

BAB II

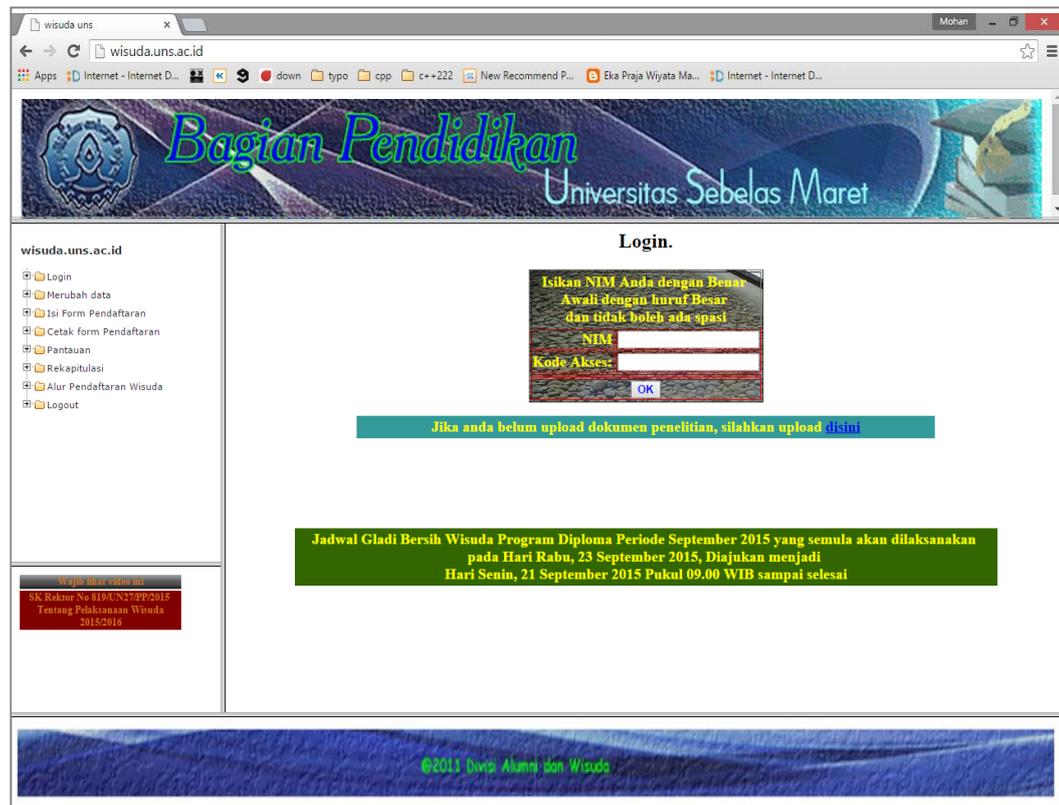
LANDASAN TEORI

2.1 Tinjauan Pustaka

Sistem pelayanan wisuda merupakan sistem yang hampir ada pada setiap universitas. Dibutuhkan suatu tinjauan pustaka agar dapat merancang dan membangun sistem pelayanan wisuda dengan baik dan benar, sehingga diperlukan referensi atau literatur yang berkaitan dengan sistem pelayanan wisuda sebagai pembanding dalam melakukan penelitian. Dalam hal ini diambil beberapa sumber referensi dari sistem-sistem pelayanan wisuda yang sudah ada sebelumnya. Selain untuk bahan acuan, kajian pustaka ini dilakukan agar keaslian dan keabsahan penelitian tentang sistem pelayanan wisuda dapat dipertanggungjawabkan. Namun setelah dilakukan observasi, masih sedikit sekali jurnal penelitian yang membahas sistem pelayanan wisuda, hanya ditemukan satu penelitian ilmiah yang membahas tentang sistem pelayanan wisuda, oleh sebab itu dalam tinjauan pustaka ini juga akan disajikan referensi sistem pelayanan wisuda dalam bentuk lain berupa sistem-sistem pelayanan wisuda yang sudah diimplementasikan oleh beberapa universitas negeri maupun swasta. Berikut ini akan dibahas beberapa literatur tentang sistem pelayanan wisuda tersebut:

2.1.1 Sistem Pelayanan Wisuda Universitas Sebelas Maret (UNS)

Sistem pelayanan wisuda Universitas Sebelas Maret (UNS) dapat diakses melalui alamat <http://wisuda.uns.ac.id>. Secara tampilan sistem pelayanan wisuda Universitas Sebelas Maret terlihat sederhana karena pengguna atau *user* langsung disuguhkan dengan *form login* yang dilengkapi dengan informasi singkat jadwal gladi bersih wisuda tepat di bagian bawahnya. Sedangkan menu utama tersemat di sebelah kiri dan terdiri dari menu *login*, menu merubah data, menu isi *form* pendaftaran, menu cetak *form* pendaftaran, menu pantauan, menu rekapitulasi, menu alur pendaftaran wisuda dan *logout*. Tampilan sistem pelayanan wisuda UNS dapat dilihat pada Gambar 2.1.



