

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Sistem**

Menurut Jogiyanto (2005:2), Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu.

Suatu sistem mempunyai maksud tertentu. Ada yang menyebutkan maksud dari suatu sistem adalah untuk mencapai suatu tujuan (*goal*) dan ada yang menyebutkan untuk mencapai suatu sasaran (*objectives*).

#### **2.2 Informasi**

Menurut Jogiyanto (2005:8), Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Sumber dari informasi adalah data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian (*event*) adalah sesuatu yang terjadi pada saat tertentu.

#### **2.3 Sistem Informasi**

Menurut Herlambang (2005:121), Data adalah fakta-fakta atau kejadian-kejadian yang dapat berupa angka-angka atau kode-kode tertentu. Data masih belum mempunyai arti bagi penggunanya. Untuk dapat mempunyai arti data diolah sedemikian rupa sehingga dapat digunakan oleh penggunanya. Hasil pengolahan data inilah yang disebut sebagai informasi. Secara ringkas, informasi adalah data yang telah diolah dan mempunyai arti bagi penggunanya. Sehingga

sistem informasi dapat didefinisikan sebagai prosedur-prosedur yang digunakan untuk mengolah data sehingga dapat digunakan oleh penggunanya.

## 2.4 Manajemen Kinerja

Menurut Wibowo (2007:7), Manajemen kinerja adalah manajemen tentang menciptakan hubungan dan memastikan komunikasi yang efektif. Manajemen kinerja memfokuskan pada apa yang di perlukan oleh organisasi, manajer, dan pekerja untuk berhasil.

Kinerja berasal dari pengertian *performance* yaitu hasil kerja atau prestasi kerja. Namun, sebenarnya kinerja mempunyai makna yang lebih luas, bukan hanya hasil kerja, tetapi termasuk bagaimana proses pekerjaan berlangsung. Kinerja merupakan hasil pekerjaan yang mempunyai hubungan kuat dengan tujuan strategis organisasi, kepuasan konsumen, dan memberikan kontribusi pada ekonomi. Dengan demikian, kinerja adalah tentang melakukan pekerjaan dan hasil yang dicapai dari pekerjaan tersebut.

Manajemen mencerminkan sikap para eksekutif yang menginginkan agar komputer tersedia untuk semua pemecah masalah perusahaan. Ketika Sistem Informasi Manajemen berada pada tempatnya dan berfungsi seperti yang diinginkan, maka dapat membantu manajer dan pemakai lain di dalam dan di luar perusahaan untuk mengidentifikasi dan memahami masalah.

Pada divisi manajer *Business Performance* terdapat 4 (empat) bagian kerja yaitu :

### 1. *Asman Quality Management*

Pengelolaan usulan perubahan dokumen, pelaksanaan AMI, penyelesaian temuan AMI, mendata hasil dan penyelesaian temuan, *monitoring* implementasi proses bisnis.

### 2. *Fraud Management*

Melaksanakan kebijakan *fraud* dan rekomendasi temuan *fraud*, data *support* deteksi *fraud*, dan menindaklanjuti temuan *fraud* lokal.

### 3. *Performance*

Mendukung penyusunan parameter performansi, melaksanakan pengukuran kinerja, *deployment* target unit, usulan RKAP, dan mendukung analisa lingkungan bisnis eksternal dan internal.

### 4. *Partnership*

*Support and monitoring* kerjasama dan potensi kemitraan

## 2.5 Analisa dan Desain Sistem Informasi

Menurut Jogiyanto (2005:129), Analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya.

Tahap analisis sistem dilakukan setelah tahap perencanaan sistem dan sebelum tahap desain sistem. Tahap analisis merupakan tahap yang kritis dan sangat penting, karena kesalahan di dalam tahap ini akan menyebabkan juga kesalahan di tahap selanjutnya.

