BAB II

TINJAUAN PUSTAKA

II. 1. Tinjauan Perusahaan

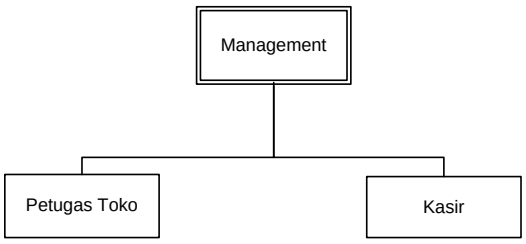
Tinjauan perusahaan terdiri dari profil singkat perusahaan, struktur organisasi dan fungsi perusahaan.

II. 1.1. Profil Singkat Perusahaan

Arsefa Collection yang beralamat di jalan K.H Wahid Hasyim 141 Bandung dirintis oleh Ibu Erna Hermina pada tahun 2005. Awalnya Arsefa Collection hanya menjual kebaya untuk kalangan tertentu saja yaitu rekanan–rekanan dari jasa rias pengantin, namun seiring meningkatnya pesanan untuk kebaya, maka Arsefa Collection pun mulai menjual kebayanya ke toko–toko terutama toko–toko di Pasar Baru Bandung yang sudah cukup terkenal di kalangan konsumen. Visi dari Arsefa Collection yaitu menjadi pusat penyedia kebaya terdepan bagi para pengusaha perias pengantin dan toko–toko penjual kebaya. Sedangkan misi Arsefa Collection diantaranya mendistribusikan koleksi kebaya–kebayanya ke pengusaha rias pengantin dan ke toko–toko penjual kebaya dengan tetap menjaga kualitas produk, menjaga hubungan baik dengan para konsumen serta memberikan pelayanan yang terbaik.

II.1.2. Struktur Organisasi dan Fungsi Perusahaan

Dalam menjalankan kegiatan usahanya, Arsefa Collection mempunyai struktur organisasi dengan jabatan dan tugas masing-masing pada setiap bagian, sehingga kegiatan operasional dapat berjalan dengan baik dan lancar.



Gambar 2.1 Struktur Organisasi Arsefa Collection

Secara umum tugas dan fungsi organisasi dari masing-masing bagian adalah sebagai berikut :

a. *Management*

1. Memiliki wewenang dalam menentukan kebijakan perusahaan.
2. Menetapkan rencana kerja organisasi dan tujuan yang akan dicapai secara keseluruhan.
3. Menetapkan target atau memberikan intruksi pada masing–masing bagian.

b. Kasir

1. Menangani transaksi penjualan.
2. Membuat laporan penjualan.
3. Melakukan pendataan barang masuk maupun keluar.

c. Petugas Toko

1. Melayani dan bertatap muka langsung dengan pelanggan.
2. Pemeliharaan barang terhadap barang yang ada di *counter*.
3. Membuat nota transaksi.
4. Melaporkan ke pemilik apabila stok barang pada *counter* kurang atau habis.

II.2. Landasan Teori

Landasan teori terdiri dari konsep dasar sistem, karakteristik sistem, klasifikasi sistem, konsep dasar informasi, sistem informasi, fungsi sistem informasi, konsep dasar *web*, *website*, *web browser*, *PHP*, *e-commerce*, *Object Oriented Program (OOP)*, *Model–View–Controller (MVC)*, Fasilitas dan *Software* Pendukung.

II.2.1. Konsep Dasar Sistem

Sistem dalam pengertian yang paling umum adalah sekumpulan benda yang memiliki hubungan diantara mereka. Kata *sistem* sendiri berasal dari bahasa Latin (systēma) dan bahasa Yunani (sustēma) adalah suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau sistem.

II.2.2. Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu :

1. Komponen-komponen

   Komponen sistem atau elemen sistem dapat berupa :

   a) Elemen-elemen yang lebih kecil yang disebut *sub sistem*, misalkan sistem komputer terdiri dari sub sistem perangkat keras, perangkat lunak dan manusia.

   b) Elemen-elemen yang lebih besar yang disebut *supra sistem*. Misalkan bila perangkat keras adalah sistem yang memiliki sub sistem CPU, perangkat I/O dan memori, maka supra sistem perangkat keras adalah sistem komputer.

2. Batas sistem

   Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batas suatu sistem menunjukkan ruang lingkup dari sistem tersebut.

3. Lingkungan luar sistem

   Lingkungan dari sistem adalah apapun di luar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Lingkungan luar yang mengutungkan merupakan energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedang lingkungan luar yang merugikan harus ditahan dan dikendalikan, kalau tidak akan mengganggu kelangsungan hidup dari sistem .

4. Penghubung

   Penghubung merupakan media perantara antar subsistem. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lainnya. *Output* dari satu subsistem akan menjadi input untuk subsistem yang lainnya dengan melalui penghubung. Dengan penghubung satu subsistem dapat berinteraksi dengan subsistem yang lainnya membentuk satu kesatuan.

5. Masukan

   Masukan adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa *maintenance input* dan *signal input*. *Maintenance input* adalah energi

yang dimasukkan supaya sistem tersebut dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran.

6. Keluaran

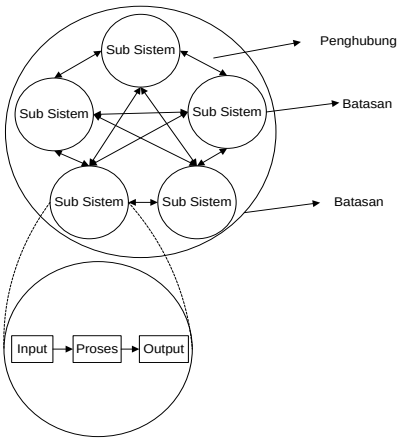
   Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lain atau kepada supra sistem.

