BAB II

TINJAUAN PUSTAKA

2.1 Konsep Dasar Sistem

Terdapat dua kelompok pendekatan yang dalam mendefinisikan sistem [5]. Pertama, pendekatan yang menekankan pada prosedur, mendefinisikan sistem sebagai berikut :

"*Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama–sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu*".

Kedua, pendekatan yang lebih menekankan pada elemen atau komponennya, mendefinisikan sistem sebagai berikut :

"*Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu*".

2.2 Konsep Dasar Informasi

Informasi (*information)* adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya [5]. Informasi juga dapat didefinisikan sebagai hasil pengolahan data sehingga menjadi bentuk yang penting bagi penerimanya dan mempunyai kegunaan sebagai dasar dalam pengambilan keputusan.

2.3 Konsep Dasar Sistem Informasi

Sistem informasi didefinisikan oleh Robert A. Leitch dari K. Roscoe Davis sebagai berikut [5]:

> *"Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi yang menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan".*

John Burch dan Gary Grudnitski mengemukakan bahwa sistem informasi terdiri dari komponen-komponen yang disebutnya dengan istilah blok bangunan (*building block*), yaitu [5]:

1. Blok masukan, input mewakili data yang masuk ke dalam sistem informasi.
2. Blok model, blok ini terdiri dari kombinasi prosedur, logikan dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.
3. Blok keluaran, merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua pemakai sistem.
4. Blok teknologi. Terdiri dari tiga bagian utama, yaitu teknisi (*humanware* atau *brainware*), perangkat lunak (*software*) dan perangkat keras (*hardware*).
5. Blok basis data.
6. Blok kendali, beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi..

2.4 Metodologi Pengembangan Sistem Informasi

System Development Life Cycle (SDLC) adalah suatu metodologi yang digunakan untuk mengembangkan, memelihara, dan menggunakan sistem informasi (O'Brien, 2001). Metodologi ini mencakup sejumlah fase atau tahapan.

Menurut Roger S. Pressman, terdapat beragam model proses pengembangan perangkat lunak, diantaranya [7]:

1. *Linear Sequential Model* (Model Sekuensial Linear)

   Model ini pertama kali dikemukakan oleh Royce. Model ini sering disebut model klasik atau *waterfall*. Model ini menyarankan pendekatan pengembangan secara sekuen dan sistematik untuk pengembangan perangkat lunak. Model ini merupakan model yang tertua. Model ini terdiri atas beberapa tahap yaitu:

   a. Rekayasa dan pemodelan sistem/informasi
   b. Analisis kebutuhan perangkat lunak
   c. Desain
   d. Generasi kode
   e. Pengujian
   f. Pemeliharaan

2. *Prototyping Model* (Model Prototipe)

   Pada model ini dimulai dengan pengumpulan informasi mengenai kebutuhan, Dimana pengembang dan pemesan bertemu dan mendefinisikan sasaran-sasaran umum, mengidentifikasi kebutuhan yang telah diketahui, dan mencari bidang-bidang yang masih memerlukan pendefinisian. Setelah itu

pengembang melakukan ”perancangan kilat” terhadap kebutuhan yang telah teridentifikasi pada pertemuan. Perancangan berfokus pada representasi yang tampak oleh pengguna. Perancangan ini menuntun pembangunan *prototype* perangkat lunak yang akan diberikan ke pemakai/pelanggan. *Prototype* itu dievaluasi oleh pemakai dan digunakan sebagai landasan untuk memperbaiki spesifikasi kebutuhan. Proses ini berulang sampai *prototype* yang dikembangkan memenuhi kebutuhan pemakai, dan pengembang telah memahami permasalahan dengan baik.

3. *Rapid Application Development (RAD) Model*

   RAD adalah sebuah model proses perkembangan perangkat lunak sekuensial linier yangn menekankan siklus perkembangan yang sangat pendek. Model RAD merupakan adaptasi “berkecepatan tinggi” dari *linear sequential model* dimana pengembangan yang cepat dapat diperoleh dengan menggunakan pendekatan konstruksi berbasis komponen.

   Pendekatan RAD melingkupi fase-fase sebagai berikut :

   a. *Bussiness modeling*
   b. *Data Modeling*
   c. *Proses Model*ing
   d. *Application generation*
   e. *Testing and turnover*

