

BAB 2

LANDASAN TEORI

2.1 Pengertian Sistem

Sistem adalah sekumpulan komponen yang mengimplementasikan model dan fungsionalitas yang dibutuhkan. Komponen-komponen tersebut saling berinteraksi di dalam sistem guna mentransformasi input yang di berikan kepada sistem tersebut menjadi output yang berguna dan bernilai bagi aktornya (Irwanto, p2).

2.1.1 Pengertian Sistem Komputer

Menurut *webmaster's dictionary* (Pressman, 2005, p154) definisi dari sistem komputer adalah :

“a set or arrangement of elements that are organized to accomplish some predefined goal by processing information” yang artinya suatu set atau kumpulan elemen – elemen yang terorganisir untuk mencapai beberapa tujuan yang telah ditetapkan oleh pengelola informasi.

2.1.1.1 Element Sistem Komputer

Ada enam buah elemen dasar yang membangun sebuah sistem komputer (Pressman, 2005, p155) diantaranya :

a. *Software*

Program komputer, struktur data, dan dokumentasi yang saling terkait satu sama lain yang berfungsi mempengaruhi metode logis, prosedur, dan kontrol yang dibutuhkan.

b. *Hardware*

Perangkat elektronik yang mampu menyediakan kemampuan komputasi, perangkat – perangkat yang terkait satu sama lain (misalnya jaringan, perangkat telekomunikasi) yang mengaktifkan aliran data , dan perangkat elektromekanis (misalnya sensor, motor) yang dapat memberikan fungsi eksternal.

c. *People*

Pengguna atau orang yang menggunakan *software* dan *hardware*.

d. *Database*

Tempat penyimpanan informasi yang tersusun secara rapi dalam jumlah yang besar dan dapat di akses melalui *software*.

e. *Documentation*

Informasi yang bersifat deskriptif (misalnya, buku panduan, *online help*) yang menggambarkan pengguna atau pengoperasian sistem.

f. Procedures

Langkah – langkah yang menetapkan penggunaan dari masing – masing elemen sistem atau konteks *procedural* dimana sistem berada.

2.1.2 Pengertian Sistem Informasi Penjualan

Sistem Informasi Penjualan adalah suatu sistem informasi yang mengorganisasikan serangkaian prosedur dan metode yang dirancang untuk menghasilkan, menganalisa, menyebarkan dan memperoleh informasi guna mendukung pengambilan keputusan mengenai penjualan (Anonim1 : 2011).

2.1.2.1 Pengertian Penjualan

Penjualan antara lain terjadi dari transaksi, penjualan barang atau baik secara tunai maupun kredit. Konsep penjualan berpendapat bahwa para konsumen, jika dibiarkan sendiri cenderung tidak akan membeli produk -produk yang dihasilkan organisasi tersebut. oleh karena itu, organisasi tersebut harus melakukan kegiatan penjualan yang agresif dan usaha promosi yang gencar (Mulyadi, 2001, p204).

2.1.2.2 Pengertian Informasi

Informasi adalah pengetahuan yang didapatkan dari pembelajaran, pengalaman, atau instruksi. Namun demikian, istilah ini memiliki banyak arti bergantung pada konteksnya, dan secara umum berhubungan erat dengan konsep seperti arti, pengetahuan, negentropy, komunikasi, kebenaran, representasi, dan rangsangan mental.

Dalam bidang ilmu komputer, informasi adalah data yang disimpan, diproses, atau ditransmisikan. Penelitian ini memfokuskan pada definisi informasi sebagai pengetahuan yang didapatkan dari pembelajaran, pengalaman, atau instruksi dan alirannya (Anonim2 : 2011).

2.1.2.3 Pengertian Sistem Informasi

Sistem informasi adalah sekumpulan komponen pembentuk sistem yang mempunyai keterkaitan antara satu komponen dengan komponen lainnya yang bertujuan menghasilkan suatu informasi dalam suatu bidang tertentu. Dalam sistem informasi diperlukannya klasifikasi alur informasi, hal ini disebabkan keanekaragaman kebutuhan akan suatu informasi oleh pengguna informasi. Kriteria dari sistem informasi antara lain, fleksibel, efektif dan efisien (Anonim3 : 2011).

2.2 *E-Commerce*

Menurut Turban (2004, p3) *E-Commerce* adalah proses pembelian, penjualan, transfer, dan pertukaran produk, layanan, atau informasi melalui jaringan komputer. Jenis transaksi *E-Commerce* (Turban,2004,p7-8) :

a. *Bisnis ke bisnis*

E-commerce model dimana yang terlibat seluruhnya adalah pelaku bisnis atau organisasi.

b. *Bisnis ke konsumen*

E-commerce model dimana pelaku bisnis menjual barang konsumen perorangan.

c. *Bisnis untuk bisnis ke konsumen*

E-commerce model dimana bisnis menyediakan beberapa produk atau layanan kepada bisnis klien yang mempertahankan pelanggan sendiri.

d. *Konsumen ke bisnis*

E-commerce model dimana individu menggunakan internet untuk menjual produk atau jasa kepada organisasi atau individu yang mencari penjual untuk menawar pada produk atau layanan yang mereka butuhkan.

e. *Konsumen ke konsumen*

E-commerce model dimana konsumen menjual langsung ke konsumen lainnya.