Menurut Jogiyanto (2005:195), Setelah tahap analisis sistem selesai dilakukan, maka analisis sistem telah mendapatkan gambaran dengan jelas apa yang harus dikerjakan. Tiba waktunya sekarang bagi analisis sistem untuk memikirkan bagaimana membentuk sistem tersebut. Tahap ini disebut dengan desain sistem. Desain sistem dapat dibagi dalam dua bagian, yaitu desain secara umum dan desain secara terinci. Desain secara umum disebut juga dengan desain konseptual atau desain logikal atau desain secara makro. Desain sistem terinci disebut juga dengan desain sistem secara fisik atau desain internal.

## 2.6 World Wide Web (WWW)

Menurut Sidik dan Pohan (2007:1), *World Wide Web* (WWW), lebih dikenal dengan *web*, merupakan salah satu layanan yang didapat oleh pemakai komputer yang terhubung ke internet.

*Web* pada awalnya adalah ruang informasi dalam internet, dengan menggunakan teknologi *hypertext*, yang digunakan untuk menemukan informasi dengan mengikuti link yang disediakan dalam dokumen *web* yang ditampilkan dalam *browser*.

## 2.7 Monitoring

Menurut Casley dan Kumar (1989:76), *Monitoring* merupakan pengidentifikasian kesuksesan atau kegagalan secara nyata maupun potensial sedini mungkin dan sewaktu-waktu bisa menyelesaikan operasinya dengan tujuan meninjau kemajuan dan mengusulkan langkah supaya dijalankan untuk merai dan mewujudkan tujuan. *Monitoring* dapat juga diartikan sebagai penilaian yang terus menerus terhadap fungsi kegiatan-kegiatan proyek di dalam konteks

jadwal-jadwal pelaksanaan terhadap penggunaan input-input proyek oleh kelompok sasaran di dalam konteks harapan-harapan rancangan praktek manajemen yang baik dikarenakan merupakan bagian integral dari manajemen sehari-hari

Aplikasi adalah penggunaan atau penerapan suatu konsep yang menjadi pokok pembahasan. Aplikasi dapat diartikan sebagai program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu.

Sehingga aplikasi monitoring dapat diartikan sebagai program komputer yang dibuat untuk melakukan penilaian yang terus menerus terhadap fungsi kegiatan-kegiatan proyek.

## 2.8 Key Performance Indicator

Menurut Haryanti (2008:11), *Key Performance indicator* adalah indikator yang merepresentasikan kinerja dari proses yang dilaksanakan. *Key Performance Indicator* merupakan sekumpulan ukuran mengenai aspek kinerja yang paling kritis, yang menentukan kesuksesan organisasi pada masa sekarang dan masa yang akan datang. *Key Performance Indicator* digunakan memprediksi peluang kesuksesan atau kegagalan dari proses-proses yang dilaksanakan organisasi, sehingga KPI dapat digunakan sebagai alat untuk meningkatkan kinerja organisasi secara dramatis. Contoh dari penjelasan diatas adalah penentuan parameter nilai dalam *trend* mitra aktif. membuat *user* dapat dengan mudah mengetahui kondisi mitra yang masi menjalin kemitraan.

## 2.9 Media Penyajian Data

Media penyajian data bergantung pada jenis informasi dan pesan yang

ingin disampaikan, serta kebutuhan dan selera pengguna. Perancang dapat memilih bentuk teks, grafik, atau mengkombinasikan keduanya. Pemilihan media penyajian yang kurang tepat akan mengurangi efektifitas penyampaian informasi, dan menimbulkan persepsi yang salah mengenai informasi yang disajikan. Ada dua prinsip dalam memilih media penyajian informasi oleh Few (Hariyanti, 2008:15) yaitu:

- a. Media dapat memberikan persepsi yang benar tentang informasi yang disajikan.
- b. Media dapat menyajikan informasi sesuai dengan tujuan, tanpa memakan banyak tempat.

Setiap jenis media memiliki kekuatan dan kelemahan tersendiri dalam menyajikan suatu jenis informasi. Media yang paling banyak digunakan dalam menyajikan informasi. Tabel dapat menyajikan data dengan lebih detail, menyajikan angka-angka dengan format yang lebih presisi. Tabel dapat menampilkan data kuantitatif maupun kualitatif. Namun demikian, informasi yang disajikan dalam tabel tidak dapat dipahami secara cepat dan sekilas.