4. *Evalutionary Software Process Model*

   Model evolusioner adalah model perulangan. Model ini dicirikan dengan pengembang mengembangkan versi-versi sistem yang lebih lengkap sedikit

demi sedikit. Model telah mempertimbangkan untuk mengakomodasikan evolusi produk secara lengkap. Model ini terdiri dari:

a. *Incremental model*, model ini mengkombinasikan antara *linear sequential model* dengan filosofi iteratif pada *prototyping*. Pada masing-masing sekuen linear menghasilkan perangkat lunak yang semakin meningkat kompleksitasnya.
b. *Spriral model*, model ini diusulkan oleh Boehm. Model ini menggabungkan antara sifat alami iterasi dari *prototyping* dengan aspek sistematik dan terkendali dari *linear sequential model*. Model ini memberi peluang untuk pengembangan cepat.
c. Model rakitan komponen, model rakitan komponen menggabungkan beberapa karakter model spiral. Model ini bersifat evolusioner, sehingga membutuhkan pendekatan iteratif untuk menciptakan perangkat lunak. Tetapi model rakitan komponen merangkai aplikasi dari komponen perangkat lunak sebelum dipaketkan (kadang-kadang disebut “kelas”).
d. *Concurent development model*, model perkembangan konkuren disebut juga rekayasa konkuren. Model proses yang konkuren dapat disajikan secara skematis sebagai sederetan aktivitas teknis mayor, tugas-tugas dan keadaannya yang lain.

5. Model Formal

Model formal mencakup sekumpulan aktivitas yang membawa kepada spesifikasi matematis perangkat lunak komputer. Model ini memungkinkan perekayasa perangkat lunak untuk mengkhususkan, mengembangkan, dan

memverifikasi sitem berbasis komputer dengan menggunakan notasi matematis yang tepat.

6. Teknik Generasi Keempat

   Bentuk “teknik generasi keempat“ (4GT) mencakup serangkaian alat bantu perangkat lunak yang luas yang secara umum memiliki satu hal: masing-masing memungkinkan perekayasa perangkat lunak untuk mengkhususkan beberapa karakteristik perangkat lunak pada suatu tingkat yang tinggi. Alat bantu tersebut kemudian secara otomatis memunculkan kode sumber yang berdasarkan pada spesifikasi perekayasa.

Meskipun jumlah tahapan dalam SDLC dalam berbagai literatur berbeda-beda, namun pada prinsipnya secara keseluruhan semua proses yang dilakukan sama saja. Tahapan-tahapan dalam SDLC adalah sebagai berikut:

1. Analisis Sistem, tahapan ini dimulai karena adanya permintaan terhadap sistem baru. Tujuan utama analisis sistem adalah untuk menetukan hal-hal detail tentang yang akan dikerjakan oleh sistem yang diusulkan (dan bukan bagaimana caranya). Analisis sistem mencakup studi kelayakan dan analisis kebutuhan.
2. Desain Sistem. Tahapan ini dibagi kedalam dua subtahapan, yakni perancangan konseptual dan perancangan fisik. Target akhir tahapan ini adalah menghasilkan rancangan yang memenuhi kebutuhan yang ditentukan selama tahapan analisis sistem. Hasil akhirnya berupa spesifikasi rancangan yang sangat rinci sehingga mudah diwujudkan pada saat pemrograman.

3. Implementasi Sistem, pada tahap ini *programmer* harus mampu mengimplementasikan desain sistem kedalam bahasa pemrograman, untuk kemudian dilakukan pengujian.
4. Pengembangan dan Pemeliharaan Sistem, tahap ini dilakukan untuk mendeteksi kesalahan-kesalahan sistem yang tidak terdeteksi pada masa pengujian sistem.

2.5 Basis Data

2.5.1 Definisi Basis Data

Basis data (*database*) terdiri atas dua kata, yaitu basis dan data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia, barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi atau kombinasinya [4].

Basis data sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti[4]:

1. Himpunan kelompok data atau arsip yang saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redudancy*) yang tidak perlu, untuk memenuhi berbagai kebutuhan.

3. Kumpulan *file* atau tabel yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.

Basis data dan lemari arsip sesungguhnya memiliki prinsip kerja dan tujuan yang sama. Prinsip utamanya adalah pengaturan data/arsip. Sedangkan tujuan utamanya adalah kemudahan dan kecepatan dalam pengambilan kembali data/arsip. Perbedaanya hanya terletak pada media penyimpanan yang digunakan. Basis data bukan hanya sekedar penyimpanan data secara elektronis. Artinya, tidak semua penyimpanan data secara elektronis bisa disebut basis data. Yang sangat ditonjolkan basis data adalah pengaturan, pemilihan, pengelompokan maupun pengorganisasian data yang akan kita simpan sesuai fungsi dan jenisnya.