f. *Peer to peer*

Teknologi yang memungkinkan jaringan komputer untuk berbagi data dengan langsung satu sama lain.

g. *Mobile commerce*

E-commerce yang dilakukan dalam lingkungan nirkabel.

h. *Intrabusiness EC*

E-commerce kategori yang mencakup semua kegiatan organisasi internal yang melibatkan pertukaran barang, jasa, atau informasi antar berbagai unit dan individu dalam sebuah organisasi.

i. *Bisnis ke karyawan*

E-commerce model dimana organisasi memberikan layanan, informasi atau produk kepada karyawan individu.

j. *Collaborative commerce(C-commerce)*

E-commerce model dimana individu atau grup berkomunikasi secara online.

k. *Nonbusiness EC*

Lembaga atau instansi (akademik,religious,social,pemerintah) menggunakan EC untuk mengurangi pengeluaran mereka atau untuk meningkatkan operasi umum dan layanan pelanggan.

l. *E-learning*

Penyampaian informasi untuk tujuan pelatihan atau pendidikan.

m. *Exchange to exchange*

E-commerce model dimana pertukaran elektronik secara formal dapat terhubung ke satu sama lain untuk tujuan pertukaran informasi

2.2.1 Keuntungan *E-Commerce*

Menurut Turban (2004, p17-19) keuntungan yang didapat dengan adanya *e-commerce* adalah :

2.2.1.1 Keuntungan Terhadap Organisasi

- a. Jangkauan global.
- b. Pengurangan biaya produksi, proses, distribusi, dan penyimpanan.
- c. Perbaiki rantai suplai
- d. Bisnis akan selalu buka(24/7/365) di web,tanpa penambahan waktu dan biaya.
- e. Cara berbisnis yang baru, melalui e-commerce dapat memberikan keuntungan strategis dan meningkatkan laba.

2.2.1.2 Keuntungan Terhadap Konsumen

- a. Memberikan konsumen untuk berbelanja atau melakukan transaksi lainnya sepanjang tahun, 24 jam sehari.
- b. Menyediakan banyak pilihan bagi konsumen.
- c. Dalam kasus produk digital, e-commerce memungkinkan untuk pengiriman lebih cepat.
- d. Pada banyak negara, bisnis online dibebaskan dari pajak penjualan.

2.2.1.3 Keuntungan Terhadap Masyarakat

- a. Berkurangnya polusi udara karena dengan e-commerce pembeli dapat melakukan transaksi tanpa harus datang ke toko.
- b. Beberapa barang bisa dijual dengan harga lebih murah sehingga memungkinkan orang yang kurang mampu bisa membeli.
- c. Orang yang tinggal di wilayah pedesaan kini dapat menikmati produk dan layanan yang tersedia di masa lalu.
- d. pelayanan publik, seperti perawatan kesehatan, pendidikan dan distribusi pelayanan sosial pemerintah, dapat dilakukan dengan mengurangi biaya dan meningkatkan kualitas.

2.2.2 Keterbatasan E-Commerce

Menurut Turban (2004, p19) resiko yang muncul dari penerapan *e-commerce* adalah :

2.2.2.1 Keterbatasan Teknologi

- a. Kurangnya standar universal yang diterima untuk kualitas, keamanan dan kehandalan.
- b. Bandwidth telekomunikasi tidak mencukupi.
- c. Alat pengembangan piranti lunak masih berkembang.
- d. Membutuhkan biaya tambahan untuk membuat web server.

- e. Rangka pemenuhan bisnis kepada pembeli berskala besar memerlukan gudang otomatis khusus.

2.2.2.2 Keterbatasan Non teknologi

- a. Keamanan *privacy* pelanggal kurang diperhatikan.
- b. Kurangnya kepercayaan dalam *e-commerce* karena pembeli dan penjual tidak bertemu secara langsung.
- c. Beberapa konsumen lebih suka menyentuh dan melihat secara langsung barang yang akan dibeli.
- d. Peningkatan jumlah penipuan yang ada di internet.
- e. Sulit untuk mengukur beberapa keuntungan dari *e-commerce*, seperti iklan.

2.3 Internet

Internet adalah kumpulan jaringan komputer di seluruh dunia yang saling terhubung. internet terdiri dari jaringan yang terpisah tetapi saling berhubungan yang diantaranya milik komersial, organisasi pendidikan dan pemerintah dan *internet service provider* (Connoly, 2002, p944).

2.3.1 WWW (World Wide Web)

Sistem berbasis hypermedia yang menyediakan sarana informasi *browsing* di internet dengan cara yang tidak berurutan menggunakan *hyperlink*.(Connoly, 2002, p948).

