

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Teori Pendukung**

Landasan teori/kajian pustaka yang penulis pakai dalam pembuatan sistem adalah sebagai berikut :

##### **2.1.1 E-Commerce**

E-commerce atau bisa disebut Perdagangan elektronik atau e-dagang adalah penyebaran, pembelian, penjualan, pemasaran barang dan jasa melalui sistem elektronik seperti internet atau televisi, www, atau jaringan komputer lainnya. E-commerce dapat melibatkan transfer dana elektronik, pertukaran data elektronik, sistem manajemen inventori otomatis, dan sistem pengumpulan data otomatis.

Industri teknologi informasi melihat kegiatan e-dagang ini sebagai aplikasi dan penerapan dari e-bisnis (e-business) yang berkaitan dengan transaksi komersial, seperti: transfer dana secara elektronik, SCM (supply chain management), e-pemasaran (e-marketing), atau pemasaran online (online marketing), pemrosesan transaksi online (online transaction processing), pertukaran data elektronik (electronic data interchange /EDI), dll.

E-dagang atau e-commerce merupakan bagian dari e-business, di mana cakupan e-business lebih luas, tidak hanya sekedar perniagaan tetapi mencakup juga pengkolaborasi mitra bisnis, pelayanan nasabah, lowongan pekerjaan dll.

Selain teknologi jaringan www, e-dagang juga memerlukan teknologi basisdata atau pangkalan data (databases), e-surat atau surat elektronik (e-mail), dan bentuk teknologi non komputer yang lain seperti halnya sistem pengiriman barang, dan alat pembayaran untuk e-dagang ini.

E-commerce pertama kali diperkenalkan pada tahun 1994 pada saat pertama kali banner-elektronik dipakai untuk tujuan promosi dan periklanan di suatu halaman-web (website). Menurut Riset Forrester, perdagangan elektronik menghasilkan penjualan seharga AS\$12,2 milyar pada 2003. Menurut laporan yang lain pada bulan oktober 2006 yang lalu, pendapatan ritel online yang bersifat non-travel di Amerika Serikat diramalkan akan mencapai seperempat trilyun dolar US pada tahun 2011.

Dalam banyak kasus, sebuah perusahaan e-commerce bisa bertahan tidak hanya mengandalkan kekuatan produk saja, tapi dengan adanya tim manajemen yang handal, pengiriman yang tepat waktu, pelayanan yang bagus, struktur organisasi bisnis yang baik, jaringan infrastruktur dan keamanan, desain situs web yang bagus, beberapa faktor yang termasuk:

1. Menyediakan harga kompetitif
2. Menyediakan jasa pembelian yang tanggap, cepat, dan ramah.
3. Menyediakan informasi barang dan jasa yang lengkap dan jelas.
4. Menyediakan banyak bonus seperti kupon, penawaran istimewa, dan diskon.
5. Memberikan perhatian khusus seperti usulan pembelian.

6. Menyediakan rasa komunitas untuk berdiskusi, masukan dari pelanggan, dan lain-lain.
7. Mempermudah kegiatan perdagangan

Perusahaan yang terkenal dalam bidang ini antara lain: eBay, Yahoo, Amazon.com, Google, dan Paypal. Untuk di Indonesia, bisa dilihat tradeworld.com, bhineka.com, fastncheap.com, dll.

Faktor Pendukung E-Commerce :

1. Cakupan yang luas .Proses transaksi yang cepat.
2. E-Commerce dapat mendorong kreatifitas dari pihak penjual secara cepat dan tepat dan pendistribusian informasi yang disampaikan berlangsung secara periodik.
3. E-Commerce dapat menciptakan efesiensi yang tinggi, murah serta informatif.
4. *E-Commerce* dapat meningkatkan kepuasan pelanggan, dengan pelayanan yang cepat, mudah, aman dan akurat.

Karakteristik *E-Commerce*:

1. Terjadinya transaksi antar dua belah pihak.
2. Adanya pertukaran barang, jasa dan informasi.
3. Internet merupakan medium utama dalam proses atau mekanisme perdagangan tersebut.

