

BAB 2

LANDASAN TEORI

2.1 *Internet*

Internet adalah singkatan dari *Interconnection Networking*, atau sering disebut juga sebagai *cyberspace*. Menurut Ellsworth (1997, p.3), bahwa *internet* adalah jaringan komputer terbesar saat ini, yang menghubungkan jutaan komputer di seluruh dunia. Seorang pengguna *internet* dapat mengakses informasi dan berhubungan secara langsung dengan pengguna internet di seluruh dunia.

Menurut Meleod (2000,p.73), *Internet* memungkinkan suatu jaringan komunikasi global yang tidak hanya menghubungkan para mitra dangang tetapi juga mencakup para pelanggan. Sebagian besar kebangkitan perdagangan melalui jaringan elektronik diharapkan berasal dari perusahaan-perusahaan yang akan mempromosikan, dan dalam beberapa kasus mengirimkan produk mereka melalui internet.. *Internet* dapat digunakan untuk aplikasi bisnis apapun yang melibatkan komunikasi data, termasuk komunikasi baik di dalam perusahaan maupun dengan lingkungan.

2.2 *Website*

Website adalah keseluruhan *web page* beserta *homepage* (dimana *homepage* merupakan suatu halaman pertama dari kumpulan halaman *web* dan *file-file* pada situs *web* tersebut) yang merupakan system yang luas dari *server* yang menawarkan semua orang melalui jaringan (Hahn,1996, p.181).

2.3 *Electronic Commerce (E-commerce)*

2.3.1 *Pengertian E-Commerce*

E-commerce pada dasarnya merupakan suatu kontak transaksi perdagangan antara penjualan dan pembelian dengan teknologi internet. Jadi, sebagian proses dikomunikasikan dengan Internet (Ustandiyanto,2002,p11).

E-commerce menggambarkan proses membeli dan menjual lewat media digital (Turban,2004,p3).

E-commerce adalah proses pembelian dan penjualan serta pemasaran dan pelayanan dari produk, jasa, dan informasi melalui jaringan komputer yang beragam (O'Brien, 2003, p.23).

Menurut Rayport (2000, p.3), *e-commerce* digambarkan dengan beberapa atribut, yaitu :

1. Pertukaran informasi digital antara beberapa kelompok (*party*).

Pertukaran informasi ini dapat menggambarkan komunikasi antara dua kelompok, koordinasi aliran barang-barang dan layanan, atau transmisi pemesanan secara elektronik. Pertukaran ini juga terjadi antara organisasi atau antara individu.

2. Penggunaan teknologi.

e-commerce bersangkutan dengan transaksi yang menggunakan teknologi. Penggunaan *internet browser* sebagai *interface* dalam *World Wide Web* (WWW), mungkin merupakan penggunaan teknologi yang sudah lazim diketahui oleh para *customer*. Bahwa, *interface* lainnya (seperti ATM, *Elektronik Data Interchange* – EDI antara rekan bisnis,

dan perbankan secara elektronik melalui telepon) telah merupakan kategori umum dalam *e-commerce*. Dalam bisnis yang mengatur transaksi secara tetap antara *customer* dan pasar adalah interaksi manusia. Dalam *e-commerce*, beberapa transaksi dapat diatur menggunakan teknologi.

3. Didalamnya terdapat aktivitas intra- dan inter- organisasi yang mendukung pertukaran.

Cakupan dari *e-commerce* termasuk dalam semua aktivitas yang berbasis elektronik baik itu intra- maupun inter- organisasi yang mendukung secara langsung maupun tidak langsung pertukaran dalam transaksi. Dalam hal ini berkaitan erat dengan bagaimana hubungan suatu organisasi bisnis dengan bagian luar dari sistem (seperti *customer*, pemasok, rekan kerja, dan lain-lain) serta bagai mana operasi internal yang dilakukan untuk mengatur aktivitas-aktivitas, proses-proses serta sistem yang ada.

2.3.2 Jenis-jenis E-Commerce

1. *Business to business* (B2B), yaitu kegiatan bisnis yang terjadi antar perusahaan atau produsen;
2. *Business to consumer* (B2C), yang terjadi pelelangan, perusahaan penjual jasa dan perusahaan retail online;
3. *Consumer to Business* (C2B), yaitu kegiatan bisnis yang terjadi di antara konsumen dan produsen;

4. *Government to Business* (G2B), yaitu kegiatan bisnis yang terjadi di antara pemerintah dan perusahaan;
5. *Government to Consumer* (G2C), yaitu kegiatan bisnis yang terjadi di antara pemerintah dan konsumen.

2.3.3 Kategori *E-Commerce*

E-commerce secara umum dapat dibagi menjadi 3 kategori yaitu (Whiteley, 2000, p4);

1. *Electronic markets*

Electronic markets merupakan penggunaan teknologi informasi dan komunikasi untuk menampilkan berbagai penawaran yang tersedia dalam segmen pasar, sehingga pembeli dapat membandingkan harga (dan atribut lainnya) dari berbagai penawaran yang ada dan membuat keputusan. Contoh umum dari *electronic markets* ialah system pemasaran tiket pesawat terbang.

2. EDI (*Electronic Data Interchange*)

EDI menyediakan system standar untuk pengkodean transaksi perdagangan sehingga mereka dapat dikomunikasikan secara langsung dari satu sistem computer ke sistem computer lainnya tanpa membutuhkan perintah tertulis, tagihan, penundaan dan kesalahan dalam penanganan media kertas. EDI digunakan oleh organisasi yang memakai transaksi biasa dalam jumlah besar. Salah satu sector dimana EDI digunakan secara ekstensif adalah pertukaran di dalam

supermarket yang menggunakan EDI untuk bertransaksi dengan *supplier* mereka.

