

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 E-Commerce**

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

Menurut E. Turban, David K, J. Lee, T. Liang, D. Turban (2012,p38), Perdagangan elektronik (*electronic commerce*, disingkat EC, atau *e-commerce*) mencakup proses pembelian, penjualan, transfer, atau pertukaran produk, layanan atau informasi melalui jaringan komputer, termasuk internet. Beberapa orang memandang istilah perdagangan (*e-commerce*) hanya untuk menjelaskan transaksi yang dapat dilakukan antar mitra bisnis. Jika definisi ini digunakan, beberapa orang menyadari bahwa istilah *e-commerce* sangat sempit. Sehingga, banyak yang menggunakan istilah e-bussines sebagai istilah penggantinya. Bisnis elektronik (*electronic bussines* atau *e-bussines*) mengarah pada definisi EC yang lebih luas, tidak adanya pembelian dan penjualan barang saja. Tetapi juga layanan pelanggan, kolaborasi dengan mitra bisnis. Lainnya memandang e-bussines sebagai “aktivitas selain pembelian dan penjualan” di internet, seperti kolaborasi dan aktivitas intra bisnis.

##### **2.1.2 Jenis E-Commerce**

Menurut E. Turban, David K, J. Lee, T. Liang, D. Turban (2012,pp42-43), *e-commerce* dapat dilakukan berbagai pihak, Jenis umum dari transaksi *e-commerce* dijelaskan di bawah ini.

**1. Bisnis ke bisnis (*bussines-to-bussines*—B2B)**

Dalam transaksi b2b, baik penjual maupun pembeli adalah organisasi bisnis. Kebanyakan dari EC adalah jenis ini.

**2. Perdagangan kolaborasi (*collaborative--c-commerce*)**

Dalam c-commerce para mitra bisnis berkolaborasi (alih – alih membeli atau menjual) secara elektronik. Kolaborasi semacam ini seringkali terjadi antara dan dalam mitra bisnis di sepanjang rantai pasokan.

**3. Bisnis ke konsumen (*bussines-to-consumer*—B2C)**

Dalam B2C, penjual adalah perusahaan dan pembeli adalah perorangan. B2C disebut juga *e-tailing*.

**4. Konsumen-ke-konsumen (*consumer-to-consumer*—C2C)**

Dalam C2C, seorang menjual produk ke orang lain. (Anda juga) dapat melihat C2C digunakan sebagai “*customer-to-customer*” (pelanggan ke pelanggan). Kedua istilah ini dapat dianggap sama, dan keduanya akan digunakan untuk menjelaskan orang – orang yang menjual produk dan jasa ke satu sama lain.

**5. Konsumen-ke-bisnis (*consumer-to-bussines*—C2B)**

Dalam C2B, konsumen memberitahukan kebutuhan atas produk atau jasa tertentu, dan para pemasok bersaing untuk menyediakan produk atau jasa tersebut ke konsumen, Contohnya di *Priceline.com*, di mana pelanggan menyebutkan produk dan harga yang diinginkan, dan Priceline mencoba untuk menemukan pemasok yang memenuhi kebutuhan tersebut.

## **6. Perdagangan intrabisnis (*intraorganisasional*)**

Dalam situasi ini perusahaan menggunakan EC secara internal untuk memperbaiki operasinya, Kondisi khusus dalam hal ini disebut juga sebagai EC B2E (*business-to-its-employees*).

## **7. Pemerintah-ke-warga(*government-to-citizen—G2C*)**

Dalam kondisi ini sebuah entitas (unit) pemerintah menyediakan layanan ke para warganya melalui teknologi EC. Unit-unit pemerintah dapat melakukan bisnis dengan berbagai unit pemerintah lainnya serta dengan berbagai perusahaan (G2B).

## **8. Perdagangan *mobile* (*mobile commerce—m-commerce*)**

Ketika *e-commerce* dilakukan dalam lingkungan nirkabel, seperti dengan menggunakan telepon seluler untuk mengakses internet dan berbelanja, maka hal ini disebut *m-commerce*.

### **2.1.3 Kelebihan E-Commerce**

Menurut E. Turban, David K, J. Lee, T. Liang, D. Turban (2012,p67), kelebihan *e-commerce* dibagi menjadi 3 yaitu :

#### **1. Kelebihan *e-commerce* bagi perusahaan :**

- Ketersediaan pasar nasional dan internasional
- Penurunan biaya pemrosesan, distribusi dan penarikan informasi

#### **2. Kelebihan *e-commerce* bagi pelanggan**

- Akses ke sejumlah besar produk dan jasa, 24 jam sehari.

