

BAB II

LANDASAN TEORI

2.1 Perancangan

Menurut Rosa A.S dan M.Shalahuddin (2013,p.23) Desain atau perancangan adalah merupakan upaya untuk mengonstuksi sebuah sistem yang memberikan upaya kepuasan (mungkin informal) akan spesifikasi kebutuhan fungsional, memenuhi target, memenuhi kebutuhan secara implisit atau eksplisit dari segi performansi maupun pengguna sumber daya, kepuasan batasan pada proses desain dari segi biaya, waktu, dan perangkat.

2.1.1 Sistem

Menurut Tata Sutabri (2012,p.10) Sistem adalah sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau *variabel* yang terorganisir, saling berinteraksi, saling tergantung satu sama lain, dan terpadu. Model umum sebuah sistem adalah *input*, proses, dan *output*. Hal ini merupakan konsep sebuah sistem yang sangat sederhana sebab sebuah sistem dapat mempunyai beberapa masukan dan keluaran. Selain itu, sebuah sistem mempunyai karakteristik atau sifat-sifat tertentu yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem. Adapun karakteristik yang dimaksud adalah sebagai berikut:

1. Komponen Sistem (*Component*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai sistem yang lebih besar atau sering disebut “supra sistem”.

2. Batasan Sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisahkan.

3. Lingkungan Luar Sistem (*Environment*)

Bentuk apapun yang ada diluar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut lingkungan luar sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Dengan demikian, lingkungan luar tersebut harus tetap dijaga dan dipelihara. Lingkungan luar yang merugikan harus dikendalikan. Kalau tidak, maka akan mengganggu kelangsungan hidup sistem tersebut.

4. Penghubung Sistem (*Interface*)

Media yang menghubungkan sistem dengan subsistem lain disebut penghubung sistem atau *interface*. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu sub sistem ke subsistem lain. Bentuk keluaran dari subsistem akan menjadi masukan untuk subsistem lain melalui penghubung tersebut. Dengan demikian, dapat terjadi suatu integrasi sistem yang membentuk satu kesatuan.

5. Masukan Sistem (*Input*)

Energi yang dimasukkan kedalam sistem tersebut masukkan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*). Contoh, di dalam suatu unit sistem komputer, "Program" adalah *maintenance input* yang digunakan untuk mengoperasikan komputernya dan "Data" adalah *signal input* untuk diolah menjadi informasi.

6. Keluaran Sistem (*Output*)

Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain seperti sistem informasi. Keluaran yang dihasilkan adalah informasi. Informasi ini dapat

digunakan sebagai masukan untuk pengambilan keputusan atau hal-hal lain yang menjadi *input* bagi subsistem lain.

7. Pengolahan Sistem (*Process*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran, contohnya adalah sistem akuntansi. Sistem ini akan mengolah data transaksi menjadi laporan-laporan yang dibutuhkan oleh pihak manajemen.

8. Sasaran Sistem (*Objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat *deterministic*. Kalau suatu sistem tidak memiliki sasaran maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran dan tujuan yang telah direncanakan.

2.1.2 Informasi

Menurut Tata Sutabri (2012,p.29) Informasi adalah data yang telah diklasifikasi atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan. Sistem pengolahan informasi mengolah data menjadi informasi atau tepatnya mengolah data dari tak berguna menjadi berguna bagi penerimanya. Nilai informasi berhubungan dengan keputusan maka informasi menjadi tidak diperlukan keputusan dapat berkisar dari keputusan berulang sederhana sampai keputusan strategis jangka panjang. Nilai informasi dilukiskan paling berarti dalam *konteks* sebuah keputusan.

2.1.3 Sistem Informasi

Menurut Tata Sutabri (2012,p.46) Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk menyediakan kepada pihak luar tertentu dengan laporan- laporan yang diperlukan

2.2 Jasa Penitipan Hewan

Menurut Puji Astuti (2012) Jasa penitipan hewan merupakan tempat perawatan hewan bagi masyarakat yang bingung meninggalkan hewan kesayangannya di rumah saat mudik atau pergi dalam kurun waktu lama. Pada saat itulah, pemilik hewan peliharaan berlomba-lomba untuk mencari tempat penitipan hewan terbaik untuk menitipkan hewan peliharaan kesayangannya.