3. *Internet commerce*

Teknologi komunikasi dan informasi juga dapat digunakan untuk periklanan dan membuat satu kali penjualan dalam daerah produk dan jasa yang besar. *E-commerce* jenis ini dilambangkan dengan pemakaian Internet secara komersial. Internet dapat digunakan untuk pembelian buku yang dapat dikirimkan melalui pos atau pemesanan tiket yang dapat dilakukan oleh klien kapan saja mereka inginkan. Hal ini dapat ditandai dengan Internet bukan sebagai satu – satunya teknologi yang digunakan untuk pelayanan jenis ini dan ini bukan satu – satunya kegunaan Internet dalam *e-commerce*.

2.3.4 Keuntungan *E-Commerce*

Menurut Purbo dan Wahyudi (2001, p.2-3), keuntungan yang dapat diambil dengan adanya *e-commerce* adalah :

- a) *Revenue Stream* (aliran pendapatan) baru yang mungkin lebih menjanjikan, yang tidak dapat ditemui di sistem transaksi tradisional
- b) Dapat meningkatkan *marke exposure* (pangsa pasar).
- c) Menurunkan *operating cost* (biaya operasional).
- d) Melebarkan jangkauan (*Global Reach*).
- e) Meningkatkan *supplier management*.
- f) Meningkatkan *supplier loyalty*.

- g) Memperpendek waktu produksi.
- h) Meningkatkan *value chain* (mata rantai pendapatan).

Menurut kami keuntungan *e-commerce* tersebut adalah sebagai berikut :

1. Bagi Konsumen : harga lebih murah, belanja cukup pada satu tempat.
2. Bagi Pengelola bisnis : efisiensi, tanpa kesalahan, tepat waktu
3. Bagi Manajemen : peningkatan pendapatan, loyalitas pelanggan.

2.3.5 Kerugian atau Resiko *E-Commerce*

Menurut Purbo dan Wahyudi (2001, p.2-3), resiko yang timbul dari penerapan *e-commerce* adalah :

1. Kehilangan segi financial secara langsung karena kecurangan.
2. Pencurian informasi rahasia yang berharga.
3. Kehilangan kesempatan bisnis karena gangguan pelayanan.
4. Penggunaan akses ke sumber oleh pihak yang tidak berhak.
5. Kehilangan kepercayaan dari *customer*.
6. kerugian yang tidak terduga

Menurut kami kerugian dari *e-commerce* adalah sebagai berikut:

1. Persoalan yang tidak biasa diselesaikan seperti nama domain, *copyright*, pajak dan biaya – biaya lainnya.
2. Kurangnya peraturan dari pemerintah nasional maupun internasional dan standar industry.

3. Banyak pembeli dan penjual yang menunggu *e-commerce* itu stabil sebelum mereka ikut berpartisipasi.
4. Persepsi bahwa *e-commerce* itu mahal dan tidak terjamin keamanannya.

2.4 *Database*

2.4.1 *Pengertian Database*

Menurut Kendall (2005, p444), *database* bukanlah sebuah kumpulan file – file, tetapi *database* merupakan sebuah sumber data pusat yang dapat dibagi (*share*) ke banyak pengguna (*user*) dalam berbagai macam aplikasi. Pusat dari *database* adalah *database management system* (DBMS), yang memungkinkan pembuatan, modifikasi, dan *update database*; pengambilan data; dan pembuatan laporan dan tampilan. Sedangkan orang yang meyakinkan *database* dapat mencapai tujuannya disebut *database administrator*.

Menurut O'Brien (2003, p.145), *database* adalah koleksi yang terintegrasi dari elemen-elemen data yang berelasi secara logika. Sebuah *database* menyatukan banyak *record* yang telah disimpan sebelumnya, agar supaya sekumpulan data yang umum dapat digunakan untuk banyak aplikasi.

Menurut Date (2000,p.11), *database* merupakan sekumpulan data yang tetap yang digunakan oleh sistem aplikasi dari perusahaan.

Tujuan utama penggunaan konsep *database* (McLead,2001,p.182):

1. Mengurangi data *redundancy*

Data *redundancy* adalah duplikasi dari data, maksudnya adalah data yang sama disimpan berulang-ulang pada banyak *file*.

2. Meningkatkan data *independence*

Data *independence* adalah kemampuan untuk merubah struktur data tanpa harus mengubah aplikasi yang memproses data tersebut.

Database terdiri dari beberapa objek antara lain yaitu :

1. *Field*

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

2. *File*

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

Contoh: transaksi pada perusahaan yang *recordnya* tercatat tanggal, kode, barang dan harga.

3. *Record*

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

Contohnya: nama, alamat, nomor karyawan, nomor telepon, dan sebagainya.

4. *Entity*

Entity adalah satu kesatuan yang terdiri dari informasi yang disimpan.

Contohnya: pelanggan, karyawan, barang, dan sebagainya.

