BAB II
LANDASAN TEORI

2.1 Sistem Informasi

Pertama yang harus diketahui adalah: "Apa yang disebut sistem ?". Sistem dapat didefinisikan sebagai kumpulan dari beberapa bagian atau elemen yang saling berinteraksi untuk mencapai suatu tujuan. Ada banyak sekali sistem yang kita kenal : sistem tubuh, sistem mesin, sistem bisnis, dan sebagainya. Setiap sistem memiliki elemen-elemen yang berbeda-beda.

Ada beberapa definisi sistem informasi dari beberapa pakar sebagai acuan agar kita dapat mengerti "Apa yang dimaksud dengan sistem informasi":

> *David Kroenke* mendefinisikan sistem informasi sebagai : *kumpulan elemen-elemen secara integratif seperti perangkat keras komputer, perangkat lunak, data, prosedur dan pemakai komputer yang berinteraksi untuk memenuhi kebutuhan pengolahan data atau informasi pemakainya*[5].
>
> *James A.O'Brien* mendefinisikan sistem informasi (berbasis komputer) sebagai *suatu sistem yang menggunakan personil, langkah-langkah operasi prosedur tertentu, dan sistem pengolahan data untuk mengumpulkan dan memproses data serta menyebarkan informasi ke seluruh bagian organisasi (perusahaan) yang membutuhkan*[5].

Selanjutnya, digunakan definisi sistem informasi dari James A.O'Brien.

Suatu sistem informasi mempunyai komponen fisik sebagai berikut [5]:

1. Perangkat keras (*Hardware*): peralatan-peralatan yang digunakan untuk mengolah data.
2. Data: sesuatu yang akan diolah yang mencerminkan objek permasalahan yang perlu diselesaikan.
3. Prosedur : langkah-langkah pekerjaan yang perlu diikuti oleh personil agar sistem informasi dapat bekerja dengan baik.
4. Personil (*Peopleware* atau *Brainware*): pengelola dari sistem informasi.

Sistem informasi selalu menghasilkan informasi yang ditujukan untuk meningkatkan produktivitas kerja organisasi dan membantu proses pengambilan keputusan di level operasional maupun manajerial dalam suatu organisasi.

Informasi yang dipakai dalam setiap level berbeda-beda, semakin tinggi level organisasinya, semakin besar kerahasiaannya. Oleh karena itu, untuk mengakses suatu informasi diperlukan otoritas yang cukup, sebab semakin rendah level organisasinya, semakin detil informasi yang dimiliki.

Sistem informasi berbasis komputer merupakan sistem informasi yang menggunakan teknologi komputer sebagai pengolah datanya. Sistem informasi berbasis komputer memiliki komponen fisik yang 'lebih' dari sistem informasi manual/biasa, yaitu perangkat keras yang berteknologi komputer, perangkat lunak, pengguna sistem, basis data dan prosedur.

Dengan bantuan basis data, maka pengolahan data menjadi lebih mudah, lebih akurat dan lebih cepat, tetapi untuk membuat basis data yang handal di perlukan *entity relationship* yang baik.

2.2 Sistem Informasi Berbasis *Web*

Arsitektur interaksi *web-based* terdiri dari *client* sebagai lapisan terluar, jaringan *web server* sebagai lapisan tengah dan *web server* sebagai intinya. *Browser* bertindak sebagai perangkat lunak mediator serta merupakan antar muka antara *client* dan sistem. *Web server* memberikan koneksi dan hak akses interaksi kepada *client* dengan aplikasi yang ter-register melalui *browser* konvensional.

2.2.1 *Web Server*

Web server (atau lebih tepatnya *World Wide Web Server*) adalah *server internet* yang mampu melayani koneksi transfer data dalam protokol HTTP (*hypertext transfer protocol*). *Web server* saat ini merupakan inti dari *server-server internet* selain e-mail *server*, ftp dan news *server*. Hal ini dapat dimaklumi karena *web server* telah dirancang untuk dapat melayani beragam jenis data, mulai dari *text*, *hypertext*, gambar (*image*), suara dan sebagainya. Keunggulan ini membuat *web* dapat diterima tidak saja di kalangan institusi universitas, melainkan hampir semua perusahaan komersial saat ini telah memiliki sebuah atau beberapa *web server* sekaligus di *Internet*.