Jenis-jenis Transaksi *E-commerce* : *Busines to Busines* (B2B).*Bussines to Cunsumer* (B2C), *Consumer to Consumer* (C2C), *Consumer to Bussines*

(C2B), *Non-Bussines Electronic Commerce, Intrabussines (Organizational) Electronic Commerce.*

### 2.1.2 UML (*Unified Modelling Language*)

Menurut Munawar (2005, hal 17) *Unified Modelling language (UML)* adalah suatu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek, UML menyediakan pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi dan mengkomunikasikan rancangan mereka dengan yang lain.

Dengan menggunakan UML, dapat membuat model untuk semua jenis aplikasi peranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, dan dapat digunakan juga untuk mendefinisikan notasi dan *syntax*. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram-diagram piranti lunak, setiap bentuk mempunyai makna tertentu dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan.

UML adalah salah satu bentuk notasi atau bahasa yang sama yang digunakan oleh professional dibidang *software* untuk menggambarkan atau memodelkan sebuah sistem *software*. Sebelumnya ada banyak notasi atau bahasa lain untuk mencapai keperluan yang sama misalnya DFD (*Data Flow Diagram*). Tetapi sejak matang dan populernya teknologi pemrograman, perancangan, dan analisis berorientasi objek, UML telah menjadi *de facto standard language*.

Diagram-diagram yang terdapat dalam UML antara lain: *use case diagram*, *class diagram*, *statechart diagram*, *activity diagram*, *sequence diagram*, *collaboration diagram*, *component diagram*, *deployment diagram*.

### 2.1.2.1 Use Case Diagram

Munawar (2005, hal 109-100) menjelaskan *Use case diagram* merupakan deskripsi fungsi dari sebuah sistem perspektif pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antar *user* (pengguna) sebuah sistem dengan sistem-nya sendiri bagaimana sistem tersebut digunakan atau “apa” yang diperbuat sistem dan bukan “bagaimana”. *Use case* dapat dipresentasikan sebuah interaksi antara *actor* dengan sistem .

*Use case Diagram* dapat sangat membantu bila sedang menyusun *requirement* (Jacobsoonet all, 1992). Diagram ini menunjukkan tiga aspek dari sistem yaitu *actor*, *use case*, dan sistem atau subsistem.

*Use case* adalah abstraksi dari interaksi antara system dan actor. *Use case* bekerja dengan cara mendeskripsikan tipe interaksi antara user sebuah system dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah system dipakai. *Use case* merupakan konstruksi untuk mendeskripsikan bagaimana system akan terlihat di mata user. Sedangkan *use case diagram* memfasilitasi komunikasi diantara analis dan pengguna serta antara analis dan client.

Dilihat dari segi pandang Romi Satria Wahono dan Sri Dharyanti sebuah *use case* dapat meng-include fungsionalitas *use case* lain sebagai bagian dari proses

dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal.

Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-*extend use case* lain dengan *behaviour*-nya sendiri.

### 2.1.2.2 Activity Diagram

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi.

Romi Satria Wahono mengatakan bahwa *Activity diagrams* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

*Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behaviour* internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

### 2.1.2.3 Sequence Diagram

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah scenario. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara *object* juga interaksi antara *object*, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

Sequence diagram dilihat dari kacamata Romi Satria Wahono dan Sri Dharyanti menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Munawar mengatakan *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai

respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*. Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan *icon* khusus untuk objek *boundary*, *controller* dan *persistent entity*.

### 2.1.3 Peralatan Pendukung

#### 2.1.3.1 HTML

**Hypertext Markup Language (HTML)** adalah bahasa markup yang umum digunakan untuk membuat halaman web. Sebenarnya HTML bukanlah sebuah bahasa pemrograman. Apabila di tinjau dari namanya, HTML merupakan bahasa mark-up atau penandaan terhadap sebuah dokumen teks. Tanda tersebut di gunakan untuk menentukan format atau style dari teks yang di tandai.

Pada dasarnya elemen HTML ada 2 kategori:

1. Elemen `<head>` berfungsi untuk memberikan informasi atau mendeklarasikan dokumen tersebut.



2. Elemen `<body>` berfungsi untuk menentukan bagaimana isi suatu dokumen ditampilkan pada browser.

Sebagian besar kode HTML harus terletak di antara tag kontainer. Yaitu diawali dengan `<namatag>` dan diakhiri dengan `</namatag>`.