2.3.2 Browser

Browser adalah *software* yang dijalankan pada komputer pemakai (“*client*”) yang meminta informasi dari server web dan menampilkannya sesuai dengan file data itu sendiri. (Jill Ellsworth dan Matthew Ellsworth ,1997, p4) . Browser dapat menerjemahkan bahasa html, jsp, php, asp. Browser memiliki beberapa fitur bagi penggunanya, yaitu :

a. *Bookmark*

Fitur ini berfungsi untuk menyimpan alamat suatu web di browser sehingga dapat dengan mudah dikunjungi tanpa harus mengetikkan alamat di address box.

b. *History*

Merupakan catatan dari semua web yang pernah kita kunjungi.

c. *Image, Font & Colour*

d. *Plug-ins*

Merupakan sebuah perangkat penunjang bagi *browser* agar dapat berfungsi lebih optimal. Contoh plug-ins antara lain adalah *adobe reader plug-ins* dan flash.

e. *Cookie*

Merupakan fitur untuk menyimpan data pengguna apabila *login* di suatu web

2.3.3 *Web Server*

Menurut Jill Ellsworth dan Matthew Ellsworth (1997, p4) *web server* adalah sebuah komputer dan *software* yang menyimpan dan mendistribusikan data ke komputer lainnya melalui *internet*.

2.3.4 *HTML (HyperText Markup Language)*

Menurut Jill Ellsworth dan Matthew Ellsworth (1997, p37) HTML merupakan sistem yang digunakan untuk merancang halaman web (Connoly, 2002, p951). Dokumen HTML dapat dibuat menggunakan editor teks standar seperti *notepad*.

2.3.5 *URL (Uniform Resource Locator)*

URL adalah karakter alfanumerik unik yang mewakili lokasi atau alamat dari suatu sumber daya di internet dan bagaimana sumber daya yang harus diakses (Connoly, 2002, p952). Bagian pertama URL menentukan metode akses. Setelah tanda titik dua terdapat dua garis miring yang menunjukkan bahwa yang mengikutinya adalah alamat mesin. URL menunjuk satu “titik” di *internet*.

Contoh URL : <http://binusmaya.binus.ac.id>

2.3.6 *HTTP (HyperText Transfer Protocol)*

Menurut Connoly (2002, p949) HTTP adalah protokol yang digunakan untuk mentransfer halaman web melalui internet. HTTP mendefinisikan bagaimana klien dan server berkomunikasi. HTTP berguna

untuk mengirimkan informasi antara server dan klien. HTTP didasarkan pada paradigma merespon permintaan. Sebuah transaksi HTTP terdiri dari tahapan berikut :

- Koneksi : Klien membentuk koneksi dengan web server.
- Permintaan : Klien mengirim pesan permintaan kepada web server.
- Respon klien : Web server mengirim respon (contoh : dokumen HTML) kepada klien.
- Tutup : Koneksi ditutup oleh web server.

2.4 Database

Menurut Connolly (2002,p14) *database* adalah sekumpulan data logikal yang saling berhubungan, dan deskripsi dari data tersebut dirancang untuk kebutuhan informasi suatu organisasi. *Database* sekarang merupakan sebuah bagian integral dari kehidupan sehari-hari yang secara tidak sadar kita sedang menggunakannya.

Database terdiri dari beberapa objek, yaitu :

a. *Field*

Field adalah kumpulan dari beberapa karakteristik dari objek-objek yang ada.

b. *File*

File adalah kumpulan dari beberapa *record* yang berhubungan membentuk saling ketergantungan satu dengan yang lain.

Contoh : transaksi yang *recordnya* mencatat nama barang, kode barang, tanggal.

c. *Record*

Record adalah kumpulan dari beberapa *field* yang berhubungan satu sama lain.

Contoh : nama, alamat, kode barang, dsb.

d. *Entity*

Entity adalah satu kesatuan yang terdiri dari informasi yang disimpan.

Contoh : pelanggan, barang, karyawan.

e. *Attribute*

Attribute adalah nama dari suatu kolom relasi yang menjelaskan suatu *entity*.

contoh : nama, kode barang, dsb.

f. *Primary Key*

Primary key adalah sebuah *field* yang memiliki nilai unik yang tidak memiliki kesamaan antara *record* yang satu dengan yang lain.

g. *Foreign Key*

Foreign key adalah satu atribut atau kumpulan atribut dalam satu relasi yang berguna untuk menghubungkan *primary key* lain yang berada di tabel lain.

2.4.1 *Database Management System (DBMS)*

Menurut Connolly (2002, p16) DBMS adalah suatu sistem *software* yang memungkinkan *user* untuk menentukan, membuat, memelihara, dan mengontrol akses ke *database*.

DBMS merupakan perangkat lunak yang berinteraksi dengan program aplikasi *user* dan *database*. DBMS memiliki beberapa komponen (Connolly, 2002, p16) :

a. *Hardware*

DBMS dan aplikasi memerlukan *hardware* untuk bekerja.

b. *Software*

Komponen *Software* terdiri dari komponen perangkat lunak *DBMS* itu sendiri dan program aplikasi, bersama sama dengan sistem operasi, termasuk perangkat lunak jaringan jika *DBMS* sedang digunakan pada jaringan.

c. *Data*

Data merupakan komponen paling terpenting pada *DBMS*. *Data* bertindak sebagai penghubung antara komponen – komponen mesin dan komponen manusia.

d. *Procedures*

Prosedur mengacu pada petunjuk dan aturan yang mengatur desain dan penggunaan *database*. Pengguna sistem dan staf yang mengelola data *database* memerlukan prosedur yang terdokumentasi tentang cara menggunakan atau menjalankan sistem ini.

e. *People*

Komponen terakhir adalah orang yang terlibat dengan sistem.