2.3 Pengembangan Sistem

Pengembangan sistem (*systems development*) dapat berarti menyusun suatu sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang telah ada (Jogiyanto, 2011).

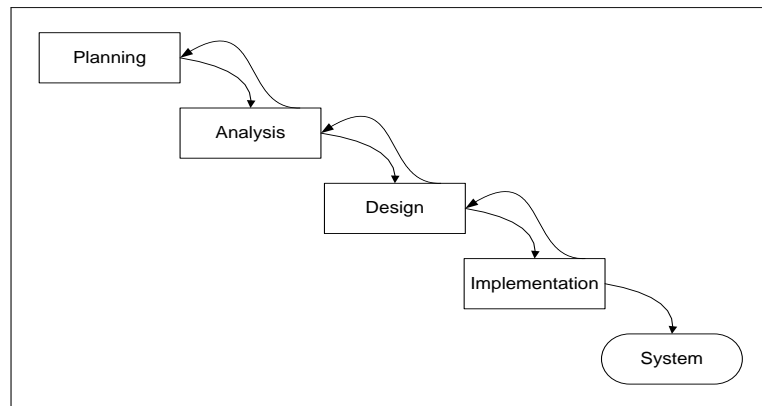
a. Metodologi Pengembangan Sistem

Dalam pengembangan sistem ini penulis menggunakan metodologi *waterfall*. Metode air terjun atau yang sering disebut metode *waterfall* sering dinamakan siklus hidup klasik (*classic life cycle*), dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan (*planning*), perencanaan (*analisis*), perancangan (*design*), serta penyerahan sistem ke para pelanggan/pengguna (*system*), yang diakhiri dengan dukungan pada perangkat lunak lengkap yang dihasilkan (Dennis, Wixom dan Roth.2012)

b. Tahapan Pengembangan Sistem

Metode System Development Life Cycle model Waterfall

Pada metode penelitian ini dilakukan rekayasa perangkat lunak yang digunakan adalah model *Waterfall* seperti pada gambar berikut ini:



Gambar 2.1 Metode Pengembangan Model *Waterfall* (Dennis dan Wixom.2003)

Keterangan:

1. *Planning (Perencanaan)*

Tahap perencanaan merupakan proses penting untuk mengetahui mengapa sistem harus dibuat dan menentukan bagaimana cara membangun sistem tersebut. Langkah pertama dari proses tersebut adalah dengan mengidentifikasi peluang apakah dapat memberikan kemungkinan biaya rendah tetapi menghasilkan keuntungan.

2. *Analysis (Analisis)*

Analisis sistem dilakukan untuk memberikan jawaban pertanyaan siapa yang akan menggunakan sistem. Apa yang akan dilakukan oleh sistem, dimana dan kapan sistem tersebut digunakan. Pada tahap ini pembuat sistem akan melakukan observasi dan pengamatan terhadap sistem yang lama, kemudian mengidentifikasi, memanfaatkan dan mengembangkan peluang, dan membangun konsep untuk sebuah sistem baru.

3. *Design (perancangan)*

Tahap perancangan dilakukan untuk menetapkan bagaimana sistem akan dioperasikan. Hal ini berkaitan dengan menentukan perangkat keras, perangkat lunak, jaringan, tampilan program, *form* dan laporan yang akan dipakai. Selain itu perlu juga menspesifikasi program, database dan file yang dibutuhkan.

4. *Implementation*

Merupakan tahap berikutnya untuk menerjemahkan data atau pemecahan masalah yang telah dirancang ke dalam bahasa pemrograman komputer yang telah ditentukan. Semua tahap ini desain perangkat lunak sebagai sebuah program lengkap atau unit program.

