesia karena adanya agen dan *reseller* yang tersebar.

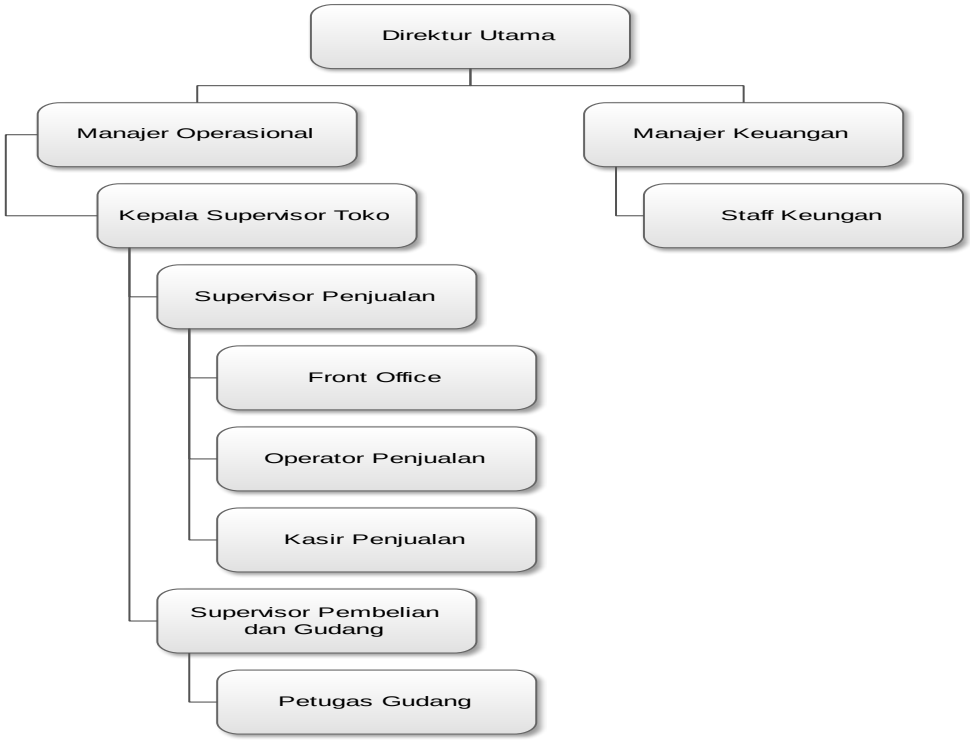
2.1.2 Visi dan Misi Perusahaan

Visi dan misi dari sepatu Gareu adalah menghasilkan produk yang *ekslusive* dengan kualitas terbaik serta membangun kemitraan, kekeluargaan dan menuju sukses bersama.

2.1.3 Struktur Organisasi

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

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 Gareu Shoes dapat dilihat pada bagan struktur organisasi pada gambar 2.1 berikut ini :



Gambar 2.1 Struktur Organisasi pada Toko Gareu Shoes

2.1.4 Deskripsi Tugas

Definisi tugas digunakan untuk mengetahui tugas, wewenang, tanggung jawab dari masing-masing bagian. Definisi tugas yang ada pada Toko Gareu Shoes adalah sebagai berikut :

1. Direktur Utama

   Direktur Utama yang juga berperan sebagai Direktur Perusahaan Toko Gareu Shoes.

   a. Memimpin keseluruhan dewan manajemen tingkat atas.
   b. Memimpin rapat umum dalam hal menentukan agenda, menjelaskan dan menyimpulkan tindakan dan kebijakan.
   c. Mengangkat karyawan atau tenaga kerja demi kepentingan perusahaan.

2. Manajer Operasional

   Tugas dan tanggung jawab Manajer Operasional adalah:

   a. Melakukan perencanaan, pengelolaan, pengawasan, dan pengendalian dalam bidang operasional secara sistematis.

b. Menyusun kebijakan dan strategi pencapaian target perusahaan.
c. Bertanggung jawab kepada Direktur Utama.

3. Kepala *Supervisor* Toko

   Tugas dan tanggung jawab *Supervisor* Toko adalah:

   a. Menjaga hubungan baik dengan para pelanggan baik dari dalam maupun dari luar kota.
   b. Bersama manajer operasional melakukan negosiasi harga pokok dengan *supplier* dan menentukan harga jual dari setiap produk yang ditawarkan pada setiap periode katalog.
   c. Menjalin hubungan baik dengan para *supplier*.
   d. Menganalisa dan memberikan arahan pengembangan desain dan warna kepada *supplier*, untuk memastikan pengembangan produk sesuai dengan kebutuhan pasar.
   e. Mengawasi setiap kinerja *Supervisor* Toko.
   f. Bertanggung jawab terhadap seluruh aset perusahan yang berada di toko.
   g. Bertanggung jawab kepada Manajer Operasional.

4. *Supervisor* Penjualan

   Tugas dan tanggung jawab *Supervisor* Penjualan adalah:

   a. Bertanggung jawab terhadap keamanan penerimaan uang di kasir.
   b. Bertanggung jawab atas data penjualan serta kesesuaian transaksi yang terjadi di kasir.
   c. Melakukan penyetoran uang hasil penjualan ke kantor pusat.
   d. Membuat laporan hasil penjualan.
   e. Bertanggung jawab atas terlaksananya proses pengiriman pesanan pelanggan baik luar dan dalam pulau jawa sesuai dengan instruksi dari pelanggan.
   f. Membuat jadwal keberangkatan pengiriman pesanan.
   g. Bertanggung jawab atas keamanan paket barang pesanan.
   h. Bertanggung jawab kepada Kepala *Supervisor* Toko.

5. Kasir Penjualan

   Tugas dan tanggung jawab Kasir Penjualan adalah:

   a. Menerima uang atau pembayaran dengan bukti yang sah.
   b. Menjaga keamanan uang kas kasir.
   c. Membuat laporan uang masuk maupun uang keluar.
   d. Bertanggung jawab kepada *Supervisor* Penjualan.

6. Operator Penjualan

   Tugas dan tanggung jawab Operator Penjualan adalah:

   a. Menerima konfirmasi pembayaran dari pelanggan.
   b. Menerima panggilan telepon yang masuk, menjawab permasalahan pelanggan.
   c. Mencatat daftar pesanan yang diterima dari pelanggan.
   d. Menjaga hubungan baik dengan pelanggan.
   e. Bertanggung jawab kepada *Supervisor* Penjualan.

7. *Front Office*

   Tugas dan tanggung jawab Petugas Penjualan adalah:

   a. Melayani pelanggan atau pengunjung Toko dengan baik dan simpatik.
   b. Bertanggung jawab kepada *Supervisor* Penjualan.

8. *Supervisor* Pembelian dan Gudang

   Tugas dan tanggung jawab *Supervisor* Pembelian dan Gudang adalah:

   a. Menjalin hubungan baik dengan *supplier*.
   b. Membuat *purchase order* sesuai rencana pembelian dan kondisi stok barang.
   c. Membuat laporan *purchase order* dan menyerahkannya kepada Manajer Operasional.
   d. Bertanggung jawab kepada *Supervisor* Kepala Toko.

9. Petugas Gudang

   Tugas dan tanggung jawab Petugas Gudang adalah:

   a. Bertanggung jawab terhadap penyimpanan barang dan tata letak barang di gudang.

b. Memeriksa barang yang diterima dari *supplier*.
c. Menjaga keamanan barang di gudang.
d. Mencatat kartu gudang.
e. Bertanggung jawab kepada *Supervisor* Pembelian dan Gudang.

