

## BAB 2

### LANDASAN TEORI

#### 2.1 Teori Umum

##### 2.1.1 *Internet*

*Internet (interconnected Network)* adalah jaringan komputer yang menghubungkan antara jaringan secara global, *internet* dapat juga disebut jaringan dalam suatu jaringan yang luas. Seperti halnya jaringan komputer local maupun jaringan komputer area, *internet* juga menggunakan protocol komunikasi yang sama yaitu TCP/IP (*Transmission Control Protocol / Internet Protocol*). Struktur protocol *internet* RFC-1122 terdiri dari 4 lapisan (*Layer*) yaitu :

a) *Application Layer*.

Lapisan Aplikasi (*Application Layer*) adalah lapisan yang berisi semua protocol dan metode yang mengatur komunikasi antar proses layanan komunikasi. Lapisan aplikasi terdiri dari : BGP, DHCP, FTP, HTTP, IMAP, LDAP, IRC, POP, RPC, SMTP, dan lain-lain.

b) *Transport Layer*

Lapisan transport (*Transport Layer*) adalah lapisan yang mengatur lalu lintas pengiriman maupun penerimaan data. Lapisan transport terdiri dari : TCP, UDP.

c) *Internet Layer*

Lapisan *Internet(Internet Layer)* adalah lapisan yang berfungsi mengatur proses fragmentasi dan defragmentasi paket

bergantung dari jenis protocol lapisan *internet*. Lapisan *internet* terdiri dari : IP (Ipv4,Ipv6), ICMP, IPSec.

d) *Link Layer*

Lapisan penghubung (*Link Layer*) adalah lapisan yang mengatur hubungan antara *host* dan *node*. *Host* sebagai pengirim dan *node* adalah lokasi tujuan dari *host*. Lapisan penghubung terdiri dari : ARP, *Tunnel* (L2TP), PPP, *Media Access Control* (*Ethernet*, DSL, ISDN). (Sibero, 2011).

### 2.1.2 W WW (World Wide Web)

Www (*World Wide Web*) atau yang dikenal juga dengan istilah web adalah suatu yang berkaitan dengan dokumen digunakan sebagai media untuk menampilkan teks, gambar, multimedia, dan lainnya pada jaringan *internet* (Sibero, 2011).

### 2.1.3 Definisi E-Commerce

*E-Commerce* dapat didefinisikan sebagai metodologi modern yang membahas kebutuhan organisasi, pedagang, dan *customer* untuk memotong biaya sambil meningkatkan kualitas barang dan jasa serta meningkatkan kecepatan pelayanan dengan menggunakan *internet*. hal itu berbeda dari perdagangan elektronik tradisional dalam cara yang memungkinkan perdagangan barang, uang, dan informasi secara elektronik dari komputer ke komputer. Bisnis dilakukan secara elektronik dan tidak lagi menjadi kebutuhan untuk mata uang fisik atau barang untuk melakukan bisnis. Melakukan bisnis *online* adalah

perniagaan elektronik, dan ada 4 bidang dimana perusahaan melakukan bisnis *online* sekarang ini:

1. *Marketing*, penjualan, dan layanan secara langsung.
2. *Online banking* dan penagihan.
3. Pengamanan informasi distribusi.
4. Rantai nilai perdagangan dan pembelian perusahaan.

(Goel, 2007, p1-4)

#### **2.1.4 Interaksi Manusia dan Komputer (IMK)**

Menurut Shneiderman dan Plaisant (2010, p22), interaksi manusia dan komputer adalah ilmu yang mempelajari implementasi dan perancangan tampilan *interface* yang dapat digunakan oleh *user* untuk berinteraksi dengan komputer.

#### **2.1.5 Faktor Manusia Terukur**

Lima faktor manusia terukur yang dapat mendukung perancangan *interface*, yaitu (Shneiderman & Plaisant, 2010, p32):

1. *Time to Learn* (Waktu belajar)

Waktu yang dibutuhkan oleh *user* untuk dapat memahami sistem.

2. *Speed of Performance* (Kecepatan kinerja)

Kecepatan *user* dalam menjalankan aksi-aksi dalam sistem.

3. *Rate of Error* (Tingkat Kesalahan)

Tingkat banyaknya kesalahan dan jenis kesalahan apa yang dilakukan oleh *user*. Pengendalian kesalahan merupakan komponen kritis dalam pembuatan *interface*.