2.4.2 MySQL

MySQL merupakan *software* yang tergolong database server dan bersifat *Open Source*. *Open Source* menyatakan bahwa software ini dilengkapi oleh source code (kode yang dipakai untuk membuat MySQL), selain tentu saja bentuk *executable*-nya atau kode yang dapat dijalankan

secara langsung dalam sistem operasi, dan bisa diperoleh dengan cara mengunduh di internet secara gratis. Hal menarik lainnya adalah MySQL juga bersifat multiplatform yang membuat MySQL dapat dijalankan pada berbagai sistem operasi.

Pengaksesan data dalam *database* dapat dilakukan dengan mudah melalui SQL (*Structured Query Language*). Data dalam *database* bisa diakses melalui aplikasi non-Web (misalnya dengan Visual Basic) maupun aplikasi Web (misalnya dengan PHP) (Abdul Kadir, 2009, p15).

2.4.3 *Entity Relationship Diagram (ERD)*

Menurut Connolly (2002, p330-351), *entity relationship diagram* digunakan untuk menggambarkan struktur logical *database* dalam bentuk diagram ERD menyediakan cara yang mudah dan sederhana untuk memahami berbagai komponen dalam desain *database*. ERD mempunyai tiga komponen yaitu :

1. *Entity*

Entity merupakan benda yang memiliki identifikasi yang berbeda. *Entity* dapat digambarkan sebagai persegi yang berisi nama dari *entity* tersebut.

2. *Relationship*

Relationship merupakan asosiasi antara *entity*. *Entity* merupakan pengikut dari *relationship*. *Relationship* dapat digambarkan dalam bentuk belah ketupat yang berisi nama dari relasi tersebut. *Relationship* dapat berupa relasi *one-to-one*, *one-to-many*, atau *many-to-many*.

a. Relasi *one-to-one* (1 atau 1..1)

Relasi dimana setiap *entity* yang ada hanya dapat mempunyai maksimal satu relasi dengan *entity* yang lain.

b. Relasi *one-to-many* (1..*)

Relasi dimana setiap *entity* yang ada dapat mempunyai satu atau lebih dari satu relasi dengan *entity* yang lain.

c. Relasi *many-to-many* (*..*)

Relasi dimana setiap *entity* dapat mempunyai lebih dari satu relasi dengan *entity* yang lain.

3. Properti / *Attribute*

Properti / *attribute* adalah sifat atau karakteristik deskriptif suatu entitas. Setiap properti / *attribute* mempunyai *key* diantaranya *Primary Key* (PK) dan *Foreign Key* (FK) untuk menunjukkan hubungan antara kedua entitas.

2.5 PHP (*Hypertext Preprocessor*)

Bahasa pemrograman PHP adalah bahasa pemrograman yang bekerja dalam sebuah web server. Script-script PHP yang anda buat harus tersimpan dalam sebuah server dan dieksekusi atau diproses dalam server tersebut. Penggunaan program PHP memungkinkan sebuah *website* menjadi lebih interaktif dan dinamis. Data yang dikirim oleh pengunjung *website* klien akan diolah dan disimpan dalam *database* web server dan bisa ditampilkan kembali apabila diakses (Madcom, 2008, p1). Berikut ini beberapa keunggulan yang dimiliki program PHP :

- PHP bersifat *free* atau gratis.
- Beberapa server seperti Apache, Microsoft IIS, PWS, AOLserver, phttpd, fhttpd, dan Xitami mampu menjalankan PHP.
- Beberapa *database* yang sudah ada baik yang bersifat gratis ataupun komersial sangat mendukung akses PHP, di antaranya MySQL, PostgreSQL, mSQL, informix, dan Microsoft SQL server.
- PHP mampu berjalan di linux sebagai platform sistem operasi utama bagi PHP, tetapi dapat juga berjalan di freeBSD, Unix, Solaris, dan yang lainnya.

2.6 Cascading Style Sheets (CSS)

Cascading Style Sheets (CSS) adalah sebuah standar yang didesain untuk memisahkan gambaran isi secara visual dan penjelasan struktural yang sebenarnya (Evans et.al., 2003, p585).

2.7 JQuery

Semenjak dirilis pertama kali pada tahun 2006 oleh John Resig, JQuery telah mencuri perhatian para developer web. Buktinya pada tahun-tahun berikutnya JQuery telah banyak digunakan oleh website terkemuka di dunia seperti Google, Microsoft, Intel, Nokia, Oracle, IBM, DELL, BBC, NBC, facebook, Amazon, Youtube dan masih banyak lagi.

JQuery adalah *Javascript Library* atau kumpulan kode/fungsi *Javascript* siap pakai, sehingga mempermudah dan mempercepat kita membuat kode *Javascript*. Secara standar apabila kita membuat kode *Javascript*, maka diperlukan kode yang panjang, bahkan kadang sangat sulit dipahami, disinilah peran JQuery sebagai

Javascript Library dimana kita bisa langsung memanggil fungsi yang terdapat didalam *Library* tersebut (Hakim, 2010, p3).

2.8 JavaScript

Javascript merupakan modifikasi dari bahasa pemrograman C++ dengan pola penulisan yang lebih sederhana, *interpreter* bahasa ini sudah disediakan ASP maupun Internet Explorer. *Javascript* terintegrasi dalam halaman HTML, *javascript* sendiri bukan merupakan bahasa pemrograman, melainkan satu bahasa *script* (Bollinger dan Natharajan, 2001, p32).