Berikut adalah contoh HTML:

```
<html>
<head>
<title>Contoh HTML</title>
</head>
<body>inilah contoh kode HTML
</body>
</html>
```

### 2.1.3.2 CSS

Menurut *wikipedia*, CSS (Cascading Style Sheet) adalah suatu bahasa stylesheet yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa markup. Untuk lebih jelasnya, CSS adalah suatu bahasa yang digunakan untuk mengatur tampilan suatu website, baik tata letaknya, jenis huruf, warna, dan semua yang berhubungan dengan tampilan. Pada umumnya, CSS tu digunakan untuk memformat halaman web yang ditulis dengan HTML ataupun XHTML. Nah sekarang anda sudah tahu kan apa yang dimaksud dengan CSS?

Sekarang kita lanjut ke cara pemakain CSS. Cara pemakain CSS ada 2 cara. Cara yang pertama adalah dengan menggabungkan CSS langsung ke dalam satu file markup kita (internal), cara yang kedua adalah dengan cara memanggil CSS tersebut (eksternal). Yang membedakan adalah cara menggunakan dan cara pemanggilan CSS tersebut. Kalau kita memakai internal CSS semua kode CSS dan markup kita masukkan dalam satu file yang sama, kalau kita memakai eksternal CSS kita perlu untuk membuat link untuk menghubungkan keduanya.

### **2.1.3.3 Java Script**

JavaScript adalah bahasa pemrograman berbasis prototipe yang berjalan di sisi klien. Jika kita berbicara dalam konteks web, sederhananya kita dapat memahami JavaScript sebagai bahasa pemrograman yang berjalan khusus untuk di browser atau halaman web agar halaman web menjadi lebih hidup. Jika dilihat dari suku katanya, JavaScript terdiri dari dua kata, yaitu Java dan Script. Java adalah bahasa pemrograman yang berorientasi objek, sedangkan Script adalah serangkaian instruksi program. JavaScript adalah bahasa `script` (bahasa pemrograman yang dapat memegang kontrol aplikasi) yang berbasis pada bahasa pemrograman Java. Namun JavaScript bukanlah bagian dari teknologi Java dari Sun. Java merupakan bahasa pemrograman berorientasi objek murni, sedangkan JavaScript digunakan secara prosedural. Cara kerja JavaScript adalah mengakses elemen pada HTML dan membuat aksi jika elemen-elemen HTML itu mengalami perubahan, misalnya berubahnya warna halaman web begitu sebuah tombol diklik.

JavaScript membuat sebuah halaman web menjadi lebih dinamis. JavaScript yang digunakan pada halaman web merupakan client side scripting yang berarti bahwa web browser mengidentifikasi dan menjalankan skrip program yang disisipkan dalam dokumen web (yang diterima dari server), dan mungkin memperbarui tampilan halaman di komputer pemakai( u s e r ) tanpa mengirimkan permintaan( r e q u e s t ) baru kepada web server. Karena JavaScript pada halaman web adalah client side scripting, maka kode JavaScript yang ditulis satu file dengan dokumen HTML akan ditampilkan ketika sebuah halaman web dilihat source code-nya. Secara fungsional, JavaScript digunakan untuk menyediakan akses skrip pada objek yang dibenamkan( e m b e d d e d ). Contoh sederhana dari penggunaan JavaScript adalah membuka halaman pop up, fungsi validasi pada form sebelum data dikirimkan ke server, merubah gambar kursor ketika melewati objek tertentu, dan lain-lain. Yang harus diperhatikan dalam pengelolaan pemrograman JavaScript diantaranya JavaScript adalah bahasa pemrograman yang case sensitive, yang artinya JavaScript membedakan huruf kecil dan huruf besar. Hal ini sama seperti bahasa pemrograman Turbo C atau C++ dimana huruf „A“ tidak sama dengan huruf „a“. JavaScript bekerja pada sisi browser, artinya untuk menampilkan halaman web, user menuliskan alamat web di address bar url. Setelah itu, browser mengambil file HTML (dengan file JavaScript yang melekat padanya jika memang ada) ke server yang beralamat di URL yang diketikkan oleh user. Selesai file diambil, file ditambahkan pada browser. Nah, setelah file JavaScript berada pada browser, barulah skrip JavaScript tersebut bekerja. Efek dari JavaScript