2.5.2 Operasi Dasar Basis Data

Operasi-operasi dasar yang dapat kita lakukan berkenaan dengan basis data dapat meliputi [4]:

1. Pembuatan basis data baru (*create database*)
2. Penghapusan basis data (*drop database*)
3. Pembuatan file/tabel baru ke suatu basis data (*create table*)
4. Penghapusan file/tabel dari suatu basis data (*drop table*)
5. Penambahan/pengisian data baru kesebuah file/tabel di sebuah basis data (*insert*)
6. Pengambilan data dari sebuah file/tabel (*retrieve* atau *search*)
7. Pengubahan data dari sebuah file/tabel (*update*)
8. Penghapusan data dari sebuah file/tabel (*delete*)

2.5.3 Sistem Manajemen Basis Data

Sistem manajemen basis data (*Database Management System / DBMS*) adalah suatu mekanisme dalam mengelola data dan informasi sedemikian rupa sehingga data dan informasi dapat digunakan oleh pengguna baik secara *single user* maupun *multiple user* sesuai dengan kepentingan dan kebutuhan pengguna.

Fungsi utama DBMS adalah menyediakan lingkungan yang nyaman dan efisien untuk digunakan dalam pengambilan dan penyimpanan informasi di basis data.

2.5.4 Arsitektur Sistem Basis Data

Terdapat dua bentuk arsitektur sistem basis data, yaitu sistem terpusat dan sistem *Client-Server*.

Sistem basis data terpusat adalah sistem basis data yang dijalankan pada sistem komputer tunggal dan tidak berinteraksi dengan sistem pada komputer lain. Pengguna terkoneksi ke komputer pusat melalui terminal.

Sistem basis data *Client-Server* adalah sistem basis data yang memisahkan program pengguna dengan program basis data disistem yang berbeda. Pengguna terkoneksi ke pusat data yang disebut server sistem melalui suatu program pengguna (*user interface*) yang terdapat pada *personal computer*. Sistem tempat program pengguna berada disebut *client system*.

2.6 Konsep Dasar Analisis Sistem

Analisis sistem (*systems análisis*) dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang untuk kedalam bagian–bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan–permasalahan, hambatan–hambatan yang terjadi dalam kebutuhan–kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan- perbaikannya [5].

Didalam tahap analisis sistem terdapat langkah-langkah dasar yang harus dilakukan oleh analis sistem sebagai berikut [5]:

1. *Identify*, yaitu mendefinisikan masalah.
2. *Understand*, yaitu memahami kerja dari sistem yang ada.
3. *Analyze*, yaitu menganalisis sistem.
4. *Report*, yaitu membuat laporan hasil analisis.

Model analisis, merupakan representasi teknis yang pertama dari sistem. Saat ini ada dua yang mendominasi landskap pemodelan analisis, yaitu:

1. Analisis terstruktur. Dengan menggunakan notasi yang sesuai dengan prinsip analisis operasional, kita menciptakan model yang menggambarkan muatan dan aliran informasi (data dan kontol), kita membagi sistem secara fungsional dan secara *behavioral*, dan menggambarkan esensi dari apa yang harus dibangun. Analisis terstruktur bukan merupakan metode tunggal yang diaplikasikan secara konsisten oleh semua yang menggunakannya.
2. Analisis berorientasi objek, mendefinisikan semua kelas yang relevan terhadap masalah beserta operasi-operasi dan atribut-atribut yang diasosiasikan dengan kelas itu, keterhubungan dikelas-kelas dan perilaku yang dimilikinya. Sasaran

analisis berorientasi objek adalah untuk mengembangkan model yang mendeskripsikan perangkat lunak yang memenuhi sekelompok kebutuhan yang didefinisikan pemesan. Analisis berorientasi objek menggunakan sejumlah pemodelan untuk memenuhi sasaran. Model analisis akan mengekspresikan informasi, perilaku dan fungsi didalam konteks model objek.

Alat – alat yang digunakan dalam analisis terstruktur adalah :

1. Diagram E-R (*Entity Relationship Diagram*)
2. Diagram konteks (*Context Diagram*)
3. Diagram arus data (*Data Flow Diagram*)

2.6.1 Diagram E-R (*Entitiy Relationship Diagram)*

Diagram E-R adalah diagram grafikal yang menggambarkan keseluruhan struktur *logic* dari sebuah basis data. Pada model ini semua data yang ada pada dunia nyata diterjemahkan atau ditransformasikan dengan memanfaatkan sejumlah perangkat konseptual menjadi sebuah diagram data.

Untuk melambangkan fungsi diatas maka digunakan simbol–simbol yang memiliki arti sebagai berikut:

a. Data Entitas

Entitas merupakan individu yang mewakili sesuatu yang nyata (eksistensinya) dan dapat dibedakan dari sesuatu yang lain. Data entitas adalah segala sesuatu baik yang nyata maupun abstrak yang datanya akan direkam

b. Atribut

Atribut mendefinisikan karakteristik (*property*) dari entitas tersebut.

c. Relasi Data

Relasi menunjukkan adanya hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berada.

d. Kardinalitas

Kardinalitas adalah penentuan jumlah kejadian dari satu entitas yang dapat terhubung dengan sejumlah kejadian pada entitas lain, juga menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain. Terdapat tiga kemungkinan kombinasi kardinalitas :

1. *Relationship one to one* (relasi satu ke satu)

   Yang artinya bahwa setiap entitas pada himpunan entitas yang satu berhubungan dengan paling banyak satu entitas pada himpunan entitas yang lainnya, begitu juga sebaliknya, entitas yang terhubung hanya memiliki satu hubungan dengan entitas yang menghubungi.

