

## **BAB 2**

### **LANDASAN TEORI**

#### **2.1 Teori Umum**

Teori – teori yang digunakan sebagai referensi pengetahuan dalam memahami hal – hal yang berkaitan dengan sistem informasi dan komponen – komponennya.

##### **2.1.1 Sistem Informasi**

Sistem informasi sangat berperan penting dalam perkembangan bisnis di zaman teknologi ini guna untuk meningkatkan produktivitas. Berikut penjabaran pengertian sistem informasi :

##### **❖ Pengertian Sistem**

Menurut Satzinger, Jackson & Burd (2005:7) sistem adalah kumpulan dari beberapa komponen yang saling berhubungan dan berkerja bersamaan untuk mencapai suatu hasil.

Menurut O'Brien (2005:22), sistem dapat didefinisikan secara sederhana sebagai sekelompok elemen yang saling berhubungan atau berinteraksi hingga membentuk satu kesatuan. Sistem adalah sekelompok komponen yang saling berhubungan, bekerja bersama untuk mencapai tujuan bersama dengan menerima input serta menghasilkan output dalam proses transformasi yang teratur. Sistem sejenis ini (kadang disebut sebagai sistem yang dinamis) memiliki tiga komponen atau fungsi dasar yang berinteraksi:

1. Input, meliputi pengambilan dan perakitan elemen yang memasuki sistem untuk diproses.
2. Processing, meliputi proses transformasi yang mengubah dari input menjadi output.
3. Output, meliputi pengiriman elemen yang telah dihasilkan dari proses transformasi ke tujuan utama.

### ❖ **Data**

Menurut Connolly & Begg (2010:70), data merupakan komponen yang paling penting dalam *Database Management System* ( DBMS ), berasal dari sudut pandang dari *end-user*. Data berperan sebagai penghubung antara mesin dengan pengguna.

### ❖ **Pengertian Informasi**

Menurut O'brien (2005:38) informasi adalah data yang telah diubah menjadi konteks yang berarti dan berguna bagi pemakai akhir tertentu.

Menurut Rainer, Turban & Potter (2007:5) informasi adalah data yang telah diorganisir sehingga memiliki makna dan nilai kepada penggunanya.

### ❖ **Pengertian Sistem Informasi**

Menurut Satzinger, Jackson & Burd (2005:7) sistem informasi adalah kumpulan dari beberapa komponen yang saling berhubungan yang berfungsi untuk mengumpulkan, memproses, menyimpan, dan menghasilkan output informasi yang dibutuhkan untuk menyelesaikan- kan tugas bisnis.

O'brien (2005:5) sistem informasi adalah kombinasi teratur apapun dari orang-orang, hardware, software, jaringan komunikasi dan sumber daya data yang mengumpulkan, mengubah dan menyebarkan sistem informasi dalam sebuah organisasi.

Menurut Suci, Purnama dan Sukadi (2011:234) sistem informasi Pendaftaran Siswa Baru adalah adalah sistem sekolah yang mengelola data calon siswa baru dengan cara memasukkan data tersebut ke dalam sistem dan menghasilkan laporan data calon siswa.

Menurut Sudarmadji (2008:161) Sistem informasi berbasis *web* sangat membantu dan mempercepat proses pendaftaran masuk sekolah serta dapat menyajikan informasi yang dibutuhkan secara cepat kepada calon siswa baru.

- Komponen Sistem Informasi Menurut O'Brien (2005:34), yaitu:
  - **People**, Manusia dibutuhkan untuk pengoperasian semua sistem informasi. Sumber daya manusia ini meliputi pemakai akhir dan pakar sistem informasi.
  - **Hardware**, Meliputi semua peralatan dan bahan fisik yang digunakan dalam pemrosesan informasi.
  - **Software**, Meliputi semua rangkaian perintah pemrosesan informasi.
  - **Network**, Menekankan bahwa teknologi informasi dan jaringan adalah komponen sumber daya dasar dari semua sistem informasi.
  - **Data**, Data dapat berupa banyak bentuk, termasuk data alfanumerik tradisional, yang terdiri dari angka dan huruf serta karakter lainnya yang menjelaskan transaksi bisnis dan kegiatan entitas lainnya.

#### ❖ Sistem Kolaborasi

Menurut Fatal (2011:7) sistem kolaborasi tidak menggantikan cara tradisional tetapi untuk memperluas kegunaan cara tradisional ke *mobile internet*.

#### 2.1.2 Database

Menurut Connolly & Begg (2010:65) *database* merupakan sekumpulan data yang berhubungan secara logical dan dideskripsikan serta dirancang untuk memenuhi kebutuhan sistem informasi yang dibutuhkan oleh suatu organisasi.

Menurut Satzinger, Jackson & Burd (2005:398) *database* adalah kumpulan dari data yang tersimpan secara terintegrasi, di-*manage* dan dikontrol secara terpusat.

Menurut Rainer, Turban & Potter (2007:6) *Database* adalah kumpulan *file* yang berhubungan atau tabel yang berisi data.

❖ **DBMS (*Database Management System*)**

Menurut Connolly & Begg (2010:66), *Database Management System* ( DBMS ) adalah suatu sistem perangkat lunak yang memungkinkan pengguna untuk mendefinisikan, membuat, memelihara, dan mengontrol akses ke *database*.

Menurut Satzinger, Jackson & Burd (2005:398) *Database Management System* (DBMS) adalah Sebuah sistem *software* yang *me-manage* dan mengontrol akses ke *database*.

Menurut Rainer, Turban & Potter (2007:5) *Database Management System* (DBMS) adalah suatu set program yang menyediakan user dengan alat untuk menambah, menghapus, menagakses, dan menganalisis data yang disimpan di satu lokasi.

Berikut adalah komponen DBMS menurut Connolly & Begg (2010:68) yang terdiri atas 5 komponen, yaitu :

1. *Hardware*

DBMS membutuhkan hardware untuk menjalankan aplikasi – aplikasinya yang meliputi PC, mainframe, dan suatu jaringan computer.

2. *Software*

Komponen perangkat lunak yang meliputi software DBMS itu sendiri, program aplikasi, sistem operasi, termasuk dan sistem jaringan.

3. *Data*

Data merupakan komponen yang paling penting dalam DBMS, berasal dari sudut pandang dari end – user. Data berperan sebagai penghubung antara mesin dan pengguna.

4. *Procedure*

Prosedur merupakan instruksi dan aturan yang mengatur perancangan dan penggunaan database.

5. *People*

Komponen terakhir adalah manusia yang berhubungan langsung dengan sistem. Komponen ini meliputi *Database Administrator*, *Database Designers*, *Application Developers*, dan *end – user*

Menurut Connolly & Begg (2010:77), DBMS memiliki beberapa keuntungan dan kerugian dalam penggunaannya, yaitu :

- a. Keuntungan
  - Kontrol terhadap redundansi data.
  - Konsistensi data.
  - Banyaknya informasi dari data yang sama.
  - Pengaksesan data oleh beberapa *user* dalam waktu yang sama.
  - Meningkatkan integritas data.
  - Meningkatkan keamanan.
  - Menetapkan standarisasi dalam penyajian data.
  - Meningkatkan aksesibilitas.
  - Meningkatkan produktivitas.
  - Meningkatkan pemeliharaan melalui data *independence*.
  - Meningkatkan keakuratan.
  - Meningkatkan fasilitas *backup* dan *recovery*.
  - Menyeimbangkan konflik dari kebutuhan yang ada
  
- b. Kerugian
  - Kompleksitas
  - Ukuran
  - Biaya
  - Tambahan *hardware*.
  - Biaya dari proses konversi.

- **MySQL**

Menurut Bunafit (2004:163) Ada beberapa fungsi yang digunakan dalam aplikasi antara PHP dan MySQL. Fungsi – fungsi tersebut sangat erat kaitannya dengan Query SQL akan tetapi, kita tidak langsung menggunakan perintah SQL pada *script* PHP. MySQL memiliki beberapa kelebihan antara lain :

1. Adanya dukungan SQL
2. Lebih handal ,cepat dan Mudah digunakan.

3. Adanya Multiplatform dan portable,MySQL dapat berjalan stabil pada berbagai sistem Operasi seperti Windows,Linux,FreeBSD,Mac Os X server,Solaris,Amiga,dan masih banyak lagi .
4. Perangkat lunak yang open source.MySQL didistribusikan sebagai perangkat lunak Open source,dibawah lisensi GPL sehingga digunakan secara gratis.
5. Multi-User.MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami crash.
6. Performance Tuning', MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana,dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
7. Ragam tipe data MySQL sangat banyak,seperti signed unsigned integer,float,double,char,text,date,timestamp,dan lain lain.
8. Perintah dan fungsi MySQL memiliki operator dan fungsi secara penuh mendukung perintah select dan where dalam perintah (query).
9. Security.MySQL memiliki beberapa lapisan keamanan seperti level subnetmask,nama host,dan izin akses user dengan sistem perizinan yang mendetail serta sandi terenkripsi.
10. Skalabilitas dan pembatasan, MySQL mampu menangani database dalam skala besar,dengan jumlah rekaman(records) lebih dari 50 juta dan 60 ribu tabelserta 5 milyar baris.selain itu batas indeks yang didapat ditampung mencapai 32 indeks pada tiap tabelnya.
11. Konektivitas , MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP ,Unix soket (UNIX),atau Named Pipes(NT).
12. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari 20 bahasa.meskipun demikian ,bahasa indonesia belum termasuk didalamnya.
13. Interface, MySQL memiliki antar muka terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application programming interface).
14. Client and Tools,MySQL dilengkapi dengan berbagai peralatan yang dapat digunakan untuk administrasi database,dan pada setiap peralatan yang ada disertakan petunjuk online.

15. Table structure,MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE ,dibandingkan database lainnya semacar PostgreSQL ataupun Oracle.

### 2.1.3 Internet

Internet sangat dimanfaatkan manusia untuk menambah fungsi alat elektronik sehingga menjadi dasar dari pola pikir manusia di zaman sekarang. Berikut adalah pengertian internet menurut pakarnya.

#### ❖ **Pengertian Internet**

Menurut Satzinger, Jackson & Burd (2005:274) *internet* adalah sebuah koleksi global dari jaringan yang menggunakan protocol jaringan yang sama TCP/IP.

Menurut O'brien (2005:261), *internet* adalah “jaringan komputer yang tumbuh cepat dan terdiri dari jutaan jaringan perusahaan, pendidikan, serta pemerintah yang menghubungkan ratusan juta komputer serta pemakainya di lebih dari 200 negara.”