4. *Retention Over Time* (Daya ingat)

Kemampuan *user* untuk mengingat informasi yang diterima dari sistem dalam jangka waktu tertentu.

5. *Subjective Satisfaction* (Kepuasan subjektif)

Keberhasilan suatu sistem yang dirancang, diukur melalui kepuasan *user* terhadap sistem tersebut.

### **2.1.6 Delapan Aturan Emas**

Menurut Shneiderman dan Plaisant (2010, p88 - 89) terdapat delapan aturan emas yang harus diaplikasikan dalam merancang *interface*:

1. Konsistensi

Dalam perancangan *interface*, diperlukan konsistensi pada setiap aksi yang dilakukan. Contohnya seperti tampilan navigasi pada sebuah *website*, ketika berpindah ke halaman lain tetap sama (*font, color, layout*, dan apapun yang digunakan didalamnya).

2. Memenuhi kegunaan yang *universal*

Perancangan fitur serta penggunaan simbol harus dapat dimengerti oleh semua kalangan *user*. Seperti penyediaan

fitur untuk *user* awam seperti *tutorial* dan penyediaan fitur untuk *user* mahir seperti *shortcut*.

3. Memberikan umpan balik informatif

Adanya respons dari sistem yang mengarahkan *user* dari setiap aksi yang dilakukan agar dapat lebih memahami instruksi yang sedang dilakukan. Respons dapat berupa umpan balik dari sistem terhadap *input* yang dimasukkan *user*.

4. Merancang dialog yang memberikan keadaan akhir

Urutan aksi dikelompokkan kedalam tiga kategori, yaitu aksi permulaan, pertengahan dan penutup. Pada saat *user* melakukan aksi penutup suatu proses, kotak dialog seharusnya muncul untuk menegaskan bahwa *user* telah selesai melakukan proses tersebut.

5. Penanganan kesalahan

Sistem yang dirancang diberikan fitur penanganan kesalahan untuk mencegah *user* melakukan kesalahan yang fatal. Contohnya : apabila *user* menekan tombol *caps lock* ketika melakukan pengisian kolom *password*, sebaiknya muncul peringatan kepada *user* bahwa ia sedang berada dalam kondisi *caps lock*.

6. Memungkinkan pengembalian aksi yang mudah

Sistem sebaiknya menyediakan fitur yang memungkinkan *user* untuk membatalkan aksi yang sudah dilakukannya

dengan mudah. Kekhawatiran *user* dapat dikurangi dengan adanya fitur ini.

7. Mendukung pusat kendali internal

Menjadikan *user* sebagai pemegang kendali dari sistem bukan sistem yang mengendalikan *user*. Ketidakmampuan *user* memperoleh informasi penting akan memberikan ketidakpuasan kepada *user*.

8. Mengurangi beban ingatan jangka pendek

Manusia memiliki batasan waktu untuk mampu mengolah informasi. Maka dibutuhkan rancangan tampilan yang sederhana agar informasi mudah dicerna.

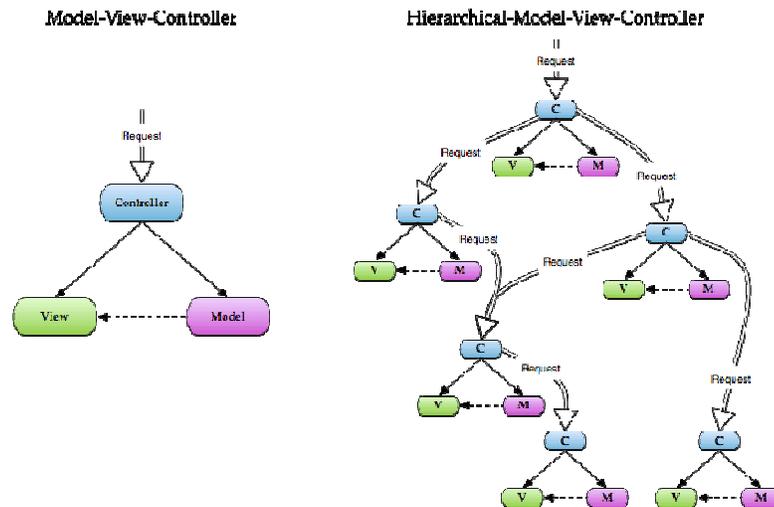
### **2.1.7 Object Oriented Analysis and Design**

*Object Oriented Analysis and Design (OOAD)* adalah Sekumpulan dari alat-alat dan tehnik untuk pengembangan sistem yang akan digunakan untuk meng-utilisasi teknologi objek untuk membangun sebuah sistem beserta piranti lunak nya. (Whitten,Bentley, 2007).