2. *Relationship one to many* (relasi satu ke banyak)

   Yang artinya bahwa setiap entitas pada himpunan A dapat berhubungan dengan lebih dari satu entitas pada himpunan entitas B, tetapi tidak berlaku untuk kebalikannya, dimana entitas B hanya dapat berhubungan dengan paling banyak satu entitas pada himpunan A.

3. *Relationship many to many* (relasi banyak ke banyak)

   Yang artinya bahwa setiap entitas pada himpunan A dapat berhubungan dengan lebih dari satu entitas pada himpunan B, begitu juga sebaliknya,

dimana entitas B dapat berhubungan dengan lebih dari satu entitas pada himpunan A.

Objektif utama dari pembuatan Diagram E-R adalah untuk menunjukkan objek-objek (himpunan entitas) apa saja yang ingin dilibatkan dalam sebuah basis data dan bagaimana hubungan yang terjadi diantaran objek-objek tersebut. Pada sebuah sistem yang ruang lingkupnya lebar dan kompleks, penggambaran atribut dalam sebuah Diagram E-R seringkali malah mengganggu objektif yang ingin dicapai. Kita dapat memisahkan pendeklarasian atribut-atribut ini dari Diagram E-R dan menyatakannya dalam sebuah kamus data. Kamus data berisi daftar atribut yang diapit kurung kurawal (‘{‘ dan ’}’). Atribut yang berfungsi sebagai *key* juga dibedakan dengan yang bukan *key* dengan menggarisbawahi atribut tersebut [4].

2.6.2 Diagram Konteks

Diagram konteks adalah arus data yang berfungsi untuk menggambarkan keterkaitan aliran-aliran data antara sistem dengan bagian-bagian luar (kesatuan luar). Kesatuan luar ini merupakan sumber arus data atau tujuan data yang berhubungan dengan sistem informasi tersebut. Diagram konteks memberikan batasan yang jelas mengenai besaran-besaran entitas yang berada diluar sistem yang sedang dibuat, artinya diagram ini mengggambarkan secara jelas batasan-batasan dari sebuah sistem yang sedang dibuat.

Diagram konteks bisa disebut dengan "*Model sistem pokok (fundamental system model) mewakili keseluruhan elemen software dengan masukan (input) dan keluaran (output) yang diidentifikasi dengan anak panah masuk dan keluar memperlihatkan sumber data*". (Roger S Pressman, 1997).

2.6.3 Diagram Arus Data (*Data Flow Diagram*)

Data Flow Diagram adalah "*Teknik grafis yang menggambarkan aliran informasi dan perubahan yang dipergunakan sebagai perpindahan data dari masukan (input) ke keluaran (output)*". (Roger S Pressman, 1997).

Beberapa simbol yang digunakan di DFD untuk maksud mewakili [5]:

1. *External entitiy* (kesatuan luar) atau *boundary* (batas sistem), merupakan kesatuan (*entity*) dilingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainnya yang berada dilingkungan luarnya yang akan memberikan input atau menerika output dari sistem. Disimbolkan dengan suatu notasi kotak
2. *Data flow* (arus data), di DFD diberi simbol suatu panah. Arus data ini mengalir dintara proses (*process*), simpanan data (*data store*) dan kesatuan luar (*external entitiy*). Arus data ini menunjukkan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem. Didalam menggambar arus data di DFD perla diperhatikan beberapa konsep, yaitu:
    a. Konsep paket dari data (*packet of data*), bila dua atau lebih data mengalir dari suatu sumber yang sama ke tujuan yang sama, maka harus dianggap sebagai suatu arus data yang tunggal.

b. Konsep arus data menyebar (*diverging data flow*), arus data yang menyebar menunjukkan sejumlah tembusan dari arus data yang sama dari sumber yang sama ke tujuan yang berbeda.
c. Konsep arus data mengumpul (*converging data flow*), arus data yang mengumpul menunjukkan beberapa arus data yang berbeda dari sumber yang berbeda bergabung bersama-sama menuju ke tujuan yang sama.
d. Konsep sumber dan tujuan arus data, semua arus data harus dihasilkan daru suatu proses atau menuju ke suatu proses (dapat salah satu atau kedua-duanya).

3. *Process* (proses). Suatu proses adalah kegiatan atau kerja yang dilakukan oleh orang, mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses.
4. *Data store* (simpanan data), merupakan simpanan dari data yang dapat berupa suatu file atau *database*, suatu arsip atau catatan manual, suatu agenda atau buku.