## **2.10 Grafik**

Ada beberapa tipe grafik atau diagram yang dapat digunakan untuk menampilkan gambaran informasi supaya lebih jelas, antara lain:

### **5. Diagram garis**

Diagram garis digunakan untuk menunjukkan perubahan nilai dari sederetan data relatif terhadap waktu, karena diagram garis biasanya digunakan untuk menunjukkan suatu kecenderungan atau *trend*.

### **6. Diagram batang**

Diagram batang digunakan untuk menyajikan nilai relatif terhadap data yang lain. Misal, eksekutif ingin melihat grafik pendaftar per-tahun dan per-gelombang.

#### 7. Diagram roti (*pie*)

Diagram *pie* biasanya digunakan untuk menggambarkan besarnya prosentase data. Misalkan menggambarkan besarnya prosentase alasan mahasiswa keluar. (Santosa, 1994)

#### 8. Diagram piramida (*pyramid*)

Diagram *pyramid* biasanya digunakan untuk menggambarkan besarnya prosentase data dan menampilkan jumlah data yang tertinggi dan secara bertahap turun kebawah mewakili data terendah. (Nadhani, 2011) pada <http://www.fusioncharts.com/demos/gallery/#pyramid-chart>.

Grafik dapat digunakan untuk menunjukkan keterhubungan antar data, seperti perbandingan nominal, *time-series*, deviasi, korelasi, dan sebagainya. Ada berbagai macam bentuk grafik yang dapat dipilih untuk menggambarkan setiap jenis keterhubungan data, seperti yang terdapat pada tabel 2.1. Namun demikian, grafik kurang bisa menampilkan angka dengan format yang presisi.

Tabel 2.1 Keterhubungan Data dan Jenis Grafik yang Sesuai  
(Sumber: Hariyanti, 2008:16)

No	Keterhubungan Data	Jenis Grafik yang Sesuai
1	Perbandingan nominal	Grafik bar (horisontal atau vertikal) Grafik titik (jika 0 tidak termasuk dalam skala nilai)
2	<i>Time-series</i>	Grafik garis (untuk melihat tren seluruh data) Grafik bar (untuk melihat perbandingan antar nilai)

		Grafik titik yang dihubungkan dengan garis (untuk melihat nilai sekaligus tren data secara keseluruhan)
3	Bagian dari keseluruhan	Grafik bar(horisontal maupun vertikal) Grafik <i>stack bar</i> <i>Pie chart</i>
4	Deviasi	Grafik garis Grafik titik yang dihubungkan dengan garis
5	Ranking	Grafik bar(harisontal atau vertikal) Grafik titik(jika 0 tidak termasuk dalam skala nilai)
6	Distribusi frekuensi	Grafik bar vertikal/ histogram Grafik garis / polygon frekuensi(untuk menunjukkan tren data secara keseluruhan)
7	Korelasi	Grafik titik dan garis ( <i>scatter-plot</i> )

## 2.11 Dashboard

*Dashboard* adalah sebuah tampilan *visual* dari informasi terpenting yang dibutuhkan untuk mencapai satu atau lebih tujuan, digabungkan dan diatur pada sebuah layar, menjadi informasi yang dibutuhkan dan dapat dilihat secara sekilas. Tampilan *visual* disini mengandung pengertian bahwa penyajian informasi harus dirancang sebaik mungkin, sehingga mata manusia dapat menangkap informasi secara cepat dan otak manusia dapat memahami maknanya secara benar. *Dashboard* itu sebuah tampilan pada satu monitor komputer penuh, yang berisi informasi yang bersifat kritis, agar kita dapat melihatnya dengan segera, sehingga dengan melihat *dashboard* itu saja, kita dapat mengetahui hal-hal yang perlu



diketahui. Biasanya kombinasi dari teks dan grafik, tetapi lebih ditekankan pada grafik. (Few, 2006:34). Menurut Eckerson dan Hariyanti (2008:7), Tujuan dalam penggunaan *dashboard* yaitu:

1. Mengkomunikasikan Strategi

Mengkomunikasikan strategi dan tujuan yang dibuat oleh eksekutif, kepada semua pihak yang berkepentingan, sesuai dengan peran dan levelnya dalam organisasi.