10. Manajer Keuangan

Tugas dan tanggung jawab Manajer Keuangan adalah:

a. Melakukan perencanaan, pengelolaan, pengawasan, dan pengendalian dalam bidang keuangan dan akuntansi secara sistematis.
b. Memastikan ketersediaan dana operasional yang dibutuhkan oleh perusahaan untuk kegiatan operasional sehari-hari.
c. Memastikan konsolidasi keuangan yang akurat dan tepat waktu untuk keperluan pelaporan keuangan kepada Direktur Utama.
d. Bertanggung jawab kepada Direktur Utama.

11. *Staff* Keuangan

Tugas dan tanggung jawab *Staff* Keuangan adalah:

a. Membuat, memeriksa dan mengarsip faktur, nota *supplier*, laporan hutang/piutang untuk memastikan status hutang/piutang.
b. Memasukan penerimaan pembayaran dari pelanggan, dan pembayaran ke *supplier* dengan tepat waktu dan akurat untuk memastikan ketepatan waktu dan keakuratan penerimaan maupun pembayaran.
c. Memeriksa laporan *rekonsiliasi* untuk memastikan data keuangan dengan benar.
d. Menyetujui konfirmasi pembayaran dan membuat konfirmasi pengiriman.
e. Memeriksa rangkuman kas kecil untuk memastikan penggunaan dan ketersediaan kas kecil yang efektif.
f. Bertanggung jawab kepada Manajer Keuangan.

2.2 Landasan Teori

2.2.1 Konsep Dasar Sistem

Terdapat dua kelompok pendekatan di dalam pendefinisian sistem, yaitu yang menekankan pada prosedur dan menekankan pada komponen atau elemen. Pendekatan sistem yang lebih menekankan pada prosedur mendefinisikan sistem sebagai berikut:

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

Pendekatan sistem yang lebih menekankan pada komponen atau elemen-elemen mendefinisikan sistem sebagai berikut:

Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu.[1]

2.2.2 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*).[1]

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*)

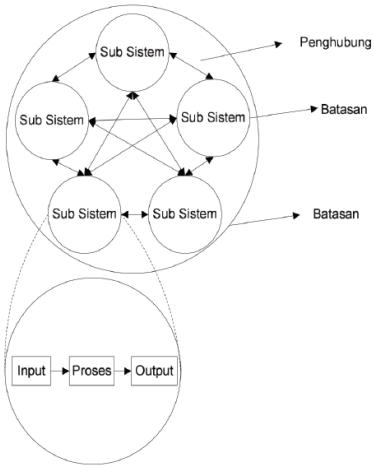
   Keluaran (*output*) merupakan hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna.

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 Suatu Sistem.[1]

2.2.3 *Internet*

Menurut Adi Nugroho "*Internet* dengan berbagai aplikasinya seperti *Web*, VoIP, E-Mail pada dasarnya merupakan media yang digunakan untuk mengefesiensikan proses komunikasi".

Sedangkan menurut tim penelitian dan pengembangan wahana komputer (2005), *internet adalah metode untuk menghubungkan berbagai komputer ke dalam satu jaringan global, melalui protokol yang disebut Transmission Control Protocol/Internet Protocol (TCP/IP)".*

Berdasarkan kedua pendapat diatas, maka dapat disimpulkan bahwa *internet* adalah suatu jaringan komunikasi antara komputer yang besar, yang

mencakup seluruh dunia dan berbasis pada sebuah *protocol* yang disebut TCP/IP (*Transmission Control Protocol/Internet Protocol*). Selain itu, internet dapat disebut sebagai sumber daya informasi yang dapat digunakan oleh seluruh dunia dalam mencari informasi.[8]

2.2.4 Sejarah *Internet*

Pada awalnya *internet* merupakan jaringan komputer yang dibentuk oleh Departemen Pertahanan Amerika Serikat di tahun 1969, melalui proyek ARPA yang disebut ARPANET *(Advanced Research Project Agency Net)*, dimana mendemonstrasikan bagaimana dengan *hardware* dan *software* komputer yang bebasis UNIX, kita bisa melakukan komunikasi dalam jarak yang tidak terhingga melalui saluran telepon. Proyek ARPANET merancang bentuk jaringan, kehandalan, seberapa besar informasi dapat dipindahkan, dan akhirnya semua standar yang ditentukan menjadi cikal bakal untuk pembangunan protokol baru yang sekarang dikenal sebagai TCP/IP *(Transfer Control Protocol / Internet Protocol).*

Tujuan awal dibangunnya proyek itu adalah untuk keperluan militer. Pada saat itu Departemen Pertahanan Amerika Serikat (*US Department Of Defense's*) membuat sistem jaringan komputer yang tersebar dengan menghubungkan komputer di daerah-daerah vital untuk mengatasi masalah bila terjadi serangan nuklir dan untuk menghindari terjadinya informasi terpusat, yang apabila terjadi perang dapat mudah dihancurkan.

Pada awalnya ARPANET hanya menghubungkan empat situs saja yaitu *Stanford Research Institute*, *University of California at Santa Barbara*, *University of Utah* dan *University of California at Los Angeles* dimana membentuk satu jaringan terpadu di tahun 1969, dan secara umum ARPANET diiperkenalkan pada bulan oktober 1972, tidak lama kemudian proyek ini berkembang pesat di seluruh daerah, dan semua unversitas di negara tersebut ingin bergabung, sehingga membuat ARPANET kesulitan untuk mengaturnya.

Oleh sebab itu ARPANET dipecah manjadi dua, yaitu "MILNET" untuk keperluan militer dan "ARPANET" baru yang lebih kecil untuk keperluan non-

militer seperti, universitas-universitas. Gabungan kedua jaringan akhirnya dikenal dengan nama DARPA *internet*, yang kemudian disederhanakan menjadi *internet*.[8]

2.2.5 *World Wide Web* (WWW)

World Wide Web merupakan jaringan dokumentasi yang sangat besar yang saling berhubungan satu dan lainnya. Satu set protokol yang mendefinisikan bagaimana sistem bekerja dan mengirimkan data, dan sebuah *software* yang membuatnya bekerja dengan mulus. *Web* menggunakan tehnik *hypertext* dan multimedia yang membuat *internet* mudah digunakan dijelajahi dan dikonstribusikan. *Web* merupakan sistem *hypermedia* yang berarea luas yang ditujukan untuk akses secara *universal*. Salah satu kuncinya adalah kemudahan tempat seseorang atau perusahaan dapat menjadi bagian dari *Web* berkonstribusi pada *Web*.[8]

Web merupakan sistem yang menyebabkan pertukaran data di *internet* menjadi mudah dan efisien. *Web* terdiri atas 2 komponen dasar;

1. *Server Web* adalah sebuah komputer dan *software* yang menyimpan dan mendistribusikan data ke komputer lainnya melalui *internet*
2. *Browser Web* adalah *software* yang dijalankan pada komputer pemakai atau *client* yang meminta informasi dari server *Web* yang menampilkannya sesuai dengan *file* data itu sendiri.

Menurut Hardjono *Web* Merupakan fasilitas *hiperteks* untuk menampilkan data berupa teks, gambar, suara, animasi, dan data multimedia lainnya.

Ada dua kategori dalam perograman *Web*, yaitu pemrograman *Server Web* dan *Client Web*. Pada pemrograman *Server Side*, perintah-perintah program (*script*) dijalankan di *server Web*, kemudian hasil dikirimkan ke *browser* dalam bentuk HTML biasa.

