

BAB II

LANDASAN TEORI

2.1 Apa Itu PHP

Menurut Agus Saputra (2011, p.1) PHP atau yang memiliki kepanjangan *PHP Hypertext Preprocessor* merupakan suatu bahasa pemrograman yang difungsikan untuk membangun suatu website dinamis. PHP menyatu dengan kode HTML, maksudnya adalah beda kondisi. HTML digunakan sebagai pembangun atau pondasi dari kerangka layout web, sedangkan PHP difungsikan sebagai prosesnya sehingga dengan adanya PHP tersebut, web akan sangat mudah di-*maintenance*.

PHP berjalan pada sisi server sehingga PHP disebut juga sebagai bahasa *Server Side Scripting*. Artinya bahwa dalam setiap/untuk menjalankan PHP, wajib adanya web server.

PHP ini bersifat open source sehingga dapat dipakai secara cuma-cuma dan mampu lintas platform, yaitu dapat berjalan pada sistem operasi Windows maupun Linux. PHP juga dibangun sebagai modul pada web server apache dan sebagai binary yang dapat berjalan sebagai CGI.

2.1.1 Keunggulan PHP

Ada beberapa alasan yang menjadi dasar pertimbangan mengapa menggunakan PHP.

1. Mudah dipelajari, alasan tersebut menjadi salah satu alasan utama untuk menggunakan PHP, Pemula pun akan mampu untuk menjadi web master PHP.
2. Mampu Lintas *Platform*, artinya PHP dapat/ mudah diaplikasikan ke berbagai *platform OS(Operating Sytem)* dan hampir semua browser juga mendukung PHP.

3. *Free* alias Gratis, bersifat *Open Source*.
4. PHP memiliki tingkat akses yang cepat.
5. Didukung oleh beberapa macam web server, PHP mendukung beberapa web server, seperti Apache, IIS, Lighttpd, Xitami.
6. Mendukung database, PHP mendukung beberapa database, baik yang gratis maupun yang berbayar, seperti MySQL, PostgreSQL, mSQL, Informix, SQL server, Oracle.

2.1.2 Skrip Dasar PHP

PHP sebagai alternatif lain memberikan solusi sangat murah (karena gratis digunakan) dan dapat berjalan diberbagai jenis platform. PHP adalah skrip bersifat server-side yang ditambahkan ke dalam HTML. PHP sendiri merupakan singkatan dari Personal Home Page Tools. Skrip ini akan membuat suatu aplikasi dapat diintegrasikan ke dalam HTML sehingga suatu halaman web tidak lagi bersifat statis, namun menjadi bersifat dinamis. Sifat *server side* berarti pengerjaan skrip dilakukan di server, baru kemudian hasilnya di kirimkan ke browser.

Sintaks dasar PHP meliputi bagaimana cara memulai suatu struktur pemrograman PHP. Ada empat cara untuk memulai pemrograman PHP, diantaranya:

- `<?php ?>`
- `<? ?>`
- `<script language="php"> </script>`
- `<% %>`

dari beberapa sintaks dasar tersebut, yang paling banyak digunakan adalah cara yang pertama dan yang kedua dari atas.

Cara penulisan skrip PHP ada dua macam, yaitu *Embedded Script* dan *Non Embedded Script*. Contohnya:

- Embedded Script

```
<html>

<body>

<? php echo "Belajar";

?>

</body>

</html>
```

- Non Embedded Script

```
<?php

echo "<html>";

echo "<body>";

echo "Belajar PHP";

echo "</body>";

echo "</html>";

?>
```

Dari contoh menjelaskan bahwa skrip PHP dapat berupa *embedded script* yaitu meletakkan tag PHP diantara tag-tag HTML sedangkan *non embedded script* yaitu semua tag HTML diletakkan dalam tag PHP. Semua kode PHP menyerupai dengan kode bahasa C, walaupun tidak sepenuhnya sama.