2. Memonitor dan Menyesuaikan Pelaksanaan Strategi

Memonitor pelaksanaan dari rencana dan strategi yang telah dibuat. Memungkinkan eksekutif untuk mengidentifikasi permasalahan kritis dan membuat strategi untuk mengatasinya.

3. Menyampaikan Wawasan dan Informasi ke semua pihak

Menyajikan informasi menggunakan grafik, simbol, bagan dan warna yang memudahkan pengguna dalam memahami dan mempersepsi informasi secara benar.

## 2.12 Interaksi Manusia dan Komputer

Menurut Rizky (2007:3), Interaksi Manusia dan Komputer (IMK) dideskripsikan sebagai sebuah disiplin ilmu yang mempelajari desain, evaluasi, implementasi dan sistem komputer interaktif untuk dipakai oleh manusia, beserta studi tentang faktor- faktor utama dalam lingkungan interaksinya. Deskripsi IMK menurut Galitz (2002) dalam Rizky (2007:3) adalah suatu ilmu yang mempelajari perencanaan dan desain tentang cara manusia dan komputer saling bekerjasama sehingga manusia merasa puas dengan cara yang paling efektif.

Menurut Rizky (2007:6), Komponen-komponen penting dalam IMK yaitu interaksi, manusia, dan komputer. Interaksi adalah komunikasi yang terjadi antara manusia dan komputer. Jenis-jenis komunikasi tersebut antara lain *command entry, menus and navigation, forms and spreadsheets, question and answer dialogue, natural language dialogue, windows icon menu pointer*, dan *direct manipulation*. Komponen selanjutnya yaitu manusia yang dalam hal ini adalah pengguna yang dapat berupa seorang atau sekelompok pengguna yang bekerja dalam sebuah tim atau organisasi dan saling berkaitan dalam mengerjakan tugas tertentu. Manusia dalam konteks IMK yang juga harus diperhatikan adalah komputer. Komputer diartikan sebagai perangkat keras ataupun perangkat lunak dari berbagai macam jenis yang nantinya berinteraksi dengan unsur manusia.

Galitz (2002) dalam Rizky (2007:26) menjelaskan bahwa sebelum memulai sebuah proses desain *interface*, terdapat beberapa tip desain yang harus diperhatikan, antara lain:

1. Memenuhi kaidah estetika.

Sebuah desain dapat disebut baik secara estetika jika (1) di dalamnya terdapat perbedaan yang jelas dan kontras antar elemen dalam sebuah tampilan. Misalnya tampilan tombol yang berbeda warna dengan tampilan *textbox*, (2) terdiri dari beberapa kelompok yang jelas antara *input* dan tombol proses, (3) antar elemen dan kelompok tampilan dipisah dengan alignment yang rapi, (4) sederhana dan tidak terlalu banyak aksesoris yang terkesan sia-sia.

2. Dapat dimengerti.

Sebuah desain harus dapat dimengerti dengan cepat dari segi tampilan secara visual, fungsi yang akan ditonjolkan, penggunaan kata-kata yang singkat dan

jelas baik dalam tampilan maupun dalam perintah. Penggunaan metafora atau pemisalan yang berlebihan dalam sebuah fungsi harus dihindari.