Gambar 2.1 Sistem Pelayanan Wisuda Universitas Sebelas Maret (UNS)

2.1.2 Sistem Informasi Pendaftaran Wisuda *Online* UIN Syarif Hidayatullah Jakarta

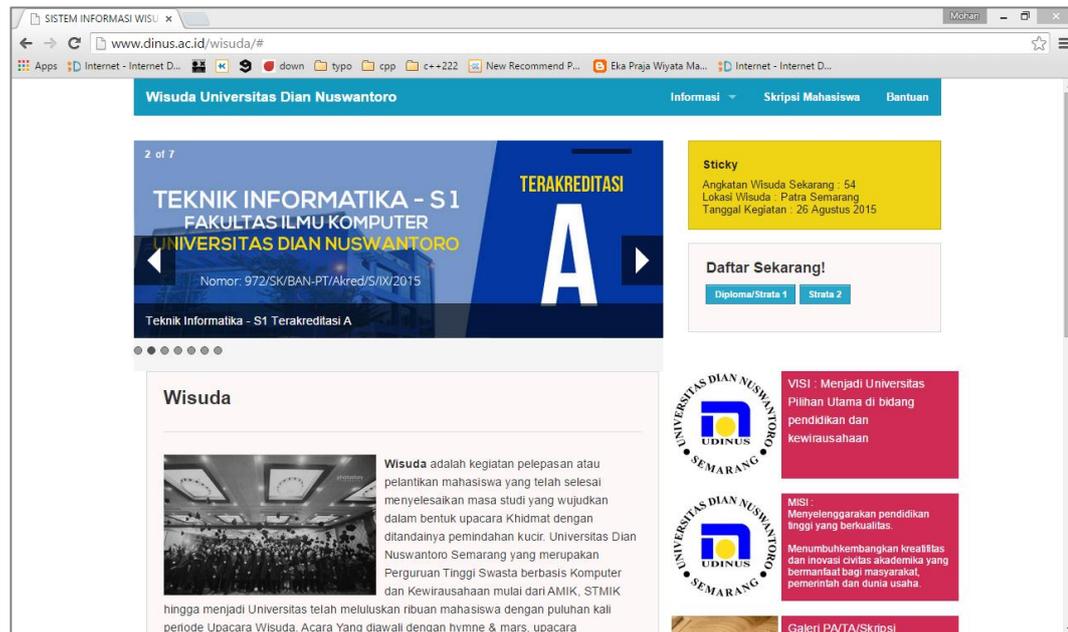
Menurut Dwi Martha Jingga (2009) Sistem Informasi Pendaftaran Wisuda *Online* merupakan salah satu solusi yang tepat bagi Universitas Islam Negeri (UIN) Syarif Hidayatullah yang dapat mempermudah dalam pengaksesan sistem yang selama ini sering mengalami hambatan dalam melihat informasi wisuda seperti jadwal wisuda dan persyaratan wisuda dan mempersingkat pencatatan data calon wisuda serta meningkatkan pelayanan kepada mahasiswa karena, universitas dapat memberikan informasi-informasi yang berkaitan dengan pendaftaran wisuda. Adapun menu yang ada pada sistem tersebut adalah: menu *home*, menu jadwal wisuda, menu persyaratan wisuda, menu nama wisudawan, menu registrasi, menu *about*, menu *login* calon wisuda. Berikut ini tampilan dari Sistem Informasi Pendaftaran Wisuda *Online* UIN Syarif Hidayatullah Jakarta ditunjukkan oleh Gambar 2.2.