Adapun pada *Client Side*, perintah program dijalankan pada *browser Web* sehingga ketika *client* meminta dokumen *script*, maka *script* dapat di-*download* dari *server* kemudian dijalankan pada *browser* yang bersangkutan.[8]

2.2.6 *E-Commerce*

Electronic Commerce (E-Commerce) merupakan konsep baru yang biasa digambarkan sebagai proses jual beli barang atau jasa pada *internet* atau proses jual beli atau pertukaran produk, jasa dan informasi melalui jaringan informasi termasuk *internet*. *E-Commerce* merupakan kegiatan bisnis yang dijalankan secara elektronik melalui suatu jaringan internet atau kegiatan jual beli barang atau jasa melalui jalur komunikasi digital.[8]

2.2.4.1 Keuntungan *E-Commerce*

E-commerce menawarkan kepada perusahaan keuntungan jangka pendek dan jangka panjang. *E-commerce* tidak hanya membuka pasar baru bagi produk atau jasa yang ditawarkan, mencapai konsumen baru, tetapi juga dapat mempermudah cara perusahaan melakukan bisnis. Perbedaan antara proses perdagangan secara manual dengan menggunakan *e-commerce* dapat jelas terlihat, dimana pada proses dengan *e-commerce* terjadi efisiensi pada penggunaan fax, pencetakan dokumen, *entry* ulang dokumen, serta jasa kurir. Efisiensi tersebut akan menunjukkan pengurangan biaya dan waktu/kecepatan proses.[8]

Keuntungan *e-commerce* terbagi dalam tiga bagian diantaranya adalah :

1. Keuntungan bagi perusahaan
    a. Memperpendek jarak, perusahaan-perusahaan dapat mendekatkan diri dengan konsumen. Dengan hanya mengklik link-link yang ada pada situs-situs, konsumen dapat menuju perusahaan dimana pun saat itu berada.
    b. Perluasan pasar, jangkauan pemasaran menjadi semakin dan tidak terbatas oleh area geografis dimana perusahaan berada.
    c. Efisien, sangat memangkas biaya-biaya operasional. Perusahaan-perusahaan yang berdagang dengan cara *e-commerce* tidak membutuhkan kantor tidak membutuhkan kantor dan toko yang besar, menghemat kertas-kertas yang digunakan untuk transaksi.

2. Keuntungan bagi konsumen
    a. Efektif, konsumen dapat memperoleh informasi tentang produk yang dibutuhkannya dan bertransaksi dengan cepat dan murah.
    b. Aman secara fisik, konsumen tidak perlu mendatangi perusahaan tersebut.
    c. Fleksibel, konsumen dapat melakukan transaksi dari berbagai lokasi, baik dari rumah, kantor, warnet atau tempat-tempat lainnya.
3. Keuntungan bagi masyarakat umum
    a. Mengurangi polusi dan pencemaran lingkungan, konsumen tidak perlu melakukan perjalanan-perjalanan ke toko-toko, dimana hal ini dapat mengurangi jumlah kendaran yang berlalu-lalang dijalanan. Berkurangnya kendaraan di jalanan berarti menghemat bahan bakar, dan mengurangi tingkat polusi udara.
    d. Membuka peluang kerja baru, bagi yang tidak buta akan teknologi, muncul pekerjaan-pekerjaan baru seperti pemogram komputer, perancang *Web*, ahli dibidang basis data, analisis sistem dan sebagainya.
    e. Menguntungkan dunia akademis, berubahnya pola hidup masyarakat dengan hadirnya *e-commerce*, seperti yang berkaitan dengan pola hidup dunia maya. Selain itu dampak langsung hadirnya *internet* secara langsung akan menantang kiprah ilmuan di bidang komputer.
    f. Meningkatkan kualitas sumber daya manusia, masyarakat tidak akan menjadi gatek (gagap teknologi) sehingga pada gilirannya akan merangsang untuk mempelajari teknologi.

2.2.4.2 Mekanisme *E-Commerce*

Transaksi elektronik (pihak yang menawarkan barang atau jasa melalui *internet*) dengan e-*commerce*, (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, memiliki banyak tipe dan variasi yaitu: [8]

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 telepon, 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 *E-Mail* 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.

2.2.4.3 Karakteristik *E-Commerce*

Berbeda dengan transaksi perdagangan biasa, transaksi e-*commerce* memiliki beberapa karakteristik yang sangat khusus, yaitu: [8]

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 sistempembayaran 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*.

2.2.4.4 Klasifikasi *E-Commerce*

Penggolongan *e-commerce* yang lazim dilakukan ialah berdasarkan sifat transaksinya antara lain: [8]

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 barang.

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 barang melainkan hanya beberapa barang 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 dan menyepakati suatu transaksi.

2.2.7 *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. [10]

2.2.8 SSL (*Server Side Includes*)

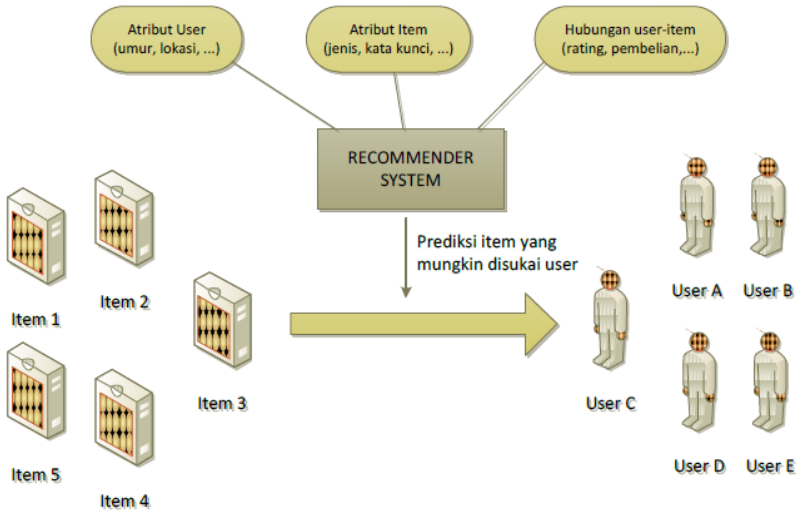
Secure Sockets Layer atau yang disingkat SSL adalah sebuah protokol keamanan data yang digunakan untuk menjaga pengiriman data antara *Web server* dan pengguna situs *Web* tersebut. SSL umumnya sudah terpasang didalam mayoritas *browser Web* yang ada (IE, *Netscape*, *Firefox*, dll), sehingga pengguna situs *Web* dapat mengidentifikasi tingkat keamanan situs *Web* tersebut yang menggunakan protokol keamanan SSL ini. [1]

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.

2.2.9 *Smart Recommendation System*

Smart Recommender system merupakan sebuah metoda untuk menampilkan informasi mengenai suatu hal (film, musik, buku, berita, gambar, dan sebagainya) yang sesuai dengan minat *user*. *Recommender system* akan membandingkan profil *user* dengan referensi yang dimilikinya lalu menampilkan informasi kepada *user* berdasarkan prediksi yang dilakukan sebelumnya.[9]



Gambar 2.3 *Recommender System*.[9]

Ada dua tipe pengumpulan data untuk membangun *recommender system* sebenarnya.[9]

1. Secara eksplisit
   a. Meminta *user* untuk merating sebuah *item*.
   b. Meminta *user* untuk merelarating sekumpulan *item.*
   c. Meminta *user* untuk memilih salah satu *item* dari beberapa *item* yang diberikan.
   d. Meminta *user* untuk membuat daftar *item* yang dia suka.
2. Secara implisit
   a. Mengobservasi *item* yang sedang dilihat oleh *user* secara *online*.
   b. Menganalisis jumlah *user* yang melihat suatu *item*.
   c. Menyimpan catatan pembelian *user*.

