

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Tinjauan Pustaka**

(Setiyawan, Purnama, & Sukadi, 2012) dalam penelitiannya yang berjudul pembuatan sistem informasi akademik berbasis web pada Sekolah Menengah Atas Negeri 1 Ngadirojo. Sistem pendataan siswa yang sekarang dirasakan masih banyak kekurangan yang terjadi sehingga dibutuhkan pembuatan sistem informasi pengolahan data siswa dan akademik berbasis web pada SMAN 1 Ngadirojo agar menjadi sarana pengolahan data akademik yang efektif dan efisien. Alat bantu pengembangannya menggunakan *flowchart* dan *diagram konteks*, dan metode pengembangan sistem yang digunakan adalah *waterfall*.

(Marlena & Sasongko, 2010) dalam penelitiannya yang berjudul pembuatan *website* profil Sekolah Menengah Pertama (SMP) Negeri 2 Kartasura. Selama ini di sekolah SMP Negeri 2 Kartasura penyampaian informasi yang diberikan kepada masyarakat hanya secara sederhana yaitu melalui papan pengumuman. Dengan penyampaian informasi yang demikian dirasakan masih banyak memiliki kekurangan, sehingga dilakukan penelitian ini untuk memudahkan pihak sekolah dalam menyebarkan informasi serta memberikan kemudahan kepada masyarakat dalam mengakses informasi khususnya mengenai profil sekolah. Aplikasi dibangun berbasis *web* dengan menggunakan bahasa pemrograman PHP, dan *MySQL* sebagai *databasenya*. Metode pengembangan sistem yang digunakan adalah *waterfall*.

(Winoto & Tj, 2012) dalam penelitiannya berjudul pembuatan *website* profil Sekolah Dasar Negeri 03 Kalisoro. SDN 03 Kalisoro adalah salah satu instansi pemerintah yang belum memiliki fasilitas *website* untuk memberikan informasi kepada warga sekolah maupun kepada masyarakat luas. Oleh karena itu dilakukan penelitian ini agar instansi sekolah dapat mempromosikan dan memberikan informasi tentang profil sekolah, fasilitas sekolah secara *online* melalui *website* yang dapat diakses setiap saat sehingga lebih efektif dan efisien. Aplikasi yang

dibangun berbasis *web* dengan menggunakan *software Incomedia X5, Adobe photosop CS4, & Mozilla Firefox 4.*

(Suhartanto, 2012) dalam penelitiannya yang berjudul pembuatan *website* Sekolah Menengah Pertama Negeri 3 Delanggu dengan menggunakan PHP dan *MySQL*. Selama ini pemberitahuan informasi yang terbaru masih ditempel di papan pengumuman, melalui *speaker* yang dipasang tiap kelas, surat selebaran yang dibagikan kepada siswa dan masih menggunakan media informasi spanduk dan pamflet sebagai media promosi kepada publik. Sehingga dilakukan penelitian ini agar bisa memberikan informasi secara cepat, akurat, dan efisien kepada siswa, guru, dan juga masyarakat umum secara *online* agar keunggulan SMPN 3 Delanggu semakin dikenal oleh masyarakat luas dan dapat membantu pihak sekolah memaksimalkan dalam promosi penerimaan siswa baru. Aplikasi yang dibangun berbasis *web* dengan menggunakan PHP yang ditulis dengan menggunakan bahasa C, dan *MySQL* sebagai *database*, dan metode penelitian yang digunakan adalah rancangan *site map*.

Pada penelitian *website* sekolah SMA Negeri 1 Tebing Tinggi Barat Kabupaten Kepulauan Meranti difokuskan pada penyampaian informasi-informasi terkini dari sekolah. Metode pengembangan sistem yang digunakan adalah *waterfall* dalam *website* ini menggunakan *bootstrap* atau lebih dikenal dengan *responsive*, keunggulan dari *responsive* lebih sedikit mengonsumsi memori dibanding dengan *website* yang belum *responsive*. Tampilan *website responsive* lebih baik dan dapat menyesuaikan lebar layar ketika diakses di berbagai perangkat seperti *PC Dekstop, Tablet PC, dan smartphone*. *Website responsive* memiliki lebih banyak menggunakan CSS dibanding *script* seperti *JavaScript, JQuery*, dan sebagainya. Aplikasi ini dibangun dengan menggunakan bahasa pemrograman PHP, HTML, *JavaScrip*, CSS, dan *MySQL* sebagai *database*-nya

## **2.2 Landasan Teori**

### **2.2.1 Sistem Informasi**

Menurut Fat pengertian sistem adalah sebagai berikut : “Sistem adalah suatu benda nyata atau abstrak (*a set of thing*) yang terdiri dari bagian-bagian atau komponen-komponen yang saling berkaitan, berhubungan, berketergantungan, saling mendukung, yang secara keseluruhan bersatu dalam satu kesatuan *Unity* untuk mencapai tujuan tertentu secara efisien dan efektif”.