5. *System*

Tahapan ini, merupakan hasil sistem yang telah dibuat dalam bentuk perangkat lunak yang telah dipasang dan digunakan, termasuk didalamnya proses pemeliharaan dan perbaikan kesalahan. Perangkat lunak yang telah selesai dibuat dapat mengalami perubahan-perubahan atau penambahan sesuai dengan permintaan *user* atau perubahan sistem.

2.4 Aplikasi dan *Mobile*

Aplikasi adalah program yang di gunakan orang untuk melakukan sesuatu pada sistem komputer dan *mobile* dapat di artikan sebagai perpindahan yang mudah dari suatu tempat ke tempat yang lain, misalnya terminal telepon yang dapat berpindah dari satu tempat ke tempat lain tanpa terjadi pemutusan atau terputus komunikasi.

Menurut Imam Mulhim (2014), *Mobile* dapat diartikan sebagai perpindahan yang mudah dari satu tempat ke tempat yang lain, misalnya telepon *mobile* berarti bahwa terminal telepon yang dapat berpindah dengan mudah dari satu tempat ke tempat lain tanpa terjadi pemutusan atau terputusnya komunikasi. Sistem aplikasi *mobile* merupakan aplikasi yang dapat digunakan walaupun pengguna berpindah dengan mudah dari satu tempat ketempat lain lain tanpa terjadi pemutusan atau terputusnya komunikasi. Aplikasi ini dapat diakses melalui perangkat nirkabel seperti pager, seperti telepon seluler dan *PDA*. Adapun karakteristik perangkat *mobile* yaitu:

1. Ukuran yang kecil : Perangkat *mobile* memiliki ukuran yang kecil. Konsumen menginginkan perangkat yang terkecil untuk kenyamanan dan mobilitas mereka.
2. Memory yang terbatas : Perangkat *mobile* juga memiliki memory yang kecil, yaitu *primary (RAM)* dan *secondary (disk)*.
3. Daya proses yang terbatas : Sistem *mobile* tidaklah setangguh rekan mereka yaitu *desktop*.
4. Mengonsumsi daya yang rendah : Perangkat *mobile* menghabiskan sedikit daya dibandingkan dengan mesin *desktop*
5. Kuat dan dapat diandalkan : Karena perangkat *mobile* selalu dibawa kemana saja, mereka harus cukup kuat untuk menghadapi benturan-benturan, gerakan, dan sesekali tetesan-tetesan air.
6. Konektivitas yang terbatas : Perangkat *mobile* memiliki *bandwith* rendah, beberapa dari mereka bahkan tidak tersambung.

2.5 Unified Modeling Language (UML)

Rosa dan Salahudin (2011,p.131), menjelaskan bahwa *unified modeling language (UML)* menguraikan salah satu *standar* bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat aplikasi dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.


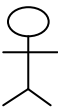

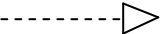
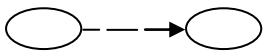
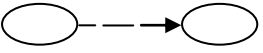
Kesimpulan metode *UML (Unified Modeling Language)* adalah merupakan sebuah metode atau sebuah bahasa yang digunakan dalam menterjemahkan, menjelaskan, memodelkan, mendefinisikan suatu sistem dengan bentuk simbol-simbol tertentu yang bertujuan untuk memberikan penjelasan-penjelasan detail dari sebuah sistem.

2.5.1 Use Case Diagram

Rosa dan Salahudin (2011,p.131) menguraikan bahwa *use case* diagram merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem

informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Simbol-simbol yang digunakan untuk pembuatan *use case* diagram dapat dilihat pada tabel 2.1 berikut

Tabel 2.1 Simbol *Use Case Diagram*



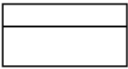


Simbol	Keterangan
<i>Use Case</i> 	Menggambarkan bagaimana seseorang akan menggunakan atau memanfaatkan sistem.
Aktor 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar
<i>Asosiasi</i> 	Komunikasi antara <i>use case</i> dan aktor yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki
<i>Generalisasi</i> 	Sebagai penghubung antara aktor- <i>use case</i> atau <i>use case-use case</i> .
<<Include>> 	<i>Include Relationship</i> (relasi cakupan) : Memungkinkan suatu <i>use case</i> untuk menggunakan
<<Extend>> 	<i>Extend Relationship</i> : Memungkinkan relasi <i>use case</i> memiliki kemungkinan untuk memperluas fungsionalitas