Gambar 2.2 Sistem Informasi Pendaftaran Wisuda *Online* UIN Syarif Hidayatullah Jakarta

2.1.3 Sistem Informasi Wisuda Universitas Dian Nuswantoro (UDINUS)

Sistem Informasi Wisuda Universitas Dian Nuswantoro (UDINUS) dapat diakses melalui alamat <http://www.dinus.ac.id/wisuda>. Sistem informasi wisuda UDINUS menyuguhkan informasi wisuda yang cukup sederhana tapi juga menampilkan informasi yang cukup jelas. Di halaman utama *user* dapat mendapatkan informasi jumlah angkatan wisuda yang berlangsung disertai jadwal dan tempat diadakannya wisuda. Seperti sistem informasi wisuda pada umumnya, Sistem Informasi Wisuda Universitas Dian Nuswantoro (UDINUS) terdapat tombol pendaftaran wisuda yang dibedakan menjadi dua yaitu: pendaftaran wisuda diploma/Strata 1 dan pendaftaran jenjang strata 2 dan terdapat 3 menu utama yang terdiri dari: menu informasi, skripsi mahasiswa dan bantuan. Berikut ini tampilan utama Sistem Informasi Wisuda Universitas Dian Nuswantoro (UDINUS) yang ditunjukkan oleh Gambar 2.3.

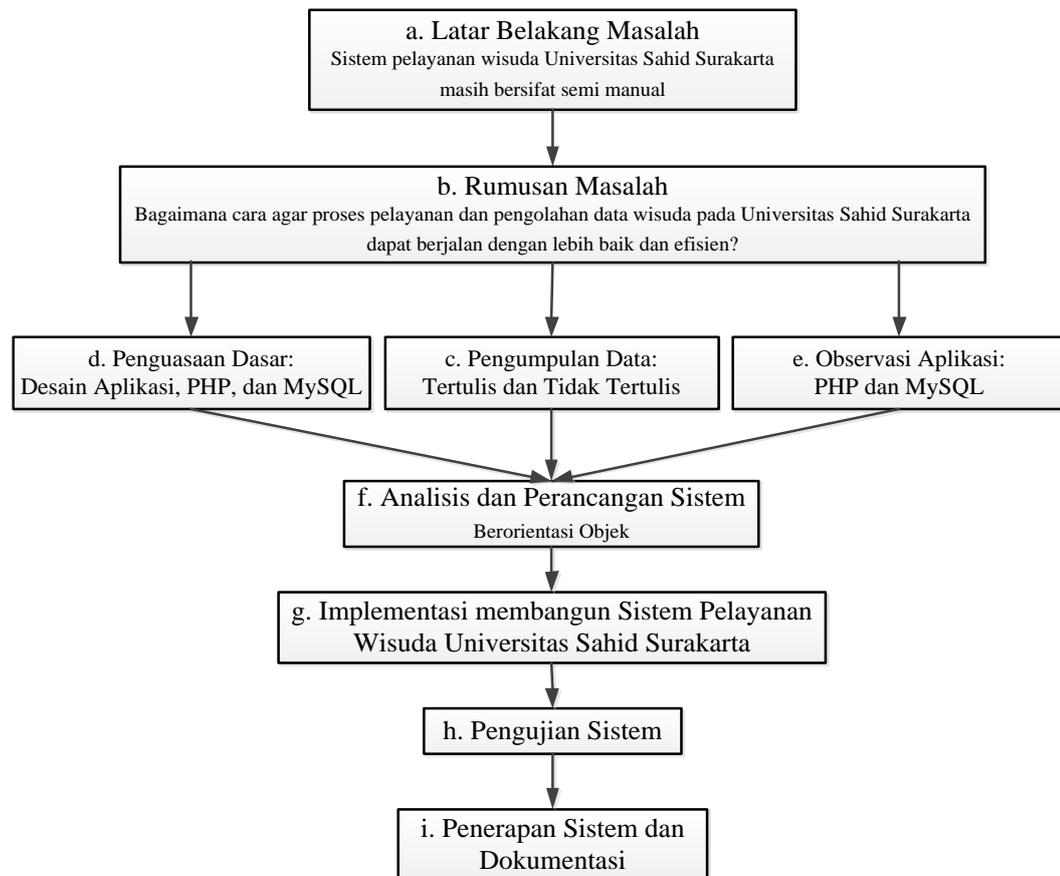


Gambar 2.3 Sistem Informasi Wisuda Universitas Dian Nuswantoro (UDINUS)

Dari tiga referensi di atas dapat disimpulkan bahwa rata-rata sistem pelayanan wisuda sudah baik, namun belum memberikan informasi secara jelas tentang hal-hal yang menyangkut wisuda, seperti statistik lulusan wisuda, jumlah kuota wisuda yang akan berlangsung, daftar wisudawan terbaik dan lain sebagainya, dan dari segi antar muka pun terlihat relatif sederhana. Sehingga dengan adanya referensi/literatur di atas dapat dibangun sistem pelayanan wisuda di Universitas Sahid Surakarta dengan menambah beberapa fitur dan penyegaran di bagian *layout* tampilan agar sistem dapat memberikan pelayanan wisuda dengan lebih optimal.