Web server pada umumnya melayani data dalam bentuk file HTML (*hypertext markup language*). Dari file ini kemudian dapat dikaitkan ke file HTML lainnya, ke file gambar, ke file suara dan segala jenis file yang dipublikasikan di *internet*.

Ada beberapa *web server* komersil dan *web server* gratis yang dapat kita pergunakan, tetapi ada dua perangkat lunak dalam bidang *web server* yang menguasai hampir 70% pasar dunia yaitu *Apache's Web Server* dan *Microsoft's Internet Information Server*.

Ketika kita mempergunakan *web* untuk menampilkan halaman *web* yang kita inginkan, maka secara otomatis kita telah berhubungan dengan *Web Server*. Proses untuk menyerahkan URL (*Uniform Resource Locator*) disebut sebagai permintaan pada *server* (*request to server*). *Server* akan menginterpretasikan URL (*Uniform Resource Locator*) yang dimaksud dan kemudian mengalokasikan halaman yang sesuai kemudian mengirimkannya ke *browser*. Proses ini biasa disebut respon ke *browser*. *Browser* kemudian akan mengambil kode-kode yang telah diterima dari *server* dan kemudian dikompilasi agar dapat menampilkan halaman yang dimaksud. Interaksi ini menunjukkan *browser* sebagai *client* dan seluruh interaksi lainnya sebagai hubungan *client-server* (*client-server relationship*).

2.2.2 *Client-Server*

Web Server bertugas untuk menyimpan, menginterpretasikan dan mendistribusikan data dan *client* (*browser*) berfungsi sebagai pengakses *server* untuk mendapatkan data. Untuk mengetahui lebih dalam mengenai *client-server* kita perlu untuk mengetahui tentang bagaimana *internet* itu sendiri bekerja.

Internet dirancang untuk membawa informasi dari satu tempat ke tempat lain. Proses ini dilakukan dengan menggunakan *networking protocols* (TCP/IP) untuk melakukan pemindahan informasi. *Networking protocols* adalah metode sederhana untuk memberikan paket informasi yang bisa dikirim melalui telepon, kabel atau media lainnya melalui *node - node* sampai informasi tersebut dapat tiba pada tempat yang dituju.

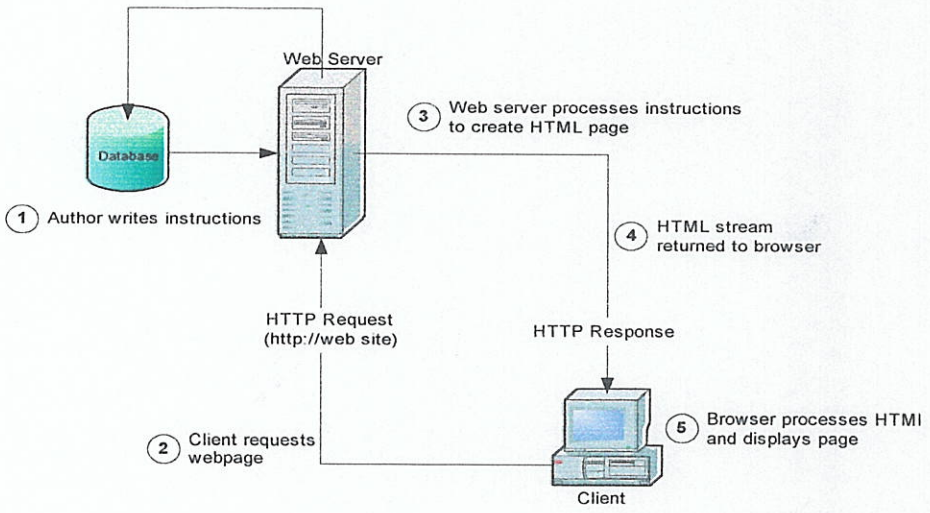
Ketika *user* melakukan permintaan pada *browser* untuk mengambil halaman *web*, *browser* kemudian memaketkan instruksi/pesan ini dengan menggunakan *protocol* yang disebut *Transmission Control Protocol* (TCP). TCP adalah *transport protocol* yang menyediakan format transmisi untuk instruksi yang diminta dan memastikan seluruh instruksi atau pesan yang dipaketkan adalah benar.