#### ❖ **Fasilitas Internet**

Menurut Yuhefizar (2008:10) *internet* menyediakan beberapa fasilitas utama, yaitu

1. *Website*

Kumpulan halaman *web* yang mengandung informasi. Informasi yang disajikan diolah sedemikian rupa sehingga mampu memberikan *multiflier* efek seperti *online shopping*, *e-government*, dan *e-banking*.

2. *Mailing List*

Media untuk membentuk ruang diskusi di internet yang dapat diikuti oleh siapa saja yang mempunyai minat terhadap suatu topik.

3. *File Transfer Protocol ( FTP )*

Fasilitas untuk mengirim dan mengambil *file* atau *folder* antar komputer yang terhubung dengan jaringan internet.

4. *Chatting*

Fasilitas untuk mengobrol secara *online* baik secara teks maupun grafik.

Berikut adalah beberapa jenis jaringan *internet* yang ada sebagai berikut :

❖ **LAN ( *Local Area Network* )**

Menurut Satzinger, Jackson, & Burd (2005:272) LAN adalah sebuah jaringan computer yang mencakup jarak local, contohnya seperti lokasi didalam gedung yang sama.

❖ **WAN ( *Wide Area Network* )**

Menurut Satzinger, Jackson, & Burd (2005:273) WAN adalah sebuah jaringan computer yang terbentang mencakupi area yang luas seperti kota, provinsi, atau negara.

#### **2.1.4 Website**

Menurut Sardi (2004:4) *website* adalah sekumpulan dokumen yang dipublikasikan melalui jaringan *internet* atau *intranet* sehingga dapat diakses oleh *user* melalui *web browser*.

Adapun kriteria *website* yang baik dikemukakan oleh Suyanto (2009:61) adalah

##### **A. Usability**

*Usability* melibatkan pertanyaan “dapatkah *user* menemukan cara untuk menggunakan situs web tersebut dengan efektif (*doing things right*)” atau *usability* adalah sebagai suatu pengalaman pengguna dalam berinteraksi dengan aplikasi atau situs *web* sampai pengguna dapat mengoperasikannya dengan mudah dan cepat. Situs *web* harus memenuhi lima syarat untuk mencapai tingkat *usability* yang ideal, yaitu:

- 1) Mudah untuk dipelajari letakkan isi yang paling penting pada bagian atas halaman agar pengunjung dapat menemukannya dengan cepat.
- 2) Efisien dalam penggunaan jangan menggunakan *link* yang terlalu banyak. Sediakan seperlunya dan hantarkan pengunjung untuk mencapai informasi yang diperlukan dengan cepat dan mudah. Hantarkan informasi yang *user* butuhkan dengan sedikit mungkin klik.

- 3) Mudah untuk diingat situs jangan terlalu banyak melakukan perubahan yang mencolok, khususnya pada navigasi.
- 4) Tingkat kesalahan rendah hindari *link* yang tidak berfungsi (*broken link*) atau halaman masih dalam proses pembuatan (*under construction*).
- 5) Kepuasan pengguna Sebuah website seharusnya enak untuk digunakan. *User* harus dapat menemukan apa yang mereka cari, mengunduhnya dengan cepat, mengetahui kapan mereka selesai, dan mudah untuk berbagi *site* kepada teman mereka.

#### B. *System* navigasi

Navigasi dapat ditampilkan dalam berbagai media yaitu teks, *image* ataupun animasi. Navigasi dari *images* dapat menawarkan banyak sekali variasi, misalnya dengan ikon, *image*, pengguna huruf dan bentuk yang lebih bebas. adapun cara membuat navigasi yang baik yaitu:

1. Rencanakan dengan benar sebelum membuat Navigasi atau struktur situs *web* nantinya akan sulit di ubah. Jadi sebelum membuat navigasi sebuah situe, tentukan konsepnya.
2. Kelompokkan link navigasi dan aturlah seperlunya halaman dengan mock up navigasi elemen, jenis *content* yang ada di halaman dan prioritasnya.
3. Buatlah tampilan navigasi berbeda dari tampilan lainnya Buatlah navigasi yang logika susunannya mudah dipahami. Kata-kata yang menjadi navigasi menerangkan tujuan navigasi. Gunakan katakata yang singkat, tepat dan jelas. gunakan istilah-istilah yang biasa dipergunakan atau mudah dimengerti pengunjung, jangan membuat image yang berkesan sebagai tombol, padahal bukan tombol atau link.
4. Buat navigasi yang memungkinkan *user feedback* seperti meletakkan home atau *back*

5. Jangan sampai navigasi tidak berfungsi (*broken link*, baik internal atau eksternal), dan navigasi yang belum ada isinya jangan di cantumkan.

#### 6. Jaga konsistensi

Posisi daerah navigasi harus tetap dan struktur navigasi harus konsisten dengan struktur isi, yang di urutkan berdasarkan logika pengunjung. bila perlu letakan pada tempat yang sama di tiap halaman, gunakan warna yang sama dan tempatkan pada tempat yang mudah dilihat.

### C. Content

*Content* yang baik akan menarik, relevan, dan pantas untuk target *audience web* tersebut. Gaya penulisan bahasa yang dipergunakan harus sesuai dengan web dan target audience. Hindari kesalahan dalam penulisan, termasuk tata bahasa dan tanda baca, di tiap halaman, *header* dan judul. Cara membuat content yang baik adalah sebagai berikut :

1. Kenalilah *audience*, tulislah dengan gaya mereka dan sesuaikan dengan isinya
2. Jaga *content* agar tetap *up to date*. Ini akan meningkatkan daya lekat situs web yang membuat user sering kembali.
3. Dahulukan kualitas di atas kuantitas. Usahakan kualitas content tercapai lebih dahulu dan kemudian jika memungkinkan baru mengejar kuantitas. Hal itu akan mendorong user untuk mengunjungi situs kita lagi.

### D. Loading time

Sebuah penelitian yang dilakukan oleh *Zona Research* (april 1999) menyatakan bahwa 80% pengunjung akan menutup *browser* bila halaman *web* yang ia buka tidak tampil dalam 7-8 detik. Penelitian *Jupiter Media metrix* mengatakan bahwa 40% pengunjung akan kembali mengunjungi situs yang tampil lebih cepat. Sebuah situs *web* yang tampil lebih cepat kemungkinan besar akan kembali dikunjungi, apalagi bila dengan konten dan tampilan yang menarik. Waktu *download* memang tidak hanya dipengaruhi desain tetapi juga koneksi, *server*, dan lain-lain. Namun demikian

desainer *web* harus memperhitungkan desain yang dibuatnya agar dapat tampil lebih cepat dengan menggunakan ukuran yang sekecil mungkin.

#### E. *Interactivity*

Interaktivitas adalah apa yang melibatkan penguana situs web sebagai user experience dengan situs web itu sendiri. Dasar dari interaktivitas adalah hyperlinks (*links*) dan mekanisme *feedback*.Gunakan hyperlink untuk membawa pengunjung ke sumber berita, *topic* lebih lanjut, *topic* terkait, atau lainnya.seperti *link* yang berbunyi *more info about this*, *related links* dan lain-lain.sedangkan untuk mekanisme *feed back*, contohnya seperti *critiques*,*comments*, *question*. Keuntungannya *user* dapat memberitahu kritik/saran agar situs *web* kita bisa lebih baik.

#### F. *Compatibility*

Situs *web* harus kompatibel dengan berbagai perangkat tampilannya, harus memberikan alternative bagi *browser* yang tidak dapat melihat situsnya.

#### G. *Functionality*

Seberapa baik sebuah *web* bekerja dari aspek teknologinya, ini bisa melibatkan *programmer* dengan *script*-nya, misalnya *HTML*, *PHP*, *Coldfusion*, *CGI*, *SSL*, dan lain – lain.

#### H. *Accesibility*

Halaman *web* harus bisa dipakai oleh setiap orang, baik anak – anak, orang tua, dan orang muda, termasuk orang cacat. Ada berbagai hambatan yang ditemui dari sisi pengguna untuk bisa menikmati halaman web itu. Untuk hambatan fisik, bagaimana memaksimalkan pengguna konten ketika satu atau lebih indera dimatikan atau dikurangi kerjanya, terutama untuk *user* dengan kekurangan indera penglihatan. Selain itu ada juga hambatan infrastruktur, seperti akses *internet* yang lambat, spesifikasi *computer*, penggunaan *browser*, dan lain – lain yang dapat mempengaruhi akses seseorang.

## I. *Graphic Design*

Kepuasan visual seseorang user secara subyektif melibatkan bagaimana desainer visual situs *website* tersebut membawa mata user menikmati dan menjelajahi situs *web* dengan melalui *layout*, warna, bentuk, dan tipografi. Grafik membuat halaman menjadi indah tetapi bisa juga memperlambat akses dengan semakin besarnya ukuran *file*. Desain yang baik setidaknya memiliki komposisi warna yang baik dan konsisten, *layout* grafik yang konsisten, teks yang mudah dibaca, penggunaan grafik yang memperkuat isi teks, dan secara keseluruhan membentuk suatu pola yang harmonis.

Berikut adalah hal – hal berkaitan dengan *website* :

- **WWW (*World Wide Web*)**

Menurut Williams & Sawyer (2007:17) *World Wide Web* adalah komponen *internet* yang berupa *multimedia*. *Internet* memang telah lahir lebih dari 35 tahun lalu, tetapi yang mempopulerkan *internet* adalah *World Wide Web*. *Web* didefinisikan sebagai sistem interkoneksi komputer *internet* yang mendukung dokumen – dokumen berformat *multimedia*.

- ***Web Browser***

Menurut Sunarto (2005:23), *web browser* adalah sebuah program aplikasi yang digunakan untuk menjelajahi dunia maya dan menampilkan *web page* yang ada di *internet*.

Menurut Comer (2007:209) *web browser* adalah program yang menampilkan dan menyediakan akses dokumen *hypermedia* yang ada di *internet*.

Berdasarkan teori diatas dapat disimpulkan bahwa *web browser* adalah program yang dapat menampilkan dokumen *hypermedia* dan *web page* di dunia maya *internet*.

- ***Web Server***

Menurut Supardi (2009:181) *web server* adalah perangkat lunak yang mengatur atau mengelola program berdasarkan permintaan *browser* dan mengirimkan hasil dari permintaan tersebut.