### **2.1.8 Hierarchical Model View Controller (HMVC)**

*Hierarchical Model View Controller (HMVC)* adalah ekstensi langsung ke pola MVC yang berhasil memecahkan banyak masalah skalabilitas yang telah disebutkan. HMVC pertama kali dijelaskan dalam *posting* blog berjudul HMVC: Pola berlapis untuk mengembangkan tingkatan *client* yang kuat di situs web *JavaWorld*.

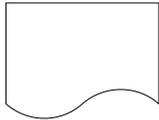
pada bulan Juli 2000. Sebagian besar artikel berkonsentrasi pada manfaat menggunakan HMVC dengan antarmuka pengguna grafis. (Freyssinet, 2010)

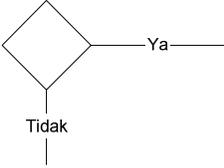
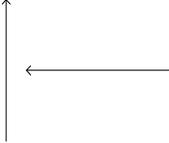


**Gambar 2.1** Hierarchical Model View Controller (HMVC)

### 2.1.9 DAD

Menurut Mulyadi (2001,pp58-63), diagram aliran dokumen adalah suatu model yang menggambarkan aliran dokumen dan proses untuk mengolah dokumen dalam suatu proses. Berikut ini adalah tabel yang menjelaskan komponen-komponen dari diagram aliran dokumen :

Simbol	Keterangan
	<p><b>Dokumen</b></p> <p>Simbol ini digunakan untuk menggambarkan semua jenis dokumen, yang merupakan formulir untuk merekam data terjadinya suatu transaksi.</p>

Simbol	Keterangan
	<p><b>Keputusan</b></p> <p>Simbol ini menggambarkan keputusan yang harus dibuat dalam proses pengolahan data. Keputusan yang dibuat ditulis dalam simbol.</p>
	<p><b>Garis Alir</b></p> <p>Simbol ini menggambarkan arah proses pengolahan data.</p>
	<p><b>Persimpangan Garis Alir</b></p> <p>Jika dua garis alir bersimpangan, untuk menunjukkan arah masing-masing garis, salah satu garis dibuat sedikit melengkung tepat pada persimpangan kedua garis tersebut.</p>
	<p><b>Pertemuan Garis Alir</b></p> <p>Simbol ini digunakan jika dua garis alir bertemu dan salah satu garis mengikuti garis lainnya.</p>
	<p><b>Proses</b></p> <p>Simbol ini untuk menunjukkan tempat-tempat dalam sistem informasi yang mengolah atau mengubah data yang diterima menjadi data yang mengalir keluar. Nama pengolahan data ditulis didalam simbol.</p>
	<p><b>Mulai / Berakhir (terminal)</b></p> <p>Simbol ini untuk menggambarkan awal dan akhir suatu sistem akuntansi</p>

**Gambar 2.2** Komponen diagram aliran dokumen

### 2.1.10 *Storyboard*

Menurut Dastbaz (2003, p134), *storyboard* dibuat untuk menyampaikan dan memvisualisasikan *user interface*. Dengan adanya *storyboard* maka akan didapatkan *feedback* dari para klien dan user yang nantinya akan menunjang implementasi dari prototype sistem tersebut.

*Storyboard* merupakan teknik yang sangat ampuh untuk memvisualisasikan rancangan *interface* sebuah sistem interaktif *multimedia* sebelum sistem tersebut diimplementasikan secara total

### 2.1.11 *Waterfall Model*

*Waterfall Model* menyarankan pendekatan yang sistematis dan sekuensial dalam pengembangan piranti lunak. Tahapan – tahapan yang ada pada *Waterfall Model* yang dilakukan secara berurutan sebagai berikut:

- *Communication*

Kegiatan ini melibatkan komunikasi dan kerjasama dengan pelanggan dan mencakup pada pengumpulan kebutuhan dan kegiatan lain yang terkait.