7. Pengolah

   Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah yang akan merubah masukan menjadi keluaran. Suatu sistem produksi akan mengolah masukan berupa bahan baku dan bahan-bahan yang lain menjadi keluaran berupa barang jadi.

8. Sasaran atau tujuan

   Suatu sistem pasti mempunyai tujuan atau sasaran. Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.



Gambar 2.2. Karakteristik sistem

II.2.3. Klasifikasi Sistem

Menurut Jogiyanto. HM dalam bukunya **Analisis dan Desain,** Sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya adalah sebagai berikut :

a) Sistem abstrak *(abstract system)* adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Sedangkan sistem *fisik (physical system)* merupakan sistem yang ada secara fisik.
b) Sistem alamiah *(natural system)* adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia. Sedangkan sistem buatan *manusia (human made system)* melibatkan interaksi antara manusia dengan mesin.
c) Sistem tertentu *(deterministic system)* beroperasi dengan tingkah laku yang sudah dapat diprediksi. Sedangkan sistem tak tentu *(probabilistic system)* adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probalilitas.
d) Sistem tertutup *(closed system)* merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luar. Sedangkan sistem terbuka *(open*

system) adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luar.

II.2.4. Konsep Dasar Informasi

Secara umum informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata yang digunakan untuk pengambilan keputusan.

Menurut Raymond Mcleod, "Informasi adalah data yang telah diolah menjadi bentuk yang memiliki arti bagi si penerima dan bermanfaat bagi pengambilan keputusan saat ini atau mendatang"

Sumber dari informasi adalah data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian adalah sesuatu yang terjadi pada saat tertentu. Di dalam dunia bisnis, kejadian-kejadian yang sering terjadi adalah transaksi perubahan dari suatu nilai yang disebut transaksi. Kesatuan nyata adalah berupa suatu obyek nyata seperti tempat, benda dan orang yang betul-betul ada dan terjadi.

II.2.5. Pengertian Sistem Informasi

Ada beberapa pengertian sistem informasi menurut beberapa sumber :

a. John F. Nash (1995:8)

   Sistem Informasi adalah Sistem Informasi adalah kombinasi dari manusia, fasilitas atau alat teknologi, media, prosedur dan pengendalian yang bermaksud menata jaringan komunikasi yang penting, proses atas transaksi-transaksi tertentu dan rutin, membantu

manajemen dan pemakai intern dan ekstern dan menyediakan dasar pengambilan keputusan yang tepat.

b. Henry Lucas (1988:35)

Sistem Informasi adalah suatu kegiatan dari prosedurprosedur yang diorganisasikan, bilamana dieksekusi akan menyediakan informasi untuk mendukung pengambilan keputusan dan pengendalian di dalam organisasi

c. John F.Nash dan Martil B.Robert (1988:35)

Sistem Informasi adalah kombinasi dari orang-orang, fasilitas, teknologi, media, prosedur-prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi pentingm, memproses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen yang lainnya terhadap kejadian-kejadian internal.

Dari ketiga pengertian sistem informasi di atas, maka dapat disimpulkan bahwa sistem informasi menyediakan informasi untuk membantu pengambilan keputusan manajemen, operasi perusahaan dari hari ke hari dan informasi yang layak untuk pihak luar perusahaan.

II.2.6. Fungsi Sistem Informasi

Adapun beberapa kemampuan sistem informasi adalah sebagai berikut :

a) Melaksanakan komputasi sistem, bervolume besar, dengan kecepatan tinggi.
b) Menyediakan komunikasi dalam organisasi atau antar organisasi yang murah akurat dan cepat.

c) Menyimpan informasi dalam jumlah yang sangat besar dalam ruang yang kecil tetapi mudah diakses.
d) Memungkinkan pengaksesan informasi yang sangat banyak di seluruh dunia dengan cepat dan murah.
e) Meningkatkan efektivitas dan efisiensi orang-orang yang bekerja dalam kelompok dalam suatu tempat atau pada beberapa lokasi.
f) Menyajikan informasi dengan jelas yang menggugah pikiran manusia.
g) Mengotomasikan proses-proses bisnis yang semiotomatis dan tugas-tugas yang dikerjakan secara manual.

II.2.7. Konsep Dasar *Web*

Konsep dasar *web* terdiri dari Pengertian *Internet,* Sejarah *Internet, Website Dan Web Browser.*

II.2.7.1. Pengertian *Internet*

Internet berasal dari kata *Interconnection Networking* yang mempunyai arti hubungan sistem dengan berbagai tipe yang membentuk sistem jaringan yang mencakup seluruh dunia (jaringan sistem global) dengan melalui jalur telekomunikasi seperti telepon, *radio link,* satelit dan lainnya.

Dalam mengatur integrasi dan komunikasi jaringan sistem ini digunakan sistem yaitu *TCP/IP (Transmission Control Protokol/Internet Protokol). TCP (Transmission Control Protokol)* bertugas memastikan bahwa semua hubungan bekerja dengan benar, sedangkan *IP (Internet Protokol)* yang mentransmisikan data dari satu sistem ke sistem lain. *TPC/IP* secara umum berfungsi memilih rute

terbaik transmisi data, memilih rute sistem jika suatu rute tidak dapat digunakan, mengatur dan mengirimkan paket-paket pengiriman data.