Menurut Smitdev & Zaki (2008:21) *web server* adalah sebuah *hardware* dan *software* yang bertugas melakukan interpretasi *HTTP request* untuk kemudian melakukan respon atas *HTTP request* tersebut dalam bentuk kode HTML, dan kode – kode lainnya yang bisa dipahami *browser*. Ada banyak sekali *web server* yang bisa dipakai antara lain :

1. *Apache Tomcat*
2. *Microsoft Windows Server 2003 Internet Information Service ( IIS )*
3. *Lighttpd*
4. *Sun Java System Web Server*
5. *Xitami Web Server*
6. *Zeus Web Server*

- ***Web Service***

Menurut Al-Fedaghi (2011:57) *Web service* adalah teknologi yang berbasiskan dari konsep layanan yang memanfaatkan penggunaan teknologi *computing*.

- **Hypertext Transfer Protocol ( HTTP )**

Menurut Sunarto (2005:22) HTTP adalah sebuah protocol yang menentukan aturan main antara *software client* dan *software server* dalam penyediaan dokumen yang diminta *browser*.

- ***Uniform Resource Locator ( URL )***

Menurut Suyanto (2009:195) URL adalah alamat sebuah *file* yang dapat diakses di *internet*, berisi nama protocol *internet* yang dibutuhkan untuk mengakses *file*, nama computer yang berisi *file* tersebut, dan direktori dimana *file* tersebut berada. Secara umum, URL dibagi menjadi tiga bagian. Bagian pertama menunjukkan protocol yang dipakai seperti *http://* atau *https://*. Bagian kedua menunjukkan alamat host seperti *www.yahoo.com* . Bagian ketiga menunjukkan *path* direktori *file*, yaitu tempat dimana *file* yang ingin diakses disimpan dalam *server*.

- **Transmission Control Protocol / Internet Protocol ( TCP / IP )**

Menurut Blank (2004:2) TCP/IP adalah kumpulan *protocol* yang memungkinkan komunikasi antar komputer.

### 2.1.5 Programming

Dalam proses membuat program maka diperlukan berbagai *tools* yang harus digunakan terutama bahasa pemrograman itu sendiri beserta *tools* pendukung lainnya.

- **Java Script**

Menurut Brooks (2007:3) *java script* adalah bahasa pemrograman yang diinterpretasikan bukan di *compile*, diadopsi dari bahasa C/C++ yang dikembangkan menjadi bahasa pemrograman *web client side*. *Java script* didesain untuk bekerja sama dengan HTML membuat *web page* yang interaktif.

Menurut Farland & Sawyer (2008:1) *java script* adalah bahasa pemrograman yang digabungkan dengan HTML untuk membuat halaman *web* yang beranimasi, interaktif dan memiliki *visual effect* yang dinamis.

- **HTML (Hypertext Markup Language)**

Menurut Jogyanto (2005:129) HTML adalah bahasa pemrograman yang digunakan untuk membuat sebuah halaman web. Kode – kode ini menerjemahkan bagaimana tampilan sebuah halaman web.

- **CSS**

Menurut Saleh & Rubianto (2008:32) CSS adalah suatu bahasa *stylesheet* yang mengatur tampilan suatu dokumen. Pada umumnya CSS digunakan untuk memformat halaman *web* yang ditulis dengan HTML dan XHTML. Dengan CSS, tampilan *website* akan lebih cantik dan konsisten. Ada dua cara menuliskan kode CSS, yaitu :

1. Internal dengan cara menuliskan langsung diantara *tag* HTML/XHTML.
2. External dengan cara menyimpan CSS kedalam *file* yang terpisah kemudian dipanggil didalam *Web* saat dibuka.

- ***PHP (Hypertext Preprocessor)***

Menurut Suryatiningsih (2009:141) PHP adalah bahasa *scripting* yang menyatu dengan HTML dan dijalankan pada *server side*. Semua *syntax* yang diberikan akan sepenuhnya dijalankan pada *server* sedangkan yang dikirimkan ke *browser* hanya berupa hasilnya saja.

- ***XAMPP***

Menurut Riyanto (2010:1) paket PHP dan MYSQL berbasis *Open Source* yang dapat digunakan sebagai *tool* pembantu pengembangan aplikasi berbasis PHP.

## 2.2 Teori Khusus

Teori – teori yang digunakan sebagai acuan pola kerja dan pola pikir untuk mencapai tujuan pembuatan aplikasi *website* serta didalamnya termasuk menganalisa dan merancang sistem informasi yang dibutuhkan.

### 2.2.1 *Rapid Application Development ( RAD )*

Dalam pembangunan aplikasi *website* sistem informasi SMP Tarsisius 2 digunakan metode *rapid application development* sebagai acuan dalam tahap – tahap pengerjaannya. Berikut gambaran *rapid application development (RAD)*.

#### 2.2.1.1 Pengertian *Rapid Application Development ( RAD )*

Menurut Whitten & Bentley (2007:98) *Rapid Application Development ( RAD )* adalah sebuah strategi pengembangan sistem yang menekankan kecepatan dalam pengembangan melalui keterlibatan pengguna dalam pembangunan secara cepat, iteratif, dan *incremental* dari suatu serangkaian *prototype* dari suatu sistem yang dapat berkembang menjadi suatu sistem akhir atau versi tertentu.

Ide dasar *Rapid Application Development ( RAD )* adalah sebagai berikut :

1. Untuk lebih mengefektifkan pengguna dalam keterlibatan kegiatan menganalisa, mendesain dan membangun.
2. Untuk mengorganisasikan pengembangan sistem agar lebih fokus, dan lebih menerlibatkan *system owner, users, analysts, designer, dan builder*.
3. Untuk mempercepat kegiatan analisa dan perancangan kebutuhan sistem melalui pendekatan pembangunan iteratif.
4. Untuk mengurangi waktu yang dibutuhkan pengguna untuk melihat proses kerja sistem yang dikembangkan.



Mengasumsikan kesalahan dapat ditemukan dan diatasi pada proses iteratif berikutnya.

3. Jarang sekali namun terdapat beberapa kemungkinan proses bisnis perlu didesain ulang untuk merepresentasikan keterlibatan aplikasi dalam sistem.
4. Dalam setiap *iteration cycle*, beberapa desain *prototype* dan beberapa fungsi dari bagian sistem dibangun dan di uji coba. Nantinya, aplikasi yang sudah selesai akan menjadi hasil dari *final iteration process*.
5. Setelah setiap *prototype* dan fungsional sistem dikembangkan dan di uji coba, pengguna diberikan kesempatan untuk mencoba menggunakan. Harapan dari pengguna terhadap *prototype* akan menjadi kebutuhan sistem baru dan memberikan tanggapan terhadap gambaran proses bisnis.
6. Setelah setiap *prototype* dan fungsional sistem dikembangkan dan di uji coba, *system analysts* dan *designers* akan *me-review* untuk mendapatkan tanggapan teknis dan arah pengembangan untuk proses berikutnya.
7. Berdasarkan tanggapan yang telah didapatkan, *system analysts* akan mengidentifikasi tujuan atau arah pembenaran sistem yang akan dilakukan atau kebutuhan sistem.
8. Berdasarkan tanggapan yang didapatkan, *designers* akan mengidentifikasi tujuan atau arah perubahan desain dan pembenaran arsitektur aplikasi.
9. Nantinya sistem akan dapat dinilai siap untuk diimplementasikan.

### **2.2.1.2 Keuntungan dan Kerugian *Rapid Application Development***

Menurut Whitten & Bentley (2007:100) *Rapid application development ( RAD )* memberikan beberapa keuntungan dalam pemanfaatannya dalam membangun *prototype website* namun juga memiliki kekurangan yang dapat menyebabkan kerugian. Berikut adalah penjabaran lebih lanjut mengenai keuntungan dan kerugian *rapid application development* dalam pemanfaatannya.

Keuntungan	Kerugian
<ol style="list-style-type: none"> <li>1. Sangat berguna untuk proyek dimana kebutuhan sistem yang akan dikembangkan belum sepenuhnya tepat atau tidak menentu.</li> <li>2. Mendorong antusias <i>end-user</i> untuk berpartisipasi dalam proyek tersebut.</li> <li>3. Proyek memiliki visibilitas dan dukungan yang tinggi yang dikarenakan keterlibatan pengguna.</li> <li>4. Solusi berbasis <i>software</i> lebih cepat dibandingkan solusi berbasis bisnis model.</li> <li>5. Kesalahan dan kelalaian dapat diketahui lebih cepat pada <i>prototype</i> ini dibandingkan pada <i>system model</i>.</li> <li>6. Uji coba adalah suatu keharusan dari suatu produk yang mendasari dari pendekatan <i>prototype</i>.</li> <li>7. Pendekatan secara <i>iterative</i> lebih wajar karena perubahan adalah factor yang diharapkan dalam</li> </ol>	<ol style="list-style-type: none"> <li>1. Beberapa isu yang mengatakan RAD lebih membutuhkan banyak tenaga, dukungan, dan pengembangan yang menyebabkan meningkatnya biaya yang diperlukan.</li> <li>2. RAD dapat menyelesaikan masalah yang keliru jika masalah tersebut diabaikan dan disingkat.</li> <li>3. <i>Prototype</i> berbasis RAD dapat memungkinkan berkurangnya semangat dari <i>analysts</i> dibandingkan yang lainnya.</li> <li>4. Terkadang para <i>stakeholder</i> melihat <i>prototype</i> sebagai pembuangan tenaga dan waktu.</li> <li>5. Menekankan pada kecepatan penyelesaian proyek dapat berdampak buruk bagi kualitas produk.</li> </ol>

proses pengembangan.	
----------------------	--

Tabel 2.1 Keuntungan dan Kerugian *Rapid Application Development ( RAD )*

( Sumber : Whitten &amp; Bentley, 2007, p100)

### 2.2.2 *Object Oriented Analysis and Design (OOA&D)*

Dalam menganalisis proses bisnis dan perancangan sistem informasi yang efektif terhadap suatu proses bisnis akan membutuhkan sebuah metode *Object Oriented Analysis and Design* untuk memudahkan dalam penganalisisan dan perancangan.

#### 2.2.2.1 *Pengertian Object Oriented Analysis and Design*

Menurut Satzinger, Jackson & Burd (2005:60) mendefinisikan *Object Oriented Analysis* (OOA) adalah semua jenis objek yang melakukan pekerjaan dalam suatu sistem dan menunjukkan interaksi apa yang dibutuhkan pengguna untuk menyelesaikan tugas.

Menurut Satzinger, Jackson & Burd (2005:60) mendefinisikan *Object Oriented Design* (OOD) adalah suatu cara untuk menentukan semua tipe *object* yang harus berkomunikasi dengan orang – orang dan perangkat didalam sistem menggambarkan bagaimana *object* berinteraksi untuk menyelesaikan tugasnya dan memperbaiki definisi masing – masing tipe dari *object* sehingga dapat diimplementasikan dengan sebuah bahasa dan lingkungan khusus

Dari definisi diatas dapat disimpulkan bahwa *Object Oriented Analysis and Design* (OOA&D) adalah cara untuk menentukan semua *object* yang berinteraksi dengan sistem untuk menyelesaikan tugas dan memperbaiki definisi *object* tersebut sehingga dapat diimplementasikan dalam penggambaran khusus.