### 3. Kelebihan *e-commerce* bagi masyarakat

- Dengan mudah dan nyaman memberikan layanan informasi, serta berbagai produk ke orang-orang di kota, di desa, dan berbagai Negara berkembang.

#### 2.1.4 Kekurangan E-commerce

Menurut E. Turban, David K, J. Lee, T. Liang, D. Turban (2012,p68), *e-commerce* memiliki beberapa keterbatasan, secara teknologi dan nonteknologi, yang telah memperlambat pertumbuhan dan penerimanya. Keterbatasan teknologi meliputi kurangnya standar keamanan yang diterima secara universal, *bandwidth* telekomunikasi yang tidak cukup dan mahalnya akses. Keterbatasan nonteknologi meliputi persepsi bahwa EC tidak aman, segi hukumnya yang belum lengkap, serta kurangnya penjual dan pembeli besar yang penting.

## 2.2 Aplikasi Web

Menurut Pressman, Roger S. (2010, p8) Aplikasi web yang disebut "*webapps*," adalah kategori perangkat lunak jaringan-sentris yang mencakup beragam aplikasi. Dalam bentuknya yang paling sederhana, *webapps* dapat menjadi *link hypertext* informatif yang menggunakan teks dan grafis yang terbatas. Namun, dengan munculnya Web 2.0, *webapps* berkembang menjadi lingkungan komputerisasi canggih yang tidak hanya menyediakan fitur *stand-alone* (berdiri sendiri), fungsi komputasi, dan konten kepada pengguna akhir, tetapi juga terintegrasi dengan *database* perusahaan dan aplikasi bisnis.

### **2.2.1 Web Sever**

Menurut Shelly, Woods, Dorin (2010,p4) Halaman Web disimpan di server Web, atau host, yang merupakan komputer yang menyimpan dan mengirim (melayani) halaman Web dan files lainnya yang telah direquest. Setiap komputer yang memiliki web server dan terhubung ke Internet dapat bertindak sebagai serverWeb. setiap situs web disimpan dan dan dijalankan satu web server atau lebih. Sebuah situs Web yang besar mungkin tersebar ke beberapa server di lokasi geografis yang berbeda.

### **2.2.2 HTML ( Hyper Text Markup Language)**

Menurut Shelly, Woods, Dorin (2010,p8), Halaman web dibuat menggunakan Hypertext Markup Language (HTML), yang merupakan bahasa *authoring* yang digunakan untuk membuat dokumen *World Wide Web* (www). HTML menggunakan set instruksi khusus yang disebut tag atau markup untuk menentukan struktur dan tata letak dokumen Web, dan menentukan bagaimana halaman ditampilkan di browser

### **2.2.3 XAMPP**

Menurut Nugroho, B (2008, p2) XAMPP adalah suatu bundel web server yang populer digunakan untuk coba-coba di Windows karena kemudahan instalasinya. Bundel program open source tersebut berisi antara lain server web Apache, interpreter PHP, dan basis data MySQL.

### **2.2.4 Hypertext Preprocessor (PHP)**

Menurut P. MacIntrye, B. Danchilla, M. Gogala (2011,pxviii) PHP Hypertext adalah bahasa pengembangan web yang paling populer di pasar saat ini. PHP hanyalah sebuah generator markup HTML. Jika Anda melihat source

code dari PHP pada halaman web, Anda hanya akan melihat tag HTML, mungkin beberapa JavaScript juga, tapi tidak ada kode PHP mentah.

## **2.3 Database**

### **2.3.1 Pengertian Database**

Menurut Connolly, Thomas M., Begg, Carolyn E (2008,p7), database adalah sekumpulan data yang terhubung secara logical (deskripsi dari data tersebut) yang dirancang untuk memenuhi kebutuhan informasi dari sebuah organisasi.

### **2.3.2 Database Management System (DBMS)**

Menurut Connolly, Thomas M., Begg, Carolyn E (2008,p9), DBMS adalah sebuah sistem perangkat lunak yang memungkinkan pengguna untuk mendefinisikan, membuat, dan memelihara basis data yang menyediakan akses untuk mengontrol basis data.