Ada beberapa cara untuk menyajikan rekomendasi, yaitu:

1. Rekomendasi *Non-Personalized*

   Rekomendasi *non-personalized* akan merekomendasikan *item* yang memiliki tingkat popularitas yang tinggi berdasarkan *rating user* lain atau data transaksi. Contoh *output* dari teknik *non-personalized* misalnya "*20-most popular software*".
2. Rekomendasi *Demographic*

   Rekomendasi *demographic* memanfaatkan fitur/atribut *user*. Teknik ini mencari *user-user* yang memiliki fitur yang mirip dan merekomendasikan *item* yang disukai satu *user* kepada *user* lain yang fiturnya mirip.
3. Rekomendasi *Content-Based*

   Kebalikannya dengan rekomendasi *demographic*, rekomendasi *content-based* memanfaatkan fitur dari *item*. Teknik ini akan mencari kemiripin dari setiap *item*. Jika *user* X memilih *item* A dan *item* A mirip dengan *item* N, maka sistem akan merekomendasikan *item* N kepada *user* X.
4. Rekomendasi *User-Based Collaborative*

   Rekomendasi *user-based collaborative* memanfaatkan *rating user* atau data transaksi. Sistem mencari *user-user* yang memiliki korelasi yang tinggi kemudian merekomendasikan *item-item* yang disukai oleh *user-user* itu. Misalnya *user* X menyukai *item* A, *item* B, dan *item* C sementara *user*

Y menyukai *item* B, *item* C dan *item* D. Maka sistem akan merekomendasikan *item* D pada *user* X dan *item* A pada *user* Y.

5. Rekomendasi *Item-Based Collaborative*

   Mirip seperti rekomendasi *user-based collaborative*, rekomendasi *item-based collaborative* memanfaatkan *rating user* atau data transaksi. Yang membedakan adalah korelasi yang dicari. Rekomendasi *item-based collaborative* mencari korelasi diantara *item-item* yang dipilih *user* kemudian merekomendasikan *item-item* yang berkolerasi itu pada *user* yang lain.

3.2.9.1 Konsep Dasar *Collaborative Filtering*

Collaborative filtering merupakan proses penyaringan atau pengevaluasian item menggunakan opini pelanggan lain.[9] *Collaborative filtering* melakukan penyaringan data berdasarkan kemiripan karakteristik pelanggan sehingga mampu memberikan informasi yang baru kepada pelanggan karena sistem memberikan informasi berdasarkan pola satu kelompok pelanggan yang hampir sama. Perbedaan minat pada beberapa anggota kelompok menjadikan sumber informasi baru yang mungkin bermanfaat bagi anggota kelompok lainnya.

Secara umum proses pemberian rekomendasi terdiri atas tiga langkah, yaitu: penemuan *similar user*, pembuatan ketetanggaan (*neighborhood*), dan penghitungan prediksi berdasarkan tetangga yang dipilih. *Collaborative filtering* menghasilkan prediksi atau rekomendasi bagi pengguna atau pelanggan yang dituju terhadap satu item atau lebih. *Item* dapat terdiri atas apa saja yang dapat disediakan manusia seperti misalnya buku, film, seni, artikel, atau tujuan wisata. *Rating* dalam *collaborative filtering* dapat berbentuk dari model-model sebagai berikut:

a. *Model rating skalar* yang terdiri atas *rating numerik* seperti 1 (satu) sampai 5 (Lima).
b. *Model rating biner* dengan memilih antara setuju atau tidak setuju, atau dapat pula baik atau buruk.
c. *Mating unary* dapat mengindikasikan bahwa pengguna telah mengobservasi atau membeli *item* atau me-*rating item* dengan positif.

Tidak tersedianya *rating* mengindikasikan tidak terdapat informasi yang menghubungkan pengguna dengan item. *Rating* dapat dikumpulkan secara *eksplisit*, *implisit*, ataupun gabungan antara *eksplisit* dan *implisit*. *Rating eksplisit* yaitu *rating* yang didapatkan pada saat pelanggan/pengguna diminta menyediakan opini terhadap item tertentu. *Rating implisit* yaitu *rating* yang didapatkan melalui aksi yang dilakukan pelanggan.

3.2.9.2 *Algoritma Collaborative Filtering*

Schafer membagi *algoritma collaborative filtering* ke dalam dua kelas yang berbeda menurut teori dan kepraktisannya, yaitu *algoritma non-probabilistik* dan *algoritma probabilistik*. Suatu algoritma dianggap *probabilistik* bila algoritma tersebut berdasarkan model probabilistik. *Algoritma* tersebut mewakili distribusi probabilitas saat menghitung prediksi *rating* atau daftar rangking rekomendasi. *Algoritma non-probabilistik* yang terkenal yaitu *nearestneighbours algorithm*. *Algoritma* ini dibagi menjadi dua kelas yaitu *user-based* dan *item-based.* [9]

3.2.9.3 *User-Based Collaborative Filtering*

User-based nearest neighbour algorithm menggunakan teknik statistika untuk menemukan sekumpulan pengguna, dikenal sebagai tetangga (*neighbour*), yang memiliki sejarah setuju dengan pengguna yang menjadi sasaran. Setelah sekumpulan tetangga terbentuk, sistem menggunakan algoritma yang berbeda untuk menggabungkan kesukaan *neighbours* untuk menghasilkan prediksi atau rekomendasi N-teratas untuk *active user*. [9]

3.2.9.4 *Item-To-Item Collaborative Filtering*

Item-based collaborative filtering merupakan metode rekomendasi yang didasari atas adanya kesamaan antara pemberian *rating* terhadap suatu produk dengan produk yang dibeli. Dari tingkat kesamaan produk, kemudian dibagi dengan parameter kebutuhan pelanggan untuk memperoleh nilai kegunaan produk. Produk yang memiliki nilai kegunaan tertinggilah yang kemudian dijadikan rekomendasi.[9] Metode ini muncul sebagai solusi untuk beberapa permasalahan pada *user-based collaborative filtering* yaitu pada masalah keterbatasan (*sparsity*) dan skalabilitas serta masalah waktu dan memori.

Pada metode ini akan diketahui nilai *similaritas* antar *item* dengan tingkat persebaran *rating* kecil dan nilai *similaritas* antar item cenderung lebih jarang berubah dibandingkan dengan nilai *similaritas* antar pengguna. *Item-based collaborative filtering* melakukan *similaritas* dengan membentuk suatu model *similaritas* secara *offline* yang secara otomatis akan menghemat waktu dan memori yang digunakan untuk penghitungan pada saat pengguna mengakses halaman situs.

3.2.9.5 *Item-Based* Collaborative Filtering

Item-based collaborative filtering memanfaatkan *rating user* atau data transaksi untuk membuat rekomendasi. Teknik ini akan mencari korelasi diantara *item-item* yang dipilih *user* kemudian merekomendasikan *item-item* yang berkolerasi itu pada *user* yang lain.