2.5.2 Class Diagram

Class diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (*atribut/properti*) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (*metode/fungsi*). *Class*

diagram menggambarkan struktur dan deskripsi *class*, *package* dan beserta hubungan satu sama lain seperti *containment*, pewarisan, *asosiasi*, dan lain-lain. Simbol-simbol yang digunakan untuk pembuatan *class diagram* dapat dilihat pada tabel 2.2 berikut:

Tabel 2.2 *Simbol Class Diagram*

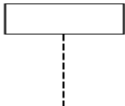
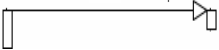
SIMBOL	NAMA	KETERANGAN
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.


	<p style="text-align: center;"><i>Dependency</i></p>	<p>Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri</p>
	<p style="text-align: center;"><i>Association</i></p>	<p>Apa yang menghubungkan antara objek satu dengan objek lainnya</p>

2.5.3 Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa pesan yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi *vertikal* (waktu) dan dimensi *horizontal* (objek-objek yang terkait). Simbol-simbol yang digunakan untuk pembuatan *class diagram* dapat dilihat pada tabel 2.3 berikut :

Tabel 2.3 Simbol Sequence Diagram


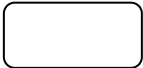
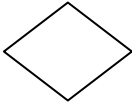

GAMBAR	NAMA	KETERANGAN
	<p style="text-align: center;"><i>LifeLine</i></p>	<p>Objek <i>entity</i>, antarmuka yang saling berinteraksi.</p>
	<p style="text-align: center;"><i>Message</i></p>	<p>Spesifikasi dari komunikasi antar objek yang memuat informasi-</p>


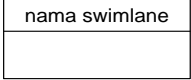
		informasi tentang aktifitas yang terjadi
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas.

2.5.4 Activity Diagram

Activity Diagram menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. (Rosa A.S dan Salahudin (2011,p.134). Simbol-simbol yang digunakan untuk pembuatan *activity diagram* dapat dilihat pada tabel 2.4 di bawah ini:

Tabel 2.4 Simbol Activity Diagram

Simbol	Keterangan
Status Awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan 	Asosiasi percabangan dimana ada pilihan aktivitas lebih dari satu. Tabel 2.5 (Lanjutan)
Penggabungan 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.

Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas.

2.6 Database

Menurut (Date dalam Muiz, 2007) mengatakan bahwa sistem basis data pada dasarnya dapat dianggap sebagai tempat atau lokasi untuk sekumpulan berkas data yang sudah terkomputerisasi dengan tujuan untuk memelihara informasi, dan juga memuat informasi tersebut, terutama apabila informasi tersebut sedang dibutuhkan.

Sistem basis data memiliki pengertian yang tidak terlalu banyak. Hal ini dikarenakan banyak para pakar yang sering mengkaitkan sistem basis data dengan *DBMS* atau *database management system* (Sistem manajemen basis data).

2.6.1 Fungsi Database

Berikut adalah fungsi dari *Database*:

- a. Mengelompokkan data, *database* bertujuan untuk mengelompokkan data agar mudah dipahami.
- b. Menghindari terjadinya duplikasi atau inkonsistensi data.
- c. Memudahkan dalam menyimpan, mengakses, dan memperbarui, serta menghapus data.
- d. Menjamin kualitas data dan informasi yang diakses sesuai dengan yang dimasukkan (*Integritas data*)
- e. Menjadi solusi dalam proses penyimpanan sebuah data, terutama data yang besar.
- f. Menunjang kinerja aplikasi yang membutuhkan sebuah penyimpanan data.