2.2 Kerangka Pemikiran

Berikut ini merupakan kerangka pemikiran yang digunakan dalam perancangan dan pembuatan sistem pelayanan wisuda pada Universitas Sahid Surakarta yang disusun sedemikian rupa berdasarkan aturan-aturan yang berlaku sehingga menjadi bagan terstruktur yang terdiri dari 10 bagian yang dijelaskan oleh Gambar 2.4 berikut.



Gambar 2.4 Kerangka Pemikiran

Keterangan:

a. Latar Belakang Masalah

Pokok permasalahan yang mendasari perlunya dibangun suatu sistem pelayanan wisuda pada Universitas Sahid Surakarta.

b. Perumusan Masalah

Perumusan masalah adalah rumusan dari hasil simpulan masalah dan solusi untuk menyelesaikan permasalahan yang didapatkan berdasarkan pokok permasalahan yang dibahas pada latar belakang masalah sebelumnya.

c. Pengumpulan Data Tertulis dan tidak Tertulis

Mengumpulkan semua data yang dibutuhkan, baik melalui observasi, wawancara/*interview* dan literatur pustaka yang berkaitan dengan sistem pelayanan wisuda di Universitas Sahid Surakarta.

d. Penguasaan Dasar (Desain Aplikasi, PHP dan MySQL)

Melakukan beberapa percobaan *trial and error* sistem pelayanan wisuda sederhana dengan tujuan agar lebih menguasai bahasa pemrograman PHP dan *database* MySQL sehingga diharapkan dapat menghasilkan hasil yang optimal.

e. Observasi Aplikasi

Mengamati beberapa aplikasi sistem pelayanan wisuda yang sudah ada, baik dari karya ilmiah, buku, atau *internet* yang bisa dijadikan referensi dalam membangun aplikasi sistem pelayanan wisuda.

f. Analisis dan Perancangan Sistem

Menganalisa dan merancang sistem pelayanan wisuda yang akan dibangun seperti apa, bagaimana desainnya, apa saja isinya, sehingga sistem tersebut dapat membantu memecahkan masalah yang ada di Universitas Sahid Surakarta.

g. Implementasi Membangun Sistem Pelayanan Wisuda di Universitas Sahid Surakarta.

a) Implementasi *Database*

Membuat *database* dari data-data yang telah didapatkan sesuai dengan kebutuhan sistem menggunakan MySQL.

b) Implementasi Sistem

Membuat sistem pelayanan wisuda dengan dasar *database* dan perancangan sebelumnya menggunakan bahasa pemrograman PHP.

h. Pengujian Sistem

Pengujian sistem dilakukan bertujuan untuk mengetahui kelayakan sistem pelayanan wisuda apakah sudah memenuhi syarat layak guna atau belum.

i. Penerapan Sistem dan Dokumentasi

Pada tahap akhir ini, sistem pelayanan wisuda diimplementasikan dan digunakan di Universitas Sahid Surakarta dan dibuat dokumentasi dari keseluruhan kegiatan penyusunan Tugas Akhir.

2.3 Teori-Teori Pendukung

Penyusunan Tugas Akhir memerlukan suatu referensi pendukung yang digunakan sebagai landasan teori agar penelitian dapat berjalan dengan benar dan tidak meyimpang dari kaedah ilmu pengetahuan yang ada. Landasan teori diperoleh dari berbagai sumber dan literatur yang mempublikasikan pendapat beberapa ilmuwan yang digunakan sebagai pendukung pembahasan masalah dalam penelitian Tugas Akhir. Berikut ini beberapa diantaranya:

2.3.1 Sistem

Sistem (*system*) dapat didefinisikan dengan pendekatan prosedur dan dengan pendekatan komponen, dengan pendekatan prosedur, sistem dapat didefinisikan sebagai kumpulan dan prosedur-prosedur yang mempunyai tujuan tertentu. Contoh: sistem yang didefinisikan dengan pendekatan prosedur ini adalah sistem akuntansi. Sistem ini didefinisikan sebagai kumpulan dan prosedur-prosedur penerimaan kas, pengeluaran kas, penjualan, pembelian dan buku besar. Dengan pendekatan komponen, sistem dapat didefinisikan sebagai kumpulan dari komponen yang saling berhubungan satu dengan yang lainnya membentuk satu kesatuan untuk mencapai tujuan tertentu. Contoh sistem yang didefinisikan sebagai kumpulan dari program perangkat keras dan perangkat lunak (Jogiyanto, 2003).