5. Attribute

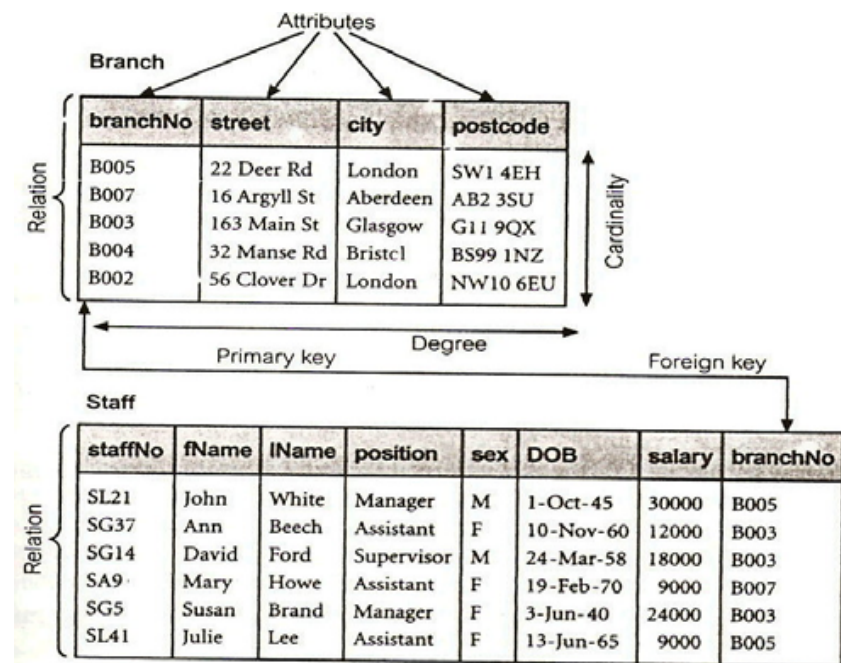
Attribute adalah nama dari suatu kolom relasi yang menjelaskan suatu *entity*. Contohnya nama, nomor karyawan, nomor telepon, dan sebagainya.

6. Primary Key

Primary Key adalah sebuah *field* yang mempunyai nilai unik yang tidak memiliki kesamaan antara *record* yang satu dengan *record* yang lain. Contohnya: nomor karyawan, kode barang, dan lain-lain.

7. Foreign Key

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



Gambar 2.1 Sistem Data Modelling

Tujuan efektivitas *database* antara lain meliputi :

1. Meyakinkan bahwa data dapat *dishare* di antara user dalam berbagai macam aplikasi.
2. Memelihara data secara tepat dan konsisten.
3. Meyakinkan bahwa seluruh data yang diperlukan untuk aplikasi yang ada saat ini dan di masa depan tersedia.
4. Memungkinkan *database* untuk berkembang sejalan dengan pertumbuhan *user*.
5. Memungkinkan *user* untuk menggunakan data sesuai pandangan pribadi mereka tanpa peduli dengan cara penyimpanan data secara fisik.

2.4.2 Entity Relationship Diagram (ERD)

Menurut Connolly (2002,p.330-351), *Entity Relationship Diagram* digunakan untuk menggambarkan struktur logical database dalam bentuk diagram ERD menyediakan cara yang sederhana dan mudah 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 1 (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* lainnya.

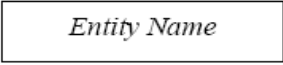
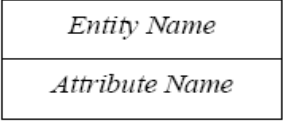
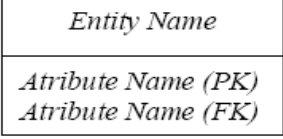
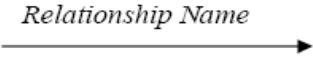
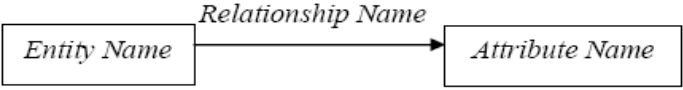
3. Properti / *atribut*

Properti atau *atribut* adalah sifat atau karakteristik deskriptif suatu entitas. Setiap properti atau atribut mempunyai *key* di antaranya *Primary Key* (PK) dan *Foreign Key* (FK) untuk menunjukkan hubungan antar kedua entitas.

Menurut Whitten, et.al.(2004, p.260-267), *Entity Relationship Diagram* adalah model aktual dari notasi pemodelan data yang menggambarkan data dalam bentuk *entities* dan *relationships*. Beberapa konsep model data adalah sebagai berikut:

- a. *Entity* adalah sekelompok orang, tempat, kejadian, atau konsep yang perlu ditangkap dan disimpan datanya.
- b. *Primary Key* adalah atribut tunggal atau jamak yang unik, digunakan untuk mengenali satu buah *entity* tunggal.
- c. *Foreign Key* adalah *primary key* dari sebuah *entity* yang diletakkan dalam *entity* lainnya untuk mengenali hubungan kedua datanya.
- d. *Relationship* adalah asosiasi alami bisnis yang ada diantara satu *entity* atau lebih.

Tabel 2.2 Notasi UML untuk perancangan ERD

Notasi UML	Keterangan
	Entitas
	Entitas dengan atribut <i>primary key</i>
	Entitas dengan beberapa atribut <i>Primary Key</i> diberi tanda PK; <i>Foreign Key</i> diberi tanda FK
	Relasi yang ditandai dengan nama relasi dan panah penunjuk
	<i>Binary Relationship</i>

2.4.3 *MySql*

Menurut anonim^[4], MySQL adalah sebuah perangkat lunak sistem manajemen basis data *SQL* atau dapat disebut juga *Database Management System* (DBMS) yang *multi-thread* dan *multi-user*. *Multi-thread* maksudnya *MySQL* merupakan perangkat lunak yang *open source*. Ukuran *database MySQL* lebih kecil file *database* yang lain.