Notasi yang digunakan dalam pendekatan OOA&D adalah UML (*Unified Modelling Language*).

- ***Objects***

Menurut Satzinger, Jackson & Burd (2005:60) objek merupakan hal dalam sistem *computer* yang dapat merespon pesan. Sebuah objek memiliki keadaan dan perilaku.

Keadaan dari sebuah objek adalah kondisi objek tersebut atau himpunan dari keadaan yang menggambarkan objek tersebut. Atribut adalah nilai internal dari suatu objek yang mencerminkan antara lain karakteristik objek, kondisi objek, kondisi sesaat, koneksi dengan objek lain dan identitas. Perubahan keadaan objek dicerminkan oleh perilaku (*behavior*).

*Behavior* suatu objek mendefinisikan bagaimana sebuah objek bertindak dan memberi reaksi. *Behavior* ditentukan oleh himpunan semua atau beberapa *operation* yang dapat dilakukan dalam objek itu sendiri.

- ***Attributes, Methods, and User Interface Object***

Menurut Satzinger, Jackson & Burd (2005:62) Atribut adalah karakteristik objek yang memiliki nilai seperti ukuran, bentuk, warna, lokasi, dan keterangan dari tombol atau label atau nama, alamat, dan nomor telepon dari pelanggan.

Menurut Satzinger, Jackson & Burd (2005:62) *methods* adalah perilaku atau operasi yang menggambarkan sebuah objek apa yang mampu melakukan suatu pekerjaan.

Menurut Satzinger, Jackson & Burd (2005:62) *User interface* adalah objek pengguna yang berinteraksi dengan saat menggunakan sistem, seperti tombol, menu *item*, kotak teks atau label.

- ***Classes, identity, superclass, and subclass***

Menurut Satzinger, Jackson & Burd (2005:62) kelas adalah jenis atau klasifikasi yang sama antara semua benda yang serupa.

Menurut Satzinger, Jackson, & Burd (2005:62) identitas adalah referensi unik untuk objek yang memungkinkan objek lain untuk menemukan dan mengirim pesan.

Menurut Satzinger, Jackson & Burd (2005:62) *superclass* adalah kelas umum dalam hirarki generalisasi / spesialisasi yang dapat diperpanjang dengan *superclass*.

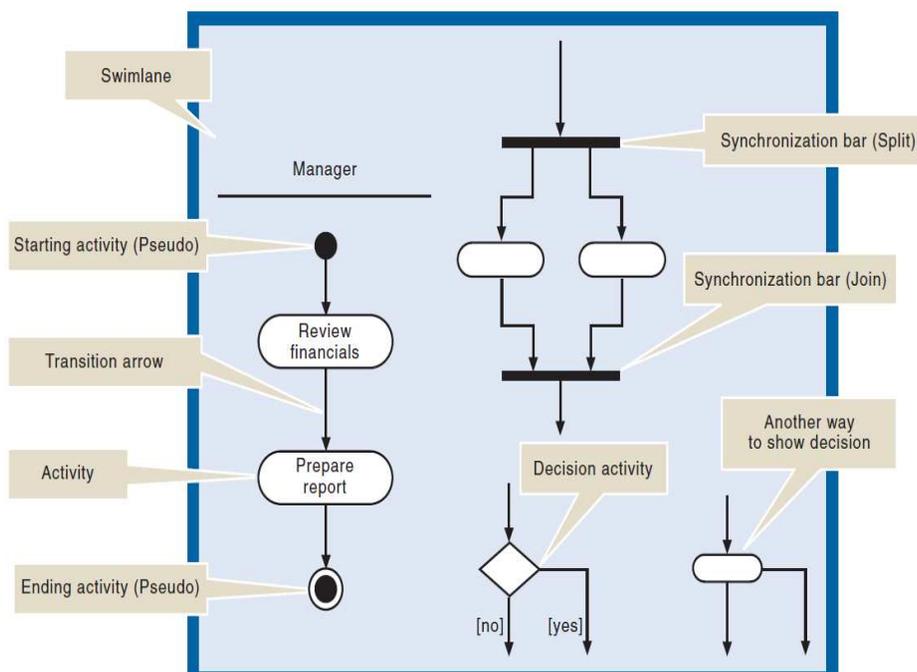
Menurut Satzinger, Jackson & Burd (2005:62) *subclass* adalah kelas khusus dalam hirarki generalisasi / spesialisasi yang berisi atribut dan metode tambahan yang membedakannya dari kelas yang lebih umum bahwa kelas itu meluas.

### 2.2.2.2 UML ( *Unified Modeling Language* )

Menurut Satzinger, Jackson & Burd (2005:48), *Unified Modeling Language* adalah suatu set standar konstruksi model dan notasi yang dikembangkan secara khusus untuk pemodelan berorientasi objek. UML terdiri dari beberapa notasi yaitu :

- ***Activity Diagram***

Menurut Satzinger, Jackson & Burd (2005:141) *Activity Diagram* adalah sebuah tipe dari *work flow* diagram yang mendeskripsikan aktifitas *user* dan tahapan - tahapan pengerjaannya secara sekuensial.



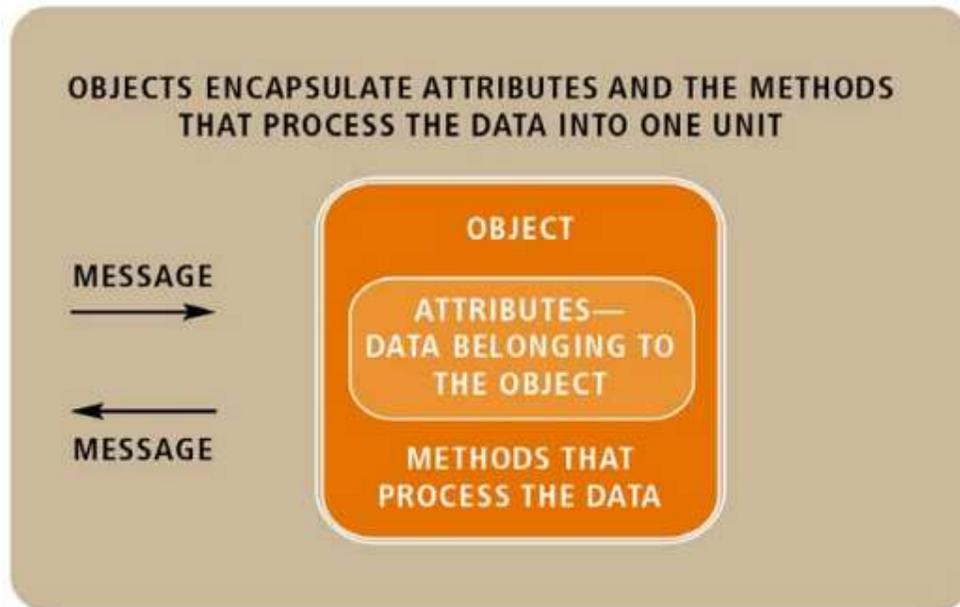
Gambar 2.2 *Activity Diagram*

(Sumber : Satzinger, Jackson, & Burd, 2005, p141)

- ***Domain Model Class Diagram***

Menurut Satzinger, Jackson & Burd (2005:184) adalah sebuah model grafikal yang digunakan di dalam pendekatan *object oriented* untuk menunjukkan *class-class* yang ada di dalam sistem

Sebuah *class* dalam *class diagram* berbentuk seperti berikut



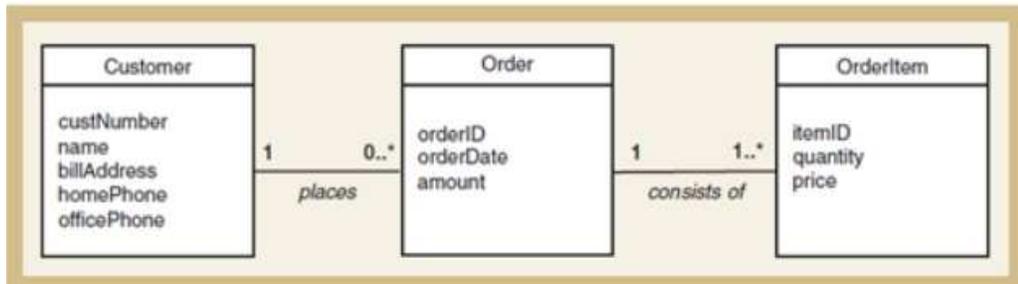
Gambar 2.3 *Class Diagram*

(Sumber : Satzinger, Jackson & Burd, 2005, p184)

- <<Stereotype Name>> diisi dengan nama kelas seperti TrPenjualan, Customer, sesuai objek yang digambarkan class tersebut.
- Attribute List diisi dengan atribut apa saja yang dimiliki class tersebut. Sebagai contoh atribut dari class Customer adalah IDCustomer, nama Customer, telepon Customer.
- Method List diisi method yang dimiliki oleh class. Sebagai contoh method dari Customer adalah getpelanggan().

- Asosiasi

Asosiasi dalam *Class Diagram* memiliki asosiasi *multiplicity*. *Multiplicity* adalah hubungan kelas satu ke banyak dalam satu arah dan satu untuk satu di arah lain. Biasanya disebut *one to many* atau *one to one*.