Untuk menampilkan nilai suatu variabel ke layar dapat menggunakan perintah yaitu echo, print maupun printf. Contohnya:

- echo

```
echo "$data";

echo $data;

echo "nilai";

echo $data. "Mahasiswa Binus";
```

- print

```
print ("$data");

print ($data);

print ("nilai");

print ($data. "Mahasiswa Binus");
```

- printf

```
$data="devie";

printf ("%s",$data);

printf ("%s Dosen Binus", $data);
```

Sebagai contoh untuk menampilkan kata “Halo” pada halaman web menggunakan PHP, maka sintaksnya sebagai berikut:

```
<?PHP

Echo "Halo";

?>
```

2.2 Apa Itu SQL

Menurut Yenie Kustiyahningsih (2010, p.145) Basis data adalah sekumpulan informasi yang diatur agar mudah dicari. Dalam arti umum basis data adalah sekumpulan data yang diproses dengan bantuan komputer yang memungkinkan data

dapat diakses dengan mudah dan tepat, yang dapat digambarkan sebagai aktivitas dari satu atau lebih organisasi yang berelasi.

MySQL merupakan suatu database. MySQL dapat juga dikatakan sebagai database yang sangat cocok bila dipadukan dengan PHP. Secara umum, database berfungsi sebagai tempat atau wadah untuk menyimpan, mengklasifikasikan data secara profesional. MySQL bekerja menggunakan *SQL Language (Structure Query Language)*. Itu dapat diartikan bahwa MySQL merupakan standar penggunaan database di dunia untuk pengolahan data.

MySQL termasuk jenis RDBMS (*Relational Database Management System*). Sedangkan RDBMS sendiri akan lebih banyak mengenal istilah seperti tabel, baris, dan kolom digunakan dalam perintah-perintah di MySQL. MySQL merupakan sebuah basis data yang mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom. Di dalam PHP telah menyediakan fungsi untuk koneksi ke basis data dengan sejumlah fungsi untuk pengaturan baik menghubungkan maupun memutuskan koneksi dengan server database MySQL sebagai sarana untuk mengumpulkan informasi.

Pada umumnya, perintah yang paling sering digunakan dalam MySQL adalah *select* (mengambil), *insert* (menambah), *update* (mengubah), dan *delete* (menghapus). Selain itu, SQL juga menyediakan perintah untuk membuat database, field, ataupun index guna menambah atau menghapus data.

2.2.1 Keunggulan MySQL

Alasan yang mengacu menggunakan MySQL adalah MySQL merupakan database yang mampu berjalan di semua sistem operasi. Selain itu, sangat mudah sekali untuk

dipelajari dan sepertinya hosting server juga banyak sekali mengadopsi MySQL sebagai standar database. Dan tentunya juga bersifat gratis atau *free*.

Saat ini MySQL juga tidak hanya gratis, semenjak MySQL dibeli oleh SUN, MySQL tidak lagi menikmati fitur-fitur barunya, karena telah dibatasi penggunaannya. Fitur-fitur tersebut hanya bisa didapat jika membeli lisensinya. Berikut beberapa kelebihan yang dimiliki oleh MySQL:

1. Bersifat *open source*, yang memiliki kemampuan untuk dapat dikembangkan lagi.
2. Menggunakan bahasa SQL(*Structure Query Language*), yang merupakan standar bahasa dunia dalam pengolahan data.
3. *Super perfomance* dan *reliable*, tidak bisa diragukan, proses databasenya sangat cepat dan stabil.
4. Sangat mudah dipelajari.
5. Memiliki dukungan support (group) pengguna MySQL.
6. Mampu lintas *platform*, dapat berjalan di berbagai sistem operasi.
7. *Multiuser*, dimana MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami konflik.

Server database MySQL mempunyai kecepatan akses tinggi, mudah digunakan dan handal. MySQL dikembangkan untuk menangani database yang besar secara cepat dan telah sukses digunakan selama bertahun-tahun sehingga membuat server MySQL cocok untuk mengakses database di internet. Dan MySQL juga merupakan sistem *client-server* yang terdiri atas *multithread* SQL server yang mendukung software *client* dan *library* yang berbeda.

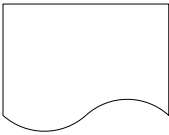
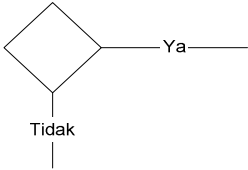
Fitur utama MySQL adalah ditulis dalam bahasa C dan C++, bekerja dalam berbagai *platform*, menyediakan mesin peyimpan transaksi dan nontransaksi, mempunyai *library* yang dapat ditempelkan pada aplikasi yang berdiri sendiri sehingga aplikasi tersebut dapat digunakan pada komputer yang tidak mempunyai jaringan dan mempunyai sistem password yang fleksibel dan aman, dapat menangani basis data dalam skala besar.