II.2.7.2. Sejarah *Internet*

Internet merupakan jaringan sistem yang dibentuk oleh Departemen Pertahanan Amerika Serikat di tahun 1969, melalui proyek ARPA yang disebut ARPANET *(Advanced Research Project Agency Network),* di mana mereka mendemonstrasikan bagaimana dengan *hardware* dan *software* yang berbasis UNIX, kita 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 mereka tentukan menjadi cikal bakal pembangunan sistem baru yang sekarang dikenal sebagai *TCP/IP.*

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

II.2.7.3. *WEBSITE*

Situs web (bahasa Inggris: *website*), atau sering disingkat dengan istilah situs adalah sejumlah halaman web yang memiliki sistem saling terkait, terkadang disertai pula dengan berkas-berkas gambar, video, atau jenis-jenis berkas lainnya. Sebuah situs web biasanya ditempatkan setidaknya pada sebuah server web yang

dapat diakses melalui jaringan seperti internet, ataupun *(Local Area Networking)* melalui alamat internet yang dikenali sebagai URL. Gabungan atas semua situs yang dapat diakses publik di internet disebut pula sebagai *World Wide Web* atau lebih dikenal dengan singkatan WWW.

Sebuah halaman web merupakan berkas yang ditulis sebagai berkas teks biasa *(plain text)* yang diatur dan dikombinasikan sedemikian rupa dengan instruksi-instruksi berbasis HTML (*Hyper Text Markup Language)*, atau XHTML, kadang-kadang pula disisipi dengan sekelumit bahasa skrip. Berkas tersebut kemudian diterjemahkan oleh peramban web dan ditampilkan seperti layaknya sebuah halaman pada monitor komputer.

Halaman-halaman web tersebut diakses oleh pengguna melalui protokol komunikasi jaringan yang disebut sebagai HTTP, sebagai tambahan untuk meningkatkan aspek keamanan dan aspek privasi yang lebih baik, situs web dapat pula mengimplementasikan mekanisme pengaksesan melalui protokol HTTPS.

II.2.7.4. *WEB BROWSER*

Web browser adalah program untuk menampilkan halaman yang berbentuk kode HTML. Semua halaman web ditulis dengan bahasa HTML. Walaupun beberapa file mempunyai ekstensi yang berbeda (contoh .html, .php, .php3), outputnya tetap HTML. HTML adalah medium yang selalu dikirimkan ke *web browser* baik halaman itu berupa halaman statis, sebuah *script* (seperti PHP), ataupun yang dibuat oleh program CGI *(Common Gateway Interface).*

HTML adalah bukan bahasa bit level seperti C++ atau Pascal, tapi hanya sebagai bahasa *mark-up* untuk menjelaskan suatu dokumen. HTML adalah versi

yang sederhana dari SGML *(Standardized Generalized Markup Language)*, yaitu bahasa untuk pertukaran data. Kode HTML ditulis dengan mode ASCII. Format ASCII sering disebut teks, yang bisa dibuat dengan perangkat lunak pengolah kata biasa. Hal ini cukup menguntungkan karena bisa dibaca dan ditulis oleh berbagai platform seperti IBM, Mac, Unix.

II.2.8. PHP

Personal Home Page (PHP) merupakan bahasa *script* yang tertanam dalam HTML untuk eksekusi bersifat *server side*, PHP digunakan untuk mengekstraksi data/informasi yang dikehendaki oleh pengguna basis data dan menampilkannya pada halaman web.

Dengan menggunakan PHP maka *maintenance* suatu situs *web* menjadi lebih mudah. Proses *update* data dapat dilakukan dengan menggunakan aplikasi yang dibuat dengan menggunakan *script* PHP. PHP/FI merupakan nama awal dari PHP. PHP/FI adalah *Form Interface*, dibuat pertama kali oleh Rasmus Lerdoff. PHP, awalnya merupakan program CGI yang dikhususkan untuk menerima input melalui form yang ditampilkan dalam web.

PHP secara dasar dapat mengerjakan semua yang dapat dikerjakan oleh program CGI, seperti mendapatkan data dari form, menghasilkan isi halaman web yang dinamis, dan menerima *cookies*. Kemampuan (*Feature*) PHP yang paling diandalkan dan signifikan adalah dukungan kebanyak *database*. Membuat halaman web yang menggunakan data dari *database* dengan sangat mudah dapat dilakukan.

II.2.9. *E-Commerce*

Bahasan *e-commerce* yang dijelaskan meliputi pengertian *e-commerce*, sejarah *e-commerce* dan klasifikasi *e-commerce.*

II.2.9.1. Pengertian E-Commerce

E-commerce atau *Electronic Commerce* merupakan salah satu teknologi yang berkembang pesat dalam aktivitas internet di dunia. Menurut Laudon (1998), *E-Commerce* adalah suatu proses membeli dan menjual produk-produk secara elektronik oleh konsumen dan dari perusahaan ke perusahaan dengan komputer sebagai perantara transaksi bisnis.