yang bekerja pada sisi browser ini, JavaScript dapat merespon perintah user dengan cepat, dan membuat halaman web menjadi lebih responsif. JavaScript melakukan apa yang tidak bisa dilakukan oleh HTML, PHP, dan CSS dalam menangani hal-hal yang membutuhkan respon cepat terhadap aksi dari user. Misalnya, fungsi validasi pada form. Ketika user mengisi sebuah form yang divalidasi menggunakan JavaScript, user mengetikkan data lalu mengklik submit. Sebelum data dikirimkan ke server, data akan dicek terlebih dahulu pada browser menggunakan fungsi JavaScript yang ada pada halaman web. Sehingga jika memang data yang diisikan tidak valid, daripada membuang-buang waktu dengan mengirim data ke server baru divalidasi di server, lalu server mengirimkan respon balik mengenai ketidakvalidan input data user, lebih baik pengecekan validasi data form dilakukan secara lokal di browser menggunakan fungsi JavaScript. Untuk lebih mempercepat dan menghemat waktu, implementasi terpopuler saat ini dari pemrograman JavaScript adalah teknik AJAX (Asynchronous JavaScript and XML). Teknik ini sering digunakan oleh aplikasi berbasis web seperti Gmail, Google Reader, dan lain-lain. Teknik AJAX ini membuat proses pertukaran data antara server dan browser terjadi di belakang layar sehingga interaksi antara user dan aplikasi web semakin responsif.

Contoh kode:

```
<script type="text/javascript">
```

```
<!--
```

```
document.write("JavaScript is not Java");  
  
-->  
  
</script>
```

#### 2.1.3.4 Bahasa Pemrograman PHP (*Personal Home Page*)

Seiring dengan kebutuhan yang semakin kompleks dan bermunculannya beberapa bahasa pemrograman web yang bersifat *server-side*, perkembangan dunia web saat ini mengarah ke pengembangan situs web dinamis dimana sebuah web dapat berinteraksi dengan pengunjung. Tampilan dan isi dapat ter-update secara otomatis dan berubah sesuai dengan kondisi yang ditentukan.

Salah satu bahasa pemrograman web dinamis yang banyak digunakan saat ini adalah PHP (*Hypertext Preprocessor*). Saat ini PHP sudah banyak digunakan oleh para programmer web. Menurut Eko Priyo utomo (2008, hal 3) Keunggulan utama PHP dibandingkan dengan bahasa pemrograman web lainnya adalah kecepatannya dalam *parsing* kode atau pemrosesan kode. Hal ini disebabkan karena proses pemrograman dilakukan di sisi *server* sehingga pengguna hanya tinggal melihat hasil prosesnya saja. Selain itu ada dukungan database yang beragam, sehingga menjadikan PHP semakin disukai oleh orang banyak untuk membangun banyak program aplikasi yang digunakan di berbagai perusahaan.

#### Sejarah PHP

Dilihat dari perkembangannya, bahasa pemrograman ini memiliki perkembangan yang sangat cepat dengan jumlah pemakai yang terus bertambah.

## 1. PHP/FI

Ini merupakan cikal bakal PHP yang sekarang. Pertama dibuat oleh Rasmus Lerdorf pada 1995, pada awalnya dia menamakan script ini "*Personal Home Page Tool*" yang merupakan bahasa sederhana dari bahasa pemrograman C dimana *Personal Home Page Tool* ini dapat berkomunikasi dengan database dan bersifat *Open Source*. Pada awalnya Rasmus membuat bahasa pemrograman ini bertujuan untuk menyimpan data pengunjung yang melihat biodata pada web sitenya.

Perkembangannya, pada pertengahan tahun 1997 pemakai bahasa bahasa PHP semakin banyak, terlihat dari jumlah statistik domain yang menggunakan PHP hampir lebih dari 50.000 web site. Kemudian karena perkembangannya yang sangat pesat, Rasmus mengembangkan bahasa pemrograman ini, dan pada bulan November tahun 1997 muncul PHP/FI versi 2.0 yang merupakan cikal bakal PHP 3.