Sebelum paket data dikirim melalui jaringan, terlebih dahulu harus diberi alamat. Untuk itu ada *protocol* kedua yaitu *Hypertext Transfer Protocol* (HTTP) yang bertugas untuk memberikan label alamat pada paket data, agar TCP/ IP dapat langsung mengirimkan alamat tersebut. HTTP adalah *protocol* yang digunakan oleh *World Wide Web* dalam mengirim informasi dari satu mesin ke

mesin yang lain. Jika kita melihat URL yang diawali dengan http://, menandakan bahwa *internet protocol* yang dipergunakan adalah HTTP.

Kita dapat mengilustrasikan TCP/IP sebagai sebuah kantor pos dan HTTP sebagai perangko dan data pada surat yang memastikan dimana surat tersebut harus dikirim. Pesan yang lewat dari *browser* ke *web server* dikenal dengan HTTP *request*. Ketika *web server* menerima permintaan, *web server* kemudian akan memeriksa di tempat penyimpanan untuk menemukan halaman yang sesuai.

Jika *web server* menemukan halaman yang dituju, maka kemudian akan dikirimkan *file* HTML (menggunakan TCP) yang berisikan apa yang diinginkan oleh *user*, mengalamatkan bingkisan ke *browser* (menggunakan HTTP) dan mengirimkannya melalui jaringan. Namun jika *web server* tidak dapat menemukan halaman yang diminta, maka akan dihasilkan halaman yang berisi pesan kesalahan. Pesan yang dikirim dari *web server* ke *browser* disebut HTTP *response* .



Gambar 2.1 Ilustrasi proses *server request* dan *response*

Jadi dapat diambil kesimpulan bahwa sistem informasi berbasis *web* adalah sistem informasi yang memanfaatkan *web server* sebagai media penghubung antara *database* dan *client*, dengan tujuan agar dapat diakses oleh semua pengguna *internet*.

2.3 Desain Perangkat Lunak

Desain perangkat lunak merupakan salah satu bagian teknis dari desain sistem informasi, proses desain sangat penting karena menentukan kualitas dari perangkat lunak. Desain juga merupakan cara untuk menerjemahkan kebutuhan *customer* menjadi desain perangkat lunak yang baik.

Dasar-dasar proses desain sistem informasi [5]:

1. Abstraksi

   Desain perangkat lunak merupakan proses *refinement* dari solusi kebutuhan perangkat lunak, dari abstraksi yang dekat dengan solusi permasalahan (*high level*) dan abstraksi yang dekat dengan representasi perangkat lunak (*low level*).

2. *Refinement*

   Merupakan strategi *top-down* desain dari abstraksi global sampai abstraksi detil yang siap untuk dikodekan.

3. Modularitas

   Merupakan strategi untuk mengurangi kompleksitas permasalahan, melokalisir perubahan, memudahkan proses implementasi, dan menghemat

resource yang ada yaitu dengan memecah solusi kebutuhan perangkat lunak menjadi komponen-komponen fungsional yang terpisah.

4. Hirarki kontrol

   Merepresentasikan hirarki komponen program (modul) dan menyatakan hirarki kontrol dengan menggunakan notasi diagram pohon.

5. Struktur data

   Merupakan representasi hubungan logika antara elemen-elemen individual data.

6. Prosedur perangkat lunak

   Menekankan pada proses detil dari setiap modul, yang di dalamnya terkandung urutan kejadian, titik keputusan, operasi pengulangan, dan organisasi / struktur data kejadian.

7. *Information hiding*

   Merupakan konsep enkapsulasi data dan prosedur pada suatu modul, dimana data dan prosedur dalam suatu modul tidak dapat diakses melalui modul lain.