Pengertian Sistem menurut Indrajit (2001) mengemukakan bahwa sistem mengandung arti kumpulan-kumpulan dari komponen-komponen yang dimiliki unsur keterkaitan antara satu dengan lainnya.

Pengertian Sistem menurut Murdick, R.G (1991) Suatu sistem adalah seperangkat elemen yang membentuk kumpulan atau prosedur-prosedur atau bagan-bagan pengolahan yang mencari suatu tujuan bagian atau tujuan bersama dengan mengoperasikan data dan atau barang pada waktu rujukan tertentu untuk menghasilkan informasi dan atau energi dan atau barang.

Definisi Sistem menurut Lani Sidharta (1995), “Sistem adalah himpunan dari bagian-bagian yang saling berhubungan yang secara keseluruhan merupakan suatu kesatuan secara fungsional”.

Sistem Informasi adalah kerangka kerja yang mengkoordinir sumber daya (manusia, komputer) untuk mengubah masukan (*input*) menjadi keluaran (*output*) berupa informasi guna mencapai sasaran. Sistem informasi terdiri dari komponen-komponen yang disebut blok bangunan atau *building block* (Jogiyanto, 2005).

Dengan demikian sistem merupakan kumpulan dari beberapa bagian yang memiliki keterkaitan dan saling bekerja sama serta membentuk suatu kesatuan untuk mencapai suatu tujuan dan sasaran dalam ruang lingkup yang sempit.

### **2.2.2 Pengembangan Sistem Berbasis Web**

Pengembangan sistem berbasis web adalah aplikasi yang sejak awal dirancang untuk dieksekusi di lingkungan berbasis *web*. Definisi ini mengungkapkan dua aspek penting dari aplikasi ini (Simarmata, 2009) sebagai berikut:

1. Suatu aplikasi *web* dirancang dapat berjalan di dalam lingkungan berbasis *web*. Artinya aspek-aspek hipermedia dalam kaitannya dengan *hypertexts* dan multimedia di dalam kombinasi dengan kelola aplikasi tradisional harus diperhitungkan di seluruh hidup aplikasi.
2. Aplikasi *web* adalah suatu aplikasi yang tidak hanya berupa sekumpulan halaman-halaman *web*.

### **2.2.3 Javascript**

*Javascript* adalah kode untuk menyusun halaman web yang memungkinkan dijalankan pada sisi klien. *Javascript* adalah bahasa yang digunakan agar dokumen HTML yang ditampilkan dalam *browser* menjadi lebih interaktif. *Javascript* merupakan bahasa *script*, bahasa yang tidak memerlukan kompilier untuk menjalankannya, cukup dengan interpreter. Tidak perlu proses kompilasi terlebih dahulu agar program dapat dijalankan. Oleh karena itu, *JavaScript* dapat dijalankan didalam aplikasi *browser* yang merupakan sebuah aplikasi interpreter, asalkan *browser* tersebut telah mendukung penggunaan *JavaScript*.

### **2.2.4 CSS (*Cascading Style Sheet*)**

Menurut Bunafit Nugroho (2013:1), *Cascading Style Sheet* adalah bahasa *stye sheet* yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa *markup*. CSS bekerja sebagai pelengkap pada elemen HTML yang kesemuanya itu dapat dikendalikan dengan menggunakan dengan menggunakan sebuah bahasa *script* CSS. Penggunaan CSS dilakukan untuk memperluas kemampuan HTML dalam memformat dokumen *web* atau untuk memperindah tampilan *web*. Penulisan kode CSS disisipkan pada *tak* HTML. Kode CSS ditulis dengan *tak* `<style>` dan `</style>` dengan mendefinisikan suatu *style* baru yang kemudian dapat digunakan berulang kali.