Menurut Connolly, Thomas M., Begg, Carolyn E (2008,p12), komponen utama DBMS dibedakan menjadi :

#### **1. Perangkat keras (hardware)**

Perangkat keras diperlukan untuk menjalankan DBMS dan aplikasi. Perangkat keras dapat berkisar dari komputer pribadi, mainframe tunggal, sampai jaringan komputer. Perangkat keras tertentu diperlukan sesuai dengan kebutuhan organisasi dan DBMS yang digunakan. Beberapa DBMS hanya berjalan pada perangkat keras atau sistem operasi tertentu, sementara DBMS yang lain berjalan diberbagai perangkat keras dan sistem operasi. Sebuah DBMS membutuhkan

jumlah minimum memori utama dan ruang disk untuk menjalankannya, tetapi konfigurasi minimal ini belum tentu memberikan kinerja yang diinginkan.

## **2. Perangkat lunak (software)**

Komponen perangkat lunak terdiri dari perangkat lunak DBMS dan program aplikasi itu sendiri, sistem operasi, termasuk perangkat lunak jaringan jika DBMS dijalankan melalui jaringan. Biasanya program aplikasi ditulis dalam bahasa pemrograman generasi ketiga, seperti 'C', COBOL, Fortran, ada, atau Pascal, atau menggunakan bahasa pemrograman keempat, seperti SQL, tertanam di dalam bahasa pemrograman generasi ketiga. Target DBMS memiliki alat generasi keempat sendiri yang memungkinkan pengembangan aplikasi yang cepat melalui penyediaan bahasa query non-prosedural. generator laporan, generator bentuk, generator grafik, dan generator aplikasi. Penggunaan alat generasi keempat dapat meningkatkan produktivitas secara signifikan dan menghasilkan program yang lebih mudah dipelihara.

## **3. Data**

Merupakan komponen terpenting dari DBMS karena data merupakan penghubung antara komputer dengan manusia

## **4. Prosedur**

Prosedur adalah panduan dan instruksi dalam membuat desain dan menggunakan basis data. Pengguna dari sistem dan *staff* dalam mengelola basis data membutuhkan prosedur dalam menjalankan

sistem dan mengelola basis data itu sendiri. Prosedur didalam basis data dapat berupa login didalam basis data, penggunaan sebagian fasilitas DBMS, cara menjalankan dan memberhentikan DBMS, membuat salinan *backup* basis data, memeriksa piranti keras dan piranti lunak yang sedang berjalan, mengubah struktur basis data, meningkatkan kinerja atau membuat arsip data pada *secondary storage*.

## **5. Orang**

Pengguna yang berkaitan dengan sistem

### **2.3.3 Entity Relationship Diagram (ERD)**

#### **1. Entitas**

Menurut Connolly, Thomas M., Begg, Carolyn E (2008,p372), entitas adalah konsep dasar dari ERD, yang merepresentasikan sekumpulan objek (orang, tempat, barang, konsep, *event*) dengan muatan yang sama di dalam *database*. Entitas diidentifikasi mempunyai keberadaan yang independen oleh sebuah perusahaan dan perorangan.

#### **2. Relationship**

Menurut Connolly, Thomas M., Begg, Carolyn E (2008,p374) *relationship* adalah sekumpulan hubungan yang berarti antara satu atau lebih entitas, dimana setiap tipe *relationship* diberi nama yang menggambarkan fungsinya. Sedangkan *relationship occurrence* adalah hubungan yang dapat diidentifikasi secara unik.



### 3. Atribut

Menurut Connolly, Thomas M., Begg, Carolyn E (2008,p379), atribut adalah sebuah sifat dari entitas atau *relationship*. Sebagai contoh, entitas staff mungkin dapat menjelaskan atribut sebagai berikut: noStaff, nama, posisi, dan gaji. Setiap atribut menyimpan nilai yang menjelaskan setiap *entry occurrence* dan menggambarkan bagian utama dari data yang disimpan di dalam *database*.

#### 2.3.4 SQL


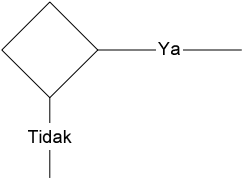
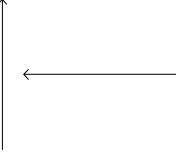
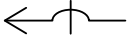

Menurut Connolly, Thomas M., Begg, Carolyn E (2008,pp46-47) SQL adalah *Structured Query Language*. SQL ini adalah suatu bahasa permintaan yang terstruktur. SQL merupakan bahasa yang berada di database, atau biasa disebut dengan MySQL. MySQL merupakan *database server* dari SQL. MySQL adalah *database management system* (DBMS) dan SQL merupakan perintah dalam DBMS.