II.2.9.2. Sejarah *E-Commerce*

Menurut M.Suyanto dalam bukunya "Strategi Periklanan pada *E-Commerce",* sejarah *E-commerce* atau *Electronic commerce* bermula pada awal tahun 1970-an dengan munculnya inovasi *Electronic Fund Transfer (EFT).* Pada saat itu tingkat aplikasinya masih sangat terbatas pada perusahaan-perusahaan besar, lembaga keungan dan segelintir perusahaan kecil yang nekat. Kemudian muncullah *Electronic Data Interchange (EDI)* yang berkembang dari transaksi keuangan ke proses transaksi lain serta memperbesar jumlah perusahaan yang berperan serta, mulai lembaga-lembaga keuangan sampai perusahaan manufaktur.

Aplikasi-aplikasi yang lain kemudian menyusul, yang memiliki jangkauan dari perdagangan saham hingga sistem reservasi perjalanan. Dengan adanya komersialisasi internet di awal tahun 1990-an, serta pesatnya pertumbuhan yang mencapai hingga jutaan pelanggan potensial, maka muncullah istilah *electronic commerce (e-commerce).* Satu alasan bagi pesatnya perkembangan

teknologi tersebut adalah adanya perkembangan jaringan, protokol, perangkat lunak, dan spesifikasi.

Dengan adanya komersial *Internet* di awal tahun 1990-an maka muncul istilah *Electronic Commerce*. Alasan bagi pesatnya perkembangan teknologi tersebut karena perkembangan jaringan, *software*, meningkatnya persaingan dan berbagai tekanan bisnis.

II.2.9.3. Klasifikasi *E-Commerce*

E-Commerce terbagi kedalam beberapa klasifikasi berdasarkan sasaran penggunaanya, klasifikasi *e-commerce* diantaranya:

a) Business to Business (B2B)

E-Commerce tipe ini meliputi transaksi antar organisasi yang dilakukan di *Electronic market.*

b) Business to Customer (B2C)

Business to Customer (B2C) merupakan transaksi eceran dengan pembeli perorangan.

c) Customer to Customer (C2C)

Customer to Customer (C2C) merupakan *e-commerce* dimana konsumen menjual secara langsung ke konsumen lain. Atau mengiklankan jasa pribadi di Internet.

d) Customer to Business (C2B)

Customer to Business (C2B) merupakan *e-commerce* dimana perseorangan yang menjual produk/layanan ke organisasi, perseorangan yang mencari penjual, berinteraksi dan menyepakati suatu transaksi.

e) Nonbusiness E-commerce

Lembaga non bisnis seperti akademis, organisasi, orgasnisasi keagamaan, organisasi sosial dan lembaga pemerintahan yang menggunakan berbagai tipe *E-Commerce* untuk mengurangi biaya guna meningkatkan operasi dan layanan publik.

f) Intrabusiness (organiszational) E-commerce

Termasuk kategori ini adalah semua aktivitas intern organisasi, biasanya dijalankan di internet yang melibatkan pertukaran barang, jasa/informasi.

II.2.10. *Object Oriented Program (OOP)*

Object Oriented Programming (OOP) adalah suatu metode pemrograman yang berbasiskan pada objek, secara singkat pengertian dari OOP adalah koleksi objek yang saling berinteraksi dan saling memberikan informasi satu dengan yang lainnya. Semua data dan fungsi di dalam paradigma ini dibungkus dalam kelas-kelas atau objek-objek. Konsep dasar dari Pemrograman Berorientasi Objek Pemrograman orientasi-objek menekankan konsep berikut :

1. *Object*

Object adalah sebuah struktur yang menggabungkan data dan prosedur untuk bekerja bersama-sama.

2. *Class*

Class adalah kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebuah *class* merupakan dasar dari modularitas dan struktur dalam pemrograman berorientasi objek.

3. *Abstraction*

Abstraction adalah kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari "pelaku" abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan. Proses, fungsi atau metode dapat juga dibuat abstrak, dan beberapa teknik digunakan untuk mengembangkan sebuah pengabstrakan.

4. *Encapsulation*

Encapsulation memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak; hanya metode dalam objek tersebut yang diberi ijin untuk mengakses keadaannya. Setiap objek mengakses *interface* yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.

5. *Polymorphism*

Polymorphism adalah kemampuan dua buah objek yang berbeda untuk merespon pesan permintaan yang sama dalam suatu cara yang unik.

II.2.10.1. *Unified Modeling Language (UML)*

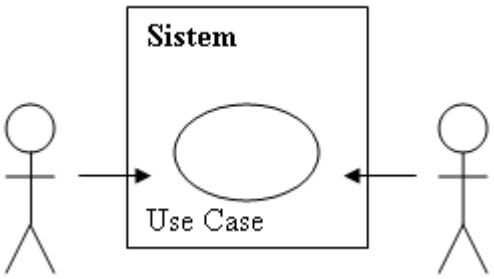
Unified Modeling Language (UML) adalah sebuah bahasa (alat bantu) yang sudah dijadikan sebagai standar di lingkungan sistem perangkat lunak yang dapat dimanfaatkan dalam proses visualisasi, merancang dan mendokumentasikan sebuah perangkat lunak. Dengan memanfaatkan *Unified Modeling Language*

proses pemodelan perangkat lunak akan lebih mudah untuk terjemahkan, meskipun dalam hal pengerjaannya perangkat lunak tersebut dijalankan pada jenis *hardware,* sistem operasi, jaringan, serta ditulis dengan menggunakan bahasa pemrograman yang berbeda. Meskipun tahapan pemodelan pada *Unfied Modeling Language* menggunakan fungsional *class* dan *operation* sebagai konsep dasarnya, *Unfied Modeling Language* tetap dapat mendokumentasikan notasi dan *syntax* serta semantik-nya.