2.6.2 Manfaat Database

Berikut adalah manfaat *database*:

- a. Sebagai komponen utama atau penting dalam sistem informasi, karena merupakan dasar dalam menyediakan informasi.
- b. Menentukan kualitas informasi yaitu cepat, akurat, dan *relevan*.
- c. Mengatasi kerangkapan data (*redundancy data*).
- d. Menghindari terjadinya *inkonsistensi* data.
- e. Mengatasi kesulitan dalam mengakses data.
- f. Menyusun format yang standar dari sebuah data.
- g. Melakukan perlindungan dan pengamanan data.

2.7 Perangkat Lunak Pendukung

2.7.1 Adobe dreamweaver cs 5

Menurut Sigit (2010,p.1) *Adobe Dreamweaver* adalah sebuah *HTML editor profesional* untuk mendesain *web* secara *visual* dan mengelola situs atau halaman *web*. Saat ini terdapat *software* dari kelompok *Adobe* yang belakangan banyak digunakan untuk mendesain suatu *web*. Versi terbaru dari *Adobe Dreamweaver* memiliki beberapa kemampuan bukan hanya sebagai *software* untuk desain *web* saja, tetapi juga menyunting kode serta pembuatan aplikasi *web*. Antara lain: *JSP, PHP, ASP, XML*, dan *ColdFusion*.

2.7.2 Notepad++

Selain menggunakan aplikasi *Dreamweaver*, peneliti juga menggunakan aplikasi *notepad++* dikarenakan tampilan *interface* aplikasi yang lebih mudah dipahami dalam melihat *source* program. *Notepad++* merupakan sebuah penyunting teks dan penyunting kode sumber yang tersedia dalam beberapa sistem operasi seperti *Windows, Linux*, dan *Mac OS X*. *Notepad++* menggunakan komponen *Scintilla* untuk dapat menampilkan dan menyunting teks dan berkas kode sumber berbagai bahasa

pemrograman. *Notepad++* didistribusikan sebagai perangkat lunak bebas. Proyek ini dilayani oleh *Sourceforge.net* dengan telah diunduh lebih dari 27 juta kali dan dua kali memenangkan penghargaan *SourceForge Community Choice Award for Best Developer Tools*.

2.7.3 XAMPP

Library IBM (2004), *XAMPP* merupakan aplikasi server yang menggabungkan beberapa aplikasi *server* yang biasa digunakan di *web server*. Berikut beberapa komponen-komponen yang terdapat pada *XAMPP*, yaitu *Apache (web server)*, *MySQL (database server)*, *Filezilla FTP server*, *Mercury Mail (mail server)*, *phpMyAdmin (web-based interface MySQL)*.

2.7.4 Pengertian HTML

Program adalah kata, ekspresi, pernyataan atau kombinasi yang disusun dan dirangkai menjadi satu kesatuan prosedur yang berupa urutan langkah untuk menyelesaikan masalah dan di implementasikan dengan menggunakan bahasa pemrograman sehingga dapat dieksekusi oleh komputer. Pemrograman adalah proses mengimplementasikan urutan langkah untuk menyelesaikan suatu masalah dengan menggunakan suatu bahasa pemrograman. Bahasa pemrograman adalah suatu prosedur penulisan program. Sedangkan pemrograman terstruktur adalah metode untuk mengorganisasikan dan membuat kode-kode program supaya mudah untuk dimengerti, mudah ditest dan mudah dimodifikasi. Salah satu contoh pemrograman adalah pemrograman di internet.

HTML merupakan salah satu unsur penting dalam pemrograman di internet. *HTML (Hypertext Markup Language)* adalah bahasa pengkodean untuk menghasilkan dokumen-dokumen *hypertext* untuk digunakan di *World Wide Web*. Namun *HTML* bukanlah sebuah bahasa pemrograman, tetapi hanyalah berisi perintah-perintah yang telah terstruktur berupa tag-tag penyusun. *HTML* merupakan pengembangan dari

standar pemformatan dokumen teks yaitu *Standard Generalized Markup Language (SGML)*. *HTML* sebenarnya adalah dokumen *ASCII* atau teks biasa, yang dirancang untuk tidak tergantung pada suatu sistem operasi tertentu.