2.6.4 Kamus Data

Kamus data (*data dictionary*) merupakan sebuah daftar yang terorganisasi dari elemen data yang berhubungan dengan sistem, dengan definisi yang tepat dan teliti sehingga pemakai dan analis sistem akan emmiliki pemahaman yang umum mengenai input, output, komponen penyimpan, dan bahkan kalkulasi *inter-mediate* [7]. Meskipun format kamus bervariasi dari peranti satu ke peranti yang lain, sebagian besar berisis informasi berikut ini:

1. *Name* – nama sebenarnya dari data atau item kontrol, penyimpanan data, atau entitas eksternal.
2. *Where-used / how-used* – suatu daftar dari proses yang menggunakan data atau item control dan bagaimana dia digunakan (misalnya, input ke proses, output dari proses, sebagai suatu penyimpanan, sebagai suatu entitas eksternal).
3. *Content description* - suatu notasi untuk merepresentasikan isi.
4. *Supplementary information* – informasi lain mengenai tipe data, harga preset (bila diketahui), batasan, dan lain-lain.

Notasi yang digunakan untuk mengembangkan deskripsi isi, dapat dilihat pada tabel II.1

Tabel II.1 Notasi Deskripsi Isi Untuk Kamus Data

Konstruk Data	Notasi	Arti
	=	Disusun atas
Berurutan	+	Dan
Pilihan	[|]	Baik ini – atau
Pengulangan	$\{ \}^n$	Pengulangan ke-n dari
	( )	Data opsional
	* *	Komentar tidak dibatasi

2.6.5 Pengujian *Black Box*

Pengujian black box berfokus pada persyaratan fungsional perangkat lunak. Pengujian black box berusaha menemukan kesalahan dalam kategori sebagai berikut [7]:

1. Fungsi-fungsi yang tidak benar atau hilang.
2. Kesalahan *interface*.
3. Kesalahan dalam struktur data atau akses *database* eksternal.
4. Kesalahan kinerja.
5. Inisialisasi dan kesalahan terminasi.

2.7 E-Commerce

Menurut David Baum pengertian *E-Commerce* dapat kita artikan sebagai berikut [6] :

> "*E-Commerce* merupakan satu set dinamis teknologi, aplikasi dan proses bisnis yang menghubungkan perusahaan, konsumen dan komunitas tertentu melalui transaksi elektronik dan perdagangan barang, pelayanan dan informasi yang dilakukan secara elektonik."

Sedangkan menurut Dian Andriana Definisi *E-Commerce* (*Electronic Commerce*) [1] :

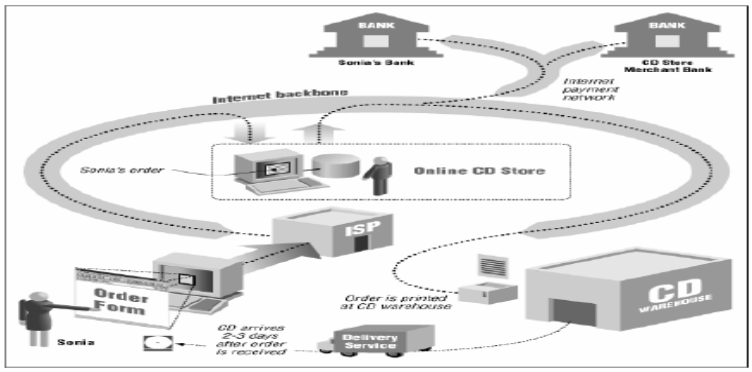
> *"E-commerce* merupakan suatu cara berbelanja atau berdagang secara online atau *direct selling* yang memanfaatkan fasilitas Internet dimana terdapat website yang dapat menyediakan layanan "*get and deliver*".

Proses yang ada dalam *E-commerce* adalah sebagai berikut :

1. Presentasi electronis (Pembuatan Web site) untuk produk dan layanan.
2. Pemesanan secara langsung dan tersedianya tagihan.
3. Otomasi account Pelanggan secara aman (baik nomor rekening maupun nomor Kartu Kredit).
4. Pembayaran yang dilakukan secara Langsung (online) dan penanganan transaksi

Keuntungan yang diperoleh dengan menggunakan transaksi melalui *E-commerce* bagi suatu perusahaan adalah sebagai berikut :

1. Meningkatkan pendapatan dengan menggunakan online channel yang biayanya lebih murah.
2. Mengurangi biaya-biaya yang berhubungan dengan kertas, seperti biaya pos surat, pencetakan, report, dan sebagainya.
3. Mengurangi keterlambatan dengan mengunakan transfer elektronik / pembayaran yang tepat waktu dan dapat langsung dicek.
4. Mempercepat pelayanan ke pelanggan, dan pelayanan lebih responsif.