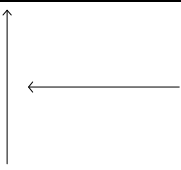
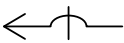



2.3 Diagram Aliran Dokumen

Menurut Mulyadi (2001, p.58-63), diagram aliran dokumen adalah suatu model yang menggambarkan aliran dokumen dan proses untuk mengolah dokumen dalam suatu proses.

Berikut ini adalah table yang menjelaskan komponen-komponen dari diagram aliran dokumen:

Tabel 2.1 Tabel Simbol-simbol Diagram Aliran Dokumen

Simbol	Keterangan
	<p>Dokumen</p> <p>Simbol ini digunakan untuk menggambarkan semua jenis dokumen, yang merupakan formulir untuk merekam data terjadinya suatu transaksi.</p>
	<p>Keputusan</p> <p>Simbol ini menggambarkan keputusan yang harus dibuat dalam proses pengolahan data. Keputusan yang dibuat ditulis dalam simbol.</p>

Simbol	Keterangan
	<p>Garis Alir</p> <p>Simbol ini menggambarkan arah proses pengolahan data.</p>
	<p>Persimpangan Garis Alir</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>Pertemuan Garis Alir</p> <p>Simbol ini digunakan jika dua garis alir bertemu dan salah satu garis mengikuti garis lainnya.</p>
	<p>Proses</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>Mulai / Berakhir (terminal)</p> <p>Simbol ini untuk menggambarkan awal dan akhir suatu sistem akuntansi</p>

2.4 Data Flow Diagram

Menurut Whitten (2004, p.344) “*Data Flow Diagram is a process model used to depict the flow of data through a system and the work or processing performed by the system*”, yang jika diartikan dalam bahasa Indonesia adalah DFD merupakan model proses yang digunakan untuk menggambarkan aliran data yang melalui sebuah sistem dan proses-proses yang dilakukan oleh sistem tersebut.

Terdapat 3 simbol yang dihubungkan oleh satu koneksi pada DFD, yaitu :

- Kotak persegi dengan ujungnya yang bulat, menggambarkan proses atau sebuah pekerjaan yang telah diselesaikan.
- Kotak menggambarkan entitas luar atau *external agents*. Simbol ini dapat digambarkan sebagai terminator yang menjelaskan batasan dari sebuah sistem.
- Kotak tanpa garis penutup di ujungnya (*the open-ended boxes*) menggambarkan penyimpanan data, berupa database atau file.
- Garis anak panah menggambarkan data flow atau aliran atau input dan output. Garis anak panah ini terhubung dari atau menuju proses.

2.4.1 Proses DFD

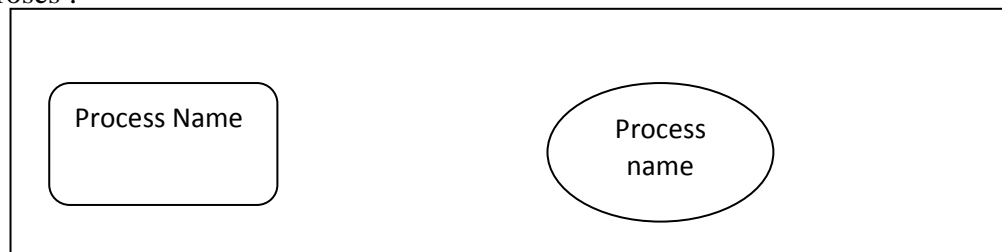
Menurut Whitten (2004, p.347), “*Proccess is a work performed by a system in response to incoming data flows or conditions. A synonym is transform.*” Jika diartikan dalam bahasa Indonesia, proses adalah sebuah kegiatan yang dilakukan oleh sebuah sistem dalam menanggapi aliran data atau sebuah kondisi yang datang ke dalam sistem tersebut. Nama lain dari proses adalah transformasi. Pemberian nama proses dilakukan dengan menggunakan kata kerja transitif (kata kerja yang

membutuhkan obyek), seperti *Menghitung gaji*, *Mencetak KRS*, *Menghitung Jumlah SKS*.