3. Kompatibilitas.

Sebuah desain *interface* harus dapat memenuhi kompatibilitas dari berbagai segi antara lain (1) kompatibilitas pengguna yaitu dapat digunakan oleh pengguna dari kalangan yang lebih luas, baik berdasarkan strata pendidikan maupun berdasarkan usia, (2) kompatibilitas penggunaan yaitu dapat memenuhi fungsi dan tujuan yang ingin dicapai dari perancangan sebuah perangkat lunak dan perangkat keras yang digunakan, (3) kompatibilitas produk yaitu agar perangkat lunak dapat berjalan dengan baik di berbagai perangkat keras yang ada dan sistem operasi yang menjadi target aplikasi.

4. Komprehensif.

Sebuah sistem yang baik akan membimbing penggunanya agar dapat dan lebih mudah memahami apa yang harus diperhatikan, bagaimana cara melakukan sesuatu, kapan dan di mana melakukan sesuatu, dan mengapa harus melakukan sesuatu.

5. Konfigurabilitas.

Sebuah sistem harus dapat dikonfigurasi ulang jika penggunanya menginginkan sesuatu berdasarkan fungsi tertentu.

6. Konsistensi.

Memiliki konsistensi dalam penempatan dan pemilihan gaya komponen visual misalnya tombol atau icon yang seragam.

7. Kontrol pengguna.

Pengguna dapat melakukan kontrol jika suatu saat terjadi kesalahan dalam proses serta pemilihan fungsi tambahan dari sebuah sistem. Hindari desain yang nantinya akan membatasi pengguna dalam memilih tampilan tertentu.

8. Efisien.

Desain dibuat seefisien mungkin, terutama dalam penempatan komponen, misalnya penempatan tombol dalam sebuah panel yang dapat menarik perhatian pengguna.

9. Mudah dikenali.

Gunakan antar muka yang sudah dikenal oleh penggunanya, misalnya penempatan icon *cut*, *copy*, *paste* secara standar dalam *toolbar*.

10. Toleransi.

Tidak ada sebuah sistem yang sempurna, karenanya terdapat beberapa toleransi kesalahan yang mungkin terjadi. Usahakan agar terjadi sebuah pesan yang dapat membimbing pengguna untuk keluar dari kesalahan yang terjadi.

11. Sederhana.

Lima cara untuk membuat desain sederhana dan tetap sesuai dengan keinginan pengguna, yaitu (1) sembunyikan komponen visual jika tidak diperlukan, (2) sediakan pilihan standar, (3) minimalkan penggunaan berbagai macam *alignment*, (4) usahakan agar fungsi yang sering digunakan terlihat, (5) perhatikan konsep konsistensi.

### 2.13 Unified Modeling Language (UML)

UML adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi obyek.

UML merupakan standar yang relatif terbuka yang dikontrol oleh *Object Management Group* (OMG), sebuah konsorsium terbuka yang terdiri dari banyak perusahaan. OMG dibentuk untuk membuat standar-standar yang mendukung interoperabilitas, khususnya interoperabilitas sistem berorientasi obyek. OMG mungkin lebih dikenal dengan standar-standar *Common Object Request Broker Architecture* (CORBA).

UML lahir dari penggabungan banyak bahasa pemodelan grafis berorientasi obyek yang berkembang pesat pada akhir 1980-an dan awal 1990. Sejak kehadirannya pada 1997, UML menghancurkan menara Babel tersebut dan menjadi sejarah (Fowler, 2004).

Tujuan UML diantaranya adalah :

1. Memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.
2. Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
3. Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan.

Menurut Sholiq (2010:52), Pemodelan bisnis adalah langkah pertama dalam proses pengembangan sistem informasi. Urutan langkah-langkah dalam permodelan bisnis adalah sebagai berikut:

1. Pemodelan bisnis.
2. Permodelan *use case system*.
3. Analisis.
4. Desain.

5. Pengkodean.
6. Pengujian.
7. Implementasi dan distribusi.

Diagram-diagram tersebut diberi nama berdasarkan sudut pandang yang berbeda-beda terhadap sistem dalam proses analisis atau rekayasa. Dibuatnya berbagai jenis diagram di atas karena:

- a. Setiap sistem yang kompleks lebih baik jika dilakukan pendekatan melalui himpunan berbagai sudut pandang yang kecil yang satu sama lain hampir saling bebas (*independent*). Sudut pandang tunggal senantiasa tidak mencukupi untuk melihat isi sistem yang lebih besar dan kompleks.
- b. Diagram yang berbeda-beda tersebut dapat menyatakan tingkatan yang berbeda-beda dalam proses rekayasa.
- c. Tujuan adanya diagram-diagram tersebut adalah agar model yang dibuat semakin mendekati realitas.