### 2.2.5 XAMPP

Menurut Wicaksono (2008:7) menjelaskan bahwa “XAMPP adalah sebuah *software* yang berfungsi untuk menjalankan *website* berbasis PHP dan Menggunakan pengolah data *MySql* di komputer lokal”. XAMPP berperan sebagai *server web* pada komputer lokal. XAMPP juga dapat disebut sebuah *Cpanel server virtual*, yang dapat membantu melakukan *preview* sehingga dapat dimodifikasi *website* tanpa harus *online*.

*Software* XAMPP dapat diperoleh secara gratis dari situs [www.apachefriends.org](http://www.apachefriends.org) atau dapat juga dicari di *google* untuk beberapa versi tertentu. XAMPP adalah perangkat lunak yang bebas, yang mendukung banyak sistem operasi dan merupakan komplikasi dari beberapa program. Fungsinya adalah sebagai *server* yang berdiri sendiri yang terdiri atas program *Apache HTTP server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan pemrograman PHP dan *Perl*.

### 2.2.6 MySQL

Menurut Aditya (2011c:61) “*MySql* adalah sebuah perangkat lunak sistem manajemen basis data SQL(bahasa inggris: *database management system*) atau DBMS uang *multithread*, *multiuser*, dengan sekitar 6 juta instalasi di seluruh dunia”. *MySql* AB membuat *MySql* tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaanya tidak cocok dengan penggunaan GPL. Tidak seperti *Apache* yang merupakan *software* yang dikembangkan komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing. *MySql* dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia yaitu ***MySql AB***. *MySql* AB memegang penuh hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan sari orang Finlandia yang mendirikan *MySql* AB adalah: David Axmark, Allan Larsson, dan Michael “**Monty**” Widenus.

## 2.2.7 Bootstrap

*Bootstrap* merupakan *framework* gratis *front-end* untuk pengembangan *web* lebih cepat dan lebih mudah, *bootstrap* termasuk HTML dan CSS berdasarkan desain *template* untuk tipografi, bentuk, tombol, tabel, navigasi, kata modal, komidi putar gambar, dan banyak *plugin JavaScript* opsional lainnya, serta *bootstrap* juga memberikan Anda kemampuan untuk dengan mudah membuat desain responsif.

## 2.2.8 Alat Bantu Perancangan Sistem

Alat bantu yang digunakan untuk mendukung keberhasilan dalam tahapan yang dilalui agar pemakaian alat-alat tersebut sesuai dengan kegunaannya maka diperlukan teori dari alat bantu perancangan sistem yang nantinya digunakan adalah sebagai berikut:


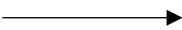
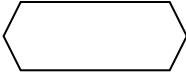
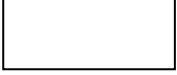


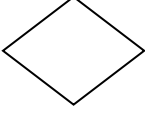
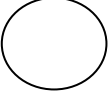
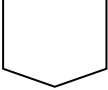
### 1. *Flowchart*

*Flowchart* adalah perangkat diagram grafik yang menyimpan dan mengkomunikasikan aliran data media dan prosedur informasi yang diperlukan dalam sistem informasi. Hal ini dilakukan dengan menggunakan berbagai simbol yang dihubungkan dengan panah-panah untuk menunjukkan kelanjutan aktifitas proses informasi. Sistem *flowchart* berfungsi penting sebagai media dan *hardware* yang digunakan dan proses yang berhubungan dengan sistem informasi fisik yang diperlukan atau diajukan.

Sistem ini banyak dipakai untuk menghubungkan struktur menyeluruh dan aliran sistem ke pengguna akhir karena sistem ini dapat menawarkan tampilan fisik yang berperan penting pada ketertarikan *hardware* dan data media. Walaupun begitu, beberapa kasus, sistem tersebut dapat digantikan dengan diagram aliran dan untuk berkomunikasi dengan pengguna akhir.

*Flowchart* adalah gambaran dalam bentuk diagram alir dari algoritma-algoritma dalam suatu program, yang menyatakan arah alur program tersebut. Berikut adalah beberapa simbol yang digunakan dalam menggambar suatu *flowchart*.