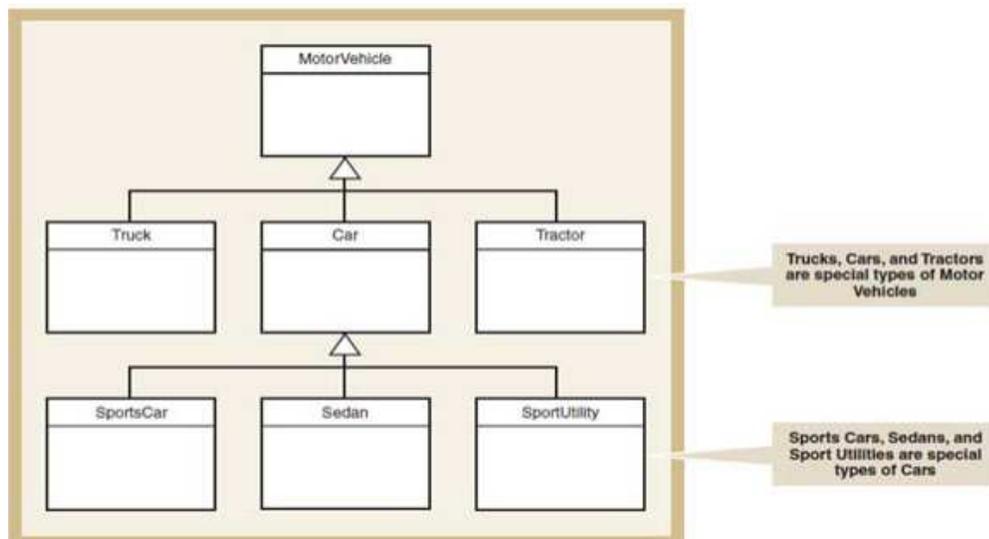


Gambar 2.4 Asosiasi *Class Diagram*

( Sumber : Satzinger, Jackson & Burd, 2005, p187))

- Generasilisasi

Menurut Satzinger, Jackson & Burd (2005:189), generalisasi adalah hierarki yang struktur atau peringkat kelas dari yang lebih umum ke *class* khusus. Generalisasi kadang – kadang disebut warisan hierarki.

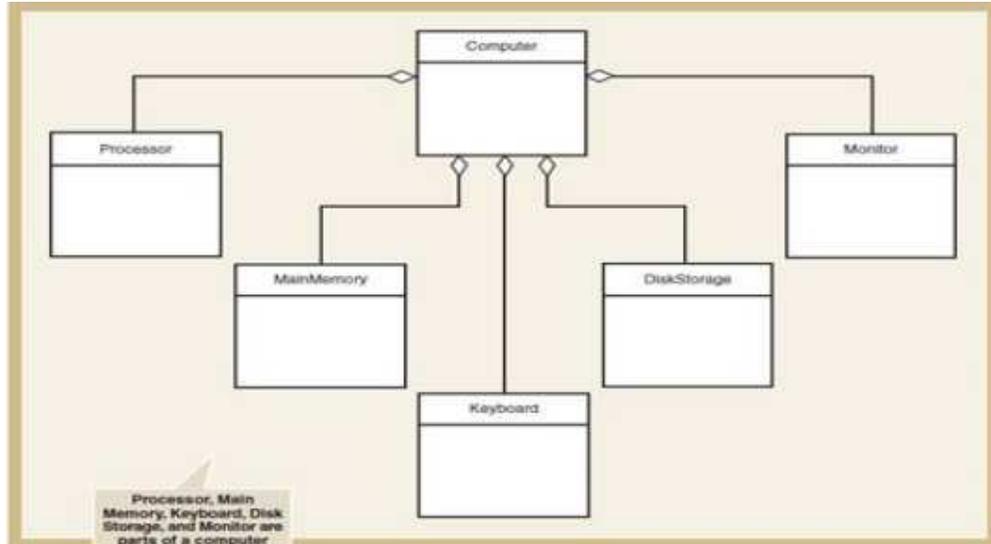


Gambar 2.5 Generalisasi *Class Diagram*

( Sumber : Satzinger, Jackson & Burd, 2005, p190)

- *Whole-Part Hierarchies*

Menurut Satzinger, Jackson & Burd (2005:190) cara lain yang menstruktur informasi adalah dengan mendefinisikan dalam hal bagian mereka.



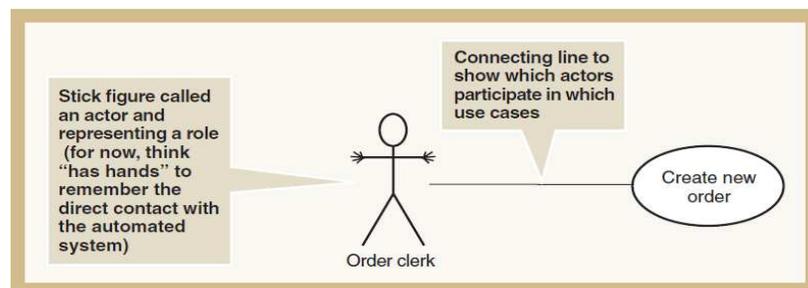
Gambar 2.6 *Whole-Part Hierarchies*

(Sumber : Satzinger, Jackson & Burd, 2005, p192)

- *Use Case Diagram*

Menurut Satzinger, Jackson & Burd (2005:215) *Use case diagram* adalah sebuah diagram yang menunjukkan urutan pesan antara *actor external* dan sistem selama *use case* berlangsung.

Komponen dari *Use Case Diagram*



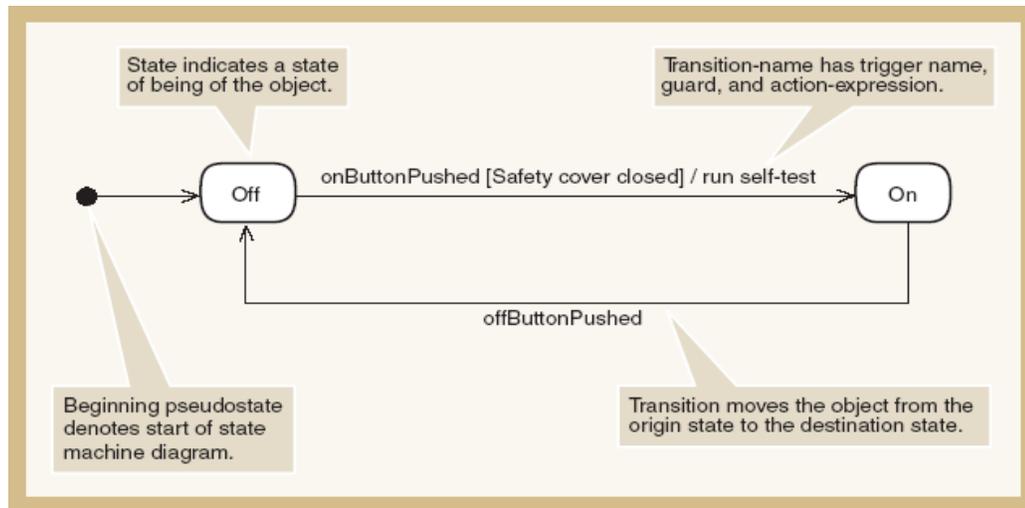
Gambar 2.7 *Komponen Use Case*

(Sumber : Satzinger, Jackson & Burd, 2005, p215)

- Aktor yang menggambarkan peran dari *user*. Digambarkan dengan gambar manusia yang bentuk tubuh yang lengkap.
- Garis Penghubung, menghubungkan aktor dengan tugas mereka dengan sistem
  - Tugas dari *user* ditulis di dalam sebuah lingkaran *ellipse*.

- ***Statechart Diagram***

Menurut Satzinger, Jackson & Burd (2005:214) *Statechart diagram* merupakan sebuah diagram yang menunjukkan siklus hidup dari sebuah objek dalam *state* dan *transition*.



Gambar 2.8 *State Chart Diagram*

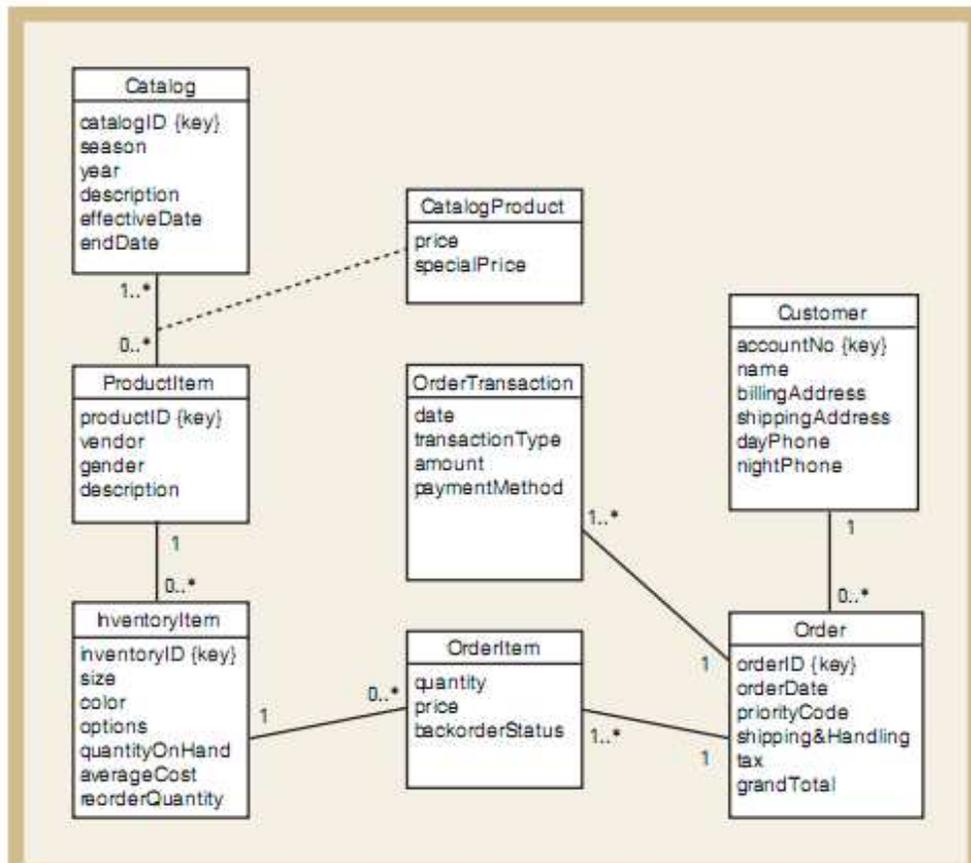
(Sumber : Satzinger, Jackson & Burd , 2005, p214)

- ***Firstcut Class Diagram***

Menurut Satzinger, Jackson & Burd (2005:311) *Firstcut Class Diagram* adalah *Class Diagram* yang dikembangkan dengan cara memperluas diagram kelas dari domain model. Hal ini membutuhkan dua langkah:

1. menguraikan jenis atribut dan informasi dari nilai awal
2. menambahkan panah navigasi visibilitas

