

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Pengertian Sistem Informasi**

Menurut Hartono (1999:1) sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Menurut Hartono (1999:8) informasi adalah data yang diolah mejadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Data merupakan bentuk yang masih mentah yang belum dapat bercerita banyak, sehingga perlu diolah lebih lanjut.

“Sistem informasi adalah suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.” (Leitch, 1983)

#### **2.2 Analisis Sistem dan Desain Sistem**

##### **2.2.1 Analisis sistem**

Analisis sistem dapat didefinisikan sebagai penguraian suatu sistem informasi yang utuh ke dalam bagian-bagian komponen dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan.

Analisis sistem dilakukan setelah tahap perencanaan sistem sebelum tahap perencanaan sistem. Langkah-langkah dasar dalam melakukan analisis (Hartono, 1999) :

- a. *Identify*, yaitu mengidentifikasi masalah.
- b. *Understand*, yaitu mengenal masalah.
- c. *Analyze*, yaitu menganalisis masalah.
- d. *Report*, yaitu membuat laporan hasil analisis.

Setelah analisis sistem dilakukan, tahap selanjutnya adalah perancangan sistem. Perancangan sistem dapat didefinisikan sebagai tahap setelah:

- a. *Identify*, yaitu mengidentifikasi masalah.
- b. *Understand*, yaitu mengenal masalah.

Perancangan sistem mempunyai dua tujuan utama, yaitu memenuhi kebutuhan kepada pemakai dan untuk memberikan gambaran yang jelas dan rancang bangun yang lengkap kepada pemrogram dan ahli teknik lainnya yang terlibat.

### **2.2.2 Desain sistem**

Desain sistem adalah tahap setelah analisis dari siklus pengembangan sistem pendefinisian dari kebutuhan-kebutuhan fungsional dan persiapan untuk rancang bangun implementasi; menggambarkan bagaimana suatu sistem dibentuk.

“Pada tahap desain secara umum, komponen-komponen sistem informasi dirancang dengan tujuan untuk dikomunikasikan kepada *user* bukan untuk pemrogram. Komponen sistem informasi yang didesain adalah model, *output*, *input*, *database*, teknologi, dan kontrol.” (Hartono, 1999).

Analisis sistem dapat mendesain model dari sistem informasi yang diusulkan dalam bentuk *physical system dan logical model*. Bagan alir sistem (*systems flowchart*) merupakan alat yang tepat digunakan untuk menggambarkan *physical system*. Simbol-simbol bagan alir sistem ini menunjukkan secara tepat arti fisiknya, seperti simbol terminal, *hard disk*, laporan-laporan.

Logical Model dari sistem informasi lebih menjelaskan kepada user bagaimana nantinya fungsi-fungsi di sistem informasi secara logika akan bekerja. Logical Model dapat digambarkan dengan menggunakan diagram arus data (*data flow diagram*). Arus dari data di DAD dapat dijelaskan dengan menggunakan kamus data (*data dictionary*). Sketsa dari *physical system* dapat menunjukkan kepada user bagaimana nantinya sistem secara fisik akan diterapkan

## 2.3 PHP

PHP singkatan dari *PHP Hypertext Preprocessor* adalah suatu bahasa yang berbentuk skrip yang ditempatkan dalam server dan diproses di server (*server side*). Hasilnya yang dikirimkan ke klien, tempat pemakai menggunakan *browser* (Kadir, 2002).

### 2.3.1 Sejarah PHP

PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama FI (Form Interpreted), yang wujudnya berupa sekumpulan script yang digunakan untuk mengolah data form dari web.

Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI, kependekan dari Hypertext Preprocessing/Form Interpreter. Dengan perlisian kode sumber ini menjadi open source, maka banyak programmer yang tertarik untuk ikut mengembangkan PHP.

Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini interpreter PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan.

Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0.

Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi. Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek

### **2.3.2 Kelebihan PHP**

Ada beberapa kelebihan bahasa pemrograman PHP, yaitu :

- PHP mudah dibuat dengan kecepatan akses tinggi.
- PHP dapat berjalan dalam *web server* yang berbeda dan dalam sistem operasi yang berbeda pula. PHP dapat berjalan di sistem operasi UNIX, Windows dan Macintosh.
- PHP diterbitkan secara gratis.

- PHP juga dapat berjalan pada *web server* Microsoft Personal Web Server, Apache, IIS, Xitami dan sebagainya.
- PHP termasuk *server-side programming*.
- PHP mampu berkomunikasi dengan berbagai database yang terkenal seperti Oracle, MySQL, Sybase, dBase, dll.

## 2.4 Javascript

Berbicara tentang web tidak lepas dari HTML, bahasa yang digunakan untuk membuat halaman web. Namun, untuk membuat halaman web yang interaktif, HTML saja tidaklah cukup. Oleh karena itu munculah CGI (Common Gateway Interface). Dengan CGI, halaman web bisa menjadi lebih interaktif. Sayangnya CGI meletakkan beban pemrosesan di server, bukan di klien. Akibatnya dengan semakin banyaknya pengguna yang mengakses server, lalu lintas jaringan semakin padat. Disamping CGI ada bahasa pemrograman lain yang juga digunakan untuk meningkatkan interaktivitas halaman web, yaitu Java. Java dikembangkan oleh Sun Microsystem dari bahasa pemrograman yang sudah populer, yaitu C++. Pada Java, beban pemrosesan dialihkan ke klien sehingga beban server dapat dikurangi. Selain itu, kemampuan java juga melebihi CGI.

Dari latar belakang inilah, Netscape Communications bersama dengan Sun Microsystems mengembangkan bahasa skrip yang diberi nama javascript. Sebenarnya javascript dikembangkan dari bahasa livescript yang khusus dirancang untuk Netscape Navigator. Dengan javascript dapat dibuat halaman web yang interaktif dan juga cerdas.

## 2.5 Database

*Database* adalah kumpulan file-file yang saling berelasi, relasi tersebut bisa ditunjukkan dengan kunci dari tiap file yang ada. Satu *database* menunjukkan satu kumpulan data yang dipakai dalam satu lingkup perusahaan, instansi. Dalam merancang aplikasi database, selain harus mengatur rutin program agar kinerjanya maksimum, juga harus memperhatikan tata letak penyimpanan data, baik secara fisik maupun logis. Perancangan *database* yang baik memberikan waktu pencarian minimum dalam menentukan lokasi *record-record* tertentu. *Database* yang baik juga harus mampu menyimpan data seefisien mungkin agar *database* tidak berkembang terlalu besar. Dengan demikian data dapat diperbarui dengan mudah.

Perancangan *database* yang baik melibatkan 7 aktivitas kunci yaitu :

1. Membuat model aplikasi.
2. Menentukan data yang diperlukan aplikasi.
3. Mengorganisir data ke dalam table.
4. Menetapkan hubungan antar table.
5. Menetapkan kebutuhan indeks, validasi dan integritas bagi data.
6. Membuat dan menyimpan query yang perlu untuk aplikasi.
7. Meninjau ulang rancangan *database*.

## 2.6 Database Management System

Basis Data (*database*) merupakan kumpulan dari data dan atau informasi yang saling berhubungan satu dengan lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Basis Data

(database) diakses atau dimanipulasi dengan menggunakan perangkat lunak paket yang disebut dengan DBMS (Database Management System).

Semua DBMS memiliki pengolah bahasa deskripsi data (data description language processor) yang digunakan untuk menciptakan *database*, serta pengolah *database* yang menyediakan isi *database* bagi pemakai. Pemakai menggunakan manipulasi data dan *query language*. Orang yang bertanggung jawab atas database dan DBMS adalah pengelola database (database administrator), atau disingkat DBA.

## 2.7 Oracle

Basis data Oracle ini pertama kali dikembangkan oleh Larry Ellison, Bob Miner dan Ed Oates lewat perusahaan konsultasinya bernama *Software Development Laboratories (SDL)* pada tahun 1977. Pada tahun 1983, perusahaan ini berubah nama menjadi Oracle Corporation sampai sekarang.