Diagram-diagram ini ditambah dengan kemampuan dokumentasi sebagai *artifacts* utama UML. *Data-flow diagram* dan tipe *diagram* lain yang tidak terdapat dalam UML tidak termasuk dalam paradigma *object-oriented*. *Activity diagram* dan *collaboration diagram* yang terdapat dalam UML menggantikan *data-flow diagram*. *Activity diagram* juga sangat bermanfaat untuk membuat *workflow*.

## 2.14 Testing dan Implementasi

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*.

Menurut Romeo (2003:3), *Testing software* adalah proses mengoperasikan software dalam suatu kondisi yang dikendalikan untuk:

1. Verifikasi.

Melakukan pengecekan atau pengetesan entitas – entitas, apakah telah berlaku sebagaimana telah ditetapkan

2. Mendeteksi error.

Untuk menentukan apakah sesuatu hal terjadi bilamana tidak seharusnya terjadi atau suatu hal tersebut terjadi dimana seharusnya mereka ada.

3. Validasi.

Apakah spesifikasi yang ditetapkan telah memenuhi keinginan atau kebutuhan pengguna yang sebenarnya?

Menurut Romeo (2003:33), Dengan adanya perkembangan teknologi internet, berkembanglah kebutuhan aplikasi berbasis *web*, baik untuk keperluan internet organisasi. Terdapat beberapa hal yang berkaitan dengan kualitas aplikasi berbasis *web*, antara lain:

1. Komplexitas Aplikasi

*Web* merupakan aplikasi yang paling berkembang saat ini, baik dari segi kompleksitas, manajemen *query* pada *database* yang sangat besar, atau metode *searching* yang ada. *Web* site lebih kompleks dari yang terlihat, karena *web* site menggunakan teknologi GUI, *Network Connectivity* dan *Database Acces*. Beberapa pengamat menyatakan bahwa teknologi *client/server* akan digantikan oleh *internet*, tapi kenyataan yang berkembang adalah teknologi

gabungan dari keduanya. Inilah alasan mengapa *client/server testing* yang dibahas sebelumnya juga berkaitan dengan subbab ini.

## 2. Keterbatasan Alat Bantu.

Hal yang tidak dapat dibantah adalah alat bantu pengembangan aplikasi berbasis *web* saat ini masih memiliki keterbatasan yang sangat mengganggu. Aplikasi *web* dibangun dengan alat bantu standar yang menghasilkan pages statis, sehingga pengguna tidak dapat dengan mudah *men-download* data ke *desktop analysis tool* seperti *excel spreadsheet*.

Produk *web* merupakan aplikasi yang paling cepat mengalami penambahan versi oleh karena itu manajemen tes yang diperlukan juga harus handal, karena hal ini berhubungan dengan kualitas dari aplikasi itu sendiri.

## 3. Kompatibilitas

*Web pages* akan terlihat berbeda jika dilihat dari *Web Browser* yang berbeda, karena perbedaan implementasi dari *HTML standart*. *Web pages* dapat diakses dari beberapa *platform* yang berbeda, seperti Win NT, Win 95, OS/2, Mac dan lain-lain. Ini artinya *testing* perlu dilakukan pada berbagai *platform* dan konfigurasi yang berbeda.

## 4. Performansi

Hal yang paling sulit untuk dites adalah pengukuran kecepatan akses. *Response Time* dari *Web*, karena hal itu bukan hal yang mudah untuk dipecahkan dengan biaya yang murah. Banyak faktor yang menjadi penyebab seperti *loads* yang tidak dapat diprediksi, *Web* yang menjadi favorit bisa menerima ribuan pengunjung per-hari dibandingkan dengan *Web* biasa yang pengunjungnya hanya ratusan.