## 2. PHP Versi 3

PHP Versi 3 merupakan versi penyempurna dari bugs-bugs pada PHP/FI versi 1.0 dan PHP/FI versi 2.0. PHP Versi 3 ini dikembangkan oleh Andi Gutmans and Zeev Suraski pada tahun 1997 yang berhasil ditulis secara sempurna pada waktu itu. Fasilitas tambahan PHP Versi 3 dibandingkan versi sebelumnya, selain tambahan fungsi-fungsi baru, juga mendukung beberapa akses ke banyak database, pengelolaan protokol, dan API. Dari versi 3 lah singkatan PHP muncul, yaitu PHP: Hypertext Preprocessor, dan pada tahun 1998 hampir 10% web site di dunia menggunakan PHP sebagai web servernya.

## 3. PHP Versi 4

Pada musim dingin di tahun 1998, menulis ulang bahasa pemrograman PHP ini untuk membuat ketangguhan bahasa pemrograman ini. Akhirnya pada pertengahan tahun 1999 diperkenalkanlah PHP versi 4.0 yang menggunakan script engine Zend untuk meningkatkan penampilan (performance) dan mempunyai dukungan yang sangat banyak terhadap ekstensi dan berbagai library beserta modul. PHP versi 4.0 ini juga mempunyai keunggulan dibandingkan versi-versi sebelumnya, diantaranya mendukung ke beberapa web server, fasilitas HTTP session, output buffer dan sistem keamanan. Pada perkembangannya, pada saat itu hampir 20% web server menggunakan bahasa pemrograman PHP sebagai interpreturnya.

#### 4. PHP Versi 5

Pada bulan Juli tahun 2005 muncul PHP versi 5.0 yang menggunakan *Zend Engine 2.0* dengan penambahan beberapa *featur* dan beberapa objek baru. PHP Versi 5 ini sangat mendukung pemrograman berbasis Object Oriented Programming alias OOP dan emang di peruntukan untuk OOP.

Standar penulisan PHP harus diawali dengan tag `<?php` dan diakhiri dengan `?>` atau bisa diawali dengan `<? Dan diakhiri ?>`. semua skrip program yang berada di antara kedua tag tersebut akan diproses sebagai skrip PHP. Tag tersebut biasanya disebut tanda untuk escaping (kabur) dari kode html. File html yang telah dibubuhi program php harus diganti ekstensi-nya menjadi **.php**. PHP merupakan bahasa pemrograman web yang bersifat server-side *HTML=embedded scripting*, di mana script-nya menyatu dengan HTML dan berada di server. Artinya adalah sintaks

### Server:

- \* Apache (web server)
- \* MySQL (database server)
- \* PostgreSQL (database server)
- \* XMail (e-mail server)
- \* SlimFTPd (ftp server)

### Interpreters:

- \* PHP (bahasa scripting)
- \* Perl (scripting language)
- \* Python (bahasa scripting)

### Graphic User Interfaces:

- \* Apache2TriadCP (panel kontrol)
- \* PhpMyAdmin (MySQL GUI)
- \* PhpPgAdmin (PostgreSQL GUI)
- \* PhpSQLiteAdmin (SQLite GUI)
- \* AWStats (webserver monitor)
- \* UebiMiau (E-mail client)
- \* PHPXmail (XMail GUI)
- \* PHPsFTPd (GUI SlimFTPd)



Seperti semua server WAMP, Apache2Triad mungkin disebut distribusi perangkat lunak webserver karena mengandung semua server, interpreter, dan user interface yang sudah dikonfigurasi dan siap digunakan.

## 2.2 Tinjauan Kasus

Kebanyakan *e-commerce* yang ada di Indonesia menjual produk-produk yang dipermudah dengan sistem penjualan secara *online*, tetapi system penjualan tersebut masih terdapat kegiatan yang bersifat manual seperti mengirim produk yang dipesan kepada customer. Kemudian juga memesan produk hanya dengan produk yang di jual dari perusahaan itu sendiri, maka dari itu penulis berusaha membuat aplikasi *e-commerce* yang lebih sederhana dengan menampilkan atau bisa melakukan pembelian dengan pesanan sendiri atau pelanggan memesan sendiri barang seperti apa yang dia inginkan. Dengan itu pelanggan bebas untuk mememesan barang seperti apa yang dia mau beli.