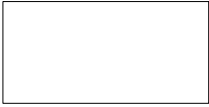

### 2.4 Diagram Aliran Dokumen (DAD)

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 :

Tabel 2.1 Tabel Simbol-simbol 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>
	<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>

Simbol	Keterangan
	<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>

## 2.5 Interaksi Manusia dan Komputer (IMK)

### 2.5.1 Pengertian IMK

Menurut B. Shneiderman, C. Plaisant (2010,p22) interaksi manusia dan komputer merupakan ilmu yang mempelajari tentang perancangan, implementasi dan evaluasi sistem komputer interaktif untuk digunakan oleh manusia, serta studi fenomena-fenomena yang berhubungan dengannya. IMK menangani perancangan dan evaluasi antarmuka pemakai (*user interface*). Antarmuka pemakai adalah bagian dari komputer yang bertujuan untuk memungkinkan interaksi antara manusia dengan komputer secara langsung.

### **2.5.2 Lima Faktor Manusia Terukur**

Demi tercapainya tujuan dari IMK, maka perancangan interface sebaiknya tidak lupa untuk mengikutsertakan evaluasi terhadap lima (5) faktor terukur dari manusia sebagai berikut (B. Shneiderman, C. Plaisant, 2010, p32):

#### **1. Waktu untuk belajar**

Ukuran berapa lama seorang user untuk mempelajari fungsi-fungsi di dalam sebuah aplikasi hingga pada akhirnya dapat menggunakan dengan baik.

#### **2. Kecepatan performa**

Ukuran berapa lama suatu fungsi atau serangkaian tugas di dalam aplikasi tersebut dilakukan.

#### **3. Tingkat *error* yang dilakukan pengguna**

Ukuran berapa banyak dan jenis error yang dilakukan oleh user di dalam melakukan serangkaian tugas.

#### **4. Daya ingat pengguna**

Ukuran berapa lama user mempertahankan ingatan dan pengetahuannya setelah beberapa jam, hari, atau bahkan minggu.

#### **5. Kepuasan subjektif**

Ukuran seberapa puas user atas berbagai aspek dari suatu sistem.

### 2.5.3 Delapan Aturan Emas Desain Antarmuka

Menurut B. Shneiderman, C. Plaisant (2010, p88-89), dalam IMK terdapat delapan aturan emas (Eight Golden Rules) yang digunakan dalam perancangan antarmuka pemakai yaitu:

#### 1. Konsistensi

Konsistensi sangat diperlukan di dalam banyak hal, seperti: urutan aksi, istilah-istilah yang digunakan dalam prompt, menu, layar bantuan, warna, tata letak, huruf capital, dan font.

#### 2. Memenuhi kegunaan universal

Kenali kebutuhan pengguna yang beragam dan desain untuk fleksibilitas dan memfasilitasi transformasi konten. Hal-hal seperti perbedaan pengguna pemula dan ahli, rentang usia, *disability*, dan keragaman jenis pengguna lainnya dapat memperkaya *user interface* dan meningkatkan kualitas sistem yang dirasakan.

#### 3. Memberikan umpan balik yang informatif

Untuk setiap aksi yang dilakukan oleh pengguna, sebaiknya harus ada umpan balik dari sistem. Untuk aksi yang minor namun sering dipakai, umpan balik sebaiknya bersifat sederhana. Sebaliknya, umpan balik yang lengkap diperlukan bagi aksi mayor namun jarang digunakan.

#### 4. Desain kotak dialog yang memiliki keadaan akhir

Urutan dari aksi-aksi harus diorganisasikan secara teratur apakah termasuk di dalam urutan awal, tengah, atau akhir. Kotak dialog akan mempermudah pengguna untuk mengingat urutan aksi yang telah

dilakukannya. Hal ini akan membuat para pengguna dapat merencanakan aksi apa yang akan dilakukan berikutnya.

#### **5. Memberikan pencegahan kesalahan dan penanganan yang sederhana**

Perancangan sistem yang baik sangat penting, sistem yang tidak memungkinkan penggunanya untuk dapat melakukan kesalahan yang serius. Jika sebuah error terjadi, maka sistem harus mendeteksinya, kemudian menawarkan mekanisme penanganan error yang sederhana namun terjamin dapat berhasil.