- *Planning*

Membuat rencana untuk pekerjaan rekayasa piranti lunak yang menjelaskan tentang teknikal yang harus dilaksanakan, resiko yang kelak mungkin terjadi, sumber daya yang akan diperlukan, produk yang akan diproduksi, dan jadwal kerja.

- *Modelling*

Pembuatan model yang membantu pengembang dan pelanggan untuk lebih mengerti kebutuhan piranti lunak dan rancangan yang menggambarkan kebutuhan itu.

- *Construction*

Menggabungkan *code generation* dan pengujian yang diperlukan untuk menemukan *error* dalam *code*.

- *Deployment*

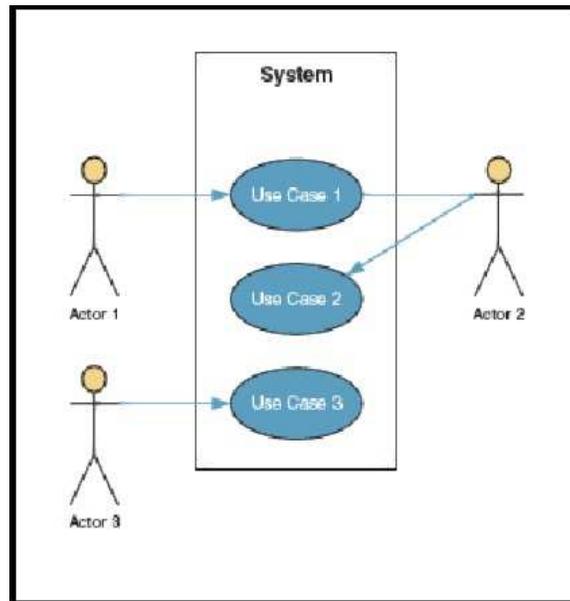
Piranti lunak dikirim ke pelanggan yang akan mengevaluasi produk yang dikirim dan memberi umpan balik berdasarkan evaluasi tersebut. (Pressman, 2005)

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

Menurut Whitten dan Bentley (2007, p371), UML adalah kumpulan model yang digunakan untuk menggambarkan hubungan atau interaksi antara suatu sistem terhadap suatu objek atau *user*. UML terdiri atas beberapa diagram yang membahas bagaimana *user* berinteraksi dengan sistem, bagaimana *user* menjalankan aktivitas-aktivitasnya, bagaimana aliran aktivitas *user* berjalan sehingga tujuan kegiatan tersebut tercapai.

Whitten dan Bentley (2007, p382), jenis-jenis diagram UML:

## 1. Use Case Diagram.



**Gambar 2.3** Use Case Diagram

(Whitten & Bentley, 2007, p246)

Menurut Whiten dan Bentley (2007, p382), *Use Case Diagram* menggambarkan interaksi yang terjadi antara sistem dan sistem, eksternal dan *user*. Dengan kata lain, diagram ini menjelaskan dengan nyata tentang siapa yang akan menggunakan sistem dan bagaimana harapan *user* untuk berinteraksi dengan sistem.

### a. Actor

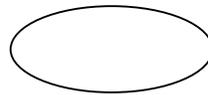
Menspesifikasikan seperangkat peran yang *user* dapat perankan ketika berinteraksi dengan sistem.



**Gambar 2.4** Simbol *Actor* dalam *Use Case Diagram*

b. *Use Case*

Mendeskripsikan seperangkat aksi-aksi berurutan yang ditampilkan dalam sebuah sistem.



**Gambar 2.5** Simbol *Use Case* dalam *Use Case Diagram*

c. *Association*

Sebuah relasi *structural* yang menghubungkan antara *actor* dengan *Use Case*.



**Gambar 2.6** Simbol *Association* dalam *Use Case Diagram*

d. Sistem *Boundary*

Menggambarkan batasan dari sistem yang akan dibuat yang mengelilingi sejumlah *Use Case*.



**Gambar 2.7** Simbol Sistem *Boundary* dalam  
*Use Case Diagram*

e. *Use case Narrative*

Menurut Whitten & Bentley (2007, p265), tingkat penggunaan *use case* lebih tinggi dibandingkan dengan *use case diagram*. *Use case narrative* digunakan sebagai tambahan dalam bentuk teks untuk menjelaskan urutan-urutan langkah dari setiap interaksi.