Ada beberapa hal yang perlu diperhatikan dalam menggambarkan proses :

- Proses harus memiliki input dan output
- Proses dapat dihubungkan dengan komponen *terminator*, *data store* atau proses melalui alur data
- Sistem / bagian / divisi / departemen yang sedang dianalisis oleh professional sistem digambarkan dengan komponen proses.

Berikut adalah symbol-simbol yang digunakan dalam menggambarkan sebuah proses :

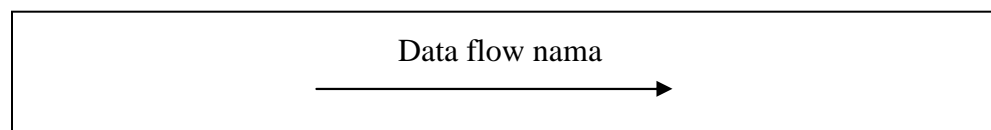


Gane & Sarson shape Demarco/Yourdon shape

Gambar 2.1 Simbol-simbol dari proses(Whitten, 2004, p347)

2.4.2 Data Flow

Menurut Whitten (2004, p357), aliran data atau *data flow* adalah sebuah aliran data yang mewakili sebuah input data ke dalam proses atau output data dari sebuah proses. Aliran data ini juga digunakan untuk mewakili kreasi, pembacaan, penghapusan, atau pembaharuan data di dalam sebuah *file* atau basis data.



Gambar 2.2 Simbol dari data flow(Whitten, 2004, p357)

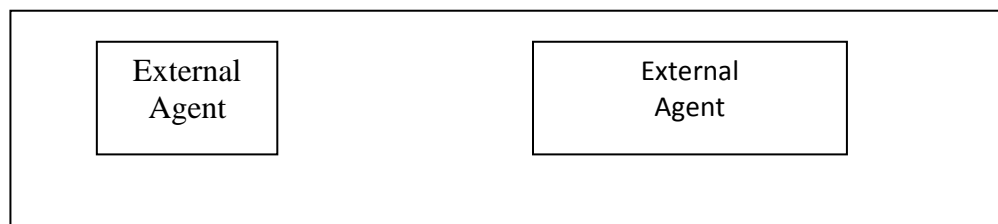
2.4.3 External Agent

Menurut Whitten (2004, p.363), “*External Agent is an outside person, organization unit, system, or organization that interact with a system*”. Jika diartikan dalam bahasa Indonesia, unit pada organisasi, sistem, atau organisasi dimana mereka berhubungan dengan sistem.

Entitas luar ini perlu diberi nama sesuai dengan dunia luar yang berkomunikasi dengan sistem yang sedang dibuat modelnya, dan biasanya menggunakan kata benda misalnya *Bagian Penjualan, Dosen, Mahasiswa*.

Ada 3 hal penting yang perlu diperhatikan perihal entitas luar, yaitu :

- Entitas luar merupakan bagian/lingkungan luar sistem.
- Alur data yang menghubungkan entitas luar dengan berbagai proses sistem, menunjukkan hubungan sistem dengan dunia luar
- Profesional sistem tidak berhak mengubah isi atau cara kerja organisasi atau prosedur yang berkaitan dengan entitas luar.



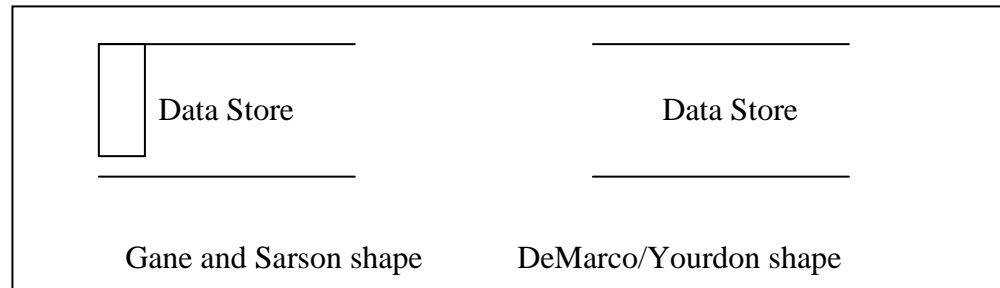
Gane and Sarson shape

DeMarco/Yourdon shape

Gambar 2.3 Simbol dari external agent(Whitten, 2004, p365)

2.4.4 Data Store

Data store adalah sebuah penyimpanan data-data. Data store menyimpan data yang akan digunakan untuk masa mendatang.