2.3.2 Pelayanan

Pelayanan adalah setiap tindakan atau kegiatan yang dapat ditawarkan oleh suatu pihak kepada pihak lain, yang pada dasarnya tidak berwujud dan tidak mengakibatkan kepemilikan apapun (Philip Kotler, 2008). Sedangkan menurut Tjipto, Fandy, & Bregorius Candra (2005) menyatakan bahwa pelayanan merupakan proses yang terdiri atas serangkaian aktivitas *intangibile* yang biasa (namun tidak harus selalu) terjadi pada interaksi antara pelanggan dan karyawan, jasa dan sumber daya, fisik atau barang, dan sistem penyedia jasa, yang disediakan sebagai solusi atas masalah pelanggan. Berdasarkan pengertian-pengertian diatas maka dapat disimpulkan bahwa pelayanan merupakan suatu bentuk sistem, prosedur atau metode tertentu diberikan kepada orang lain, dalam hal ini, kebutuhan

pelanggan tersebut dapat terpenuhi sesuai dengan harapan atau keinginan pelanggan dengan tingkat persepsi mereka.

2.3.3 Wisuda

Menurut Kamus Besar Bahasa Indonesia (KBBI) Edisi III Wisuda merupakan peresmian atau pelantikan yang dilakukan dengan upacara khidmat. Sedangkan menurut Dwi Martha Jingga (2009) wisuda adalah suatu proses pelantikan kelulusan mahasiswa yang telah menempuh masa belajar pada suatu universitas. Proses wisuda biasanya diawali dengan prosesi masuknya rektor dan para pembantu rektor dengan dekan-dekannya guna mewisuda para calon wisudawan dan wisudawati.

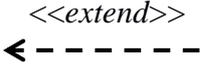
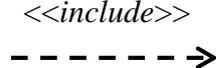
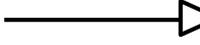
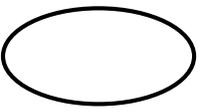
2.3.4 Analisis dan Perancangan Sistem

Analisis sistem didefinisikan bagaimana memahami dan menspesifikasikan dengan detail apa yang harus dilakukan oleh sistem. Sementara sistem desain diartikan sebagai menjelaskan dengan detail bagaimana bagian-bagian dari sistem informasi diimplementasikan, sehingga analisis dan desain sistem informasi (ANSI) bisa didefinisikan sebagai proses organisasional kompleks dimana sistem informasi berbasis komputer diimplementasikan, bisa diringkas *Analysis* mendefinisikan masalah (*From requirements to specification*), *Design* memecahkan masalah (*From specification to implementation*) (Hanif Fattah, 2007). Analisis dan perancangan sistem pada Tugas Akhir ini menggunakan metodologi analisis dan perancangan sistem berorientasi objek yang terdiri dari:

2.3.4.1 Use Case Diagram

Use Case Diagram adalah teknik untuk merekam persyaratan fungsional sebuah sistem. *Use Case* mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan (Martin Fowler, 2005). *Use Case Diagram* dibuat untuk memvisualisasikan atau menggambarkan hubungan antara *Actor* dan *Use Case*. Simbol-simbol *Use Case* ditunjukkan oleh Tabel 2.1.

Tabel 2.1 Simbol-Simbol *Use Case*

No.	Gambar	Nama	Keterangan
1.		<i>Association</i>	Jalur komunikasi antara aktor dan <i>use case</i> berpartisipasi.
2.		<i>Extend</i>	<i>Insertion</i> tambahan ke <i>Use Case</i> yang tidak diketahui.
3.		<i>Include</i>	<i>Insertion</i> tambahan ke <i>Use Case</i> yang secara eksplisit menggambarkan <i>insertion</i> .
4.		<i>Use Case Generalization</i>	Hubungan antara <i>Use Case</i> satu dengan <i>Use Case</i> Lainnya.
5.		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan berinteraksi dengan <i>Use Case</i> .
6.		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.

2.3.4.2 Class Diagram

Class Diagram mendeskripsikan jenis-jenis objek dalam sistem dan berbagi macam hubungan statis yang terdapat diantara mereka. *Class Diagram* juga menunjukkan properti dan operasi sebuah *class* dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut (Martin Fowler, 2005).

Diagram kelas mempunyai 3 macam *relationships* (hubungan) sebagai berikut:

a. *Association*

Suatu hubungan antara bagian dari dua kelas, terjadi *association* antara dua kelas jika salah satu bagian dari kelas mengetahui yang lainnya dalam

melakukan suatu kegiatan. Di dalam diagram, sebuah *association* adalah peghubung yang menghubungkan dua kelas.