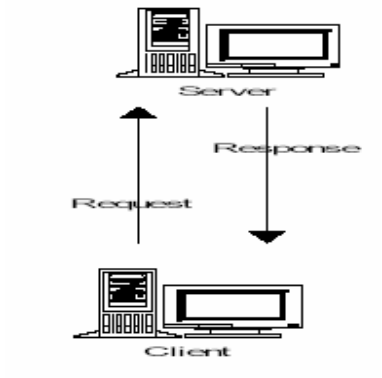


Gambar II.1 Contoh Aplikasi E-Commerce :

Pembelian CD dengan Kartu Kredit

2.7.1 Arsitektur dan Konfigurasi Sistem

Arsitektur dasar dari aplikasi web ini adalah arsitektur *clientIserver*. Artinya pemrosesan aplikasi ini dijalankan melibatkan kedua sisi yakni sisi mesin server pusat dan sisi *client*. Hal ini berbeda dengan misalnya aplikasi Microsoft Word yang hanya melibatkan satu sisi saja yaitu sisi *client*. Atau bagi pengguna mesin VAX yang hanya menggunakan sisi server saja sedangkan sisi *client* hanya *dumb terminal* saja yang tidak melakukan pemrosesan apapun di sisi *client.*

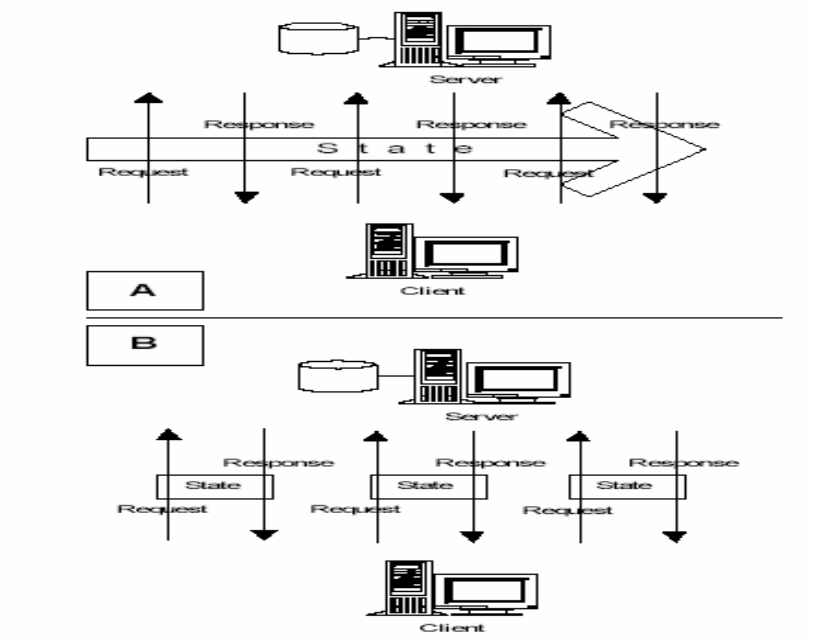


Gambar II.2 Arsitektur *Client/Server*

2.7.2 Stateless Web Server

Untuk aplikasi *E-Commerce* ini w*eb* server harus dapat mengingat siapa / identitas pengguna yang sedang melakukan browsing setiap halaman. Pada dasarnya aplikasi web dan protocol HTTP adalah *stateless*. Artinya setiap merespon sebuah *request* HTTP, server akan selesai bekerja (*complete*) dan tidak melakukan pencatatan apa yang telah dilakukan oleh pengguna sebelumnya dan terhadap siapa identitas pengguna. Server memperlakukan informasi permintaan (*request*) secara serial, satu persatu pada saat *request* masuk. Tidak ada koneksi

permanen (*persistence*) yang berjalan setelah sebuah halaman telah selesai dilayani / dikerjakan.



Gambar II.3. Perbandingan State dalam Sistem Aplikasi: A.*State* yang kontinyu dalam aplikasi *desktop*, dan B.*Stateless* protokol dalam aplikasi web

Agar sebuah situs web mempunyai memori / *state*, dalam hal ini aplikasi ini mampu mengingat ‘siapa memesan apa’, beberapa informasi yang mengidentifikasi pengguna harus dikirim dengan setiap *request* halaman web. Informasi tersebut disimpan dengan menggunakan *session.* Session tersebut dipergunakan untuk merekam / *tracking* aktivitas pengguna yang melalui sejumlah halaman pada website, misalnya pada jenis aplikasi *Shopping Cart* (kereta belanja). Direkam pula informasi identitas pengguna yang memiliki kereta belanja tersebut.