Oracle adalah DBMS hasil produksi dari Oracle Corporation yang berpusat di USA, yang sekaligus berperan sebagai vendor yang mengkhususkan diri pada produk-produk DBMS untuk berbagai jenis komputer dan mendukung system operasi dari berbagai vendor. Mulai dari PC yang berbasis Windows atau Linux sampai dengan komputer mainframe baik yang IBM compatible maupun yang berasal dari vendor lain seperti SUN Microsystems, Hewlett Packard, dan lain-lain. Sifat kompatibilitas dengan berbagai platform itulah yang membuat Oracle menjadi salah satu DBMS yang banyak digunakan dewasa ini. Oracle pertama kali diproduksi sekitar tahun 1984 dan terus mengalami perkembangan sampai sekarang ini (William, 2008)

## 2.8 Software Testing

Beberapa definisi tentang testing :

1. Menurut Hetzel 1973 :

Testing adalah proses pemantapan kepercayaan akan kinerja program atau sistem sebagaimana diharapkan.

2. Menurut Myers 1979 :

Testing adalah proses eksekusi program atau sistem secara intens untuk menemukan error.

3. Menurut Hetzel 1983(Revisi) :

Testing adalah tiap aktivitas yang digunakan untuk dapat melakukan evaluasi suatu atribut atau kemampuan dari suatu program atau sistem dan menentukan apakah telah memenuhi kebutuhan atau hasil yang diharapkan

4. Menurut Standar ANSI/IEEE 1059 :

Testing adalah proses menganalisa suatu entitas software untuk mendeteksi perbedaan antara kondisi yang ada dengan kondisi yang diinginkan(defects/ error/ bugs) dan mengevaluasi fitur – fitur dari entitas software.

Berikut ini adalah pengertian testing yang dihubungkan dengan proses verifikasi dan validasi *software*: *testing software* adalah proses mengoperasikan *software* dalam suatu kondisi yang dikendalikan, untuk :

1. Verifikasi apakah telah berlaku sebagaimana telah ditetapkan (menurut spesifikasi). Verifikasi adalah pengecekan atau pengetesan entitas-entitas, termasuk *software*, untuk pemenuhan dan konsistensi



dengan melakukan evaluasi hasil terhadap kebutuhan yang telah ditetapkan. (*Are we building the system right ?*)

2. Mendeteksi *error*. Deteksi *error*: Testing seharusnya berorientasi untuk membuat kesalahan secara intensif untuk menentukan apakah suatu hal tersebut terjadi bilamana tidak seharusnya terjadi atau suatu hal tersebut tidak terjadi dimana seharusnya mereka ada.
3. Validasi apakah spesifikasi yang telah ditetapkan sudah memenuhi keinginan atau kebutuhan dari pengguna yang sebenarnya.

*Test case* merupakan suatu tes yang dilakukan berdasarkan pada suatu inisialisasi, masukan, kondisi ataupun hasil yang telah ditentukan sebelumnya.

Adapun kegunaan dari *test case* :

1. Untuk melakukan testing kesesuaian suatu komponen terhadap spesifikasi – *Black Box Testing*.
2. Untuk melakukan testing kesesuaian suatu komponen terhadap desain – *White Box Testing*.

Testing tidak dapat membuktikan kebenaran semua kemungkinan eksekusi dari suatu program. Namun dapat didekati dengan melakukan perencanaan dan desain *test case* yang baik sehingga dapat memberikan jaminan efektifitas dari *software* sampai pada tingkat tertentu yang diharapkan.

*White Box Testing* adalah suatu metode desain *test case* yang menggunakan struktur kendali dari desain prosedural. Metode desain *test case* ini dapat menjamin :

1. Semua jalur (*path*) yang independen / terpisah dapat dites setidaknya sekali test.

2. Semua logika keputusan dapat dites dengan jalur yang salah dan atau jalur yang benar.
3. Semua *loop* dapat dites terhadap batasannya dan ikatan operasionalnya.
4. Semua struktur internal data dapat dites untuk memastikan validasinya.