Gambar 2.4 Simbol dari data store (Whitten, 2004, p370)

2.4.5 Diagram Konteks

Model proses yang digunakan untuk mendokumentasikan ruang lingkup dari sebuah sistem. Disebut juga model environmental. Sistem DFD konteks dibuat untuk membangun inisialisasi ruang lingkup proyek.

Diagram konteks berisi gambaran umum sistem yang akan dibuat. Secara kalimat, dapat dikatakan bahwa diagram konteks ini berisi “siapa saja yang memberi data ke sistem, serta kepada siapa saja informasi yang harus dihasilkan sistem.




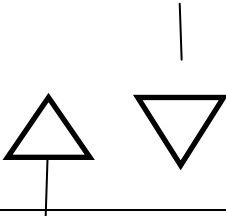
2.5 Entity Relationship Diagram(ERD)

Model *entity relationship* merupakan salah satu model yang dapat memastikan pemahaman yang tepat terhadap data dan bagaimana penggunaannya di dalam suatu organisasi (Connlly dan Begg, 2005, p342). *ER Modelling* merupakan pendekatan *top-down* pada perancangan *database* yang dimulai dengan indentifikasi *entity* dan relasi antar data yang harus direpresentasikan di dalam model, dan kemudian ditambahkan atribut dan setiap *constraint* pada *entity*, *relationship*, dan atributnya.

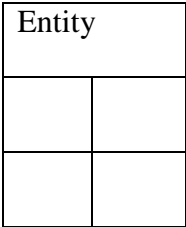

Secara umum simbol yang digunakan untuk ERD adalah sebagai berikut:

1. Persegi panjang mewakili himpunan entitas.
2. Elips mewakili atribut.
3. Jajaran genjang mewakili relasi antar entitas.
4. Garis penghubung antara entitas dengan relasi, maupun antara relasi dengan himpunan atributnya.

Tabel 2.2 Tabel Simbol ERD konseptual

Simbol	Arti
	Menunjukkan entitas
	Menunjukkan arah hubungan dari satu entitas ke entitas yang lain
	Menunjukkan satu entitas berhubungan dengan entitas yang lain
	Menunjukkan generalization, aggregation, specialization dan composition

Tabel 2.3 Tabel Simbol ERD Logical

Simbol	Arti
	Menunjukkan entitas
	Menunjukkan arah hubungan dari satu entitas ke entitas yang lain yang biasanya menentukan FK suatu entitas jika primary key suatu entitas terdapat pada entitas yang lain.

2.6 Rekayasa Perangkat Lunak

2.6.1 Definisi Rekayasa Perangkat Lunak

Menurut Fritz Baurer , rekayasa perangkat lunak adalah pengembangan dan penggunaan prinsip pengembangan suara untuk memperoleh perangkat lunak dan bekerja secara efisien pada mesin nyata (Pressman, 2002, p28).

IEEE (Institute of Electrical and Electronics Engineers) telah mengembangkan definisi tersebut menjadi lebih luas , antarlain sebagai berikut :

- Aplikasi dari sebuah pendekatan yang disiplin, sistematis kepada pengembangan, operasi, dan pemeliharaan perangkat lunak; yaitu aplikasi dari rekayasa perangkat lunak.
- Studi tentang pendekatan-pendekatan seperti pada (a).

2.6.2 Paradigma Rekayasa Perangkat Lunak

Paradigma rekayasa perangkat lunak mencakup metode, *tools*, dan prosedur. Metode rekayasa perangkat lunak memberikan teknik untuk membangun perangkat lunak. Metode-metode itu menyangkut serangkaian tugas yang luas yang menyangkut analisis kebutuhan, konstruksi program, desain, pengujian, dan pemeliharaan.