Notasi *Unified Modeling Language* merupakan sekumpulan bentuk khusus yang digunakan untuk menggambarkan berbagai diagram perangkat lunak. Setiap bentuk memiliki makna tertentu dan memiliki *Unified Modeling Language Syntax* yang bermanfaat sebagai media untuk mendefinisikan bentuk-bentuk lainnya yang dapat dikombinasikan. Notasi *Unified Modeling Language* diturunkan dari tiga notasi yang telah ada sebelumnya, seperti grady Booch OOD *(Object Oriented Design),* Jim Rumbaugh OMT *(Object Modeling Technique)* dan Ivar Jacobson OOSE *(Object Oriented Software Engineering).*

II.2.10.2. Use Case Diagram

Use Case Diagram menggambarkan fungsionalitas yang dibutuhkan oleh sebuah sistem. *Use case* harus dapat mempresentasikan sebuah interaksi antara *Actor* dengan sistem. Seorang *Actor* dapat diibaratkan dengan sebuah entitas manusia atau mesin yang berinteraksi langsung dengan sistem untuk melakukan pekerjaan-pekerjaan atau proses tertentu, seperti login, membuat sebuah daftar belanja, dan sebagainya.



Gambar 2.3 ***Use Case Diagram***

II.2.10.3. Class Diagram

Class adalah sebuah spesifikasi yang jika diinisiasi akan menghasilkan sebuah objek, kelas juga merupaka inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan atribut *(property)* pada suatu sistem, sekaligus menawarkan layanan untuk dapat memanipulasi keadaan atribut tersebut. *Class* memiliki empat area pokok, yaitu nama *class* yang disertai dengan *stereotype*, sebuah atribut, sebuah metode dan *multiplicity (dependency).* Sedangkan metode dan atribut memiliki salah satu sifat berikut :

1. *Private*, atribut yang hanya terlihat oleh kelas itu sendiri dan teman dari kelas tersebut.
2. *Protected*, atribut yang hanya terlihat oleh kelas itu sendiri, sub-kelasnya atau teman kelas tersebut.
3. *Public*, atribut yang dapat dilihat dari dalam dan dari luar paket yang mengandung kelas yang bersangkutan.
4. *Primer*, kelas yang berperan sebagai induk dari kelas lainnya.
5. *Esential*, sifat yang diturunkan dari kelas lainnya.

Selain memiliki sifat, atribut serta metode, *class* juga memiliki beberapa *multiplicity* atau sering dikenal dengan *dependency* untuk memperlihatkan fungsionalitas dari hubungan antar *class*, berikut jenis-jenis *multiplicity* yang menggambarkan hubungan antar *class*.



Gambar 2.4. *Multiplicity* Satu ke Satu



Gambar 2.5. *Multiplicity* Satu ke Banyak



Gambar 2.6. *Multiplicity* Satu ke Satu atau Banyak



Gambar 2.7. *Multiplicity* Satu atau Banyak ke Satu atau Banyak



Gambar 2.8. *Multiplicity* Banyak ke Banyak

Dari penjelasan tersebut maka dapat disimpulkan bahwa *class diagram* adalah sebuah diagram yang menggambarkan sebuah struktur dari deskripsi kelas, *package*, objek dan hubungan antar kelas seperti, *entertainment*, pewarisan, *association*, dan sebagainya.

II.2.10.4. Behaviour Diagram

Behavior diagram dapat dikelompokkan menjadi tiga diagram, yaitu :

a. *Statechart Diagram*

Statechart diagram berfungsi untuk memodelkan prilaku dinamis satu kelas satu objek.

b. *Activity Diagram*

Activity diagram memodelkan alur kerja (*work flow*) sebuah proses bisnis dan urutan aktivitas dalam suatu proses.

c. *Interaction Diagram*

Interaction diagram dibagi menjadi dua model diagram yaitu :

1. *Sequence diagram* menjelaskan interaksi objek yang disusun dalam suatu urutan waktu. Diagram ini secara khusus bersosialisasi dengan *use case. Sequence diagram*, memperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu dalam *use case.*
2. *Collaboration diagram* melihat pada interaksi dan hubungan terstruktur antar objek. Tipe diagram ini menekankan pada hubungan *(relationship)* antar objek, sedangkan *sequence diagram* menekankan pada urutan kejadian. Dalam *collaboration diagram* terdapat beberapa objek, *link*, dan *message*.

II.2.10.5. Implementation Diagram

Implementation diagram dibagi menjadi dua diagram, yaitu :

a) *Component Diagram* menggambarkan alokasi semua kelas dan objek ke dalam komponen-komponen dalam desain fisik sistem *software.* Diagram ini memperlihatkan pengaturan dan kebergantungan antara komponen-komponen *software*, seperti *source code, binary code,* dan komponen tereksekusi *(execute components)*

b) *Deployment Diagram* memperlihatkan pemetaan *software* kepada *hardware*. Dimana akan berjalan (di server/multitier, *standalone* atau lainnya), dan menggambarkan model koneksi dan kemampuan jaringan dan hal lainya yang bersifat fisik.