Pada awalnya, *item-based collaborative filtering* akan menghitung nilai kemiripan antara *item* yang satu dengan *item* yang lainnya berdasarkan *rating* yang diberikan oleh *user*. Nilai kemiripan antara dua *item* itu didapat dengan menghitung *rating* kedua *item* tersebut menggunakan rumus *Pearson Correlation* atau *Adjusted-Cosine.* [9]

Persamaan *Pearson Correlation* :

$$S_{(i,j)} = \frac{\sum_{u \in U}(R_{u,i}-\bar{R}_i)(R_{u,j}-\bar{R}_j)}{\sqrt{\sum_{u \in U}(R_{u,i}-\bar{R}_i)^2}\sqrt{\sum_{u \in U}(R_{u,j}-\bar{R}_j)^2}} \quad \text{..... (2.1)}$$

[9]

Persamaan *Adjusted Cosine* :

$$S_{(i,j)} = \frac{\sum_{u \in U}(R_{u,i}-\bar{R}_u)(R_{u,j}-\bar{R}_u)}{\sqrt{\sum_{u \in U}(R_{u,i}-\bar{R}_u)^2}\sqrt{\sum_{u \in U}(R_{u,j}-\bar{R}_u)^2}} \quad \text{..... (2.2)}$$

[9]

Keterangan :

$S_{(i,j)}$ = Nilai kemiripan antara *item* i dengan *item* j.

$u \in U$ = Himpunan *user* yang me-*rating* baik *item* i maupun *item* j.

$R_{u,i}$ = *Ratinguser* u pada *item* i.

$\bar{R}_i$ = Nilai *rating* rata-rata *item* i.

$R_{u,j}$ = Rating *user* u pada *item* j.

$\bar{R}_j$ = Nilai *rating* rata-rata *item* j.

$\bar{R}_u$ = Nilai *rating* rata-rata *user* u.

Nilai yang dihasilkan oleh rumus adalah antara -1.0 hingga +1.0. Jika nilai koefisien semakin mendekati -1 atau +1, maka hubungan antara kedua variabel itu akan semakin kuat. Jika nilai koefisiennya adalah 0, maka kedua variabel itu tidak ada hubungannya (independen).

Pada kasus *collaborative filtering* nilai koefisien lebih populer disebut *similarity* (kemiripan). Jika nilai *similarity* antara kedua *item* mendekati +1, maka kedua *item* akan semakin mirip satu sama lain. Sebaliknya, jika mendekati -1, kedua *item* itu akan semakin bertolak belakang.

Tahap berikutnya adalah menghitung prediksi. Tahapan ini dilakukan untuk memperkirakan *rating* yang akan diberikan oleh seorang *user* pada suatu *item* yang belum pernah di-*rate* oleh *user* itu. Penghitungan prediksi menggunakan rumus *weighted sum*.

Persamaan *weighted sum*:

$$P_{(a,j)} = \frac{\sum_{i \in I} (R_{a,i} \cdot S_{i,j})}{\sum_{i \in I} |S_{i,j}|} \quad \text{.....} \quad (2.3)$$

[9]

Keterangan :

$P_{(a,j)}$ = Prediksi *ratingitem* j oleh *user* a.

$i \epsilon I$ = Himpunan *item* yang mirip dengan *item* j.

$R_{u,i}$ = *Rating user* a pada *item* i.

$S_{i,j}$ = Nilai *similarity* antara *item* i dan *item* j.

Tahap terakhir adalah pembuatan rekomendasi. Pada tahap ini, teknik *collaborative filtering* berperan untuk menyediakan nilai-nilai yang akan dijadikan bahan pokok pembuatan rekomendasi. Sistem secara keseluruhan memiliki peran yang lebih besar. Sistem dapat membuat dua jenis rekomendasi, yaitu:

1. Rekomendasi yang bersifat umum

   Pembuatan rekomendasi umum memanfaatkan nilai kemiripan yang sudah dihitung sebelumnya. Setiap *user*, siapapun itu, akan mendapatkan rekomendasi yang sama.

2. Rekomendasi yang bersifat khusus

   Rekomendasi khusus bersifat personal. Artinya, setiap *user* akan mendapatkan rekomendasi yang berbeda. Pembuatan rekomendasi memanfaatkan nilai prediksi yang telah dihitung dengan menggunakan rumus *weigthed sum*.

3.2.9.6 Karakteristik Teknik *Item Based Collaborative Filtering*

Karakteristik dari teknik *item based collaborative filtering* dari sebenarnya yaitu: [9]

1. *Scalable*

   Di dunia nyata, teknik *item-based collaborative filtering* digunakan untuk membuat rekomendasi bagi jutaan pengguna dengan jutaan *item-item* yang tersedia. Salah satu situs yang memanfaatkan teknik ini adalah amazon.com.

2. *Cold Start*

   Pada tahap awal, akan sulit untuk membuat rekomendasi dengan kualitas yang baik karena sumber data yang digunakan tidak banyak.

3. *Sparsity Problem*

   Sparsity problem (masalah kekosongan data) adalah kondisi ketika lebih banyak sel yang kosong dibandingkan dengan sel yang terisi dalam suatu tabel. Teknik *item-based collaborative filtering* tidak akan berurusan dengan masalah seperti ini karena teknik ini hanya berurusan dengan sel yang terisi.

4. *Content Analysis*

   Analisis konten tidak diperlukan karena *user* akan me-*rate* suatu *item* berdasarkan pengalaman mereka.

2.2.10 Konsep Dasar *Object Oriented Programming (OOP)*

2.2.10.1 Sejarah *Object Oriented Programming (OOP)*

Pemrograman berorientasi objek atau *object oriented programming* (OOP) menurut Adi Nugroho adalah "Suatu metode pemrograman yang berbasiskan pada objek". Secara singkat pengertian dari OOP adalah koleksi dari berbagai objek yang saling berinteraksi dan memberi informasi satu dengan yang lainnya.[5]

2.2.10.2 Konsep *Object Oriented Programming (OOP)*

Pendekatan berorientasi objek adalah "*Suatu cara baru dalam berpikir serta berlogika dalam menghadapi masalah-masalah yang akan dicoba-atasi dengan bantuan komputer, mencoba melihat permasalahan lewat pengamatan dunia nyata dimana setiap objek adalah entitas tunggal yang memiliki kombinasi struktur data dan fungsi tertentu*".[5]

Suatu cara pengembangan perangkat lunak dan sistem informasi berdasarkan objek-objek yang ada di dunia nyata.

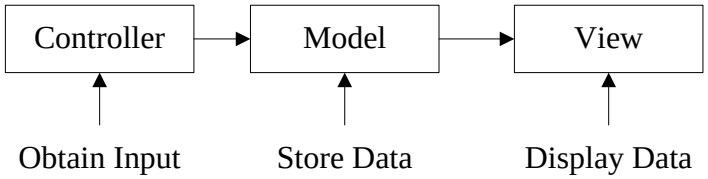
Teknologi berorientasi objek menganalogikan sistem aplikasi seperti kehidupan nyata yang didominasi objek. Berikut prinsip dasar dari pemograman berorientasi objek yang terdiri dari:

1. Abstraksi: memfokuskan pada karakteristik objek.
2. Kelas: bagian yang dapat menciptakan objek.
3. *Enkapsulasi* (Pembungkusan): Menyembunyikan banyak hal yang terdapat dalam objek yang tidak perlu diketahui objek lain
4. *Generlaisasi* (Spesialisasi)*:* pembagian kelas-kelas yang berperilaku sama.
5. *Polimorfisme*: fungsi yang sama dapat diterapkan dan dimiliki oleh kelas-kelas berlainan.