Tools rekayasa perangkat lunak memberikan topangan yang otomatis ataupun semi otomatis pada proses dan metode yang ada. Prosedur rekayasa perangkat lunak menerapkan metode dan tools sekaligus atau pendefinisian urutan aktifitas metode yang diterapkan dan apa yang diinginkan.

2.6.3 Model Proses Perangkat Lunak

Untuk menyelesaikan masalah aktual di dalam sebuah industri, rekayasa perangkat lunak atau tim rekayasa harus menggabungkan strategi pengembangan yang melingkupi lapisan proses, metode, dan alat-alat bantu. Strategi ini sering diacukan sebagai model proses. Model proses untuk rekayasa perangkat lunak dipilih berdasarkan sifat aplikasi dan proyeknya, metode dan alat bantu yang akan dipakai dan kontrol serta penyampaian yang dibutuhkan.

Model yang digunakan dalam skripsi ini adalah model *Waterfall*, karena model ini melakukan pendekatan kepada perkembangan perangkat lunak yang sistematis dan sekuensial / bertahap yang mulai pada tingkat dan kemajuan sistem seluruh analisis, desain, kode, pengujian, dan pemeliharaan.

Model ini melingkupi aktivitas-aktivitas sebagai berikut :

1. Rekayasa dan pemodelan sistem / informasi

Perangkat lunak merupakan bagian dari suatu sistem yang lebih besar, kerja dimulai dengan membangun syarat dari semua elemen sistem dan

mengalokasikan beberapa subset dari kebutuhan ke perangkat lunak tersebut. Pandangan sistem ini penting ketika perangkat lunak harus berhubungan dengan elemen-elemen lain seperti perangkat lunak, manusia, database.

2. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan diintensifkan dan difokuskan, khususnya pada perangkat lunak. Untuk memahami sifat program yang dibangun, perekrutan kerja, dan antar muka yang diperlukan. Baik kebutuhan untuk sistem maupun perangkat lunak didokumentasikan dan dilihat lagi dengan pelanggan.

3. Desain

Desain perangkat lunak adalah proses multi langkah yang berfokus pada empat atribut sebuah program yang berbeda ; struktur data, arsitektur perangkat lunak, representasi *interface*, dan detail (algoritma) procedural.

4. Generasi kode

Desain harus diterjemahkan ke dalam bentuk mesin yang bisa dibaca. Langkah pembuatan kode melakukan tugas ini. Jika desain dilakukan dengan cara yang lengkap, pembuatan kode dapat diselesaikan secara mekanis.

5. Pengujian

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

6. Pemeliharaan

Perangkat lunak akan mengalami perubahan setelah disampaikan kepada pelanggan. Perubahan akan terjadi karena kesalahan-kesalahan ditentukan karena perangkat lunak harus disesuaikan untuk mengakomodasi perubahan-perubahan di dalam lingkungan eksternalnya, atau karena pelanggan membutuhkan perkembangan fungsional atau unjuk kerja. Pemeliharaan perangkat lunak mengaplikasi lagi setiap fase program sebelumnya dan tidak membuat yang baru lagi.

2.7 Delapan Aturan Emas Perancangan Interface

Menurut Shneiderman (2005, p75-77), delapan aturan emas dalam perancangan *interfacem* meliputi :

1. Berusaha keras untuk konsisten.

Konsistensi yang dimaksud adalah konsistensi dari aksi-aksi dalam situasi tertentu, yaitu konsistensi menu, warna, *layout*, *fonts*, dan sebagainya.

2. Memungkinkan pengguna untuk sering menggunakan jalan pintas (*shortcut*).

Bagi *user* yang sudah ahli menggunakan sistem, ia membutuhkan jumlah interaksi yang lebih singkat dengan memanfaatkan fasilitas *shortcut*.

3. Memberikan umpan balik yang informatif

Sistem harus menyediakan umpan balik untuk setiap aksi pemakai. Isyarat-isyarat seperti suara dan tampilan visual harus ditampilkan untuk setiap interaksi pemakai untuk memberi tahu bahwa aplikasi memberikan respon masukan dari pemakai.