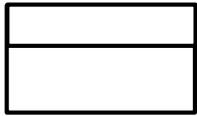
b. *Aggregation*

Suatu *association* dimana salah satu kelasnya merupakan bagian dari suatu kumpulan, *aggregation* memiliki titik pusat yang mencakup keseluruhan bagian sebagai contoh: *Order detail* merupakan kumpulan dari *order*.

c. *Generalization*

Suatu hubungan turunan dengan mengamsusikan satu kelas merupakan satu *super class* (kelas super) dari kelas yang lain. *Generalization* memiliki tingkatan yang berpusat pada *super class*, contohL *Payment* adalah *super class* dari *Cash, Check, Credit*. Simbol *Class Diagram* dapat dilihat pada Tabel 2.2 di bawah ini.

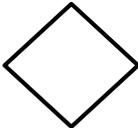
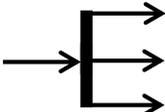
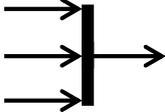
Tabel 2.2 Simbol-Simbol *Class Diagram*

No.	Gambar	Nama	Keterangan
1.		<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>).
2.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
3.		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4.		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
5.		<i>Depedency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.

2.3.4.3 Activity Diagram

Activity Diagram adalah teknik untuk menggambar logika prosedural, proses bisnis, dan jalur kerja, dalam beberapa hal diagram ini memainkan peran mirip sebuah diagram alir, tetapi perbedaan prinsip antara diagram ini dan notasi diagram alir adalah diagram ini mendukung *behavior* paralel (Martin Fowler, 2005). Adapun simbol-simbol yang digunakan dalam *activity diagram* ditunjukkan oleh Tabel 2.3.

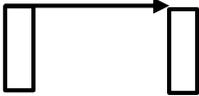
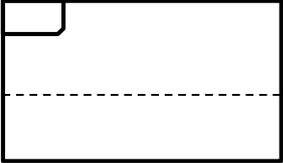
Tabel 2.3 Simbol-Simbol *Activity Diagram*

No.	Gambar	Nama	Keterangan
1.		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2.		<i>Decision Node</i>	Suatu <i>point</i> pada <i>activity diagram</i> yang mengindikasikan suatu kondisi ada perbedaan transisi.
3.		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4.		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan.
5.		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.
6.		<i>Join Node</i>	Beberapa aliran masukan tertentu berubah menjadi satu aliran.

2.3.4.4 Sequence Diagram

Sequence Diagram adalah penjabaran *behavior* sebuah skenario tunggal. *Sequence Diagram* menunjukkan sejumlah objek contoh dan pesan-pesan yang melewati objek-objek ini di dalam *Use Case* (Martin Fowler, 2005). Sedangkan simbol-simbol *Sequence Diagram* dapat dilihat pada Tabel 2.4.

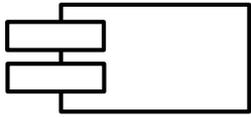
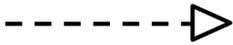
Tabel 2.4 Simbol-Simbol *Sequence Diagram*

No.	Gambar	Nama	Keterangan
1.		<i>Life Line</i>	Objek <i>entity</i> antarmuka yang saling berinteraksi.
2.		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
3.		<i>Actor</i>	Pengguna di luar sistem.
4.		<i>Fragment</i>	Menggambarkan batas grafis suatu diagram.

2.3.4.5 Component Diagram

Component Diagram merupakan bagian fisik dari sebuah sistem, karena menetap di komputer tidak berada dianalisis. *Component* terhubung melalui antar muka yang digunakan dan dibutuhkan (Martin Flower, 2005). Simbol *Component Diagram* dapat dilihat pada Tabel 2.5.

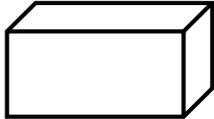
Tabel 2.5 Simbol-Simbol *Component Diagram*

No.	Gambar	Nama	Keterangan
1.		<i>Component</i>	<i>Physical</i> dari sebuah sistem.
2.		<i>Interface</i>	A <i>name state</i> informasi yang menjadi ciri dari perilaku.
3.		<i>Usage</i>	Situasi dimana satu elemen membutuhkan lain fungsi yang benar.
4.		<i>Realization</i>	Hubungan antara spesifikasi dan implementasinya.