Berdasarkan aspek teknis desain dibagi menjadi [5] :

1. Desain data

   Konsep yang mendasari desain data adalah *information hiding* dan abstraksi data. Menurut *Wasserman* proses desain adalah memilih representasi logika dari objek data (struktur data) yang diidentifikasikan

selama tahap pendefinisian dan tahap spesifikasi kebutuhan (analisis) yang paling efisien dan sederhana.

2. Desain arsitektur

   Memiliki tujuan membangun struktur modul program dan merepresentasikan hubungan antar modul.

3. Desain prosedural

   Pada tahap ini dilakukan transformasi komponen terstruktur menjadi deskripsi prosedural perangkat lunak (algoritma).

4. Desain antarmuka

2.4 Rekayasa perangkat lunak

Rekayasa perangkat lunak terdiri dari sekumpulan metode, *tool*, dan prosedur yang disebut paradigma rekayasa perangkat lunak. Paradigma rekayasa perangkat lunak yang digunakan penulis adalah: "*Classic Life Cycle* "(Daur Hidup Klasik) atau model *waterfall.* Definisi rekayasa perangkat lunak oleh Fritz Bauer: *pembentukan dan penggunaan prinsip-prinsip rekayasa yang baik untuk memperoleh perangkat lunak yang ekonomis, handal, dan dapat bekerja secara efisien pada mesin nyata.*

Rekayasa perangkat lunak melibatkan tiga elemen kunci [8] :

1. *methods* : memberikan teknik dalam pengembangan perangkat lunak
2. *tools* : memberikan dukungan terhadap metode
3. *procedures* : menghubungkan *methods* dengan *tools*

Metode ini membutuhkan pendekatan yang sistematis dan sekuensial dalam pengembangan perangkat lunak. Pemodelan ini menyangkut aktivitas-aktivitas berikut ini :

1. Rekayasa sistem dan analisis (*sistem engineering and analysis*)

   Perangkat lunak merupakan bagian dari sistem yang lebih besar, oleh karena itu pekerjaan dimulai dari pembentukan kebutuhan-kebutuhan dari semua elemen sistem dan mengalokasikan suatu subset ke dalam pembentukan perangkat keras. Hal ini penting karena perangkat lunak harus berkomunikasi dengan perangkat keras, manusia, dan basis data. Rekayasa sistem dan analisis menekankan pada pengumpulan kebutuhan pada level sistem dengan sedikit desain dan analisis.

2. Analisis kebutuhan perangkat lunak (*software requirement analysis*)

   Proses pengumpulan kebutuhan diintensifkan ke perangkat lunak, harus dapat dibentuk domain informasi, fungsi-fungsi yang dibutuhkan, performansi dan antarmuka. Hasilnya harus didokumentasikan dan di-*review* ke *customer.*

3. Desain (*design*)

   Ada empat atribut untuk program, yaitu struktur data, arsitektur perangkat lunak, prosedur *retail,* dan karakteristik antarmuka. Proses desain mengubah kebutuhan-kebutuhan menjadi bentuk karakteristik yang dimengerti perangkat lunak sebelum dimulai penulisan program. Desain

juga harus didokumentasikan dengan baik dan menjadi bagian konfigurasi perangkat lunak.

4. Penulisan program atau pengkodean (*coding*)

   Setelah proses desain selesai, hasil desain harus diubah menjadi bentuk yang dimengerti oleh mesin, yaitu dengan melakukan langkah penulisan program. Jika desainnya detil, maka proses pengkodean dapat dicapai secara mekanis.

5. Pengujian (*testing*)

   Setelah kode program selesai, dan program dapat berjalan, proses pengujian dimulai. Pengujian difokuskan pada logika internal dari perangkat lunak, fungsi-fungsi eksternal dan mencari segala kemungkinan kesalahan, selain itu juga untuk memeriksa apakah hasil program sesuai dengan yang diinginkan.