2.2.11 *Model-View-Controller* (MVC)

Pendekatan *Model-View-controller* adalah sebuah jalan untuk mengembangkan komponen dengan cara memisahkan antara penyimpanan data dan penanganan dari representasi visual data. Komponen untuk penyimpanan dan

penanganan data, dikenal sebagai *model*, yang berisi konten yang sebenarnya dari suatu komponen. Kemudian komponen untuk menampilkan data, dikenal sebagai *view* dan komponen *controller* adalah sebuah komponen yang umumnya bertanggung jawab untuk memperoleh data. [1]



Gambar 2.4 MVC (*Model View Controller*).[5]

Gambar 2.3 menjelaskan bahwa *controller* memperoleh data dan menyimpan dalam *model*, kemudian *view* menampilkan data yang disimpan di dalam *Model*

Memisahkan komponen ke dalam *model* dan *view* mempunyai dua keuntungan besar:

1. Dapat membuat beberapa tampilan yang memungkinkan data dibagi melalui *model* yang sama.
2. Dapat menyederhanakan proses penulisan aplikasi yang komplek dan membuat komponen, serta mudah untuk melakukan pemeliharaan .
3. Perubahan terhadap *view* dapat dilakukan tanpa mempengaruhi Model dan begitu juga sebaliknya

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.[5]

2.2.11.1 *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.[5]

2.2.11.2 *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.[5]

2.2.11.3 *Controller*

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

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 *pengguna*.

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 tiga 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.[5]

2.2.12 *Tools* Pendukung

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

3.2.12.1 *Unifield Modeling Language* (UML)

Menjelaskan tentang pengertian UML, konsep Dasar UML, beserta diagram-digram dalam penggambaran model objek.

A. Pengertian UML

Berikut ini definisi *Unified Modeling Language* (UML) menurut para ahli:

1. Menurut Sholiq "*Unified Modeling Language (UML) adalah bahasa yang telah menjadi standard untuk visualisasi, menetapkan, membangun dan mendokumentasikan artifak suatu sistem perangkat lunak*".
2. Menurut Adi Nugroho "*Unified Modeling Language (UML) adalah alat bantu analisis serta perancangan perangkat lunak berbasis objek*".

Berdasarkan beberapa pendapat yang dikemukakan diatas dapat ditarik kesimpulan bahwa "*Unified Modeling Language* (UML) adalah sebuah bahasa yang berdasarkan *grafik* atau gambar untuk menvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis OO (*Object Oriented*)". [4]

B. Konsep Dasar UML

Abstraksi konsep dasar UML yang terdiri dari *structural classification*, *dynamic behavior*, dan *model management*, bisa dipahami dengan mudah apabila melihat gambar diatas dari *diagrams*. *Main concepts* bisa dipandang sebagai *term* yang akan muncul pada saat membuat diagram. Dan *view* adalah kategori dari diagaram tersebut. Untuk menguasai UML, sebenarnya cukup dua hal yang harus diperhatikan. [12]

1. Menguasai pembuatan diagram UML.
2. Menguasai langkah-langkah dalam analisa dan pengembangan dengan UML.

C. Bagian-Bagian UML

Bagian-bagain utama dari UML adalah *view, diagram, model element,* dan *general mechanism.*[2]

1. *View*

   View digunakan untuk melihat sistem yang dimodelkan dari beberapa yang berbeda. *View* bukan mlihat grafik, tapi merupakan suatu abstraksi yang berisi sejumlah diagram. Beberapa jenis *view* dalam UML antara lain: *Use Case View, Logical View, Component View, Concurrency View dan Deployment View.*

   a. *Use Case View*

      Use Case View mendeskripsikan fungsionalitas sistem yang seharusnya dilakukan sesuai yang diinginkan *external actor*. *Actor* yang berinteraksi dengan sistem dapat berupa pengguna atau sistem lainnya.

   b. *Logical View*

      Logical View mendeskripsikan bagaimana fungsionalitas sistem, struktur statis (*class,object* dan *relationship*) dan kolaborasi dinamis yang terjadi.

   c. *Component View*

      Component View mendeskripsikan implementasi dan ketergantungan modul komponen yang merupakan tipe lainnya dari *code module* diperlihatkan dengan struktur dan ketergantungannya juga alokasi sumber daya komponen dan informasi *administrative* lainnya.

   d. *Councurrency View*

      Councurrency View membagi sistem kedalam proses dan prosesor, *view* ini digambarkan dalam diagram dinamis dan diagram implementasi serta digunakan untuk pengembang (*developer*), pengintegrasi (*integrator*) dan penguji (*tester*).

e. *Deployment View*

Deployment View mendeskripsikan fisik dari sistem seperti komputer dan perangkat (*nodes*) dan bagaimana hubungan dengan lainnya.

2. *Diagram*

Diagram merupakan bagian dari suatu *View* tertntu dan ketika digambarkan biasanya dialokasikan untuk *View* tertentu ada beberapa jenis diagram anatara lain. [12]

a. *Use Case Diagram*

Use Case Diagram adalah deskripsi fungsi yang disediakan oleh sistem dalam bentuk teks sebagai dokumentasi dari *Use Case Symbol* namun dapat juga dilakukan dalam *Activity Diagram*.

b. *Class Diagram*

Class Diagram menggambarkan struktur statis *class* didalam sistem *class* merepresentasikan sesuatu yang ditangani oleh sistem *class* dapat berhubungan dengan yang lain melalui berbagai cara *associated* (terhubung satu sama lain), *specialed* (satu *class* merupakan spesialisasi dari *class* lainnya), atau *package* (grup bersama sebagai satu unit).

c. *Statechart Diagram*

Statechart Diagram menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari *stimuli* yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*).

d. *Activity Diagram*

Activity Diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

e. *Sequence Diagram*

Sequence Diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

f. *Collaboration Diagram*

Collaboration Diagram juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*. Setiap *message* memiliki *sequence number*, di mana *message* dari level tertinggi memiliki nomor satu. *Messages* dari level yang sama memiliki *prefiks* yang sama.

g. *Component Diagram*

Component Diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) diantaranya. Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time*. Umumnya komponen terbentuk dari beberapa *class* atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain

h. *Deployment Diagram*

Deployment Diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, *server* atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi *server*, dan hal-hal lain yang bersifat fisikal Sebuah *node* adalah *server*, *workstation*, atau piranti keras lain yang

digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya. Hubungan antar *node* (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini.

D. *Indikator Multiciplicity*

Walaupun *multiciplicity* ditentukan untuk *class, multiciplicity* menentukan banyaknya objek yang terhubung satu dengan yang lainnya. Indikator *multiciplicity* terdapat pada masing-masing akhir garis relasi, baik pada asosiasi dan *agregasi*, beberapa contoh *multiciplicity* adalah: [12]

Tabel 2.1 Notasi *Multiciplicity* dalam *UML*

Simbol	Keterangan
*	Banyak
0	Nol
1	Satu
0..*	Antara nol sampai banyak
1..*	Antara satu sampai banyak
0..1	Noa atau satu
1..1	Tepat satu

E. *Businees Modeling*

Dalam dunia bisnis dan industri, terdapat banyak sistem manual dan otomatis yang muncul secara regular setiap sistem tersebut meiliki satu atau banyak *workflow*. *Business Modeling* menggambarkan semua *workflow* yang terjadi dalam suatu organisasi. Secara formal, didefinisikan *Business Modeling* sebagai segala teknik pemodelan yang digunakan untuk menggambarkan model sebuah bisnis. *Business Modeling* dapat digunakan untuk meninjau, meningkatkan, dan membuat sebuah bisnis. Adapun tujuan *Business Modeling* adalah:[4]

1. Memahami struktur dan dinamika organisasi.
2. Memahami masalah-masalah dalam mencapai target organisasi dan menemukan potensi untuk kemajuan organisasi.
3. Yakin bahwa para *costumer, end user* dan *developer* mempunyai sebuah pemahaman yang benar mengenai organisasi.