**Tabel 2.1** Simbol *flowchart* dan Penjelasannya

Simbol	Nama	Fungsi
	Terminator	Permulaan / akhir program
	Garis Alir ( <i>Flow Line</i> )	Arah aliran program
	<i>Preparation</i>	Proses inisialisasi / pemberian harga awal
	Proses	Proses perhitungan / Proses pengolahan data
	<i>Input / Output</i> data	Proses input / output data, parameter, informasi
	<i>Predefined</i> <i>Process</i> (Sub Program)	Permulaan sub program / Proses menjalankan sub program
	<i>Decision</i>	Penyeleksian data untuk pilihan selanjutnya
	<i>On Page</i> <i>Connector</i>	Penghubung bagian flowchart yang ada pada satu halaman
	<i>Off Page</i> <i>Connector</i>	Penghubung bagian flowchart yang ada pada halaman berbeda

## 2. *Site Map*

*Site map* adalah susunan menu atau hirarki menu dari suatu situs yang menggambarkan isi dari setiap halaman dan *link* atau navigasi tiap halaman suatu situs *web*. Susunan *sitemap* situs sangat dipengaruhi oleh tujuan pembuatan situs *website*. *Sitemap* dapat dibuat dalam bentuk *flowchart*, dalam bentuk tampilan pohon (*tree-view*). Dari situ akan terlihat struktur, hirarki, dan isi halaman per halamannya. *Flowchart* ini sangat membantu untuk menggambarkan isi setiap halaman dan *link* atau navigasi dan mempermudah mengatur kode dan *file*-nya di antara halaman tersebut.

## 3. *Unified Modeling Language (UML)*

*Unified Modeling Language (UML)* merupakan metode pengembangan perangkat lunak (sistem informasi) dengan menggunakan metode grafis serta merupakan bahasa visualisasi, spesifikasi, konstruksi serta dokumentasi [Adin05].

*Unified Modeling Language (UML)* adalah bahasa yang telah menjadi standar untuk visualisasi, menetapkan, membangun dan mendokumentasikan arti suatu sistem perangkat lunak [Hend07].

Jadi dapat disimpulkan *Unified Modeling Language (UML)* adalah metode pemodelan secara visual untuk sarana perancangan sistem perangkat lunak.

Jenis-jenis diagram UML dan beberapa contoh diagramnya adalah sebagai berikut:

### 1. *Use Case Diagram*

*Use case diagram* yaitu salah satu jenis diagram pada UML yang menggambarkan interaksi antara sistem dan aktor, *use case diagram* juga dapat mendeskripsikan tipe interaksi antara si pemakai sistem dengan sistemnya.

### 2. *Activity Diagram*

*Activity diagram* yaitu salah satu jenis diagram pada UML yang dapat memodelkan proses-proses apa saja yang terjadi pada sistem.

### 3. *Class Diagram*

*Class diagram* yaitu salah satu diagram pada UML yang digunakan untuk menampilkan kelas-kelas maupun paket-paket yang ada pada suatu sistem yang



nantinya akan digunakan. Jadi diagram ini dapat memberikan sebuah gambaran mengenai sistem maupun relasi-relasi yang terdapat pada sistem tersebut.

#### **4. *Entity Relationship Diagram (ERD)***

ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. ERD untuk memodelkan struktur data dan hubungan antar data, untuk digunakan beberapa notasi dan simbol.

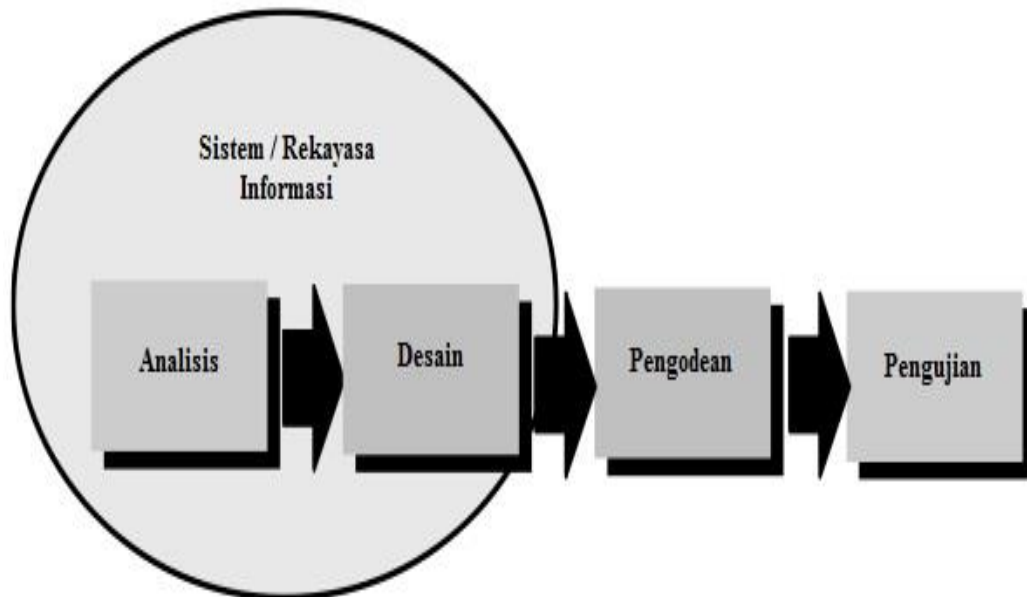
Menurut salah satu para ahli, Brady dan Loonam (2010). *Entity Relationship Diagram (ERD)* merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh sistem analisis dalam tahap analisis persyaratan proyek pengembangan sistem