Perintah-perintah dalam yang digunakan dalam dalam *MySQL* tidak *case sensitive*, jadi baik huruf kapital maupun huruf kecil tidak dibedakan. Setiap perintah harus diakhiri tanda *semi colon* (;).

Sintaks-sintaks bahasa My SQL yaitu :

1. *CREATE DATABASE[IF NOT EXIST] namadatabase*

Perintah ini digunakan untuk membuat sebuah *database*. Jika pernyataan *IF NOT EXIST* tidak disertakan, maka bila *database* dengan nama yang sama telah diinput sebelumnya pada server, akan terjadi *error*.

2. *DROPDATABASE [IF EXIST] namadatabase*

Perintah ini digunakan untuk menghapus sebuah *database*. Penggunaan perintah ini dengan bijaksana karena MySQL tidak akan memberi peringatan.

3. *CREATE DATABASE [IF NOT EXIST] namatabel(definisi)*

Perintah *create table* mempunyai sintaks yang cukup kompleks.

4. *ALTER [IGNORE] TABLE namatabel spesifikasi [spesifikasi]*

Digunakan untuk memodifikasi tabel yang pernah dibuat, pernyataan

IGNORE merupakan ekstensi dari ANSI SQL 92, yang digunakan untuk mengatur bagaimana perintah *ALTER TABLE* bekerja jika ada duplikat pada *primary key* sebuah tabel.

5. *DROP TABLE [IF EXISTS] namatabel*

Perintah ini digunakan untuk menghapus sebuah tabel.

6. *INSERT [INTO]namatabel [(namafield,...)] perintah SELECT*

Perintah ini digunakan untuk menyisipkan suatu data ke dalam tabel.

7. *SELECT {*/namafield}FROM namatabel [INTO table*

tujuan][WHERE kondisi] Perintah ini digunakan untuk mengambil data dari suatu table. Tanda bintang (*) menunjukkan bahwa semua *field* yang ada telah dipilih.

8. *DELETE FROM namatabel WHERE kondisi*

Perintah ini digunakan untuk menghapus sebuah *record* dari tabel.

9. *UPDATE namatabel SET kriteria WHERE kondisi*

Perintah ini digunakan untuk memperbaharui nilai suatu data.

10. *REPLACE*

Perintah *REPLACE* hampir sama dengan *INSERT*, perbedaannya jika *record* lama pada tabel memiliki nilai yang sama dengan *record* baru, maka *record* lama akan dihapus dan diganti dengan *record* yang baru.

11. *USE namatabel*

Perintah ini digunakan untuk memilih *database* yang akan digunakan.

12. *SHOW DATABASES [LIKE kondisi]*

Perintah ini digunakan untuk menampilkan informasi tentang *database* yang sedang digunakan.

2.5 Security

Dalam pengembangan aplikasi berbasis web dengan menggunakan PHP kerap kali kita menggunakan koneksi menggunakan database. Dalam menghubungkan kita menuliskan perintah koneksi menggunakan query dan perintah lainnya dalam query pula tentunya. Tapi terkadang tanpa disadari rentannya suatu situs diakibatkan penggunaan query yang tidak menggunakan keamanan yang baik. Kerentanan pada query salah satunya dengan melakukan injeksi pada query di dalam source code.

Dikerenakan query yang dilewatkan oleh PHP ke database MySQL ditulis layaknya SQL, maka melakukan kemungkinan resiko untuk orang lain melakukan injeksi SQL menggunakan parameter di MySQL dikenal sebagai SQL injection. Dengan memasukan malicious code melalui parameter di form, maka kemungkinan bagi attacker mematahkan keamanan server Anda (Anonim2).

2.6 Interaksi Manusia dan Komputer

Interaksi manusia komputer dalam bahasa Inggris, *Human Computer Interaction* (HCI) adalah suatu ilmu yang mempelajari bagaimana interaksi antara manusia dengan komputer dan pengaruh komputer dalam pengembangannya untuk berinteraksi dengan manusia. (Schneiderman, 2005, p8).

Interaksi manusia dengan computer melibatkan perancangan antara muka pemakai yang memiliki 8 faktor yang harus diperhatikan, yaitu:

1. Berusaha keras untuk konsisten.

Ada beberapa bentuk konsistensi, diantaranya adalah konsisten dalam penggunaan bentuk dan ukuran *font*, pemberian warna tulisan dan latar

belakang, penggunaan *layout*, dan penggunaan terminology. Konsistensi dapat memudahkan pemakai dalam menggunakan dan menjalankan aplikasi, mempelajari hal yang baru dengan lebih cepat dan lebih dapat memfokuskan pada tugas karena mereka tidak memerlukan waktu untuk mengingat perbedaan – perbedaan dalam interaksi.

2. Memungkinkan *frequent user* menggunakan *shortcuts*.

Jika suatu tindakan sering dilakukan, pemakai tentunya ingin agar jumlah interaksi untuk melakukannya dikurangi untuk meningkatkan kepraktisannya. *Shortcut* seperti tombol – tombol khusus, *hidden commands* dan fasilitas makro akan disukai oleh *frequent knowledgeable users*, karena dapat memberikan waktu respon dan waktu tampilan yang cepat.

3. Memberikan umpan balik yang informatif.

Sistem harus selalu menyediakan umpan balik untuk setiap aksi pemakai. Isyarat – isyarat seperti suara dan tampilan *visual* harus ditampilkan untuk setiap interaksi pemakai yang dapat digunakan untuk mengkonfirmasi bahwa piranti lunak memberikan dari suatu tampilan yang mati atau suatu tampilan yang tidak memberikan respon apapun terhadap masukan.