4. Merancang dialog untuk menghasilkan keadaan akhir

Urutan aksi harus tersusun dalam bagian awal, tengah, dan akhir suatu grup. Umpan balik penyelesaian aksi yang informatif, memberikan kepada pengguna

perasaan lega, sinyal untuk mendapatkan kemungkinan rencana dan pilihan dari pikiran pengguna, serta indikasi bahwa cara tersebut jelas mempersiapkan aksi berikutnya.

5. Menyediakan pencegahan kesalahan dan penanganan kesalahan yang sederhana

Pengguna desain antarmuka yang terbaik pun, pemakai tetap dapat membuat kesalahan. Kesalahan ini dapat secara fisik (secara sengaja merujuk ke perintah dan data yang salah) dan secara mental (membuat keputusan yang salah mengenai perintah dan data yang dipilih). Maka sistem didesain sedemikian rupa agar pengguna tidak membuat kesalahan serius. Jika terjadi kesalahan, sistem harus bisa mendeteksi dengan menawarkan mekanisme penanganan yang sederhana dan mudah dimengerti.

6. Mengijinkan pembalikan aksi dengan mudah

Sedapat mungkin aksi harus dapat diulang. Hal ini dapat mengurangi, menghilangkan kecemasan karena pengguna tahu bahwa kesalahan dapat dihilangkan. Dengan begitu mendorong kita untuk menjelajahi pilihan-pilihan yang tidak dikenal

7. Mendukung pusat kendali internal.

Pengguna yang sudah berpengalaman menginginkan suatu perasaan bahwa mereka menguasai sistem dan sistem harus merespon semua keinginan pengguna. Tindakan sistem yang mengejutkan, data yang rumit, kesalahan dalam memperoleh informasi penting dan ketidakmampuan untuk melakukan tindakan yang diinginkan akan membuat kecemasan dan ketidakpuasaan.

8. Mengurangi beban ingatan jangka pendek

Manusia mempunyai keterbatasan dalam menghafal. Maka tampilan sebaiknya dibuat sederhana, tampilan informasi dikelompokkan, frekuensi gerakan window dikurangi, serta diberikan waktu latihan yang cukup untuk kode, dan urutan aksi. Untuk lebih memudahkan, sebaiknya dibuat akses online untuk mencari istilah, singkatan maupun urutan aksi yang sering dipakai namun sulit dihafal.

2.8 Electronic Commerce (E-Commerce)

2.8.1 Pengertian E-Commerce

Menurut Ustandiyanto (2002, p11) merupakan suatu kontak transaksi perdagangan antara penjual dan pembeli dengan menggunakan teknologi internet. Jadi, sebagian proses dikomunikasikan dengan internet.

2.8.2 Jenis-Jenis E-Commerce

- a. *Business to business* (B2B), merupakan suatu kegiatan bisnis yang terjadi antar perusahaan atau antar produsen.
- b. *Business to pelanggan* (B2C), merupakan kegiatan bisnis yang terjadi pada saat pelelangan , perusahaan penjual jasa serta retail online.
- c. *Pelanggan to business* (C2B), merupakan kegiatan bisnis yang terjadi di antara konsumen dan produsen.
- d. *Government to business* (G2B), merupakan kegiatan bisnis yang dilakukan antara pemerintah dan pengusaha.
- e. *Government to Pelanggan* (G2C), merupakan kegiatan bisnis yang terjadi di antara pemerintah dan konsumen.

2.8.3 Manfaat *E-commerce*

Menurut Elias (2002, p13) Dengan menggunakan transaksi melalui *e-commerce* diharapkan suatu perusahaan mendapatkan keuntungan sebagai berikut:

- Biaya lebih murah

Dengan melakukan *e-business* di *internet* mengurangi biaya; mengurangi masalah logistic.

- Ekonomis

E-commerce hemat. Tidak seperti lingkungan *brick-and-mortar*, di dalam *e-commerce* tidak memerlukan biaya penyewaan tempat, asuransi, atau investasi infrastuktur.

- Keuntungan maksimum

E-commerce dimaksudkan untuk memaksimalkan pendapatan.

- Pelayanan yang lebih baik

E-commerce dimaksudkan untuk mempercepat pelayanan. Pelayanan berbasis *web* lebih memuaskan pelanggan. Untuk perusahaan yang berbisnis dengan perusahaan lain, menambahkan pelayanan di dalam *web* menambah keuntungan bersaing.