#### **2.2.9 Metode Pengembangan Sistem**

*System Development Life Cycle* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model metodologi yang digunakan orang untuk mengembangkan sistem sebelumnya.

##### **1. *Waterfall***

*Waterfall* sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial dan terurut dimulai dari analisis, desain, pengkodean, pengujian, dan tahap pendukung (*support*) atau pemeliharaan (*maintenance*). **Gambar 2.1** Berikut adalah tahapan-tahapan dalam metode tersebut:



**Gambar 2.1** Ilustrasi Model *Waterfall*

1. Analisis

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini diperlukan untuk didokumentasikan.

2. Desain

Desain perangkat lunak adalah proses *multi* langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengkodean. Tahap ini menerjemahkan kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program tahap selanjutnya.

3. Implementasi

Desain harus diterjemahkan ke dalam program perangkat lunak. Hasil dari tahap ini adalah komputer sesuai dengan desain yang telah dibuat.

#### 4. Pengujian

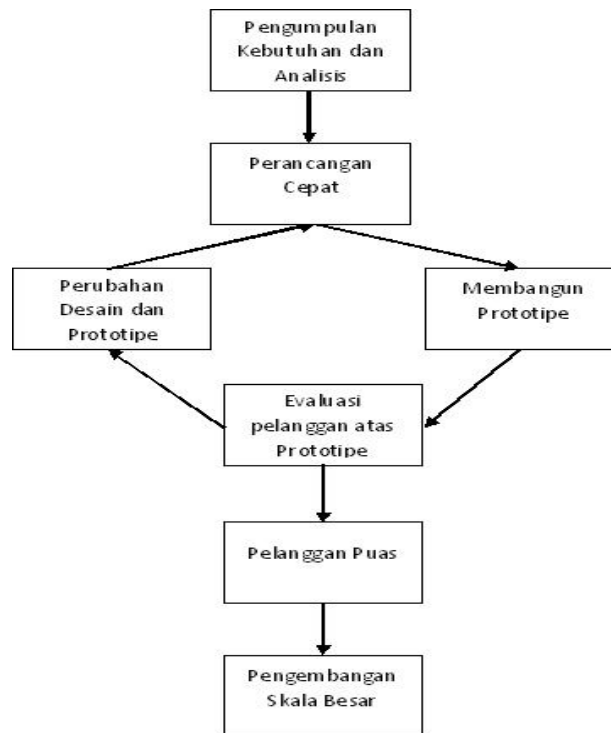
Pengujian berfokus pada perangkat lunak dari segi logika dan fungsional dan memastikan bahwa semua bagian sudah diuji. Ini dilakukan untuk meminimalisir kesalahan dan memastikan keluaran yang dihasilkan sesuai dengan apa yang diinginkan oleh pengguna.

#### 5. Pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirim ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru.

#### **2.2.10 Prototype**

*Prototype* adalah bagian dari produk yang mengekspresikan logika maupun fisik antarmuka eksternal yang ditampilkan. Konsumen potensial menggunakan *prototype* dan menyediakan masukan untuk tim pengembang sebelum pengembangan skala besar dimulai. Melihat dan mempercayai menjadi hal yang diharapkan untuk dicapai dalam *prototype*. Dengan menggunakan pendekatan ini, klien dan tim pengembang dapat mengklarifikasikan kebutuhan dan interpretasi mereka. **Gambar 2.2** Tahap-tahap dari model *prototype* adalah sebagai berikut:



**Gambar 2.2** Langkah-langkah pengembangan *prototype*.

### 2.2.11 Metode Pengujian Sistem

Pengujian sistem menyajikan anomali yang menarik bagi rekayasa perangkat lunak pada proses perangkat lunak, perancang berusaha membangun perangkat lunak dari konsep abstrak ke implementasi yang dapat dilihat, baru kemudian dilakukan pengujian.