II.2.11. Model – View – Controller (MVC)

Model-View-Controller atau MVC adalah sebuah metode untuk membuat sebuah aplikasi dengan memisahkan data *(Model)* dari tampilan *(View)* dan cara bagaimana memprosesnya *(Controller).* MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, antarmuka pengguna, dan bagian yang menjadi kontrol dalam sebuah aplikasi web. Bagian–bagian dari MVC adalah :

1. Model

Model mewakili struktur data. Biasanya model berisi fungsi-fungsi yang membantu seseorang dalam pengelolaan basis data seperti memasukkan data ke basis data, pembaruan data dan lain-lain.

2. View

View adalah bagian yang mengatur tampilan ke pengguna. Bisa dikatakan berupa halaman web.

3. Controller

Controller merupakan bagian yang menjembatani *model* dan *view. Controller* berisi perintah-perintah yang berfungsi untuk memproses suatu data dan mengirimkannya ke halaman web.

II.2.12. Fasilitas dan *Software* Pendukung

Fasilitas dan *Software* pendukung *meliputi Hyper Text Markup Language (HTML), Personal Home Page (PHP), Structured Query Language (SQL), Macromedia Dreamweaver, Cascading Style Sheet (CSS), SSL dan Paypal.*

II.2.12.1. Hyper Text Markup Language (HTML)

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

Secara garis besar, terdapat empat jenis elemen dari HTML :

1. *Structural*, tanda yang menentukan level atau tingkatan dari sebuah teks (contoh, <h1>Golf</h1> akan memerintahkan browser untuk menampilkan "Golf" sebagai teks tebal besar yang menunjukkan sebagai Heading 1
2. *Presentational*, tanda yang menentukan tampilan dari sebuah teks tidak peduli dengan level dari teks tersebut (contoh, boldface akan menampilkan bold. Tanda *presentational* saat ini sudah mulai digantikan oleh CSS dan tidak direkomendasikan untuk mengatur tampilan teks,
3. *Hypertext*, tanda yang menunjukkan pranala ke bagian dari dokumen tersebut atau pranala ke dokumen lain.

   contoh, Wikipedia akan menampilkan Wikipedia sebagai sebuah *hyperlink* ke URL tertentu.
4. Elemen widget yang membuat objek-objek lain seperti tombol (<button>), list (<li>), dan garis horizontal (<hr>).

II.2.12.2. Personal Home Page (PHP)

PHP adalah skrip bersifat *server-side* yang ditambahkan ke dalam HTML. Sifat *server-side* berarti pengerjaan skrip dilakukan di *server*, kemudian hasilnya dikirim ke *browser*. Keunggulan dari sifat *server-side* adalah :

1. Tidak diperlukan kompatibilitas *browser* atau harus menggunakan *browser* tertentu, karena serverlah yang mengerjakan skrip PHP.
2. Dapat memanfaatkan sumber-sumber aplikasi yang dimiliki oleh *server*, misalnya koneksi ke *database*.

3. Dapat berkomunikasi dengan layanan-layanan yang menggunakan protokol IMAP, SNMP, NNTP, POP3, HTTP, dan lain-lain.
4. Dapat melakukan koneksi dengan berbagai macam *database*, salah satunya adalah *database* MYSQL

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

Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilisan kode sumber ini menjadi sumber terbuka, maka banyak pemrogram yang tertarik untuk ikut mengembangkan PHP. Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini, *interpreter* PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan.

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

Pada pertengahan tahun 1999, Zend merilis *interpreter* PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan

kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

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

Penulisan Script PHP

Berikut ini adalah contoh penulisan script/kode dari PHP :

```
<HTML>
<HEAD>
<TITLE>contoh PHP</TITLE>
</HEAD>
<BODY>
<?php
echo ”Scripts PHP!”;
?>
</BODY>
</HTML>
```

II.2.12.3. Structured Query Language (SQL)

SQL (*Structured Query Language*) adalah sebuah bahasa yang dipergunakan untuk mengakses data dalam basis data relasional. Bahasa ini secara *de facto* merupakan bahasa standar yang digunakan dalam manajemen basis data relasional. Saat ini hampir semua server basis data yang ada mendukung bahasa ini untuk melakukan manajemen datanya.

Secara umum, SQL terdiri dari dua bahasa, yaitu *Data Definition Language* (DDL) dan *Data Manipulation Language* (DML). Implementasi DDL dan DML berbeda untuk tiap Sistem Manajemen Basis Data (SMBD), namun secara umum implementasi tiap bahasa ini memiliki bentuk standar yang

ditetapkan ANSI. Artikel ini akan menggunakan bentuk paling umum yang dapat digunakan pada kebanyakan SMBD.

1. *Data Definition Language* (DDL)

   DDL digunakan untuk mendefinisikan, mengubah, serta menghapus basis data dan objek-objek yang diperlukan dalam basis data, misalnya *tabel*, *view, user*, dan sebagainya. Secara umum, DDL yang digunakan adalah :

   a. *CREATE* untuk membuat objek baru,
   b. *USE* untuk menggunakan objek,
   c. *ALTER* untuk mengubah objek yang sudah ada, dan
   d. *DROP* untuk menghapus objek.

   DDL biasanya digunakan oleh administrator basis data dalam pembuatan sebuah aplikasi basis data.