4. Dapat menurunkan/mendapatkan *requirement software* aplikasi yang dibuat yang diperlukan dalam mencapai tujuan suatu organisasi.

Untuk mencapai tujuan tersebut, harus membuat dua jenis model bisnis, yaitu model *use case* bisnis (*Business Use Case Model*) dan model objek bisnis (*Business Object Model*) yang berisi proses-proses, peranan-peranan, dan tanggung jawab organisasi untuk mencapai targetnya.

F. Proses Bisnis

Dalam suatu organisasi mungkun terdapat banyak proses bisnis. Proses Bisnis (*Business Process)* adalah sekumpulan aktivitas yang dirancang untuk menghasilkan keluaran (*output*) tertentu bagi *costumer.* Sebuah proses bisnis menekankan pada bagaimana sebuah pekerjaan dikerjakan dalam sebuah organisasi, dengan berfokus pada produk yang dihasilkan proses tersebut. Proses disini adalah urutan aktivitas tertentu terhadap waktu dan tempat, dengan sebuah titik awal, sebuah titik akhir dan sebagai *input* dan *output* yang didefinisikan dengan baik. [4]

G. *Stereotype* Dalam Model Bisnis

Beberapa jenis elemen yang digunakan *Business Modeling* dari yaitu: [4]

1. *Business Actor*

   Business Actor (Aktor Bisnis) menggambarkan peran yang dimainkan oleh seseorang atau sesuatu yang dengannya bisnis berinteraksi sebuah *Business Actor* menggambarkan seseorang seorang *costumer* atau *partner* bisnis, tetapi sebuah sistem informasi yang berhubungan dengan bisnis dapat berperan sebagai sebuah *Business Actor* juga.

2. *Business Use Case*

   Business Use Case merupakan urutan tindakan yang dimainkan suatu bisnis yang menghasilkan sebuah mengasilkan sebuah nilai yang dapat dilihat dan dtujukan untuk suatu *Business Actor* tertentu. Setiap *business use case* mewakili suatu proses bisnis.

3. *Business Use Case Realization*

   Business Use Case Realization pandang dari dalam (*internal View*) terhadap *business use case*. Yakni menentukan bagaimana pekerjaan

(dalam *business use case*) tersebut diatur dan dimainkan untuk mencapai hasil yang diinginkan sebuah *business use case realization* menentukan *business worker* dan *business entity* yang terlihat ketika menjalankan sebuah *business use case*.

4. *Business Worker*

   Business Worker sebuah kelas yang mewakili, seorang manusia yang bertindak dalam bisnis. Sebuah *business worker* berinteraksi dengan para *worker* lainnya dan memainkan entitas bisnis saat berpatisipasi dalam realisasi suatu *business use case*.

5. *Business Entity* (Entitas Bisnis)

   Business Entity (Entitas Bisnis) menggambarkan benda yang ditangani atau digunakan oleh *business worker* selama melaksanakan suatu *Business Use Case*.

3.2.12.2 *JavaScript*

Javascript adalah bahasa pemrograman yang sederhana karena bahasa ini tidak dapat digunakan untuk membuat aplikasi ataupun *applet*. Dengan *javascript*, kita dapat dengan mudah membuat sebuah halaman *Web* yang interaktif. [11]

Sedangkan menurut Stendy B. Sakur "*Javascript* adalah pendekatan lain untuk membuat hal *Web* menjadi lebih interaktif, baik dalam deteksi maupun tanggapan ke interaksi pengguna dengan halaman *Web"*. *Javascript* dapat langsung digabungkan dengan HTML tanpa harus di-*compile* terlebih dahulu. [11]

3.2.12.3 *HyperText Markup Language* (HTML)

Menurut Stendy menyatakan *"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). [11]

3.2.12.4 *PHP (Personal Home Page)*

Menurut Stendy B. Sakur "PHP adalah bahasa *scripting* untuk sisi *server* yang dirancang secara khusus untuk *Web"*. Dalam halaman HTML dapat dimasukkan kode-kode PHP yang akan dijalankan setiap kali halaman tersebut dieksekusi. Kode-kode PHP akan diinterpretasikan pada *server Web* dan menghasilkan HTML atau output lainnya yang akan dilihat oleh pengunjung *Web.*

PHP disusun tahun 1994 dan merupakan hasil kerja keras Rasmus Lerdorf. Kemudian dilanjutkan oleh pengembang lain dan telah melewati tiga kali penyusunan ulang secara besar untuk memberikan hasil produk yang matang seperti yang ada sekarang ini. Pada Januari 2001, PHP digunakan hampir lima juta daerah di seluruh dunia, dan jumlah ini terus bertambah.

PHP merupakan produk *Open Source*. PHP awalnya berarti *Personal Home Page,* tetapi diubah dengan penamaan konvensi rekursif GNU dan sekarang PHP ialah *PHP Hypertext Preprocessor*.

Beberapa dari pesaing PHP ialah *Perl, Microsoft Active Server Pages (ASP), Java Server Pages (JSP)* dan *Allaire Cold Fusion.* Dalam perbandingan dengan produk-produk tersebut, PHP memiliki beberapa kekuatan termasuk diantaranya ialah:

1. Performa yang tinggi

   PHP sangat efisiensi menggunakan sebuah server yang tidak mahal, dapat melayani berjuta-juta permintaan per hari.

2. *Integrasi Database*

   PHP memiliki koneksi yang mengijinkan kebanyak sistem database dengan menggunakan *MySQL,* dapat melakukan koneksi langsung ke *postfreSQL, mSQL, Oracle, dbm, filePro, Hyperwave, Informix,*

InterBase, dan *Sybase database*. Menggunakan *Open Database Connectivity Standard (ODBC)*, dapat melakukan koneksi ke banyak database yang disediakan oleh *driver ODBC*, termasuk produk *Microsoft.*

3. *Library* yang *built-in*

Karena PHP dirancang untuk digunakan pada *Web*, PHP memiliki banyak fungsi yang telah dibangun untuk mendukung banyak tugas yang berguna pada *Web*. Dengan PHP, dapat menampilkan gambar GIF, koneksi dengan layanan *network* yang lain, mengirim *E-Mail*, dan membuat file PDF, semuanya hanya dengan beberpa baris kode.

4. Biaya yang rendah

PHP gratis, PHP dapat diunduh kapanpun dari *http://www.php.net* tanpa biaya.

5. Mudah dipelajari dan digunakan

Sintaks dari PHP mengambil dasar bahasa pemograman lain, utamanya C dan *Perl*. Jika telah mengetahui C atau *Perl* atau sebuah bahasa seperti C, contohnya C++ atau *java*, maka PHP dapat hampir secara langsung dapat digunakan secara produktif.

6. Portabilitas

PHP dapat digunakan pada banyak sistem operasi. Kode PHP dapat ditulis pada operasi sistem UNIX yang gratis seperti *Linux* dan *Free*BSD, operasi sistem UNIX yang komersial seperti Solaris dan IRIX, atau berbagai versi dari *Microsoft Windows*.