#### **Pengujian *Black Box***

Dalam pengujian perangkat lunak ada dua yaitu *white box testing* dan *black box testing*. Dari kedua metode itu, pada skripsi dipilih menggunakan *black box testing* karena dianggap lebih tepat dibanding *white box testing*. Perangkat lunak memerlukan tes untuk pencarian kesalahan fungsi-fungsi dalam aplikasi sehingga dalam hal ini *black box testing* lebih sesuai. Pengujian ini digunakan untuk mengetahui apakah fungsi-fungsi dalam perangkat lunak sudah sesuai dengan yang diharapkan.

Menurut Roger S. Pressman (2010), *black box testing* berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan *engineer* untuk memperoleh *input* yang sepenuhnya akan melaksanakan persyaratan fungsional untuk sebuah program. *Black box testing* berusaha untuk menemukan kesalahan dalam kategori berikut:

1. Fungsi yang tidak benar atau fungsi yang hilang.
2. Kesalahan antarmuka.
3. Kesalahan dalam struktur data atau akses *database* eksternal.
4. Kesalahan kinerja.
5. Kesalahan inisialisasi dan pemutusan kesalahan.

#### **2.2.12 Kriteria Website yang Baik.**

Menurut Suryanto (2007), setiap sesuatu diciptakan mempunyai kriteria, dalam hal ini situs juga mempunyai kriteria. Adapun kriteria situs-situs yang baik dapat dilihat dari:

##### **1. Usability**

Sebagai suatu cara untuk menggunakan situs *web* tersebut dengan efektif. Untuk mencapai tingkat *usability* yang ideal, situs *web* harus memenuhi 5 (lima) syarat, antara lain: mudah dipelajari, efisien dalam penggunaan, mudah untuk diingat, tingkat kesalahan yang rendah, dan kepuasan pengguna.

##### **2. Sistem Navigasi**

Digunakan untuk membantu pengunjung menemukan jalan yang mudah ketika menjelajahi situs *web*. Navigasi dapat ditampilkan dalam berbagai media, yaitu teks, *image*, animasi. Adapun syarat navigasi yang baik, antara lain: mudah dipelajari, tetap konsisten, memungkinkan *feedback*, muncul dalam konteks, menawarkan alternatif lain, memerlukan perhitungan waktu dan tindakan, menyediakan pesan *visual* yang jelas, menggunakan label yang jelas dan mudah dipahami, mendukung tujuan dan perilaku *user*.

### 3. *Graphic Design* (Desain Visual)

Kepuasan *visual* seorang *user* secara subyektif melibatkan bagaimana *designer visual* situs *web* tersebut membawa mata *user* menikmati dan menjelajahi situs *web* dengan melalui *layout*, warna, bentuk, dan tipografi. Grafik membuat halaman menjadi indah tetapi bisa juga memperlambat akses dengan semakin besarnya ukuran *file*.

### 4. *Contents*

Konten yang baik akan menarik, relevan, dan pantas untuk target audiens situs *web* tersebut. Gaya penulisan dan bahasa yang digunakan harus sesuai dengan *web* dan target audiens. Hindari kesalahan dalam penulisan, termasuk tata bahasa dan tanda baca di setiap halaman, *header*, dan judulnya.

### 5. *Compatibility*

Situs *web* harus kompatibel dengan berbagai perangkat tampilannya (*browser*), harus memberikan alternatif bagi *browser* yang tidak dapat menampilkan situsnya.

### 6. *Loading Time*

Sebuah situs *web* yang tampil lebih cepat kemungkinan besar akan kembali dikunjungi, apalagi memiliki konten dan tampilan yang menarik.

### 7. *Functionality*

Seberapa baik sebuah situs *web* bekerja dari aspek teknologinya, ini bisa melibatkan *programmer* dengan *script*-nya, misalnya HTML (DHTML), PHP, ASP, *ColdFusion*, CGI, SSI.

### 8. *Accessibility*

Halaman *web* harus bisa digunakan oleh setiap orang. Ada berbagai hambatan pengguna untuk bisa menikmati halaman *web* tersebut. Untuk hambatan fisik, bagaimana memaksimalkan penggunaan konten ketika satu atau lebih indera dimatikan atau dikurangi kerjanya, terutama untuk user dengan kekurangan penglihatan.