Javascript adalah bahasa *interpreter* yang tidak membutuhkan *tools* pendukung seperti *compiler* dan *debugger*. Untuk membuat *script* dan men-*debug script* tersebut yang diperlukan hanyalah sebuah *text editor* dan hanya sebuah *browser* yang mendukung *javascript*.

2.9 UML (*Unified Modeling Language*)

Menurut Martin Fowler and Kendall Scott (2000, p1) *Unified Modelling Language* atau biasa disebut dengan UML adalah sebuah "bahasa" pengganti sebuah gelombang dari metode *Objek-Oriented Analisis End Design(OOAND)* yang muncul di tahun 80an dan 90an. Secara keseluruhan menyatukan metode dari Booch, Rumbaugh (OMT), and Jacobson, tetapi pencapaian methodenya melebar dari yang diharapkan. UML menerobos melewati proses standarisasi dengan *OMG (Object Management Group)* dan sekarang sebuah standar *OMG*.

Menurut Bernd Oestereich (2002, p158) UML adalah suatu bahasa dan notasi (tanda) untuk spesifikasi, konstruksi, visualiasi dan dokumentasi dari suatu model sistem aplikasi.

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET.

2.9.1 Class Diagram

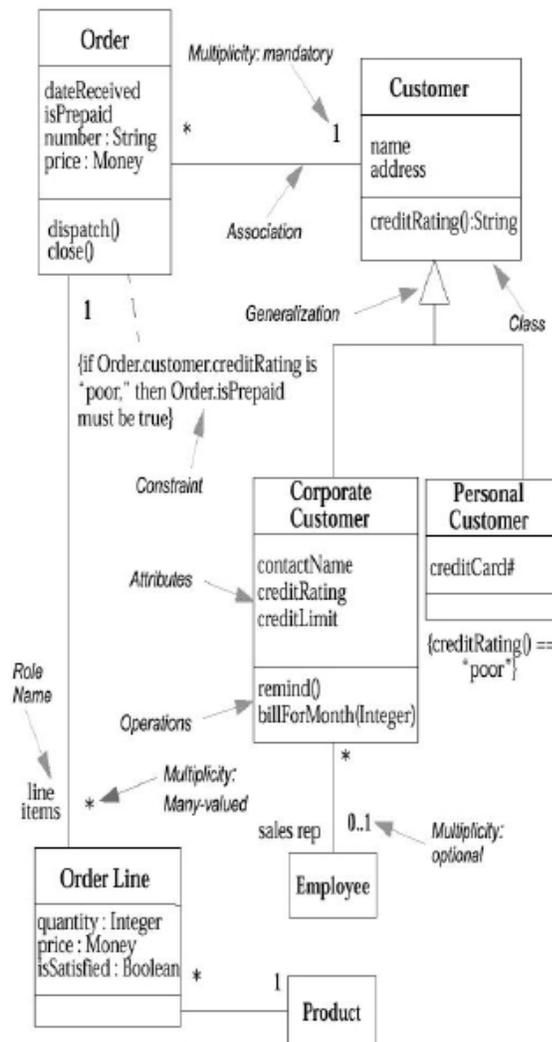
(Martin Fowler and Kendall Scott, 2000, p49) *Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class* memiliki tiga area pokok :

- Nama (dan *stereotype*)
- Atribut
- Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

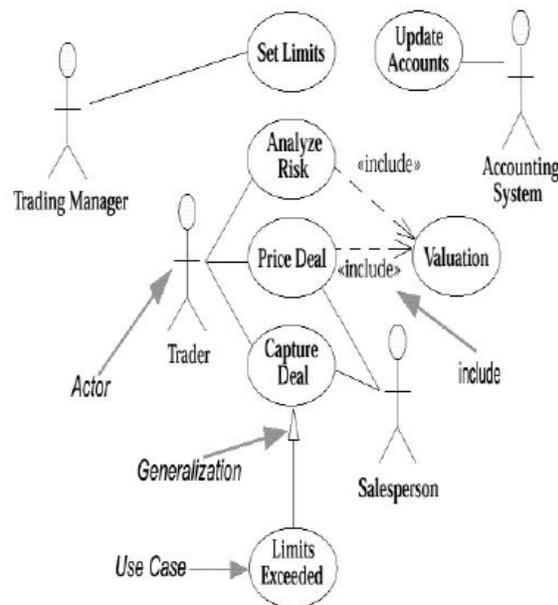
- *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan
- *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya
- *Public*, dapat dipanggil oleh siapa saja



Gambar 2.1 Class Diagram

2.9.2 Use Case Diagram

(Menurut Martin Fowler and Candle Scott, 2000, p39) *Use case* diagram adalah suatu kumpulan skenario yang disusun bersama dari tujuan user pada umumnya.