Dengan PHP, untuk penggunaan session ini mula-mula dilakukan pengaturan pada file php.ini yang menunjukkan *session* dimulai (*start*). Dengan ini PHP akan membuat suatu identifier unik dan file yang berkaitan, yang disimpan di server (lokasinya di atur di php.ini dan nilai *default*nya di direktori /tmp). Kemudian pada saat pengguna berkunjung pada halaman-halaman situs web, semua informasi variabel yang dipilih oleh pengguna akan disimpan dalam file pada server, dan semua script yang dibutuhkan untuk melacak sebagai identifier unik.

Implementasi *session* dapat mempergunakan *cookie* yang disimpan pada sisi *client,* atau dipropagasikan melalui alamat URL.

Untuk penggunaan *cookie,* yaitu dengan *passing* variabel melalui *cookie* yang menyimpan informasi semua elemen barang belanja dan harganya. Namun hal ini memiliki keterbatasan yaitu dari www.netscape.com mengenai spesifikasi cookie yang hanya mengijinkan 20 *cookie* per domain dan berukuran hanya 4 bytes per *cookie*. Cara lain adalah dengan memberi identitas / *identifier* unik pada masing-masing pengguna, suatu nilai unik yang mengidentifikasi siapa pengguna tersebut. Sehingga pada saat pengguna menambahkan satu item pada kereta belanja, informasi yang berkaitan dengan identifier unik tadi disimpan di komputer server. Jika menggunakan cookie untuk fungsi penyimpan informasi tadi, diperlukan membuat *string* unik yang akan diletakkan dalam *cookie*, dalam direktori di server akan terdapat sebuah file yang memiliki nama yang sama sebagai ID pengguna yang unik. Dalam file tersebut dapat disimpan semua variabel yang berkaitan dengan pengguna. Contohnya terdapat *array* berisi item-

item barang yang ditambahkan oleh seorang pengguna ke dalam kereta belanjanya.

Terdapat keterbatasan penggunaan *cookie,* yakni bila browser pengguna di atur untuk menolak (*reject*) *cookie*.

Metode lain yang dapat digunakan adalah dengan propagasi URL, yaitu dengan mengaktifkan *flag* enable-trans-sid dalam konfigurasi PHP, hal ini berguna agar *session id* akan secaraotomatis ditambahkan ke setiap *relative link* pada halaman-halaman web setiap kali *session* telah dimulai.

2.8 Tinjauan Perangkat Lunak

MySQL dan PHP merupakan kombinasi pasangan piranti (*tools*) yang banyak dan umum digunakan dalam pembangunan dan pengembangan situs web berbasis database dalam sajian informasinya. MySQL dan PHP dapat dikatakan merupakan pasangan software yang serasi untuk pengembangan aplikasi web berbasis database.

2.8.1 PHP (*Hypertext Prepocessor*)

PHP adalah bahasa pemrograman *serer-side*, bahasa pemrograman yang digunakan oleh server web untuk menghasilkan dokumen HTML secara *on the fly*. PHP merupakan interpreter yang dapat dieksekusi sebagai program CGI untuk server web atau dijadikan modul dari server web. PHP merupakan bahasa script selain paling populer di lingkungan pemrogram, pengembang web, di lingkungan

server web Apache, kini juga telah menjadi salah satu alternatif bahasa script di lingkungan server web di Windows.

PHP telah tersedia pada hampir semua sistem operasi jaringan yang menyediakan server web terutama server web Apache. Server web berbasis Windows non Apache juga telah mendukungnya, seperti MS IIS, PWS, atau Xitami, dari Windows 98/ME sampai dengan Windows NT4/2000 dan XP.

Portabilitas aplikasi yang dikembangkan dengan menggunakan PHP lebih mudah, dan tidak membutuhkan perubahan pada source code aplikasi, salinkan langsung ke dalam server tujuan, maka aplikasi dengan PHP langsung dapat dijalankan. Portabilitas adalah kemampuan untuk dipindahkan dari satu platform sistem operasi kepada platform sistem operasi lain, misal dari Windows ke Linux atau sebaliknya. Kemudahan portabilitas ini adalah karena samanya PHP pada semua platform.

2.8.2 My SQL

MySQL adalah *Relational Database Management Sistem* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Dimana setiap orang bebas untuk menggunakannya, tapi tidak boleh dijadikan produk turunan yang bersifat c*losed source* atau komersial.