#### **6. Memungkinkan pengembalian aksi yang mudah**

Setiap aksi yang dilakukan pengguna sebaiknya dapat dibatalkan dengan mudah, misalnya dengan penggunaan *undo*. Hal ini dimaksudkan agar pengguna tidak terlalu tegang ketika sedang mengeksplorasi suatu aplikasi.

#### **7. Mendukung pusat kendali internal (*internal locus of control*)**

Sistem harus memastikan agar pengguna benar-benar memegang kontrol akan sistem dan sistem tersebut merespon aksi yang dilakukan oleh pengguna.

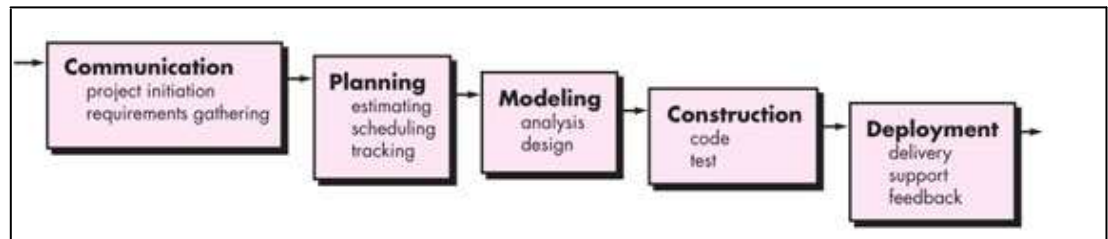
#### **8. Mengurangi beban ingatan jangka pendek**

Tampilan dan fungsi sistem sebaiknya dibuat sesederhana mungkin agar pengguna dapat mengingatnya dalam jangka waktu yang cukup lama. Selain itu, akses *online* untuk *command-syntax forms*, singkatan, kode, dan informasi lainnya juga harus disediakan oleh sistem.

## 2.6 Model Waterfall

### 2.6.1 Pengertian Waterfall

Menurut Pressman, Roger S (2008,p39), *waterfall model* memberikan pendekatan yang sistematis dan sekuensial bagi pengembangan *software* yang dimulai dari level sistem dan melalui tahapan *communication*, *planning*, *construction*, dan *deployment*.



**Gambar 2.1 Waterfall Mod**

**(Pressman, 2008,p39)**

Berikut ini adalah penjelasan dari tahapan-tahapan dari model *waterfall* :

#### 1. *Communication*

Terdiri dari *project initiation* dan *requirements gathering*. Merupakan tahapan pengumpulan kebutuhan yang dilakukan secara intensif dan focus terutama kepada software.

#### 2. *Planning*

Terdiri dari *estimating*, *scheduling*, dan *tracking*. Pada tahap ini *developer* membuat perkiraan waktu yang diperlukan. Pada tahap ini dilakukan pula penjadwalan untuk menyelesaikan tahap-tahap pembuatan *software* sesuai dengan waktu yang telah diperkirakan.

#### 3. *Modeling*

Tahap *modeling* ini terdiri dari analisis dan desain. Pada saat merancang aplikasi, biasanya ada beberapa proses yang memfokuskan

pada empat atribut bagian dari program, diantaranya adalah struktur data, arsitektur *software*, representasi *interface*, dan *detail* algoritma. Proses desain mengubah kebutuhan menjadi representasi dari *software* yang dapat dinilai untuk kualitas sebelum penulisan kode dimulai.

#### **4. Construction**

Tahap *construction* terdiri dari *coding* dan *test*. Pada tahap ini, desain diterjemahkan menjadi bentuk yang dapat dibaca oleh mesin. Setelah *coding* selesai, *testing* terhadap aplikasi dilakukan. Proses pengujian ini menekankan pada logika internal pada aplikasi, meyakinkan semua pernyataan telah diuji, dan pada fungsi eksternal.

#### **5. Deployment**

Tahap terakhir yaitu *deployment*, terdiri dari *delivery*, *support*, dan *feedback*. Pada tahap ini, pengembang menyiapkan aplikasi yang menyediakan fungsi dan fitur yang bermanfaat. Pengembang aplikasi juga menyediakan dokumentasi untuk semua fitur dan fungsi. Pada tahap ini, pengembang aplikasi mendapatkan umpan balik pada aplikasinya yang berakhir pada perubahan atau *update* fungsi dan fitur dari aplikasi tersebut.