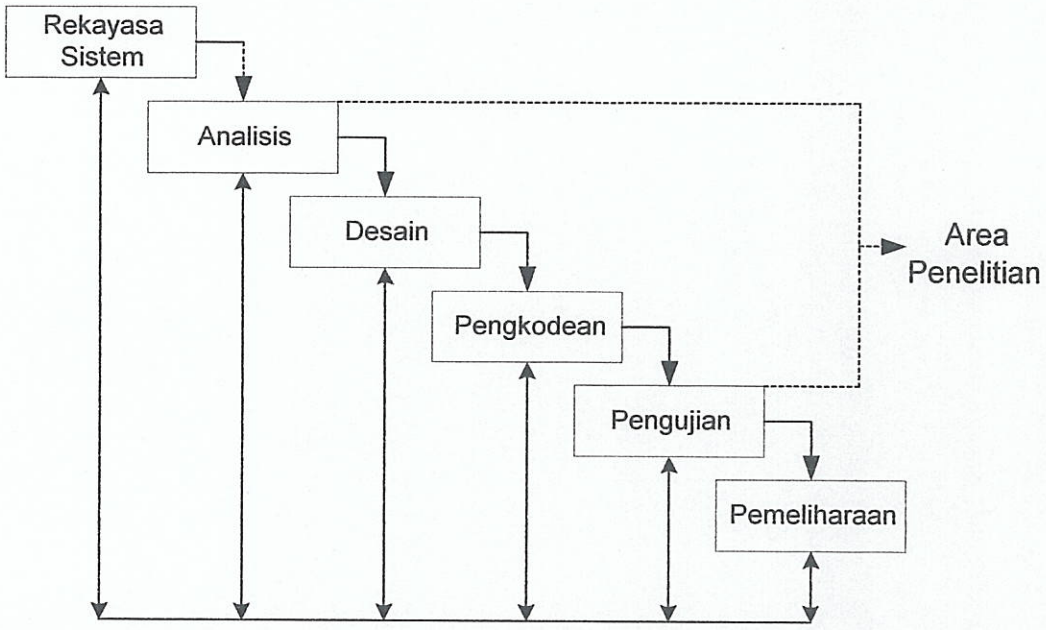
6. Pemeliharaan (*maintenance*)

   Perangkat lunak yang telah selesai dibangun diberikan pada *customer*, setelah proses ini selesai maka mungkin akan ditemukan kesalahan (*error*) ketika dijalankan, atau mungkin *customer* menginginkan penambahan fungsi. Hal ini yang membuat pemeliharaan menjadi hal penting dalam penggunaan metode ini.

Dari aktivitas yang disebutkan , dalam pengembangan perangkat lunak ini yang tidak dilakukan adalah :

1. Rekayasa Sistem

2. Pemeliharaan



Gambar 2.2 Model *Waterfall*

2.5 Metode Analisis Kebutuhan Perangkat Lunak

Metode analisis kebutuhan perangkat lunak [6]:

1. *survei* sistem kini
2. wawancara dengan pengguna dan pihak manajemen tentang :
    - kekurangan sistem kini
    - usulan pengembangan
    - kebijakan organisasi
3. survei perangkat lunak jenis lain untuk mempelajari *feature-feature* umum
4. *prototyping* dan *review*

Prinsip spesifikasi kebutuhan perangkat lunak [6] :

1. Memisahkan fungsionalitas dari implementasi
2. Harus menjabarkan lingkungan tempat sistem beroperasi
3. Harus menjabarkan sistem yang merupakan super sistem dari perangkat lunak
4. Sistem harus merupakan model kognitif
5. Harus bersifat operasional
6. Harus toleran terhadap ketidaklengkapan dan dapat dikembangkan
7. Harus teralokasi dan dapat dilepas satu sama lain (*loosely coupled*)

Metode spesifikasi kebutuhan perangkat lunak yang dibutuhkan pada skripsi ini adalah :