2. *Data Manipulation Language* (DML)

   DML digunakan untuk memanipulasi data yang ada dalam suatu tabel. Perintah yang umum dilakukan adalah:

   a. *SELECT* untuk menampilkan data

      SELECT adalah perintah yang paling sering digunakan pada SQL, sehingga kadang-kadang istilah *query* dirujukkan pada perintah *SELECT*. *SELECT* digunakan untuk menampilkan data dari satu atau lebih tabel, biasanya dalam sebuah basis data yang sama. Secara umum, perintah SELECT memiliki bentuk lengkap :

```
SELECT [nama_tabel|alias.]nama_field1 [AS alias1] [, nama_field2, ...]
FROM nama_tabel1 [AS alias1] [INNER|LEFT|RIGHT JOIN tabel2 ON
kondisi_penghubung]
[, nama_tabel3 [AS alias3], ...]
[WHERE kondisi]
```

[ORDER BY nama_field1 [ASC|DESC][, nama_field2 [ASC|DESC], ...]]
[GROUP BY nama_field1[, nama_field2, ...]]
[HAVING *kondisi_aggregat*]

b. *INSERT* untuk menyimpan data dalam tabel dipergunakan sintaks:

INSERT INTO [NAMA_TABLE] ([DAFTAR_FIELD]) VALUES

([DAFTAR_NILAI])

c. *UPDATE* untuk mengubah data yang sudah ada:

UPDATE [NAMA_TABLE] SET [NAMA_KOLOM]=[NILAI] WHERE

[KONDISI]

d. *DELETE* untuk menghapus data:

DELETE FROM [nhew andiz] [KONDISI]

II.2.12.4. Macromedia Dreamweaver

Macromedia Dreamweaver adalah sebuah HTML editor profesional untuk mendesain secara visual dan mengelola situs web maupun halaman web. Bila kita menyukai untuk berurusan dengan kode-kode HTML secara manual atau lebih menyukai bekerja dengan lingkungan secara visual dalam melakukan editing, *Dreamweaver* membuatnya menjadi lebih mudah dengan menyediakan tools yang sangat berguna dalam peningkatan kemampuan dan pengalaman dalam mendesain web.

Dreamweaver MX dalam hal ini digunakan untuk *web desain*. *Dreamweaver MX* mengikutsertakan banyak tools untuk kode-kode dalam halaman web beserta fasilitas-fasilitasnya, antara lain : Referensi HTML, CSS dan *Javascript*, *Javascript debugger* dan editor kode (tampilan kode dan *Code Inspector)* yang mengizinkan kita mengedit kode *Javascript*, XML dan dokumen teks lain secara langsung dalam *Dreamweaver*.

Teknologi *Dreamweaver Roundtrip* HTML mampu mengimpor dokumen HTML tanpa perlu memformat ulang kode tersebut dan kita dapat menggunakan *Dreamweaver* pula untuk membersihkan dan memformat ulang HTML bila kita menginginkannya.

Selain itu *Dreamweaver j*uga dilengkapi kemampuan manajemen situs, yang memudahkan kita mengelola keseluruhan elemen yang ada dalam situs.

Evolusi *Dreamweaver* salah satunya :

1. *Macromedia Dreamweaver* 3 : merupakan versi lamanya yaitu *web authoring* terbaik pada tahun 2000.
2. *Macromedia Dreamweaver* 4 : merupakan software yang lebih baik lagi dan lebih canggih pada tahun 2002.
3. *Macromedia Dreamweaver MX*: pertama kali di tampilkan pada tahun 2004, selain *software* untuk mendesain web,juga bisa untuk menyunting kode dan untuk membuat aplikasi *web* dengan menggunakan bahasa pemrograman JPS, PHP, ASP atau *Coldfusion*. Dilengkapi dengan fasilitas yang cukup lengkap untuk manajemen situs.
4. *Dreamweaver 8 MX*, *adobe Dreamweaver* cs3 dan cs4.

II.2.12.5. Cascading Style Sheet (CSS)

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

Sama halnya *styles* dalam aplikasi pengolahan kata seperti Microsoft Word yang dapat mengatur beberapa *style,* misalnya *heading,* subbab, bodytext,

footer, *images*, dan *style* lainnya untuk dapat digunakan bersama-sama dalam beberapa berkas (file). Pada umumnya CSS dipakai untuk memformat tampilan halaman web yang dibuat dengan bahasa HTML dan XHTML.

CSS dapat mengendalikan ukuran gambar, warna bagian tubuh pada teks, warna tabel, ukuran *border*, warna *border*, warna *hyperlink*, warna *mouse over*, spasi antar paragraf, spasi antar teks, margin kiri, kanan, atas, bawah, dan parameter lainnya. CSS adalah bahasa *style sheet* yang digunakan untuk mengatur tampilan dokumen. Dengan adanya CSS memungkinkan untuk menampilkan halaman yang sama dengan format yang berbeda.

Ada dua sifat CSS yaitu internal dan eksternal :

1. Internal, skrip itu dimasukan secara langsung ke halaman website yang akan didesain. Kalau halaman web yang lain akan didesain dengan model yang sama, maka skrip CSS itu harus dimasukkan lagi ke dalam halaman *web* yang lain itu.
2. Eksternal, di mana skrip CSS dipisahkan dan diletakkan dalam berkas khusus. Nanti, cukup gunakan semacam tautan menuju berkas CSS itu jika halaman *web* yang didesain akan dibuat seperti model yang ada di skrip tersebut.