7. Ketersediaan *Source code*

Source code PHP dapat diakses. Tidak seperti produk komersial, produk yang *source code*-nya tertutup, jika ada sesuatu yang hendak dimodifikasi atau ditambahkan pada PHP, dapat dilakukan dengan langsung dan gratis. [2]

3.2.12.5 My*SQL*

Menurut Betha Sidik "*database MySQL* merupakan sistem manajemen berbasis data SQL yang sangat terkenal dan bersifat *open source"*. *MySQL* dibangun, didistribusikan dan didukung oleh *MySQL AB*. *MySQL AB* merupakan perusahaan komersial yang dibiayai oleh pengembang (*developer*) *MySQL*. *MySQL* dapat didefinisikan sebagai: [2]

1. *MySQL* merupakan sistem manajemen *database*. *Database* merupakan struktur penyimpanan data. Untuk menambah, mengakses, dan memproses data yang disimpan dalam sebuah *database* komputer. Diperlukan sistem manajemen *database* seperti *MySQL Server*.
2. *MySQL* merupakan sistem manajemen *database* atau basis data terhubung *(relational database management system)*. *Database* terhubung menyimpan data pada tabel-tabel terpisah. Hal tersebut akan menambah kecepatan dan fleksibilitasnya. Kata SQL pada MySQL merupakansingkatan dari *Structured Query Language*. SQL merupakan bahasa standar yang digunakan untuk mengakses *database* dan ditetapkan oleh ANSI/ISO SQL *Standard*.
3. *MySql* merupakan *software open source*. *Open source* berarti semua diijinkan menggunakan dan memodifikasi model. Semua dapat mengunduh *software MySQL* dari *internet* dan menggunakannya tanpa membayar.
4. Dapat mempelajari *Source Code* dan menggunakannya sebagai kebutuhan.
5. *Server database MySQL* mempunyai kecepatan akses tinggi, mudah digunakan, dan andal. *MySQL* dikembangkan untuk menangani *database* yang besar secara cepat dan telah sukses digunakan selama bertahun-tahun. Konektivitas, kecepatan, dan keamanannya membuat *server MySQL* cocok untuk mengakses *database* di *internet*.

Fitur utama *MySQL* adalah:

1. Ditulis dalam bahasa C dan C++.
2. Bekerja dalam berbagai *platform* (*misalnya Mac Os X, Solaris, Sun OS, Unix, Novel Netware, Windows,* dan lain-lain).
3. Menyediakan mesin penyimpan *(engine storage)* transaksi dan *non*-transaksi *server* tersedia sebagai program yang terpisah untuk digunakan pada lingkungan jaringan *client/server*.
4. *MySQL* mempunyai *library* yang dapat ditempelkan pada aplikasi yang berjalan sendiri, sehingga aplikasi tersebut dapat digunakan pada komputer yang tidak mempunyai jaringan.
5. Mempunyai sistem *password* yang fleksibel dan aman.
6. Dapat menangani basis data dalam skala besar, basis data dalam *server MqSQL* dapat berisi 50 juta *record.*

MySQL termasuk jenis *RDBMS (Relational Database Management System)*. Pada *MySQL* sebuah *database* terdiri atas tabel-tabel. Sebuah tabel terdiri atas tabel dan kolom.

3.2.12.6 *NetBeans*

Menurut Stendy, *Netbeans* IDE adalah *arsitektur rich-client* dan menyediakan berbagai *tools* untuk mengembangkan aplikasi yang berbeda seperti *Web, enterprise, plug-in module,* dan aplikasi *mobile* dengan bahasa yang berbeda termasuk JAVA, C/C++ dan Ruby. *Netbeans* IDE kompatibel terhadap *Windows,* Linux, UNIX, dan Solaris.[11]

Fitur-fitur yang terdapat dalam *Netbeans* IDE :

1. *Additional Language*.
2. *Improve Code Eeditor*.
3. *The Netbeans Profiler*.
4. *Swing Desktop Database*.

3.2.12.7 *Rational Rose*

Menurut Adi Nugroho, *Rational Rose* adalah kakas (*tools*) pemodelan visual untuk pengembangan sistem berbasis objek yang sangat handal untuk digunakan sebagai bantuan bagi para pengembang dalam melakukan analisis dan perancangan sistem. *Rational Rose* digunakan untuk melakukan pemodelan sistem sebelum pengembang menulis kode-kode dalam bahasa pemrograman tertentu. *Rational Rose* mendukung pemodelan bisnis, yang membantu para pengembang untuk memahami sistem secara komprehensif.

Dalam *Rational Rose*, pemodelan adalah cara melihat sistem dari berbagai sudut pandang, yang mencakup semua diagram yang dikenal dalam UML, aktor-aktor yang terlibat dalam sistem, *usecase*, objek-objek, kelas-kelas, komponen-komponen, serta simpul-simpul penyebaran (*deployment node)*. [4]

2.2.13 Kuesioner

Menurut Sugiyono menyatakan bahwa "Kuesioner merupakan teknik pengumpulan data yang dilakukan dengan cara memberi seperangkat pertanyaan atau pernyataan tertulis kepada responden untuk dijawabnya". Kuesioner atau angket merupakan teknik pengumpulan data yang digunakan untuk mendapatkan informasi responden, dalam arti laporan tentang pendapat dari hal-hal yang diketahuinya.

Kuesioner disebarkan menggunakan teknik *quota sampling* dimana pemilihan sampel dengan memilih sejumlah tertentu (kuota) dan paling mudah diperoleh seperti yang dikehendaki oleh peneliti. Kuesioner disebarkan kepada 10 responden, dari hasil kuesioner tersebut akan dilakukan perhitungan agar dapat mengambil kesimpulan terhadap penilaian penerapan sistem yang baru. [15]

2.2.14 Skala Pengukuran

2.2.14.1 *Skala Likert*

Skala likert digunakan untuk mengukur sikap, pendapat, dan persepsi seseoran atau sekelompok orang tentang fenomena sosila. Dalam penelitian, fenomena sosial ini telah ditetapkan secara spesifik, yang selanjutnya disebut sebagai variabel penelitian.

Dengan *skala likert*, maka variabel yang akan diukur dijabarkan menjadi indikator variabel. Kemudian indikator tersebut dijadikan sebagai titik tolak untuk menyusun item-item instrumen yang dapat berupa pernyataan atau pertanyaan.

Jawaban setiap item instrumen yang menggunakan *skala likert* mempunyai gradiasi dari sangat positif sampai sangat negatif, yang ada berupa kata-kata antara lain:

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

Untuk keperluan analisisi kuantitiatif, maka jawaban itu dapat diberi skor, misalnya:

1. Setuju diberi skor 5
2. Setuju diber skor 4
3. Ragu-ragu diberi skor 3
4. Tidak setuju diberi skor 2
5. Sangat tidak setuju diberi skor 1

Instrumen penelitian yang menggunakan *skala likert* dapat dibuat dalam bentuk *checklist* ataupun pilihan ganda. Kemudian untuk mencari prosentasi dari masing-masing jawaban kuesioner digunakan rumus skala likert sebagai berikut:

$$Y = \frac{S}{skorideal} \times 100 \quad \text{.....(2.5)}$$

Keterangan:

Y = Nilai prosentasi yang dicari.

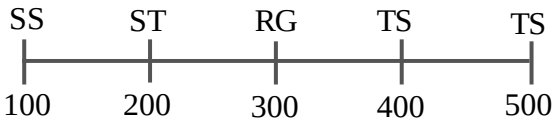
S = Total yang diperoleh S =$\sum_{i=1}^{5} n_i . r_i$

r_i = Banyak responden yang menjawab jawaban ke-i.

n_i = Nilai jawaban ke-i.

Skor ideal = Nilai tertinggi dikalikan dengan jumlah sampel atau responden.

Dibawah ini merupakan skala untuk mengmbil kesimpulan dari setiap petanyaan, datap dilihat pada Gambar 2.4. [15]



Gambar 2.4 *Skala Likert*. [15]