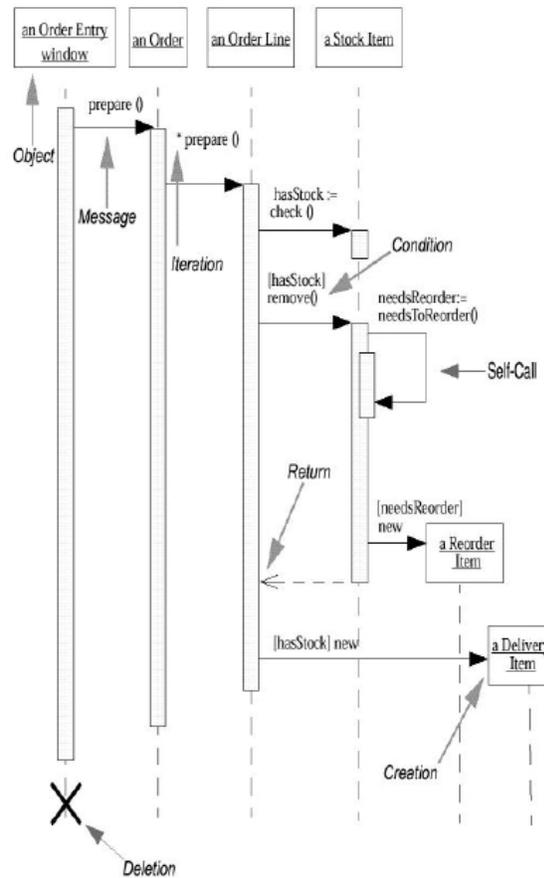


Gambar 2.2 Use Case Diagram

2.9.3 Sequence Diagram

(Martin Fowler and Kendall Scott, 2000, p67) *Sequence* diagram sebuah model yang menggambarkan bagaimana grup dari objek berkolaborasi di beberapa tindakan.

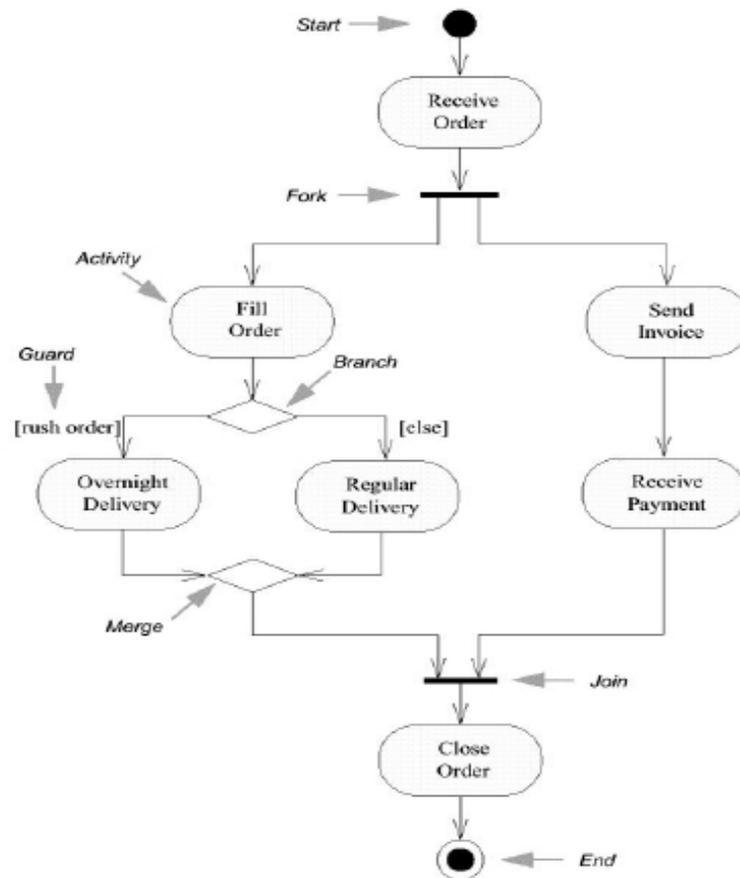
Pada *sequence diagram*, *objects* digambarkan berupa garis vertikal. Panah menunjukkan *message*. Bisa berupa sinkron *message* ataupun asinkron *message*. Urutan waktu berjalan dari atas ke bawah. (Bernd Oestereich, p261)



Gambar 2.3 Sequence Diagram

2.9.4 Activity Diagram

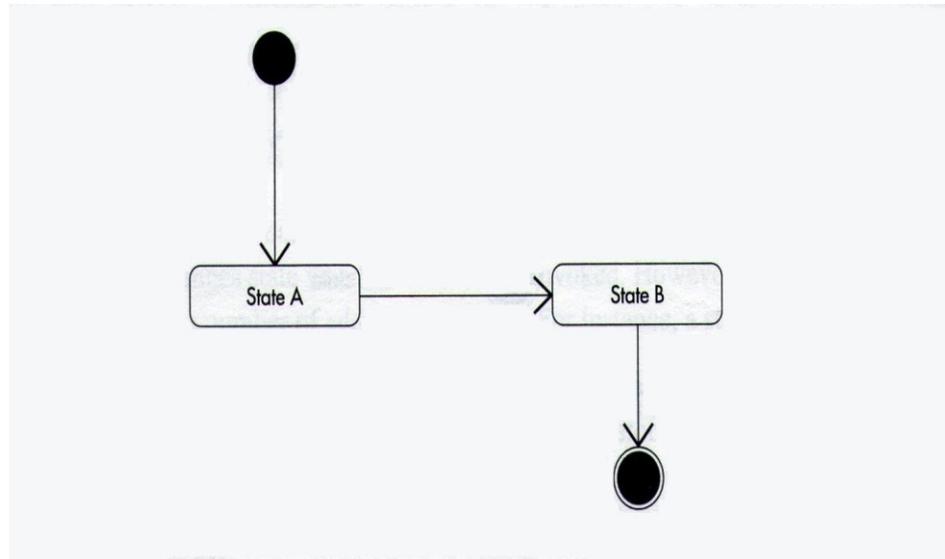
(Martin Fowler and Kendall Scott, 2000, p129) *Activity diagram* menggambarkan aliran aktivitas dalam sistem yang sedang dirancang. Bagaimana masing-masing aliran berawal, kondisi yang mungkin terjadi dan bagaimana akhirnya. *Activity diagram* juga menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi



Gambar 2.4 Activity Diagram

2.9.5 State chart

(Jeson T. Roff, 2003, p212) *Statechart* diagram digunakan untuk memodelkan *behavior / method (lifecycle)* sebuah kelas atau *object* dan memperlihatkan urutan kejadian sesaat(*state*) yang dilalui sebuah *object*, transisi dari sebuah *state* ke *state* lainnya.



Gambar 2.5 Statechart Diagram

2.10 Interaksi Manusia dan Komputer (IMK)

User interface desain adalah salah satu kriteria penting pada suatu sistem. *User interface* membantu bisnis dalam menarik pelanggan, serta menghasilkan kompetisi intensif, hak cipta desain, persaingan intelektual dan relasi internasional.

Pada tingkatan individual, *user interface* mengubah kehidupan banyak orang. *User interface* yang efektif dapat memberi keakuratan pada suatu pekerjaan. Dalam merancang *user interface*, ada kaidah yang harus dipatuhi, yaitu kaidah 8 aturan emas. (Schneiderman & Plaisant, 2010, p22).

2.10.1 Delapan Aturan Emas

Dalam Interaksi manusia dan computer ada 8 aturan emas yang harus diikuti agar tercapai tujuannya, yaitu :

a. Konsisten

Konsisten dilakukan pada urutan tindakan, perintah, dan istilah yang digunakan pada *prompt*, menu, serta layar bantuan.

b. *Shortcuts*

Ada kebutuhan dari pengguna yang sudah ahli untuk meningkatkan kecepatan interaksi, sehingga diperlukan singkatan, tombol fungsi, perintah tersembunyi, dan fasilitas makro

c. Umpan balik yang informatif

Untuk setiap tindakan, sebaiknya disediakan suatu sistem umpan balik. Untuk tindakan yang sering dilakukan dan tidak terlalu penting, dapat diberikan umpan balik yang sederhana. Tetapi ketika tindakan merupakan hal yang penting, maka umpan balik sebaiknya lebih substansial. Misalkan muncul suara ketika salah menekan tombol.

d. Adanya penutupan (keadaan akhir)

Urutan tindakan sebaiknya diorganisir dalam suatu kelompok dengan bagian awal, tengah, dan akhir. Umpan balik yang informatif akan memberikan indikasi bahwa cara yang dilakukan sudah benar dan dapat mempersiapkan tindakan berikutnya

e. Memberikan penanganan kesalahan yang sederhana

Jika kesalahan terjadi, sistem dapat mendeteksi kesalahan dengan cepat dan memberikan mekanisme yang sederhana dan mudah dipahami untuk penanganan kesalahan.

f. Pembalikan aksi

Hal ini dapat mengurangi kekawatiran *user* karena *user* mengetahui kesalahan yang dilakukan dapat dibatalkan.

g. Pusat kendali internal

User ingin menjadi pengontrol sistem dan sistem akan merespon tindakan yang dilakukan *user* daripada *user* merasa bahwa sistem mengontrol *user*.

h. Mengurangi beban ingatan jangka pendek

Memberi tampilan sederhana atau banyak tampilan yang sebaiknya disatukan, serta diberikan cukup waktu untuk kode dan urutan tindakan.

2.11 Rekayasa Perangkat Lunak (RPL)

Software engineering adalah pembentukam dan penggunaan teknik dasar untuk menciptakan suatu perangkat lunak yang reliable (dapat dipercaya) dan bekerja secara efisien.

Menurut R.S Pressman (2005, p53), rekayasa piranti lunak adalah penerapan dan pemakaian prinsip rekayasa dalam rangka mendapatkan piranti lunak ekonomis yang terpercaya dan bekerja secara efisien pada mesin

computer.Rekayasa piranti lunak ,mencakup 3 elemen yang mampu mengontrol proses perkembangan piranti lunak, yaitu:

a. Metode

Metode merupakan cara-cara teknis membangun piranti lunak yang terdiri dari perancangan proyek dan estimasi, analisis kebutuhan sistem dan piranti lunak, perancangan struktur data, arsitektur program, prosedur algoritma, pengkodean, pengujian, dan pemrograman.

b. Alat-alat bantu

Alat-alat bantu menyediakan dukungan otomatis atau semi-otomatis untuk metode-metode seperti *computer-aided software engineering* (CASE) yang mengkombinasikan piranti lunak dan piranti keras dan software engineering database (tempat penyimpanan yang mengandung informasi yang penting tentang analisis, perancangan, pembuatan program, dan pengujian).