## **2.7 Rekayasa Piranti Lunak (RPL)**

### **2.7.1 Pengertian RPL**

Menurut Sommerville, I (2011, pp 6-7), Rekayasa Perangkat Lunak atau disebut juga dengan *Software Engineering* adalah disiplin ilmu yang membahas semua aspek produksi perangkat lunak. Mulai dari tahap awal spesifikasi sistem



sampai pemeliharaan sistem setelah digunakan. Pada definisi ini, ada dua istilah kunci:.

### **1. Disiplin Rekayasa**

*Developer* membuat suatu alat bekerja. Mereka menerapkan teori, metode, dan alat bantu yang sesuai, selain itu mereka menggunakannya dengan selektif dan selalu mencoba mencari solusi terhadap permasalahan, walaupun tidak ada teori atau metode yang mendukung. *Developer* juga menyadari bahwa mereka harus bekerja dalam batasan organisasi dan keuangan, sehingga mereka berusaha mencari solusi dalam batasan-batasan ini.

### **2. Semua Aspek Produksi Perangkat Lunak**

Rekayasa piranti lunak tidak hanya berhubungan dengan proses teknis dari pengembangan perangkat lunak tetapi juga dengan kegiatan seperti manajemen proyek perangkat lunak dan pengembangan alat bantu, metode, dan teori untuk mendukung produksi perangkat lunak.

#### **2.7.2 Karakteristik RPL**

Untuk lebih mendalami rekayasa perangkat lunak, pemahaman lebih dalam mengenai perangkat lunak perlu dilakukan. Perangkat lunak lebih cocok disebut elemen sistem logika dibanding elemen sistem fisik (Pressman, Roger S, 2008, pp 4-7). Perangkat lunak memiliki tiga karakteristik utama, yaitu:

#### **1. Perangkat lunak dikembangkan ataupun direkayasa**

Walaupun terdapat beberapa kesamaan antara pemroduksian perangkat lunak dan perangkat keras, dua aktivitas tersebut memiliki dasar yang berbeda. Pada kedua aktivitas tersebut, kualitas tertinggi perangkat

dapat dicapai dengan desain yang baik. Perbedaannya adalah pada proses pembuatan perangkat keras, masalah kualitas dapat diketahui dengan lebih mudah dibanding perangkat lunak. Biaya pembuatan perangkat lunak berkonsentrasi pada rekayasa.

## **2. Perangkat lunak tidak habis dipakai**

Perangkat lunak tidak rentan terhadap keadaan lingkungan yang menyebabkan perangkat keras dapat habis dipakai. Ketika perangkat keras habis dipakai, komponen akan diganti dengan suku cadang baru. Pada perangkat lunak, tidak terdapat suku cadang. Setiap kesalahan perangkat lunak menyatakan *error* pada desain atau pada proses saat desain diterjemahkan ke dalam bentuk *code* yang dapat dilaksanakan mesin. Maka dari itu, perawatan perangkat lunak dipertimbangkan lebih kompleks dari perawatan perangkat keras.

## **3. Mayoritas perangkat lunak dibuat sesuai kriteria yang ditetapkan**

Seiring berkembangnya disiplin teknis, sejumlah koleksi komponen desain standar diciptakan. Pemasangan standar dan sirkuit yang terintegrasi diluar susunan hanyalah dua dari ribuan komponen standar yang digunakan oleh teknisi mesin dan elektro untuk mendesain sistem baru. Komponen yang dapat dipergunakan berulang telah diciptakan sehingga teknisi dapat berkonsentrasi untuk masalah desain. Pada perangkat keras, komponen yang bisa dipergunakan berulang adalah bagian alami dari proses rekayasa sementara pada perangkat lunak ini adalah awal untuk mencapai skala yang lebih besar. Sebuah komponen perangkat lunak harus didesain dan diimplementasikan sehingga dapat

digunakan di banyak program yang berbeda. Komponen modern yang dapat digunakan kembali berfungsi menyimpan data dan pemrosesan pada data, memungkinkan teknisi untuk menggunakan bagian yang dapat digunakan kembali untuk membuat aplikasi baru. Sebagai contoh, saat ini *user interface* interaktif dibuat menggunakan komponen yang bisa digunakan kembali yang memungkinkan dibuatnya *windows* grafik, menu *pull-down*, dan berbagai variasi mekanisme interaktif.