*contoh Firstcut Class Diagram :*

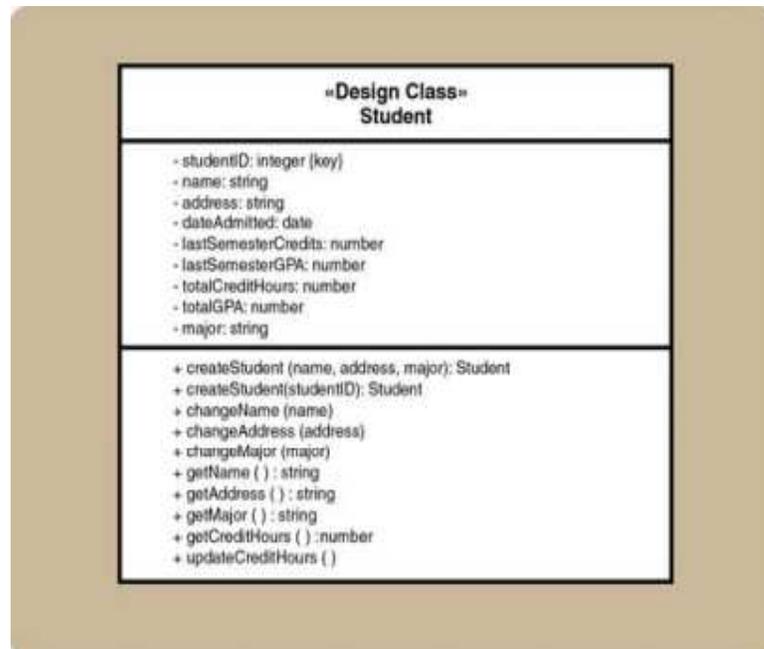


Gambar 2.9 *Firstcut Class Diagram*

(Sumber : Satzinger, Jackson & Burd , 2005, p311)

- **Update Class Diagram**

Menurut Satzinger, Jackson & Burd (2005:305) *update class diagram* adalah desain *class diagram* yang dikembangkan untuk tiap *layer*. Dalam *view layer* dan *access layer*, beberapa kelas harus dispesifik. *Domain layer* juga mempunyai beberapa kelas baru yang ditambahkan untuk *use case controller*.



Gambar 2.10 *Update Class Diagram*

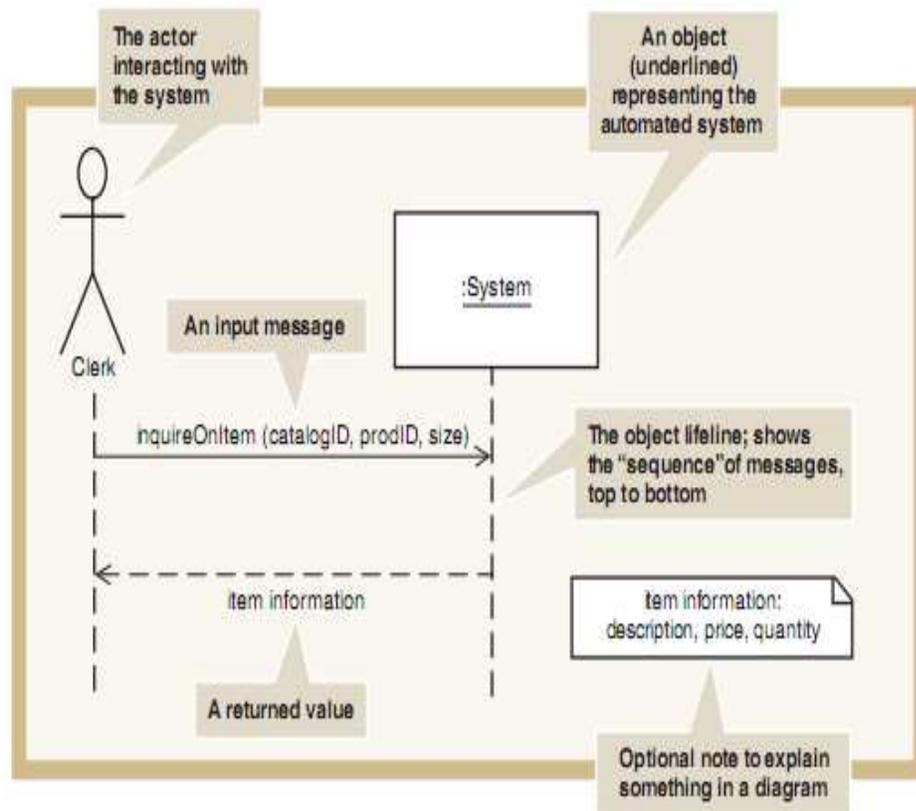
(Sumber : Satzinger, Jackson & Burd, 2005, p305)

- **System Sequence Diagram (SSD)**

Menurut Satzinger, Jackson & Burd (2005:226) mendefinisikan *system sequence diagram*, sebagai berikut :

*System Sequence Diagram* adalah *diagram* yang digunakan untuk mendeskripsikan aliran informasi *input* dan *output* dari sistem yang terotomatisasi. SSD menjelaskan data yang masuk dan data yang keluar dan mengidentifikasi interaksi antara actor dalam sistem.

Contoh *System Sequence Diagram* :



Gambar 2.11 *System Sequence Diagram*

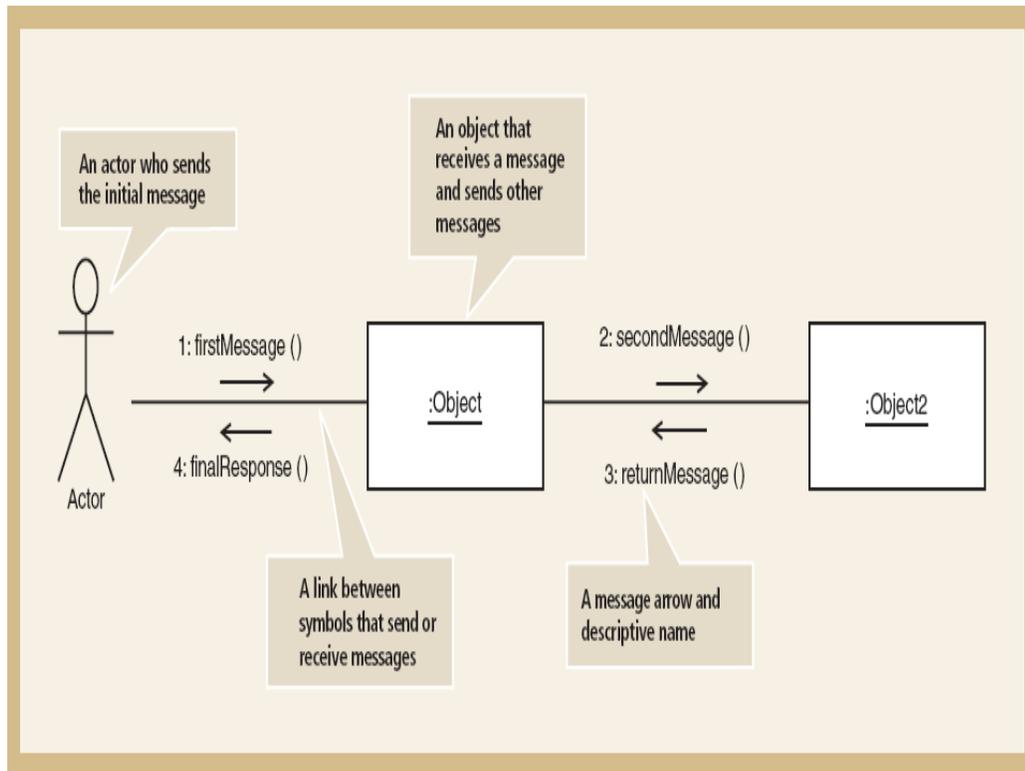
(Sumber : Satzinger, Jackson & Burd 2005, p229)

- ***Communication Diagram***

Menurut Satzinger, Jackson & Burd (2005:334) *communication diagram* berkaitan dengan *sequence diagram*, sebagai berikut :

Keduanya saling berinteraksi, dan mereka menangkap informasi yang sama. *Communication diagram* menggunakan *symbol* yang sama untuk *actor*, *object* dan pesan sebagai *Sequence diagram*. *Lifeline* dan aktivasi *lifeline* tidak digunakan. Namun, *symbol* yang berbeda, *link symbol* digunakan, yaitu sebagai berikut:

Contoh *Communication Diagram* :



Gambar 2.12 *Communication Diagram*

(Sumber : Satzinger, Jackson & Burd, 2005, p334)

*Syntax* pesan *descriptor* di *communication diagram* adalah sebagai berikut:

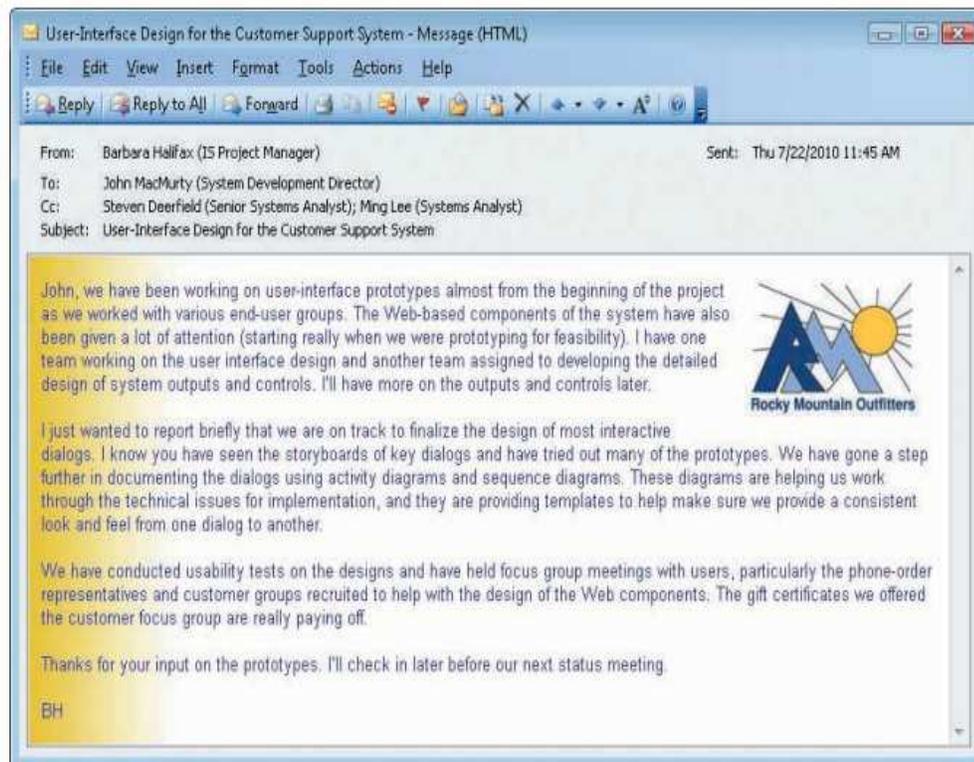
- *[true/false condition] sequence-number: return-value:= message-name (parameter-list)*
- *Connecting lines* antara *object* atau antara *actor* dan *object* membutuhkan *links*.
- Di *communication diagram*, sebuah *link* memberitahu bahwa 2 *items share a message* dimana yang satu mengirim *message* dan yang satu menerimanya. Koneksi garis esensialnya digunakan untuk membawa pesan.
- Dalam *communication diagram* harus ditambahkan angka pesan untuk mengindikasikan bahwa pesan telah dikirim. *Hirarki dot numbering scheme* digunakan ketika pesan bergantung pada pesan lain. Sebagai contoh, pesan utama sebagai berikut:
- 1: *startOrder (accountNo)*, dikirim ke *:OrderHandler*, dimana di-forward ke pesan yang sama, 1.1: *createOrder (accountNo)*, to *:Customer*. Pesan

yang kedua adalah pesan langsung dari pertama, jadi ditambahkan **1.1**, sebagai *subordinate* (bagian) dari *primary message*. Cara yang baik menentukan nomor pesan adalah dengan memiliki pesan pertama yang dimulai dari *external actor*.