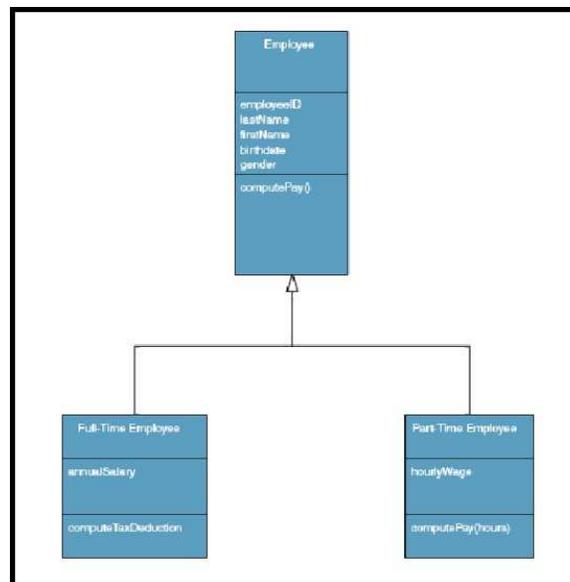
**Tabel 2.1.12.1** Tabel *Use Case Narrative*

Nama <i>Use Case</i>	
ID <i>Use Case</i>	
Sumber	
Pelaku Bisnis Utama	
Pelaku Sistem Utama	

Deskripsi		
Prakondisi		
Kegiatan	Kegiatan Pelaku	Respon Sistem
	.	
Kesimpulan		
Postkondisi		

**Gambar 2.8** *Use case Narrative*

## 2. Class Diagram

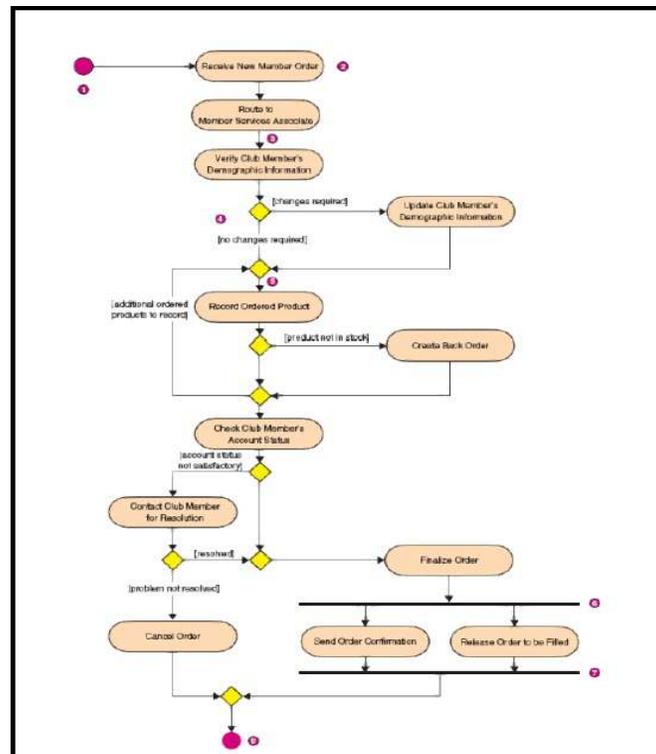


**Gambar 2.9** *Class Diagram* (Whitten & Bentley, 2007, p381)

Diagram kelas menggambarkan struktur dari *user* atau objek, sebagai contoh suatu *user* memiliki atribut-atribut seperti nama, umur, jenis kelamin, dan lain lain.

Selain memiliki atribut, *user* atau objek juga memiliki kegiatan atau fungsi yang dapat dilakukan terhadap sistem.

### 3. Activity diagram

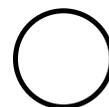


**Gambar 2.10** Activity diagram (Whitten dan Bentley, 2007, p392)

Pada diagram aktivitas, kegiatan *user* atau objek yang memiliki atribut atau fungsi dijelaskan secara detail. Penjelasan meliputi rangkaian fungsi yang dilakukan oleh *user* atau *actor* dari awal hingga akhir kegiatan sampai tujuan *user* tersebut tercapai.

#### a. Initial Node

Merepresentasikan awal dari proses.



**Gambar 2.11** Simbol *Initial Node* dalam  
*Activity diagram*

b. *Actions*

Merepresentasikan tahap individual.

Memperlihatkan banyaknya aktivitas yang terjadi dalam diagram.