4. Merancang dialog untuk menghasilkan keadaan akhir.

Urutan-urutan tindakan harus terorganisir kedalam kelompok – kelompok dengan permulaan, pertengahan dan akhir yang jelas. Umpan balik yang menginformasikan bahwa sekelompok tindakan yang telah selesai dilakukan akan memberi keputusan kepada pemakai, perasaan lega, dan perasaan tiap untuk melakukan tindakan selanjutnya.

5. Meberikan penanganan kesalahan yang sederhana.

Walaupun dengan rancangan *interface* yang terbaik, pemakai tetap saja dapat berbuat kesalahan. Oleh sebab itu sedapat mungkin system dirancang agar pengguna tidak dapat membuat error yang serius. Jika error terjadi, sistem harus mendeteksi dengan meberikan penawaran mekanisme penanganan kesalahan yang sederhana dan harus dimengerti oleh pemakainya.

6. Mengizinkan pengembalian aksi (*undo*) dengan mudah.

Sedapat mungkin setiap tindakan pemakai dapat di-*undo*. Hal ini dapat memberikan kelegaan dan keberaniaan kepada pemakai untuk menelurusi pilihan – plihan yang tidak dikenalnya.

7. Pengontrolan terletak pada pemakai sendiri.

Pemakai harus merasa bahwa ia berkuasa atas sistem dan sistem harus menuruti perintahnya. Tindakan sistem yang mengejutkan, data yang rumit, kesulitan dalam meperoleh informasi yang penting, dan ketidak mampuan untuk melakukan tindakan yang diinginkan akan membuat kecemasan dan ketidakpuasan.

8. Mengurangi hafalan pada pemakai.

Karena keterbatasan manusia dapat menghafal, maka tampilan sebaiknya dibuat sesederhana mungkin, yaitu di mna informasi –informasi yang ada dikelompokan sesuai jenisnya, frekuensi gerakan *window* dikurangi dan diberikan waktu latihan yang cukup. Jika memungkinkan, dibuatkan akses *online* untuk mencari istilah, singkatan maupun urutan aksi yang sering dipakai namun sulit untuk dihafal.

2.7 Rekayasa Perangkat Lunak

2.7.1 Definisi Rekayasa Perangkat Lunak

Definisi rekayasa perangkat lunak menurut Fritz Bauer adalah pengembangan dan penggunaan prinsip pengembangan suara untuk memperoleh perangkat lunak secara ekonomis yang *reliable* dan bekerja secara *efisien* pada mesin nyata (Pressman, 2002, p28).

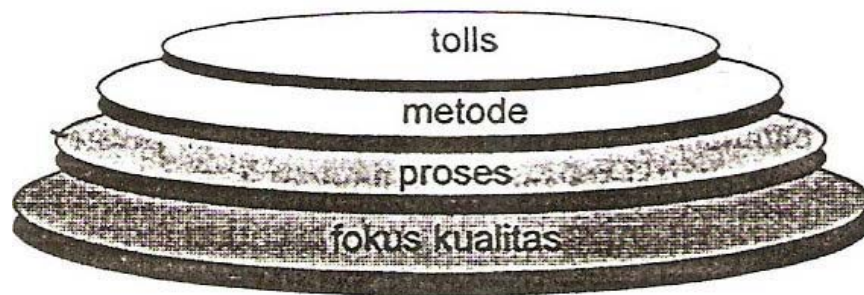
Definisi tersebut telah dikembangkan menjadi lebih *komprehensif* oleh IEEE (*Institute of Electrical and Electronics Enginners*) sebagai berikut: (1) Aplikasi dari sebuah pendekatan kuantifiabel, disiplin, dan sistematis kepada pengembangan, operasi, dan pemeliharaan perangkat lunak; yaitu aplikasi dari rekayasa perangkat lunak. (2) Studi tentang pendekatan-pendekatan seperti pada (1).

2.7.2 Lapisan RPL

Secara umum Rekayasa Perangkat Lunak dapat dibagi menjadi tiga *layer* (Pressman, 2001, p.23-24), antara lain :

1. *Process Model* adalah fondasi dari RPL yang mendefinisikan sebuah *framework* untuk sekumpulan *key process area* yang harus dibangun demi keefektifan penyampaian teknologi pengembangan RPL.
2. *Methods* menyediakan secara teknis bagaimana untuk membangun suatu perangkat lunak.
3. *Tools* menyediakan dukungan otomatis dan semi otomatis untuk *process model* dan *methods*.
4. *Quality focus* merupakan batu landasan yang menopang

tools, methods, dan process dalam RPL



Gambar 2.3 Lapisan Rekayasa Perangkat Lunak

2.7.3 Model Proses *Software*

Menurut Pressman (2001, p.29-47), ada beberapa model proses *software* yang umum digunakan, salah satunya adalah Model Sekuensial Linear Model Sekuensial Linear ini juga dikenal dengan nama “*ClassicLife Cycle*” atau “*Waterfall Model*”. Model ini melingkupi aktivitas-aktivitas sebagai berikut :

1. Rekayasa dan Pemodelan Sistem Informasi

Rekayasa dan pemodelan sistem informasi diperlukan karena perangkat lunak selalu merupakan bagian dari sebuah sistem yang lebih besar, kerja dimulai dengan membangun syarat dari semua elemen sistem dan mengalokasikan beberapa subset dari kebutuhan ke perangkat lunak tersebut. perangkat lunak harus berhubungan dengan element-element yang lain seperti perangkat lunak, manusia, dan *database*.