Adapun keuntungan dan kerugian menggunakan CSS *(Cascading Style Sheets)* adalah sebagai berikut:

1. Keuntungan Mengggunakan CSS *(Cascading Style Sheets)*
    a. ***Update* tampilan lebih mudah,** meng-*update* tampilan situs web yang terdiri dari beberapa halaman saja tentu jauh lebih mudah daripada untuk situs dengan ratusan atau ribuan halaman. CSS dapat meng-

update tampilan halaman-halaman web dalam jumlah banyak dengan mudah, karena semua tag untuk *style* ini berada pada satu file CSS saja.

b. **Beban *bandwidth* lebih kecil**, dengan CSS ukuran file web akan menjadi langsing karena *tag-tag style* dipisahkan secara mandiri. Hal ini secara signifikan akan berdampak pada proses *loading* yang lebih cepat. Selain itu, CSS hanya akan diakses satu kali oleh *browser* dan akan terus ditautkan pada halaman-halaman lain. Hal ini berbeda dengan situs yang menggunakan table karena harus di-*load* berulang-ulang.

c. **Modifikasi *web template* lebih mudah**, cara paling cepat membangun situs web ialah dengan menggunakan *template*. Namun terkadang *template* tersebut tidak sesuai dengan keinginan dan keperluan. CSS akan lebih mudah melakukan "*vermak*" pada *template* tersebut, misalnya mengganti warna latar atau mengubah jenis huruf.

d. **Lebih mudah digunakan pada *mobile phone,*** sekarang pengguna melakukan *browsing* tidak hanya melalui *browser* di PC, tetapi juga melalui *mobile phone* atau PDA. Situs *web* yang menggunakan CSS akan jauh lebih mudah digunakan oleh mereka yang *browsing* melalui *gadget-gadget* tersebut.

e. ***Search engine friendly,*** sebuah situs *web* yang dibuat dengan CSS akan lebih *search engine friendly* daripada situs-situs web yang menggunakan tabel sebagai pondasi *layout*-nya.

2. Kerugian Mengggunakan CSS *(Cascading Style Sheets) :*

Tidak semua *browser* mengartikan kode CSS dengan cara yang sama. Jadi kadang-kadang, tampilan web dengan CSS terlihat baik di *browser* yang satu, tapi berantakan di *browser* yang lain. Jadi pengguna harus memeriksa tampilan supaya terlihat baik di semua *browser* dan menambahkan kode-kode khusus *browser* tertentu jika memang dibutuhkan agar tampilan *web* pengguna terlihat baik di semua *browser*.

***II.2.12.6.* SSL**

SSL atau *Secure Sockets Layer* adalah sebuah protokol keamanan data yang digunakan untuk menjaga pengiriman data *web server* dan pengguna situs *web* tersebut. SSL merupakan protokol standar *web* untuk mengenkripsi komunikasi data antara pengguna dengan *site* yang diakses. Lalu lintas data melalui sambungan SSL akan dienkripsi sehingga akan menghindari resiko sabotase atau pencurian data. Misalnya data *username, password*, data kartu kredit dll.

Protokol SSL dan TLS berjalan pada layer dibawah *application* protokol seperti HTTP, SMTP and NNTP dan di atas layer TCP transport protokol, yang juga merupakan bagian dari *TCP/IP protokol*. Selama SSL dan TLS dapat menambahkan keamanan ke protokol apa saja yang menggunakan TCP, keduanya terdapat paling sering pada metode akses HTTPS. HTTPS menyediakan keamanan *web-pages* untuk aplikasi seperti pada *Electronic commerce*. Protokol SSL dan TLS menggunakan *cryptography public-key* dan sertifikat *public key* untuk memastikan identitas dari pihak yang dimaksud. Sejalan dengan

peningkatan jumlah *client* dan *server* yang dapat mendukung TLS atau SSL alami, dan beberapa masih belum mendukung. Dalam hal ini, pengguna dari *server* atau *client* dapat menggunakan produk *standalone*-SSL seperti halnya *Stunnel* untuk menyediakan enkripsi SSL.

II.2.12.7. Paypal

Paypal adalah salah satu alat pembayaran *(Payment procesors)* menggunakan *internet* yang terbanyak digunakan d idunia dan teraman.

PayPal mengatasi kekurangan dalam pengiriman uang tradisional seperti Cek atau *Money Order* yang prosesnya dapat memakan waktu lama. *Paypal* seperti rekening bank, pertama pembuatan *account*, lalu dana akan ditarik langsung dari kartu kredit ke *account* tersebut atau dengan dana dari transferan *account Paypal* satu ke Saldo atau *Balance Paypal* yang lain, maka dan *account Paypal* sudah dapat digunakan sudah bertransaksi.

Keuntungan menggunakan *Paypal* diantaranya:

a) Kebijaksanaan perlindungan tertulis untuk pembeli yang menggunakan *Paypal* menyatakan pembeli yang menggunakan Paypal dapat melakukan komplain dalam waktu 45 hari jika pembeli belum mendapatkan barang yang dipesan atau jika barang yang dipesan tidak sesuai deskripsi yang diberitahukan penjual.
b) *Paypal* juga melindungi penjual dari pengembalian uang atau komplain tergantung situasi dan pembuktian.

Adapun kekurangan *PayPal* yaitu karena sistem pengamanan super ekstra sehingga terkadang sedikit menyulitkan bagi pemilik *account paypal*.

Karena dengan adanya sistem pengamanan yang super ekstra ketat tersebut maka sangat memungkinkan untuk terjadinya *account limitation acces* atau dibatasinya *access* penggunaan *account paypal.* Hal ini akan terjadi jika ada kegiatan yang dianggap mencurigakan oleh pihak *paypal* yang sekiranya akan merugikan pihak *paypal*.