**Gambar 2.12** Simbol *Actions* dalam *Activity diagram*

c. Alur

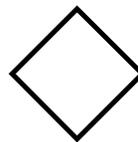
Mengindikasikan pergerakan yang melewati *action*.



**Gambar 2.13** Simbol Alur dalam *Activity diagram*

d. *Decision*

Menunjukkan kondisi pengambilan keputusan dari sebuah aktivitas yang harus dipilih.



**Gambar 2.14** Simbol *Decision* dalam *Activity diagram*

e. *Activity Final*

Menunjukkan titik akhir dari aktivitas. *Activity diagram* bisa mempunyai banyak *Activity final nodes*, atau tidak ada sama sekali.



**Gambar 2.15** Simbol *Activity Final* dalam *Activity diagram*

### 2.1.13 Database Management System

DBMS adalah sebuah piranti lunak yang memungkinkan pengguna (user) untuk mendefinisikan, membuat, merawat, dan mengatur akses ke *basisdata*. DBMS juga memperbolehkan pengguna untuk mendefinisikan *database* melalui *Data Definition Language* (DDL) dan juga untuk memasukkan, merubah, dan mengambil data dari *database* melalui *Data Manipulasi Language*. (Connolly & Beg, 2005).

**Tabel 2.1.13.1** Verifikasi Studi Literatur

No.	Nama Proses	Masalah	Verifikasi Studi Literatur
1.	Akses <i>website tracking</i> dan cek tariff.	Belum memiliki <i>website</i> dengan fitur cek tariff dan cek status pengiriman	Website adalah salah satu teknologi saat ini yang mampu menyebarkan informasi yang dibutuhkan <i>customer</i> keseluruh negeri ini.

No.	Nama Proses	Masalah	Verifikasi Studi Literatur
2.	Pencatatan Data Transaksi	Manual atau tulis tangan.	Administrasi data adalah fungsi tingkat tinggi yang bertanggung jawab untuk manajemen sumber daya data secara keseluruhan dalam sebuah organisasi. (Hoffer, Presscott, Topi, 2009,p.599).
3.	Penyimpanan data Transaksi	Kehilangan data	<i>Internet</i> dan sistem komputer terhubung dan memproses informasi digital. Informasi Digital bisa dengan mudah disimpan, ditransmisikan, diproses, digabungkan,dan diubah, singkatnya dapat dimanipulasi dengan banyak cara, tidak tergantung dari sumbernya. (Timmer,2000,p.15)

**Tabel 2.1.13.2** *Verifikasi* dengan Landasan Teori

No.	Permasalahan Yang Diidentifikasi	Sumber Identifikasi Permasalahan	Verifikasi Dengan Landasan Teori
1.	Belum tersedianya <i>website tracking</i> dan cek <i>tariff</i> , sebagai sarana informasi dan promosi yang memadai.	Sumber identifikasi permasalahan didapatkan dari wawancara dan pengamatan langsung di lapangan.	<i>E-Commerce</i> adalah proses pembelian dan penjualan serta pemasaran dan pelayanan dari produk, jasa, dan informasi melalui jaringan komputer yang beragam (O'Brien,2003,p.23).
2.	Penyimpanan data transaksi yang tidak baik	Sumber identifikasi permasalahan didapatkan dari wawancara dan pengamatan langsung di lapangan.	Administrasi data adalah fungsi tingkat tinggi yang bertanggung jawab untuk manajemen sumber daya data secara keseluruhan dalam sebuah organisasi. (Hoffer, Prescott, Topi, 2009,p.599)

**Tabel 2.1.13.3** Verifikasi dengan Landasan Teori

No.	Permasalahan Yang Diidentifikasi	Solusi Untuk Permasalahan	Verifikasi Dengan Landasan Teori
1.	Belum tersedianya <i>website tracking</i> dan cek <i>tariff</i> , sebagai sarana informasi dan promosi yang memadai.	Berdasarkan hasil wawancara dan pengamatan langsung ke lapangan, pihak <i>Avatar Kargo Indonesia</i> masih menggunakan media brosur dan <i>door to door</i> sebagai media informasi. Solusi yang dapat dilakukan adalah pembuatan <i>website tracking</i> dan cek <i>tariff</i> serta informasi jasa yang dimiliki perusahaan <i>Avatar Kargo Indonesia</i> sebagai penyedia layanan jasa.	<i>E-Commerce</i> adalah proses pembelian dan penjualan serta pemasaran dan pelayanan dari produk, jasa, dan informasi melalui jaringan komputer yang beragam (O'Brien, 2003,p.23).