2.3.4.6 *Deployment Diagram*

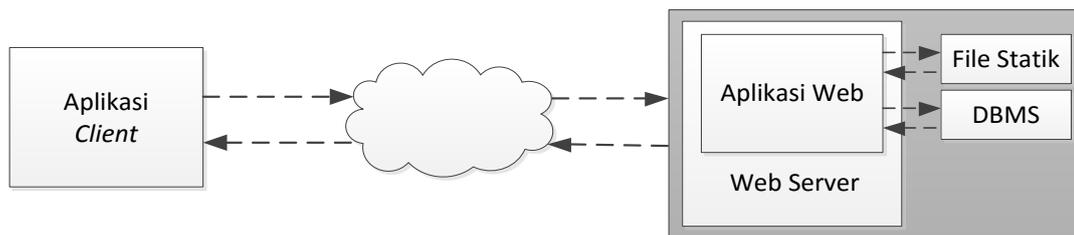
Deployment Diagram menunjukkan susunan fisik sebuah sistem, menunjukkan bagian perangkat lunak mana yang berjalan pada perangkat keras mana (Martin Fowler, 2005). *Deployment Diagram* juga menggambarkan tata letak sebuah sistem secara fisik, menampakkan bagian-bagian *software* yang berjalan pada bagian-bagian *hardware*, menunjukkan hubungan komputer dengan perangkat (*nodes*) satu sama lain dan jenis hubungannya. Simbol-simbol *Deployment Diagram* ditunjukkan oleh Tabel 2.6 berikut.

Tabel 2.6 Simbol-Simbol *Deployment Diagram*

No.	Gambar	Nama	Keterangan
1.		<i>Node</i>	Biasanya mengacu pada perangkat keras (<i>hardware</i>) dan perangkat lunak (<i>software</i>), jika di dalam <i>node</i> di sertakan komponen untuk menkonsistensikan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang sudah didefinisikan sebelumnya pada diagram komponen.
2.		<i>Dependency</i>	Sebuah <i>dependency</i> digunakan untuk me-notasikan relasi antara dua <i>component</i> .
3.		<i>Link</i>	Relasi antar <i>node</i> .

2.3.5 Aplikasi Web

Aplikasi *web* adalah aplikasi yang disimpan dan dieksekusi di lingkungan *web server*. Setiap permintaan yang dilakukan oleh *user* melalui aplikasi klien (*web browser*) akan direspon oleh aplikasi *web* dan hasilnya akan dikembalikan lagi ke hadapan *user*. Dengan aplikasi *web*, halaman yang tampil di layar *web browser* dapat bersifat dinamis, tergantung dari nilai data atau parameter yang dimasukkan oleh *user*. Komunikasi antara *web browser* dan aplikasi *web* dapat digambarkan pada Gambar 2.5.



Gambar 2.5 Gambaran Global Aplikasi Web

Seperti yang tampak pada gambar di atas bahwa aplikasi *web* dapat juga digunakan untuk mengakses *file-file* yang bersifat statis (misal: dokumen HTML, *file* gambar maupun *file* teks) (Raharjo dkk., 2010).

2.3.6 PHP

PHP adalah salah satu bahasa pemrograman skrip yang dirancang untuk membangun aplikasi *web*. Ketika dipanggil dari *web browser*, program yang ditulis dengan PHP akan di-*diparsing* di dalam *web server* oleh *interpreter* PHP dan diterjemahkan ke dalam dokumen HTML, yang selanjutnya akan ditampilkan kembali ke *web browser*. Karena pemrosesan program PHP dilakukan di lingkungan *web server*, PHP dikatakan sebagai sisi *server* (*server-side*). Oleh sebab itu, seperti yang telah dikemukakan sebelumnya, kode PHP tidak akan terlihat pada saat *user* memilih perintah “View Source” pada *web browser* yang mereka gunakan. Selain PHP, aplikasi *web* juga dapat digunakan dengan Java (JSP-*JavaServer Pages dan Servlet*), Perl, maupun ASP (Raharjo dkk., 2010).

2.3.7 MySQL

MySQL adalah *multiuser database* yang menggunakan bahasa *Structured Query Language* (SQL). MySQL dalam operasi *client-server* melibatkan *server daemon* MySQL di sisi *server* dan berbagai macam program serta *library* yang berjalan di sisi *client*. MySQL Mampu menangani data yang cukup besar. Perusahaan yang mengembangkan MySQL yaitu TcX, mengaku mampu menyimpan data lebih dari 40 *database*, 10.000 tabel dan sekitar 7 juta baris, totalnya kurang lebih 100 Gygabyte data (Sunarfrihantono, 2002).