*Communication diagram* merupakan cara yang efektif untuk mendapatkan overview dengan cepat dari *object* yang bekerja secara bersama-sama. Dalam *communication diagram* yang harus diperhatikan adalah nomornya dimana itu dipakai pada sequence sebagai pesan. *Communication diagram* sangat efektif untuk mengkolaborasi *object*.

- **User Interface**

Menurut Satzinger, Jackson & Burd (2005:441) *User Interface* adalah tempat dimana sistem informasi menangkap *input* dan menghasilkan *output*, serta terjadinya *input* dan *output* antara sistem dan lingkungannya.



Gambar 2.13 *User Interface*

(Sumber : Satzinger, Jackson, & Burd, 2005, p442)

Menurut Satzinger, Jackson & Burd (2005:454) untuk meningkatkan kegunaan dari sistem aplikasi penting untuk memiliki *interface* yang dirancang dengan baik.

Shneiderman, Plaisant, Colient & Jacobs (2010:88) mendeskripsikan panduan desain interaksi yang baik dalam “*The eight golden rules for designing interactive interface*”, yaitu :

#### 1. Konsistensi

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

#### 2. Memungkinkan pengguna untuk menggunakan *shortcut*

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

#### 3. Memberikan umpan balik yang informatif

Untuk setiap tindakan *operator*, sebaiknya disertakan suatu sistem umpan balik. Untuk tindakan yang sering dilakukan dan tidak terlalu penting, dapat diberikan umpan balik sederhana. Tetapi ketika tindakan merupakan hal yang penting, maka umpan balik sebaiknya lebih substansial. Misalnya muncul suatu suara ketika salah menekan tombol pada waktu *input* data atau muncul pesan kesalahan.

#### 4. Merancang dialog untuk menghasilkan suatu penutupan

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

#### 5. Memberikan penanganan kesalahan yang sederhana

Sedapat mungkin sistem dirancang sehingga pengguna tidak dapat melakukan kesalahan fatal. Jika kesalahan terjadi, sistem dapat mendeteksi kesalahan

dengan cepat dan memberikan mekanisme yang sederhana dan mudah dipahami untuk penanganan kesalahan.

#### 6. Mudah kembali ke tindakan sebelumnya

Hal ini dapat mengurangi kekhawatiran pengguna karena pengguna mengetahui kesalahan yang dilakukan dapat dibatalkan, sehingga pengguna tidak takut untuk mengeksplorasi pilihan – pilihan lain yang belum biasa dilakukan.

#### 7. Mendukung tempat pengendali *internal (internal locus of control)*

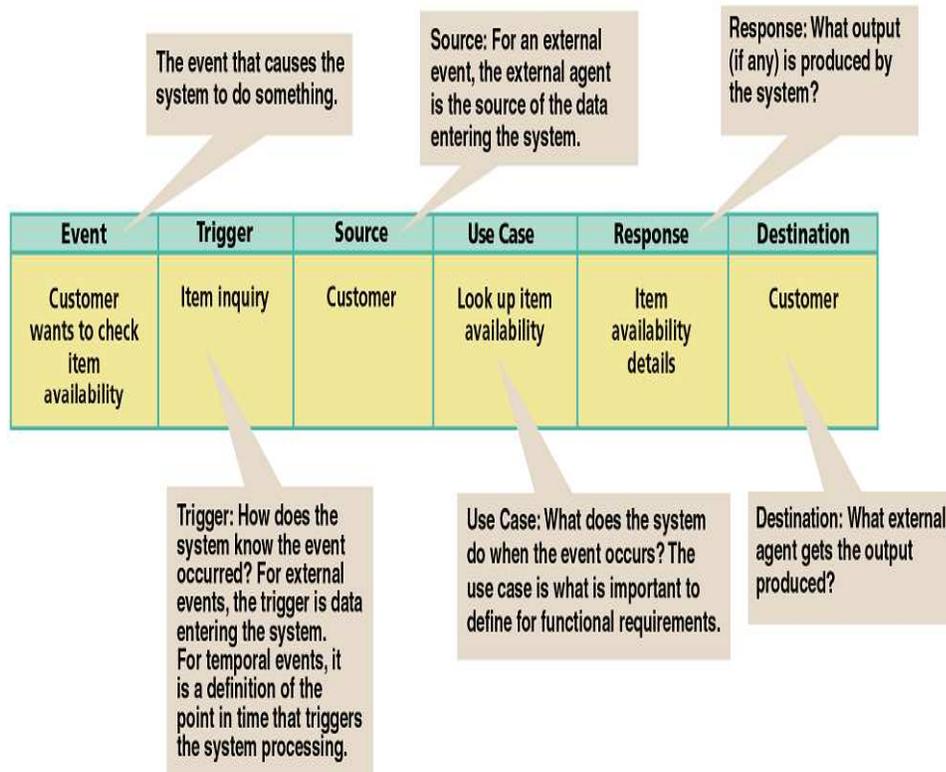
Pengguna ingin menjadi pengontrol sistem dan sistem akan merespon tindakan yang dilakukan pengguna daripada pengguna merasa bahwa sistem mengontrol pengguna. Sebaiknya sistem dirancang sedemikian rupa sehingga pengguna menjadi *inisiator* daripada responden.

#### 8. Mengurangi beban ingatan jangka pendek

Keterbatasan ingatan manusia membutuhkan tampilan yang sederhana atau banyak tampilan halaman yang sebaiknya disatukan, serta diberikan cukup waktu pelatihan untuk kode, *mnemonic*, dan urutan tindakan.

- **Event Table**

Menurut Satzinger, Jackson & Burd (2005:174), event table adalah sebuah pedoman use case daftar peristiwa dalam baris dan potongan kunci informasi setiap peristiwa dalam kolom. Di dalam event table terdapat tahapan yang harus dilakukan didalam kolom yakni berikut ini :



Gambar 2.14 *Event Tabel*

(Sumber : Satzinger, Jackson & Burd, 2005, p175)

- **Event**

Menurut Satzinger, Jackson & Burd (2005:174), event adalah kejadian pada waktu dan tempat tertentu, dapat digambarkan, dan harus diingat oleh sistem.

- **Trigger**

Menurut Satzinger, Jackson & Burd (2005:175), trigger adalah tanda yang memberitahukan sistem bahwa telah terjadi peristiwa. Untuk peristiwa eksternal, trigger merupakan datangnya data yang harus diproses oleh sistem. Contohnya,

ketika pelanggan melakukan pesanan, maka rincian pesanan baru sebagai input. Untuk peristiwa sementara, trigger merupakan titik waktu. Contohnya, pada akhir setiap hari kerja, sistem telah mengetahui waktu untuk menghasilkan laporan ringkasan transaksi.

- *Source*

Menurut Satzinger, Jackson & Burd (2005:175), source adalah agen eksternal yang memberikan data ke sistem.

- *Response*

Menurut Satzinger, Jackson & Burd (2005:175), response adalah output dari sistem. Ketika sistem menghasilkan laporan ringkasan transaksi, laporan tersebut merupakan outputs. Satu use case dapat menghasilkan beberapa responses. Contoh, ketika sistem membuat pesanan baru, maka konfirmasi pesanan diberikan kepada pelanggan, rincian pesanan diberikan kepada bagian pengiriman, dan catatan transaksi diberikan kepada bank.

- *Destination*

Menurut Satzinger, Jackson & Burd (2005:175), destination adalah tempat di mana beberapa response telah dikirim. Kadang-kadang use case tidak menghasilkan response sama sekali. Contoh, jika pelanggan ingin melakukan update informasi akun, informasi tersebut tersimpan dalam database, tapi tidak dibutuhkan output untuk dihasilkan. Mencatat informasi dalam database merupakan bagian dari use case.

- **Fully Develop Use Case Description**

Menurut Satzinger, Jackson & Burd (2005:221) use case description menjelaskan tentang suatu penggunaan kasus sistem yang mencakup seluruh urutan langkah untuk menyelesaikan suatu proses bisnis. Dan beberapa variasi dari langkah-langkah bisnis ada dalam kasus penggunaan tunggal.