MySQL sebenarnya merupakan turunan salah satu konsep utama dalam *database* sejak lama, yaitu SQL (*Structure Query Language*). SQL adalah sebuah konsep pengoperasian *database*, terutama untuk pemilihan atau seleksi dan

pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Sebagai *database server* yang memiliki konsep *database* modern, MySQL memiliki banyak keistimewaan, diantaranya:

1. *Portability*, MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FeeBSD, dan lain-lain
2. *Open Source*, MySQL didistribusikan secara *open source*, sehingga dapat digunakan secara bebas.
3. *Multi-user*, MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah. Hal ini memungkinkan sebuah *database server* MySQL dapat diakses *client* secara bersamaan
4. *Performance Tuning*, MySQL memiliki kecepatan yang tinggi dalam menangani *query*, dengan kata lain dapat memproses lebih banyak SQL persatuan waktu.
5. *Column Types*, MySQL memiliki tipe kolom yang sangat kompleks, seperti signed, unsigned, integer, float, double, char, varchar, text, date, time, timestamp, year, set dan enum.
6. *Command* dan *Function*, MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah SELECT dan WHERE dalam query.
7. *Security*, MySQL memiliki lapisan–lapisan sekuritas seperti level subnet mask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta password terenkripsi

8. *Scalability* dan *Limits*, MySQL mampu menangani *database* dengan skala besar, dengan jumlah records lebih dari 50 juta dan 60 juta tabel serta 5 miliar baris. Selain itu, batas indeks yang dapat ditampung mencapai 32 indeks pada setiap tabelnya.
9. *Connectivity*, MySQL dapat melakukan koneksi dengan *client* menggunakan protocol TCP/IP, Unix Socet (UNIX), atau Named Pipes (NT).
10. *Locallisation*, deteksi pesan kesalahan pada *client* dengan menggunakan lebih dari 20 bahasa.
11. *Interface*, terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).
12. *Client* dan *Tools*, dilengkapi dengan berbagai *tool* yang dapat digunakan untuk administrasi *database*, dan pada setiap *tool* yang ada disertakan petunjuk *online*.
13. Struktur tabel, yang lebih fleksibel dalam menangani *ALTER TABLE*, dibanding *database* lainnya semacam PostgreSQL ataupun oracle.

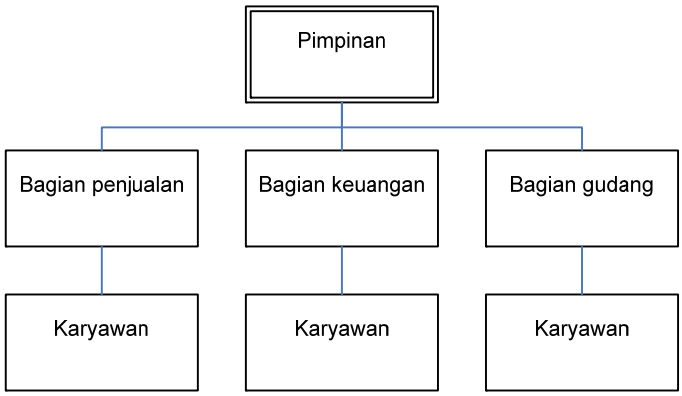
2.9 Tinjauan Tempat Penelitian

CV.HEXA merupakan perusahaan yang bergerak dalam bidang penjualan barang, khususnya produk-produk komputer dan aksesoris, beralamatkan di jalan Dipatiukur no.66 Bandung. Berdiri pada tahun 1998, yang didirikan oleh Bapak Ir. Agung Basuki. Seiring dengan waktu dan perkembangan perusahaan, Bapak Ir. Agung Basuki dibantu oleh teman-teman sejawatnya yang akhirnya perusahaan

ini mengalami perubahan menjadi CV.HEXA terhitung mulai tahun 2000 dengan pimpinannya Bapak Ir. Agung Basuki sampai sekarang.

2.9.1 Struktur Organisasi

Struktur organisasi yang diterapkan di CV.HEXA adalah sebagai berikut :



Gambar II.4. Struktur Organisasi CV.HEXA

2.9.2 Deskripsi Kerja Bagian-Bagian di CV.HEXA

Dengan melihat struktur organisasi diatas maka dapat diuraikan mengenai tugas dan fungsi dari masing-masing bagian.

1. Pimpinan

Adalah seorang yang mengatur seluruh gerak-gerik dan langkah-langkah yang harus dilakukan dalam kegiatan sehari-hari, termasuk rencana-rencana ke depan yang berhubungan dengan kemajuan dari perusahaan tersebut.

2. Bagian gudang

   Tugas utama bagian gudang sebagai berikut :

   - Mengelompokkan barang sesuai jenisnya.
   - Mengolah barang sebelum dan setelah barang itu dijual.

3. Bagian penjualan

   Tugas utama dari bagian penjualan sebagai berikut :

   - melakukan penjualan dan mengirim barang pada pembeli.
   - Membuat bukti atau faktur penjualan.

4. Bagian keuangan

   Tugas utama bagian keuangan sebagai berikut :

   - Mengelola keuangan perusahaan yang meliputi pelaksanaan administrasi keuangan atau perbendaharaan.
   - Membuat laporan tiap satu minggu sekali.
   - Serta mengontrol keuangan perusahaan.