c. Prosedur-prosedur

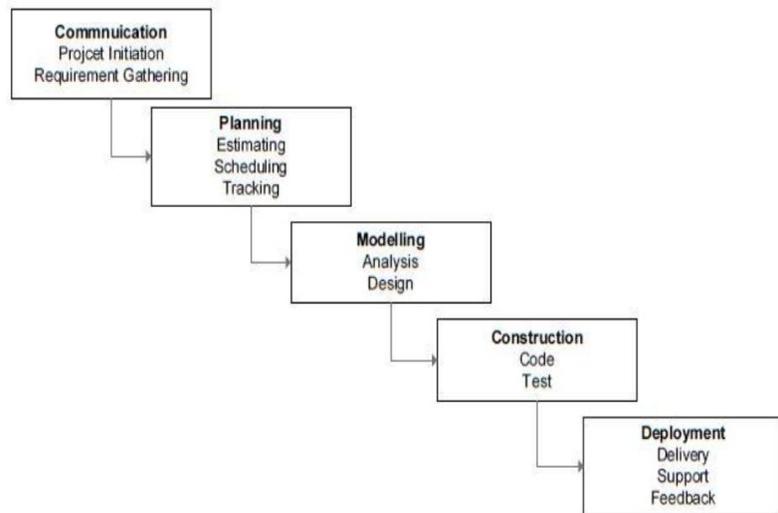
Prosedur-prosedur untuk menghubungkan alat-alat bantu dengan metode.Tujuan dari prosedur itu untuk mendapatkan piranti lunak yang efisien, berguna, dan ekonomis.

2.11.1 Model Proses Piranti Lunak

Menurut R.S Pressman (2001, p29-30) terdapat sebuah model proses yang dapat digunakan dalam merancang sebuah piranti lunak dimana perancangan dilakukan secara langsung dari tahap yang satu ke tahap selanjutnya, yaitu *Waterfall model*.

2.11.1.1 *Waterfall Model*

Waterfall paradigm yang paling tua dalam rekayasa piranti lunak. *Waterfall* model kadang disebut juga sebagai “*classic life cycle*” yang menyarankan sebuah sistematis, pendekatan yang beruntun pada perkembangan perangkat lunak yang dimulai dengan spesifikasi pelanggan pada kebutuhan dan memproses penjadwalan, pemodelan, konstruksi dan pendistribusian dimana titik puncak berpusat pada dukungan terhadap perangkat lunak yang telah selesai (Pressman, 2005, p79).



Gambar 2.6 *Waterfall*

Secara umum tahapan-tahapan dalam model *waterfall* yaitu :

a. Analisis dan perancangan sistem

Piranti lunak selalu merupakan bagian dari sistem yang lebih besar, maka aktifitas ini dimulai dengan penetapan kebutuhan dari semua elemen sistem dan mengalokasikan bagian dari kebutuhan ini ke piranti lunak. Gambaran sistem ini penting jika perangkat lunak harus berinteraksi dengan elemen-elemen lain seperti perangkat keras, manusia dan basis data.

b. Analisis kebutuhan piranti lunak

Proses pengumpulan kebutuhan khususnya ditingkatkan dan difokuskan pada piranti lunak. Untuk memahami program aslinya yang harus dibuat, seorang analis harus mengerti domain informasi untuk piranti lunak, seperti fungsi, kinerja dan antar muka yang dibutuhkan dengan peninjauan bersama pengguna.

c. Desain

Desain piranti lunak sebenarnya adalah proses bertahap yang berfokus pada empat atribut yang berbeda, terdiri dari : struktur program data, arsitektur piranti lunak, representasi antar muka, dan detil secara prosedur (algoritma). Proses desain menterjemahkan kebutuhan menjadi representasi piranti lunak yang dapat dinilai kualitasnya sebelum pengkodean dimulai. Seperti kebutuhan, desain dibuktikan dan menjadi bagian dari konfigurasi piranti lunak.

d. Kode

Aktivitas yang dihasilkan adalah memindahkan hasil perancangan menjadi suatu bentuk yang dapat dimengerti oleh mesin yaitu dengan membuat program.

e. Testing

Tahap pengujian dapat dilakukan agar keluaran yang dihasilkan oleh program sesuai dengan yang diharapkan. Pengujian dilakukan secara menyeluruh sampai semua perintah dan fungsi telah diuji.

f. Perawatan (*Maintenance*)

Suatu program yang telah selesai dibuat, mungkin saja masih memerlukan sedikit perubahan sesuai dengan lingkungannya. Untuk itu diperlukan suatu penyesuaian-penyesuaian agar program itu dapat dimanfaatkan dengan optimal.

Beberapa alasan diperlukannya tahap pemeliharaan ini antara lain :

1. Ditemukannya kesalahan-kesalahan (*error*) baru.
2. Perbedaan lingkungan program.
3. Pemakai menginginkan peningkatan fungsi.

2.12 Pseudocode

Pseudocode adalah teknik untuk menuliskan algoritma secara high level tanpa bergantung pada sebuah bahasa pemrograman. Penulisan pseudocode tidak memiliki aturan baku, biasanya dituliskan dalam bahasa inggris dan sedikit bantuan notasi matematika. Teknik ini bersifat lebih terstruktur daripada bahasa inggris biasa, tetapi tidak lebih detil daripada sebuah program (Setiadi, 2009, p4).