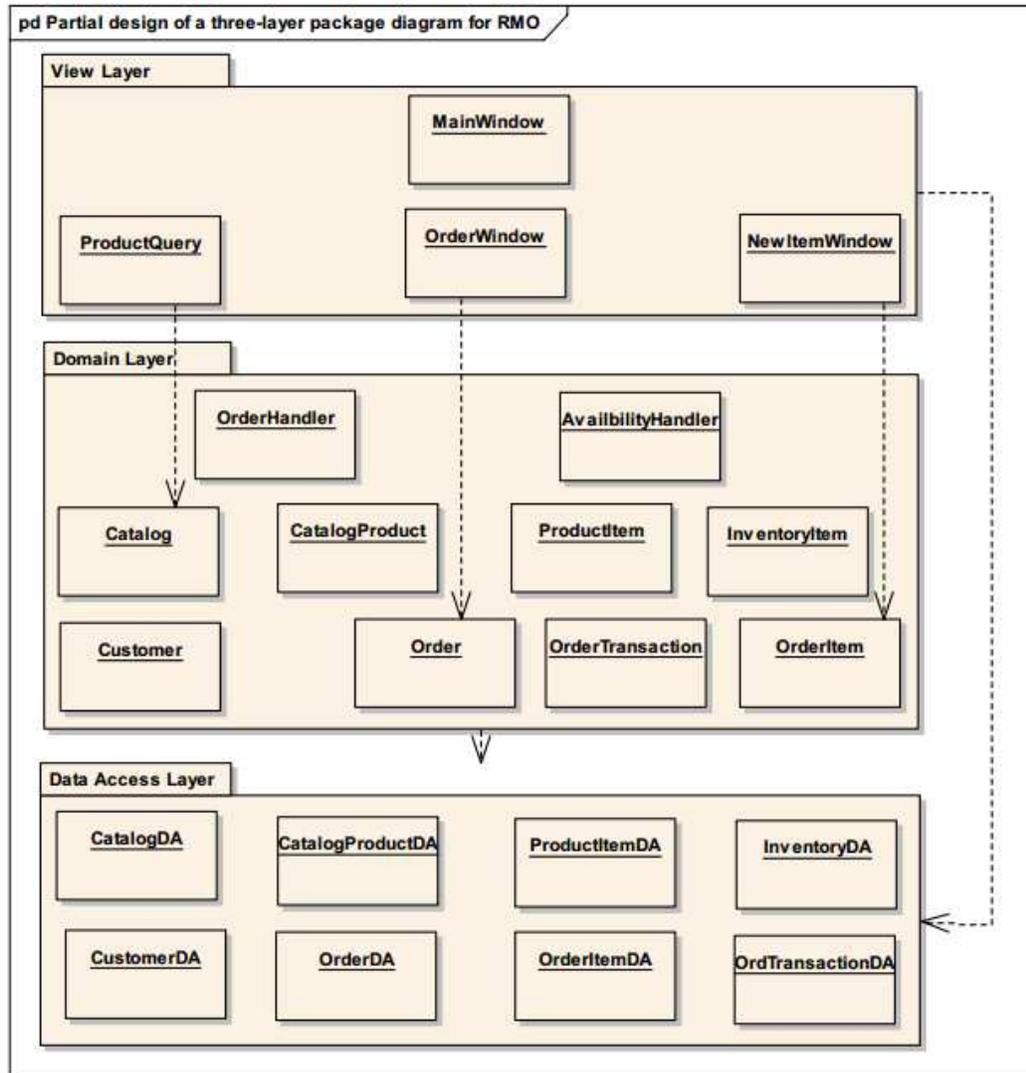
<b>Use Case Name:</b>	<i>Create new order</i>	
<b>Scenario:</b>	Create new telephone order	
<b>Triggering Event:</b>	Customer telephones RMO to purchase items from the catalog.	
<b>Brief Description:</b>	When customer calls to order, the order clerk and system verify customer information, create a new order, add items to the order, verify payment, create the order transaction, and finalize the order.	
<b>Actors:</b>	Telephone sales clerk.	
<b>Related Use Cases:</b>	Includes: <i>Check item availability</i> .	
<b>Stakeholders:</b>	Sales department: to provide primary definition. Shipping department: to verify information content is adequate for fulfillment. Marketing department: to collect customer statistics for studies of buying patterns.	
<b>Preconditions:</b>	Customer must exist. Catalog, Products, and Inventory items must exist for requested items.	
<b>Postconditions:</b>	Order and order line items must be created. Order transaction must be created for the order payment. Inventory items must have the quantity on hand updated. The order must be related (associated) to a customer.	
<b>Flow of Activities:</b>	<b>Actor</b>	<b>System</b>
	<ol style="list-style-type: none"> <li>1. Sales clerk answers telephone and connects to a customer.</li> <li>2. Clerk verifies customer information.</li> <li>3. Clerk initiates the creation of a new order.</li> <li>4. Customer requests an item be added to the order.</li> <li>5. Clerk verifies the item (<i>Check item availability</i> use case).</li> <li>6. Clerk adds item to the order.</li> <li>7. Repeat steps 4, 5, and 6 until all items are added to the order.</li> <li>8. Customer indicates end of order; clerk enters end of order.</li> <li>9. Customer submits payment; clerk enters amount.</li> </ol>	<ol style="list-style-type: none"> <li>2.1 Display customer information.</li> <li>3.1 Create a new order.</li> <li>5.1 Display item information.</li> <li>6.1 Add an order item.</li> <li>8.1 Complete order.</li> <li>8.2 Compute totals.</li> <li>9.1 Verify payment.</li> <li>9.2 Create order transaction.</li> <li>9.3 Finalize order.</li> </ol>
<b>Exception Conditions:</b>	<ol style="list-style-type: none"> <li>2.1 If customer does not exist, then the clerk pauses this use case and invokes <i>Maintain customer information</i> use case.</li> <li>2.2 If customer has a credit hold, then clerk transfers the customer to a customer service representative.</li> <li>4.1 If an item is not in stock, then customer can             <ol style="list-style-type: none"> <li>a. choose not to purchase item, or</li> <li>b. request item be added as a back-ordered item.</li> </ol> </li> <li>9.1 If customer payment is rejected due to bad-credit verification, then             <ol style="list-style-type: none"> <li>a. order is canceled, or</li> <li>b. order is put on hold until check is received.</li> </ol> </li> </ol>	

Gambar 2.15 Use Case Description

(Sumber : Satzinger, Jackson & Burd 2005, p223)

- **Package Diagram**

Menurut Satzinger, Jackson & Burd (2005:339), *Package diagram* adalah diagram tingkat lanjut yang memungkinkan *designer* mengasosiasikan beberapa kelas didalam kelompok yang berhubungan.



Gambar 2.16 *Package Diagram*

(Sumber : Satzinger, Jackson & Burd, 2005, p341 )

### 2.2.3 Flow Chart

Menurut Jogiyanto (2005:795) "Bagan alir (*flow chart*) adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika".

Ada 5 macam menurut jogiyanto (2010:795) bagan alir diantaranya;

a. Bagan alir sistem (*systems flow chart*)

Bagan alir sistem (*system flow chart*) merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Bagan menjelaskan urutan-urutan dari prosedur-prosedure yang ada dalam sistem. Bagan alir sistem menunjukkan apa yang dikerjakan sistem.

b. Bagan alir dokumen (*document flow chart*)

Bagan alir dokumen (*document flow chart*) atau disebut bagan alir formulir (*form flowchart*) atau paperwork flowchart merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya. Bagan alir dokumen ini menggunakan simbol-simbol yang sama dengan yang digunakan di dalam bagan alir sistem.

c. Bagan alir skematik (*schematic flow chart*)

Bagan alir skematik (*schematic flow chart*) merupakan bagan alir yang mirip dengan bagan alir sistem, yaitu untuk menggambarkan prosedur di dalam sistem. Perbedaannya adalah bagan alir skematik menggunakan simbol-simbol bagan alir sistem, juga menggunakan gambar - gambar komputer dan peralatan lainnya yang digunakan. Maksud penggunaan gambar-gambar ini adalah untuk memudahkan komunikasi kepada orang yang kurang paham dengan simbol-simbol bagan alir.

d. Bagan alir program (*program flow chart*)

Bagan alir program (*program flow chart*) merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program. Bagan alir program dibuat dengan menggunakan simbol-simbol sebagai berikut ini.

e. Bagan alir proses (*process flow chart*)

Bagan alir proses (*process flow chart*) merupakan bagan alir yang banyak digunakan teknik *industry*. Bagan alir juga berguna bagi analisis sistem untuk

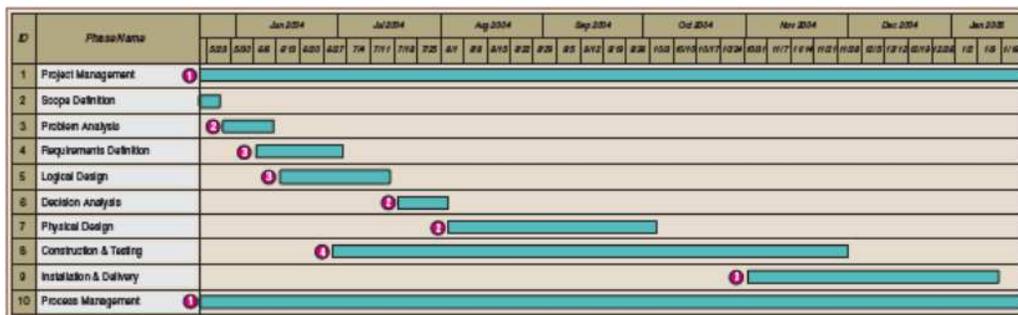
menggambarkan proses dalam suatu prosedur. Bagan alir proses menggunakan lima buah *symbol* tersendiri.

#### 2.2.4 *Persistent Object*

*Persistent object* menurut Satzinger, Jackson & Burd (2005:66) adalah obyek yang tersedia untuk dapat digunakan sepanjang waktu. *Persistent object* disebut juga *relational database* yang digunakan dalam bentuk tabel yang diisi atribut beserta dengan masing – masing nilai dari atribut tersebut. Di tiap tabel tersebut memiliki *primary key* , dimana *primary key* tersebut merupakan sebuah atribut yang unik.

#### 2.2.5 *Gantt Chart*

Whitten & Bentley (2007:73) mengemukakan penggambaran aktivitas dan *phase* pengembangan atau pembangunan sistem dapat dipaparkan melalui *gantt chart*.



Gambar 2.17 *Gantt Chart*

(Sumber : Whitten & Bentley, 2007, p73)

#### Karakteristik Gantt Chart

1. Gantt chart secara luas dikenal sebagai alat fundamental dan mudah diterapkan oleh para manajer proyek untuk memungkinkan seseorang melihat dengan mudah waktu dimulai dan selesainya tugas-tugas dan sub- sub tugas dari proyek.
2. Semakin banyak tugas-tugas dalam proyek dan semakin penting urutan antara tugas-tugas maka semakin besar kecenderungan dan keinginan untuk memodifikasi gantt chart.
3. Gantt chart membantu menjawab pertanyaan-pertanyaan “what if” saat melihat kesempatan-kesempatan untuk membuat perubahan terlebih dahulu terhadap kebutuhan.

**Keuntungan menggunakan *Gantt chart* :**

- Sederhana, mudah dibuat dan dipahami, sehingga sangat bermanfaat sebagai alat komunikasi dalam penyelenggaraan proyek.
- Dapat menggambarkan jadwal suatu kegiatan dan kenyataan kemajuan sesungguhnya pada saat pelaporan
- Bila digabungkan dengan metoda lain dapat dipakai pada saat pelaporan

**Kelemahan *Gantt Chart* :**

- Tidak menunjukkan secara spesifik hubungan ketergantungan antara satu kegiatan dan kegiatan yang lain, sehingga sulit untuk mengetahui dampak yang diakibatkan oleh keterlambatan satu kegiatan terhadap jadwal keseluruhan proyek.
- Sulit mengadakan penyesuaian atau perbaikan/pembaharuan bila diperlukan, karena pada umumnya ini berarti membuat bagan balok baru.