2. Analisis kebutuhan perangkat lunak.

Proses pengumpulan kebutuhan diintensifkan dan difokuskan, khususnya pada perangkat lunak. Untuk memahami sifat program yang dibangun, seorang perancang perangkat lunak harus memahami kebutuhan informasi, fungsi-fungsi, unjuk kerja, dan *interface* yang diperlukan.

3. Perancangan.

Design perangkat lunak sebenarnya adalah proses multi langkah yang berfokus pada empat *attribute* sebuah program yang berbeda : struktur data, arsitektur perangkat lunak, representasi *interface*, dan detail (algoritma).

4. Pengkodean.

Design harus diterjemahkan menjadi bentuk yang bisa dibaca atau dimengerti oleh komputer, biasanya dalam bentuk bahasa pemrograman.

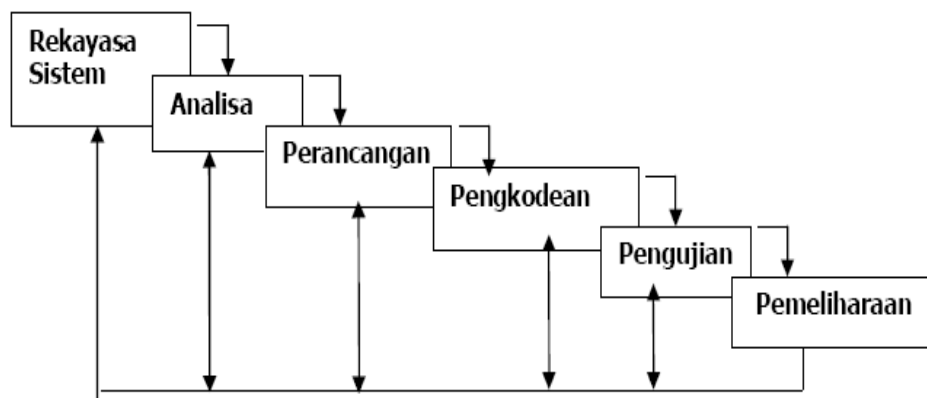
5. Pengujian.

Sekali kode dibuat, pengujian program juga dimulai. Proses pengujian berfokus pada logika internal perangkat lunak, memastikan bahwa semua pernyataan sudah diuji, dan pada eksternal fungsional yaitu mengarahkan pengujian untuk menemukan kesalahan-kesalahan dan memastikan bahwa *input* yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan.

6. Pemeliharaan.

Perangkat lunak akan mengalami perubahan setelah disampaikan kepada pelanggan. Perubahan akan terjadi karena kesalahan-

kesalahan ditentukan, karena perangkat lunak harus disesuaikan untuk mengakomodasikan perubahan-perubahan didalam lingkungan eksternalnya, atau pelanggan membutuhkan perkembangan fungsional atau untuk kerja. Pemeliharaan perangkat lunak mengaplikasi lagi setiap fase program sebelumnya dan tidak membuat yang baru lagi.



Gambar 2.4 Waterfall Model

2.8 Unified Modelling Language(UML)

Selama tahun 1990an banyak metodologi *modelling language* yang berkembang. Setiap *modelling language* tersebut mempunyai perbedaan dan keunggulan masing-masing. Menurut Jeffrey L. Whitten (2002, p.646), Pada saat itu ada tiga *modelling language* yang populer digunakan, yaitu :

1. Object Modelling Technique (OMT) yang dikembangkan oleh James Rumbaugh. *Modelling language* ini baik dalam analisis tetapi lemah dalam desain.
2. Metode Booch dikembangkan oleh Grady Booch tahun 1991 ini baik dalam

desain tetapi lemah dalam analisis.

3. *Object Oriented Software Engineering* (OOSE) dikembangkan oleh Ivar Jacobson. *Modelling language* baik dalam analisis tetapi lemah dalam bidang yang lain.

Pada tahap selanjutnya tiap *modelling language* saling mengadopsi kelebihan dari *modelling language* yang lain untuk memperbaiki kelemahannya. Sampai pada akhirnya muncul suatu *modelling language* baru yang merupakan gabungan dari semua kelebihan *modelling language* yang ada. *Modelling language* baru ini disebut *Unified Modelling Language* (UML) versi 1.0 yang dirilis tahun 1997.

Menurut anonim^[3], UML (*Unified Modelling Language*) adalah spesifikasi bahasa standar untuk permodelan objek. UML adalah bahasa permodelan umum yang meliputi sebuah notasi grafik yang digunakan untuk membuat suatu model abstrak dari suatu sistem.

UML is a standard untuk menspesifikasi, memvisualisasi, membangun, dan mendokumentasi artifak-artifak dari sistem *software*, sama baiknya sebagai permodelan bisnis dan non sistem *software*.

UML merupakan satu kumpulan konvensi permodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek. Kegunaan UML sebagai bahasa grafikal standar untuk model *software* berorientasi objek. UML berisi beberapa tipe diagram, meliputi :

1. *Use-case*

pemodelan fungsi-fungsi sistem dalam konteks peristiwa-peristiwa bisnis, siapa yang mengawalinya, dan bagaimana sistem itu merespon hal tersebut.