No.	Permasalahan Yang Diidentifikasi	Solusi Untuk Permasalahan	Verifikasi Dengan Landasan Teori
2.	Penyimpanan data transaksi	Berdasarkan hasil wawancara dan pengamatan langsung ke lapangan, pihak <i>Avatar Kargo Indonesia</i> saat ini penyimpanan data transaksi masing-masing sebatas penyimpanan pada <i>hard disk</i> . Penyimpanan ini masih memiliki resiko tinggi dalam kehilangan, misalnya terjadi kerusakan pada <i>hard disk</i> tersebut. Solusi yang dapat dilakukan adalah dengan pembuatan <i>website</i> yang	Administrasi data adalah fungsi tingkat tinggi yang bertanggung jawab untuk manajemen sumber daya data secara keseluruhan dalam sebuah organisasi. (Hoffer, Presscott, Topi, 2009,p.599)

No.	Permasalahan Yang Diidentifikasi	Solusi Untuk Permasalahan	Verifikasi Dengan Landasan Teori
		<p>memiliki <i>database</i>, data akan tersimpan pada <i>server database</i>. Ini untuk menanggulangi resiko kerusakan <i>hard disk</i> yang terjadi sewaktu-waktu sehingga data tidak akan hilang karena sudah memiliki <i>backup</i> pada <i>server</i> dan <i>database</i> dapat diakses computer manapun selama menggunakan <i>server database</i>.</p>	

## 2.2 Teori khusus

### 2.2.1 CodeIgniter

*CodeIgniter* adalah aplikasi *open source* yang berupa *framework* dengan model MVC (*Model, View, Controller*) untuk membangun *website* dinamis. Dengan menggunakan PHP *CodeIgniter*

akan memudahkan *developer* untuk membuat aplikasi *website* dengan cepat dan mudah dibandingkan dengan membuat dari awal. (Septian, 2011, p9).

### 2.2.2 *PHP*

*PHP* memiliki kepanjangan *PHP: Hypertext Processor* merupakan suatu bahasa pemrograman yang difungsikan untuk membangun suatu *website* dinamis. *PHP* berjalan pada sisi *server*, sehingga *PHP* disebut juga sebagai bahasa *Server Side Scripting*. Artinya bahwa dalam setiap / untuk menjalankan *PHP*, wajib menggunakan *web server* dalam menjalakkannya. (Saputra, Agustin, 2012,p2)

### 2.2.3 *MySQL*

*SQL* adalah sebuah bahasa komputer yang digunakan untuk mengelola dan berinteraksi dengan data dalam database relasional. *SQL* adalah bahasa yang paling universal database diimplementasikan digunakan, dan telah menjadi bahasa standar untuk manajemen database. *SQL* bekerja sama dengan *RDBMS* untuk menentukan struktur database, menyimpan data dalam database itu, memanipulasi data, mengambil data, mengontrol akses ke data, dan menjamin integritas data. meskipun bahasa lain telah dikembangkan untuk mengimplementasikan model relasional, *SQL* telah muncul sebagai pemenang (Sheldon,Moes, 2005, p10).

#### 2.2.4 *HTML5*

Menurut Pilgrim (2010, p9), *HTML5* adalah generasi berikutnya dari *HTML*, menggantikan *HTML 4.01*, *XHTML 1.0*, dan *XHTML 1.1*. *HTML5* menyediakan *fitur* baru yang diperlukan untuk aplikasi web modern. *HTML5* merupakan standarisasi *fitur* dari *platform web* yang telah digunakan oleh *web developers* selama bertahun-tahun, tetapi belum pernah diperiksa atau didokumentasikan oleh *standards committee*.

#### 2.2.5 **Grocery CRUD**

Menurut Skoumbourdis (2012) Grocery CRUD adalah librari untuk CodeIgniter berupa generator CRUD(Create, Read, Update, Delete) otomatis yang menjangkau hampir seluruh kebutuhan sistem CRUD(sederhana maupun kompleks). yang didukung banyak fitur-fitur, seperti.

- *Data grid listing* dengan *paging*, *sorting*, *searching by field* dan *search*.
- Membuat *field-field input* secara otomatis.
- Relasi basisdata.
- Validasi form dengan *client side* dan *server side*.
- Multi bahasa