2.3.8 HTML

HTML (*Hypertext Markup Language*) merupakan salah satu format yang digunakan dalam pembuatan dokumen dan aplikasi yang berjalan di halaman *web*. Oleh karena itu agar dapat membuat program aplikasi di atas halaman *web*, kita terlebih dahulu harus mengenal dan menguasai HTML. Walaupun sekarang telah banyak paket secara WYSIWYG (*What You See is What You Get*) seperti Frontpage, DreamWeaver, Netscape Composer, dan masih banyak lagi, namun kita tetap harus menguasai *tag-tag* HTML terutama yang dipergunakan untuk membuat aplikasi di internet karena mau tidak mau kita akan bekerja dalam mode *text* editor bilamana hendak menyisipkan setiap *script* program dalam *script* HTML (Sunarfrihantono, 2002).

2.3.9 CSS (*Cascading Style Sheet*)

CSS atau *Cascading Style Sheet* adalah suatu fasilitas untuk mempermudah pemeliharaan sebuah halaman *web*, dengan menggunakan CSS sebuah halaman *web* dapat diubah tampilannya tanpa harus mengubah dokumen HTML-nya (Shalahuddin dan Rosa A. S., 2008).

2.3.10 jQuery

jQuery adalah *Javascript Library* yaitu kumpulan kode/fungsi Javascript siap pakai, sehingga mempermudah dan mempercepat kita dalam membuat kode Javascript, termasuk dalam membuat kode AJAX (Hakim, 2011).

2.3.11 phpMyAdmin

phpMyAdmin disebut juga sebagai *tools* yang digunakan untuk mengakses *database* MySQL dalam bentuk tampilan *web*. *Tools* ini secara standar disertakan ketika menginstal XAMPP. Dengan adanya phpMyAdmin, semua pekerjaan akan menjadi lebih mudah, Karena sudah dapat memajemen *database* dan data yang ada didalamnya, selain itu juga dapat menjadi administrator dengan mudah (Nugroho, 2008).

2.3.12 Paket XAMPP

XAMPP merupakan paket PHP dan MySQL berbasis open source, yang dapat digunakan sebagai tool pembantu pengembangan aplikasi berbasis PHP. XAMPP mengkombinasikan beberapa paket perangkat lunak berbeda ke dalam satu paket (Riyanto, 2011).

2.3.13 *WebQual*

WebQual merupakan salah satu metode atau teknik pengukuran kualitas *website* berdasarkan persepsi pengguna akhir. Metode ini merupakan pengembangan dari *SERVQUAL* (Zeithaml et al. 1990) yang banyak digunakan sebelumnya pada pengukuran kualitas jasa. Instrumen penelitian *WebQual* tersebut dikembangkan dengan metode *Quality Function Development (QFD)* (Barnes & Vidgen, 2002). *WebQual* sudah mulai dikembangkan sejak tahun 1998 dan telah mengalami beberapa iterasi dalam penyusunan dimensi dan butir-butir pertanyaannya. Menurut teori *WebQual*, terdapat tiga dimensi yang mewakili kualitas suatu *website*, yaitu kegunaan (*usability quality*), kualitas informasi (*information quality*) dan interaksi layanan (*service interacton*).

2.3.13.1 Dimensi *WebQual*

1. Kualitas Informasi

Menurut Barnes (2003), kualitas informasi meliputi hal-hal seperti informasi yang akurat, informasi yang bisa dipercaya, informasi yang *up to date*/terbaru, informasi yang sangat detil, dan informasi yang disajikan dalam format desain yang sesuai.

2. Kualitas Interaksi

Kualitas interaksi meliputi kemampuan memberi rasa aman saat interaksi, memiliki reputasi yang bagus, memudahkan komunikasi, menciptakan perasaan emosional yang lebih personal, memiliki kepercayaan dalam menyimpan informasi pribadi pengguna, mampu menciptakan komunitas yang lebih spesifik, mampu member keyakinan bahwa janji yang disampaikan ditepati.

3. Kualitas Penggunaan

Kualitas pengguna meliputi kemudahan untuk dipelajari, kemudahan untuk dimengerti, kemudahan untuk ditelusuri, kemudahan untuk digunakan, sangat menarik, menampilkan bentuk visual yang menyenangkan, memiliki kompetensi yang baik, memberikan pengalaman baru yang menyenangkan.

Persepsi pengguna tersebut terdiri dari dua bagian, yaitu persepsi tentang mutu layanan yang dirasakan (aktual) dengan tingkat harapan (ideal). Barnes & Vidgen melakukan penelitian dengan menggunakan *WebQual* untuk mengukur kualitas *website* yang dikelola oleh OECD (*organization for Economic Cooperation and Development*). *Website* yang bermutu dari perspektif pengguna dapat dilihat dari tingkat persepsi layanan aktual yang tinggi dan kesenjangan persepsi aktual dengan ideal (*gap*) yang rendah.