2. *Interaction diagram*

Interaction diagram terdapat dua tipe yaitu : *Sequence and Collaboration Diagrams*, dimana kedua diagram ini menunjukkan proses dari sistem dalam kaitannya dengan bagaimana interaksi antar objek.

3. *Activity Diagrams*

Menunjukkan bagaimana sistem bertindak secara internal.

2.8.1 *Use-Case Diagram*

2.8.1.1 *Pengertian Use-Case Diagram*

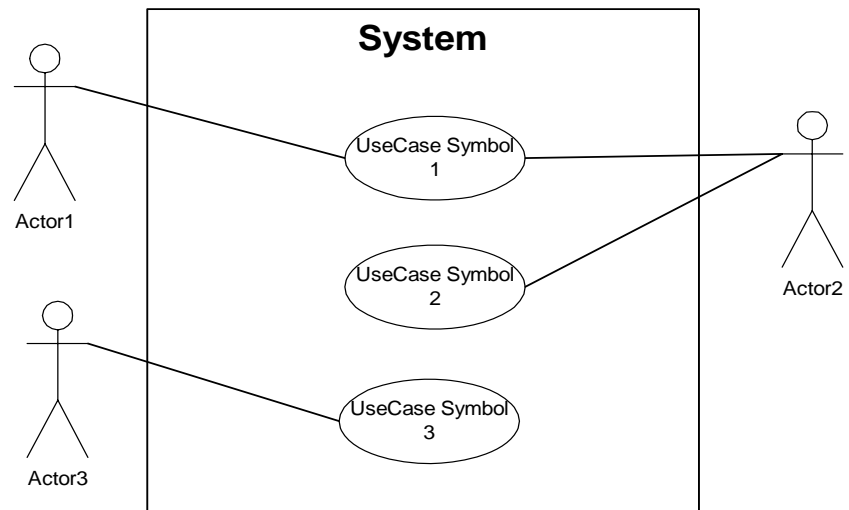
Pemodelan *Use-Case* adalah proses pemodelan fungsi-fungsi sistem dalam konteks peristiwa-peristiwa bisnis, siapa yang mengawalinya, dan bagaimana sistem itu merespon hal tersebut. Dan Pemodelan *Use-Case* terbukti menjadi alat bantu yang sangat berharga dalam menghadapi tantangan untuk menentukan apa yang harus dilakukan oleh sistem menurut perspektif pengguna dan *stakeholder*.

Use-Case Diagram merupakan salah satu alat utama yang digunakan saat menyajikan permodelan *use-case*. *Use-Case Diagram* menggambarkan fungsi-fungsi sistem dari sudut pandang pengguna eksternal dalam sebuah cara dan terminologi

yang mereka pahami dimana urutan langkah-langkah yang secara tindakan saling terkait, baik secara otomatis maupun manual, untuk tujuan melengkapinya satu tugas bisnis tunggal.

Pembuatan *Use-Case* secara luas dikenal sebagai aplikasi teknik terbaik dalam menentukan, mendokumentasikan, dan memahami persyaratan fungsional sistem informasi.

Use-Case juga bisa dijelaskan sebagai notasi UML yang menampilkan hubungan antara suatu kumpulan kasus dan beberapa actor. *Use-Case* membantu *software engineer* untuk menyampaikan tampilan gambar tingkat tinggi dari suatu sistem.



Gambar 2.6 Use-case diagram

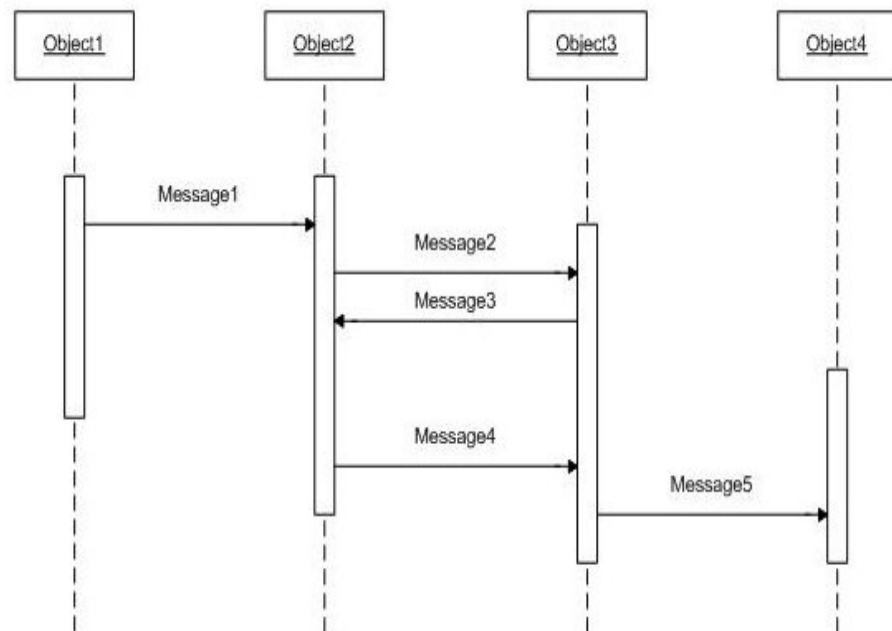
2.8.1.2 Deskripsi *Use-Case*

Sebuah *Use-Case* harus menjelaskan interaksi *user* dengan sistem, tidak hanya memperhitungkan penampilan dari sistem saja.