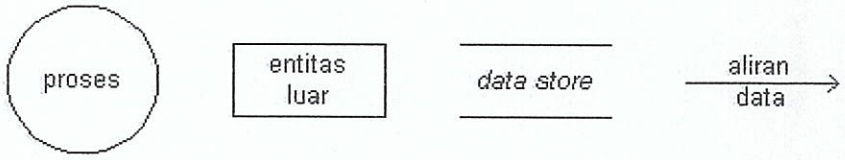
➔ Metode spesifikasi informal

1. Bahasa alami (*natural language*)
2. *Structured System Analysis*

   Metode analisis struktural adalah metode pembuatan yang didasarkan atas aliran informasi dan kandungannya. Untuk menggambarkan alir informasi tersebut digunakan *Data Flow Diagram* (DFD), yang memiliki 4 notasi dasar :

1. entitas luar : entitas luar dari suatu proses yang akan menjadi sumber atau tujuan aliran data dari suatu proses
2. proses : aktivitas yang dilakukan untuk memanipulasi data

3. aliran data : menggambarkan data yang mengalir dari entitas luar ke proses, proses ke entitas luar, proses ke proses, proses ke *data store* atau sebaliknya
4. *data store* : tempat menyimpan data



Gambar 2.3 Notasi dasar DFD

2.6 *Entity Relationship* Diagram

Entity Relationship adalah diagram yang digunakan untuk menjelaskan hubungan yang terjadi antar data yang ada di dalam sistem informasi.

Entity relationship terbentuk dari 3 komponen informasi [8] :

1. Objek : Merepresentasikan komponen utama dalam sistem.
2. Atribut : Menjelaskan properti objek dan dapat digunakan sebagai penghubung antar objek.
3. Relationship : Menjelaskan hubungan yang terjadi antar objek.

Contoh : Dalam data kepemilikan kendaraan terdapat :

Objek : 1. Pemilik
2. Kendaraan

Atribut : Atribut Pemilik
1.1. Nama

1.2. Alamat

1.3. Umur

1.4. Nomor SIM

Atribut kendaraan

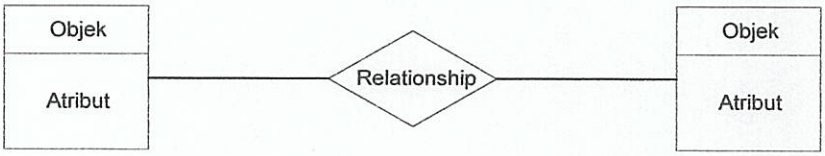
2.1. Jenis Mobil

2.2. Nomor Kendaraan

2.3. Tahun pembuatan

2.4. Warna Kendaraan

Relationship : Mempunyai gambaran seperti berikut :



Gambar 2.4 Diagram *Entity Relationship*

2.7 Paradigma Analisis dan Desain

Paradigma yang digunakan dalam analisis dan desain perangkat lunak pada skripsi ini adalah analisis dan desain struktural dengan paradigma *waterfall*. Analisis struktural adalah metode pengembangan untuk menganalisa sistem yang ada, yang dilanjutkan dengan pengembangan spesifikasi untuk sistem yang baru [1].

Dalam analisis struktural digunakan *Data Flow Diagram* (DFD) yaitu teknik grafis yang digunakan untuk menggambarkan aliran data dan informasi dan

juga transformasi yang mengalir dari *input* ke *output*. DFD juga dikenal sebagai *data flow graph* atau *bubble chart* [8].

DFD adalah peralatan grafis yang digunakan untuk mendeskripsikan dan menganalisa pergerakan data dalam sebuah sistem termasuk juga proses penyimpanan data dalam sistem. DFD adalah alat utama dan basis pengembangan komponen-komponen lain.[1]

DFD juga digunakan untuk merepresentasikan sistem atau perangkat lunak pada setiap level abstraksi. DFD dapat dipartisi menjadi beberapa level yang merepresentasikan alur informasi yang semakin detil dan juga detil fungsional, DFD juga menyediakan mekanisme untuk model fungsional dan model alur informasi.[1]

DFD level 0 yang dikenal sebagai *fundamental sistem* model atau *context model* yang merepresentasikan seluruh elemen perangkat lunak dalam sebuah model dengan *input* dan *output* data yang diindikasikan dengan anak panah. DFD level 0 akan dipartisi kembali untuk mendapatkan spesifikasi yang lebih detil pada level yang lebih tinggi yaitu DFD level 1 dan seterusnya.[1]