## 5. Kegunaan

Beberapa pengguna mungkin punya harapan sendiri-sendiri tentang bagaimana *web site* yang menarik. Seperti contohnya *Web Pages* harus dapat dengan mudah untuk disimpan. Oleh karena itu *Web pages* harus terlihat atraktif agar menarik perhatian dari pengguna. Ada beberapa pengguna yang sangat sensitif dan terganggu jika keluar atau masuk dari suatu *Web pages* tanpa suatu *permission* atau *awareness*.

## 6. Keamanan

Sistem keamanan merupakan hal yang sangat penting dalam aplikasi berbasis web, karena aplikasi ini dibangun untuk dapat diakses oleh pengguna atau aplikasi yang baik itu dalam suatu intranet ataupun extranet dengan sama baiknya. Hak akses eksternal memang dibatasi tapi tidak menutup kemungkinan terjadinya *hacking* terhadap aplikasi.

## 7. Organisasional

Telah dijelaskan diatas bahwa teknologi ini merupakan inovasi yang sangat fenomenal. Oleh karena itu mungkin dalam perkembangannya yang kurang diperhatikan adalah kendali kualitas dan standar *testing* yang baik. Yang terjadi pada pengembangan intranet yang mengambil alih semua proses pembangunan dari suatu aplikasi *web* mulai dari desain hingga proses *testing*.

Dalam beberapa organisasi intranet mambuat kekacuan karena kurangnya koordinasi. Setiap orang mempunyai *web* internal pribadi. Setiap orang punya ide sendiri-sendiri bagaimana membuat harus membuat *web*, apa isinya, dan bagaimana harus berjalan. Sehingga terjadi kekacauan pada kepemilikan dan

hak akses informasi juga pertanyaan siapa yang bertanggung jawab atas kualitas dari informasi dan maintenance dari aplikasi itu sendiri.

Tipe-tipe *testing* pada aplikasi berbasis *web*, antara lain :

1. *Content* dan *Funcionality testing*. *Testing* terhadap isi dan fitur seperti yang terdapat pada *Web site* umumnya, pastikan sudah lengkap dan berjalan sesuai dengan yang diinginkan.
2. *Feature interaction testing*. Banyak pengguna yang secara simultan mengakses satu site yang sama dan tidak boleh terjadi *interfrensi* antara mereka.
3. *Usability testing*. Melakukan *testing* apakah *Web site* sudah *user friendly*.
4. *Database testing*. Memastikan *database* dapat diakses dari *Web site* yang mempunyai kendali integritas dan kecukupan data.
5. *Security* dan *control testing*. Memastikan *site* ini aman, termasuk *account setup, billing, dan dari unauthorized acces*.
6. *Connectivity testing*. Pastikan *Web site* dapat melakukan *connection* atau *disconnection*.
7. *Interoperability testing*. Pastikan semua *Web Browser* dari semua versi dan jenis komputer yang berbeda dapat berjalan dengan baik pada aplikasi ini.
8. *Cross platform* dan *configuration testing*. Pastikan perilaku dari sistem kompatibel dalam *platform* dan konfigurasi yang berbeda.
9. *Performance* dan *Stress testing*. Ukur kemampuan, *response time* dan semua proses yang terjadi dalam keadaan *workloads* di atas rata-rata, rata-rata atau dibawah rata-rata.

10. *Internazionalization testing*. Pastikan *site* tidak membingungkan atau menyerang pengguna.
11. *Beta testing*. Undang beberapa pengguna terpilih untuk melakukan eksperimen pada *site* anda dan mintalah *feedback* pada mereka sebelum *web site* itu diluncurkan.
12. *Standart Compilance testing*. Pastikan *Web site* itu kompetibel dengan *internet standart*, apakah terlihat sama meskipun menggunakan *browser* atau *search engines* yang berbeda.