- Perbandingan perbelanjaan yang cepat

E-commerce membantu pelanggan untuk membaningka produk yang ingin dibeli.

- Mendapatkan *produktivitas*

Menelusuri *web* melalui organisasi artinya mengembangkan produktifitas.

- Kerjasama tim

E-commerce membantu orang untuk bekerja bersama. *E-commerce* telah mengubah cara organisasi berinteraksi dengan *supplier*, *vendor*, *partner*, *bisnin*, dan pelanggan.

- Pengetahuan pasar

Kelompok-kelompok kecil di dalam perusahaan yang besar dapat dibiayai dengan uang untuk mengembangkan ide baru.

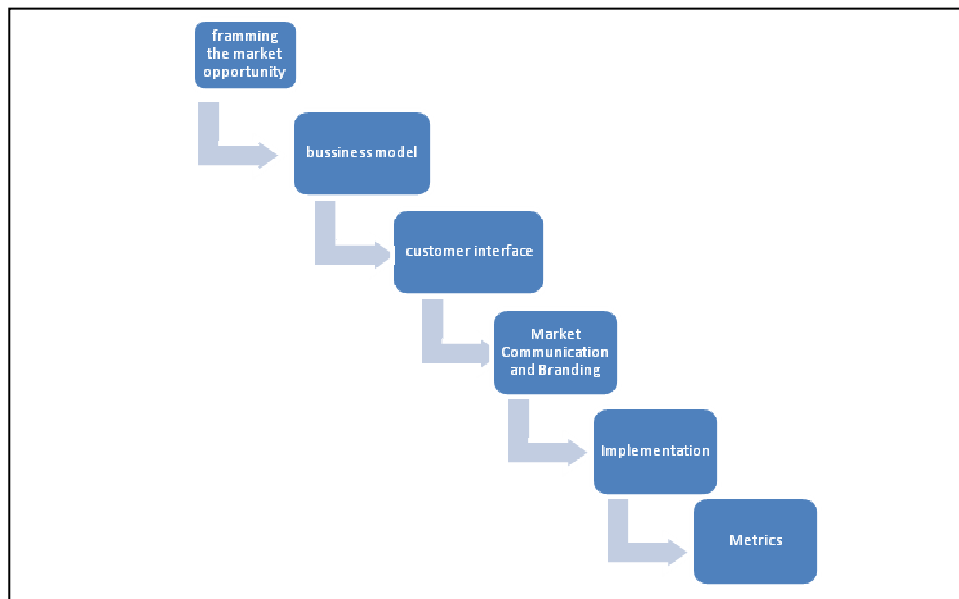
- Berbagai informasi, kenyamanan dan control

Pasar elektronik mengembangkan pertukaran informasi diantara *merchants* dan pelanggan dan dipromosikan dengan cepat, *just-in-time deliveries*. Kenyamanan pelanggan adalah penggerak utama untuk perubahan di banyak industry. Control adalah faktor penggerak utama lainnya.

- Pertukaran barang dan jasa

Pertukaran adalah bertransaksi barang yang anda punya dengan barang yang lebih anda inginkan.

2.8.4 Strategi E-commerce



Gambar 2.5 Strategi E-Commerce(Sumber: Rayport dan Jaworski, 2003 p11)

2.9 Interaksi Manusia dan Komputer

Menurut Schneiderman (2005, p15), dalam membuat suatu desain sistem, pengembang sistem harus memperhatikan beberapa factor dari manusia. Lima factor tersebut adalah :

1. *Time to Learn*, waktu yang diperlukan seorang pengguna untuk belajar bagaimana menggunakan sistem untuk keperluan tertentu.
2. *Speed of Performance*, berapa lama waktu yang dibutuhkan untuk menghasilkan apa yang dibutuhkan.
3. *Rate Errors by Users*, berapa banyak dan apa saja jenis kesalahan yang dibuat oleh pengguna dalam menggunakan sistem.
4. *Retention over Time*, bagaimana kemampuan user untuk mempertahankan pengetahuan mereka terhadap sistem setelah beberapa waktu. Hal ini berhubungan dengan lamanya waktu yang digunakan untuk belajar dan frekuensi penggunaan sistem.
5. *Subjective Satisfaction*, berapa banyak pengguna yang menyukai sistem.