HTML ini merupakan sebuah format bahasa dari *world wide web (WWW)* untuk menulis dan menampilkan dokumen. Sehingga dokumen tersebut dapat dengan mudah tersebar ke PC melalui media jaringan komputer atau internet. *HTML* ini berupa kode tag yang menginstruksikan *browser* untuk menghasilkan tampilan sesuai dengan yang diinginkan. Sehingga, untuk dapat membuka dokumen yang ditulis dengan *HTML* maka diperlukan sebuah tool yaitu *browser* misalnya *Opera*, *Mozilla*, *Google Chrome*, *Netscape*, dan banyak lagi yang lainnya. *HTML* juga dapat dikenali oleh aplikasi pembuka email ataupun dari *PDA* dan program lain yang memiliki kemampuan *browser*.

Adapun pengertian dari *HTML5* itu sendiri adalah versi terbaru dari *HTML (HyperText Markup Language)* yang mana dikembangkan oleh *W3C* atau *World Wide Web Consortium*. *HTML5* diciptakan dalam pengembangan bahasa *HTML* untuk memperbaiki konten dan memperbarui teknologi multimedia yang sudah ada pada *HTML 4* dan versi sebelumnya agar mudah dijalankan oleh *browser* dan mudah dimengerti. *HTML5* merupakan kerjasama antara *World Wide Web Consortium (W3C)* dan *Web Hypertext Application Teknologi Working Group (WHATWG)*. *WHATWG* bekerja dengan bentuk *web* dan aplikasi, dan *W3C* bekerja dengan *XHTML 2.0*. Tujuan utama pengembangan *HTML5* adalah untuk memperbaiki teknologi *HTML* agar mendukung teknologi multimedia terbaru, mudah dibaca oleh manusia dan juga mudah dimengerti oleh mesin.

Di sisi lain *HTML5* sering juga disebut dengan *Semantic Web*. *Semantic web* berasal dari bahasa Yunani (*Greek*), merupakan bahasa pembelajaran *semiotic*, yakni pembelajaran untuk memahami penanda. *Semantic* sendiri mempunyai arti bahasa

yang berfokus pada penanda untuk mengetahui arti yang terkandung di dalamnya. Jadi, yang dimaksud *semantic web* adalah bahasa pemrograman yang mempunyai penanda khusus dalam implementasinya dengan tujuan agar mampu mendeskripsikan apa yang terkandung dalam *website* tersebut. *Web semantic* ini bukan hanya dikembangkan di *Web 2.0*, namun sudah ke *Web 3.0*.

2.8 Pengertian *Mobile Application*

menurut Turban (2013), *Mobile application* juga biasa disebut dengan *mobile apps*, yaitu istilah yang digunakan untuk mendeskripsikan aplikasi internet yang berjalan pada *smartphone* atau *piranti mobile* lainnya. Aplikasi *mobile* biasanya membantu para penggunanya untuk terkoneksi dengan layanan internet yang biasa diakses pada sebuah *personal computer*.

2.8.1 Kelebihan *Mobile Application*

menurut Turban *mobile application* memiliki beberapa kelebihan antara lain :

1. *Ubiquity*: Pertukaran informasi dapat dikirimkan kapan saja dan dimana saja selama adanya akses internet.
2. *Convenience*: Fungsi dan kegunaan yang semakin banyak dengan bentuk fisik yang tetap kecil membuat orang pada kebanyakan lebih nyaman dalam melakukan akses internet menggunakan perangkat *mobile* dibanding *platform* lain.
3. *Interactivity*: Dibanding komputer *desktop*. Perangkat *mobile* lebih interaktif.
4. *Personalization*: Pengguna bisa bebas melakukan pengaturan dalam melakukan transaksi.
5. *ime Efficient*: Tidak perlu lagi melakukan *set up* seperti pada komputer.