1. Nama use-case (*use-case*), merupakan nama dari suatu *use-case*, memberikan suatu singkatan.
2. Aktor (*actor*), pelaku yang berpartisipasi di dalam suatu *use-case*.
3. Tujuan (*goals*), menjelaskan apa yang akan dicapai oleh aktor(aktor).
4. Prakondisi (*precondition*), batasan pada keadaan sistem sebelum *use-case* dapat dieksekusi.
5. Ringkasan (*summary*), deskripsi ringkasan pendek yang berisi kegiatan yang dilakukan oleh aktor (*actor*).
6. *Related use-case*, *use-case* yang saling berhubungan.
7. Langkah-langkah (*steps*), menjelaskan setiap langkah dari *use-case* dengan menggunakan dua kolom, dimana kolom sebelah kiri menunjukkan aksi dari pelaku dan kolom sebelah kanan merupakan respon yang diberikan oleh sistem.
8. *Postconditions*, sistem yang mengikuti penyelesaian dari *use-case*.

2.8.2 Sequence Diagram

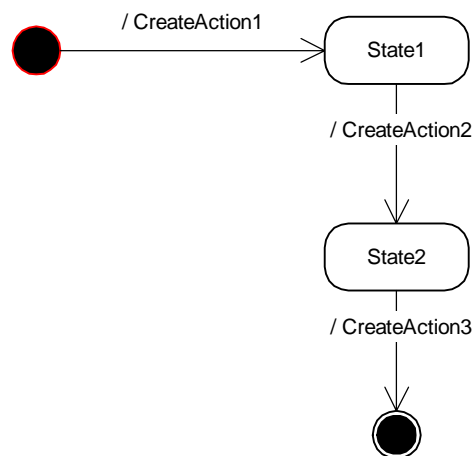
Secara grafis menggambarkan bagaimana objek berinteraksi satu dengan yang lain melalui pesan pada eksekusi sebuah *use-case*. Diagram ini mengilustrasikan bagaimana pesan terkirim dan diterima di antara objek.



Gambar 2.7 Sequence Diagram

2.8.3 Activity Diagram

Statechart diagram digunakan untuk memodelkan perilaku dinamis satu kelas atau objek. *Statechart diagram* memperlihatkan urutan keadaan sesaat (*state*) yang dilalui sebuah objek, Kejadian yang menyebabkan sebuah transisi dari suatu *state* atau *aktivitas* kepada yang lainnya. *Statechart diagram* khusus digunakan untuk memodelkan tahap-tahap diskrit dari sebuah siklus hidup objek.



Gambar 2.8 Activity Diagram

2.9 Persediaan Barang

Menurut Sofjan Assuri (2004;169), Persediaan merupakan sejumlah bahan-bahan, *part* yang disediakan dan bahan-bahan dalam proses yang terdapat dalam perusahaan untuk proses produksi, serta barang-barang jadi/produk yang disediakan untuk memenuhi dari komponen atau langganan setiap waktu.

Menurut kami Persediaan adalah barang yang dimiliki untuk dijual atau untuk diproses selanjutnya dijual.

2.10 Sistem Informasi Penjualan

2.10.1 Sistem

Menurut McLeod(20004,p9), sistem adalah sekelompok elemen-elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan.

Menurut O'Brien (2003, p8), sistem adalah sekelompok komponen yang saling berhubungan, bekerja bersama untuk mencapai tujuan bersama dengan menerima input serta menghasilkan output dalam proses transformasi yang teratur.

2.10.2 Informasi

Menurut McLeod (2001, p12), informasi adalah data yang telah diproses atau data yang telah memiliki arti dan pengertian.

Menurut Turban et.al. (2003, p15), informasi adalah kumpulan dari fakta- fakta yang terorganisasi dengan cara-cara tertentu sehingga dapat memiliki arti bagi orang yang menerimanya.

Jadi menurut kesimpulan kami, informasi adalah kumpulan data yang telah diproses sedemikian rupa sehingga bermakna bagi orang yang menggunakannya.

2.10.3 Definisi Penjualan

Menurut Romney (2003, p.157), penjualan merupakan satu set rekursif dari kegiatan bisnis dan koperasi pemrosesan informasi terkait yang dihubungkan dengan penyediaan barang dan layanan pelanggan dan penerima pembayaran dari penjualan tersebut.

Sistem penjualan adalah suatu sistem usaha yang terpadu untuk mengembangkan rencana – rencana yang diarahkan pada usaha pemuasan kebutuhan dan keinginan pembeli. Guna mendapatkan penjualan yang menghasilkan laba (Anonim1).

2.10.4 Sistem Informasi Penjualan

Menurut kesimpulan kami, sistem informasi penjualan adalah sistem yang digunakan seseorang atau perusahaan untuk mengumpulkan, memproses, menyimpan, menganalisa, dan menyebarkan informasi dari kegiatan bisnis yang dihubungkan dengan penyediaan barang, pelayanan *customer*, dan pengecekan pembayaran.

2.11 Penjualan *Online*

Era perkembangan teknologi yang semakin pesat dewasa ini membuat persaingan bisnis dalam bidang apapun menjadi lebih variatif. Banyak metode bisnis yang dilakukan oleh sebuah perusahaan untuk menarik pelanggan sebanyak-banyaknya. Mulai dari memberi bonus, memasang iklan, membagikan brosur dan pemesanan produk dan jasa melalui media elektronik seperti internet.

memasuki era globalisasi ini, sistem penjualan *online* dilakukan untuk mempermudah dalam pemesanan barang yang ditawarkan. Pembeli dapat melihat tampilan gambar dan rincian barang tanpa harus datang ke tempat penjual.