*Black Box Testing* dilakukan tanpa pengetahuan detail struktur internal dari sistem atau komponen yang dites. *Black Box Testing* berfokus pada kebutuhan fungsional pada *software*, berdasarkan pada spesifikasi kebutuhan dari *software*. Kategori *error* yang akan diketahui melalui tes ini :

1. Fungsi yang hilang atau tidak benar.
2. *Error* antar muka.
3. *Error* dari struktur data atau akses eksternal database.
4. *Error* dari kinerja atau tingkah laku.
5. *Error* dari inisialisasi dan terminasi.

Tipe – tipe testing pada aplikasi berbasis web, antara lain:

1. Content dan functionality testing. Testing terhadap isi dan fitur seperti yang terdapat pada website umumnya, pastikan sudah lengkap dan berjalan sesuai seperti yang diinginkan.
2. Feature interaction testing. Banyak pengguna secara simultan mengakses satu site yang sama dan tidak boleh terjadi interfensi antara mereka.
3. Usability testing. Melakukan testing apakah wwebsite tersebut sudah user friendly.

4. Database testing. Memastikan database dapat diakses dari website yang mempunyai integritas dan kecakupan data.
5. Security dan control testing. Memastikan site ini aman, termasuk account setup, billing, dan dari unauthorized acces.
6. Connectivity testing. Pastikan website dapat melakukan connection dan disconnection.
7. Interoperability testing. Pastikan semua web browser dari semua versi dan jenis komputer yang berbeda dapat berjalan baik dengan aplikasi ini.
8. Performasnce dan stress testing. Ukur kemampuan, respontime dan semua proses yang terjadi dalam keadaan workloads diatas rata- rata atau dibawah rata –rata.
9. Cross platform dan configuration testing. Pastikan perilaku dari sistem kompatibel dalam platform dan konfigurasi yang berbeda.
10. Internazionalization testing. Pastikan site tidak membingungkan atau menyerang sistem.
11. Beta testing. Undang beberapa pengguna terpilih untk melakukan eksperimen pada site anda dan mintalah feedback pada mereka sebelum site tersebut diluncurkan.
12. Standard compliance testing. Pastikan website itu kompatibel dengan internet standards, apakah terlihat sama meskipun menggunakan browser atau search engine.

Daftar cek testing aplikasi berbasis Web, sebagai berikut:

1. Fungsionalitas

1.1 Apakah secara umum kegunaan dari *Web site* telah jelas?  
Apakah semua telah terpenuhi?

1.2 Apakah *Web site* telah memiliki fungsi yang sesuai dengan obyektifitas dan spesifikasi yang dibutuhkan?

1.3 Apakah setiap fungsi dapat berjalan sesuai dengan yang diinginkan dalam semua spesifikasi? ( jika ada pertanyaan spesifikasi yang mana? Maka hal itu bagus anda dapat menggambarkan di spesifikasi apa saja aplikasi ini harus berjalan)

## 2 Kompatibilitas dan interoperabilitas

2.1 Apakah *site* ini sudah terlihat menarik terlihat dari berbagai *platform browser* dan sistem operasi yang berbeda?

2.2 Apakah *site* ini sudah menggunakan *text based service browser* yang memuaskan?(beberapa *browser* kadang-kadang tidak didukung GUI hanya *text based*)

## 3 Hyperlink testing

3.1 Kesalahan yang sering terjadi pada *Web site* adalah *missing links*, salah *link* atau *link out of date*. *Update* pada *Web site* juga sering mengakibatkan kesalahan itu terjadi. Bagaimanapun testing terhadap *link* itu sangat diperlukan untuk memastikan *Web site* itu berjalan sebagaimana mestinya. Tes sederhana pada *link* hanya memastikan bahwa setiap *link* yang berhubungan dengan *page* itu berjalan sebagaimana diharapkan. Testing terhadap *link* eksternal harus dilakukan

secara periodik meskipun mungkin tidak terjadi perubahan pada *Web site*.

