

BAB 2

TINJAUAN PUSTAKA

2.1 Landasan Teori

2.1.1 Multimedia

2.1.1.1 Pengertian Multimedia

Multimedia adalah kombinasi dari teks, grafik, suara, animasi, dan video yang dikemas oleh komputer atau peralatan elektronik lainnya (Vaughan,2011:1).

2.1.1.2 Multimedia Interaktif

Menurut Vaughan (2011:1), suatu karya atau proyek multimedia yang memungkinkan *end user* atau penggunanya untuk mengendalikan penyampaian dari satu atau lebih elemennya disebut dengan multimedia interaktif.

Menurut Vaughan (2011:56), sebuah perusahaan yang mempresentasikan produk dengan cara menggunakan multimedia interaktif dapat menyampaikan informasi dengan lengkap dan daya jual dalam jangkauan pengguna, termasuk informasi latar belakang, materi pemasaran, statistik harga, dan data teknis.

2.1.1.3 Elemen-Elemen Multimedia

Vaughan (2011:1-164) menjelaskan elemen-elemen yang terdapat dalam multimedia adalah:

a. Teks

Setiap kata dapat mengandung banyak arti. Dalam penggunaan teks yang terpenting adalah mengembangkan akurasi dan keringkasn dalam kata-kata tertentu yang dipilih. Didalam multimedia, teks merupakan kata-kata

yang akan digunakan pada judul, menu, dan alat bantu navigasi serta dalam naratif atau konten.

b. Gambar

Gambar dalam multimedia dibagi menjadi dua jenis, yaitu *bitmaps* dan *vector-drawn*:

1. *Bitmaps*: digunakan untuk menggambarkan objek secara realistis dan menggunakan banyak detail.
2. *Vector-drawn*: digunakan untuk menggambarkan objek seperti garis, kotak, bulat, poligon dan bentuk grafis lainnya yang secara matematis dapat menyatakan sudut, koordinat, dan jarak.

c. Suara

Suara merupakan unsur berbicara yang penting pada setiap bahasa dan menjadi salah satu elemen multimedia yang paling menarik panca indera karena suara dapat memberikan kenyamanan dalam mendengarkan musik, membuat efek suara mengejutkan, dan penggunaan musik latar belakang yang dapat meningkatkan *mood*. Beberapa musik yang enak didengar dapat menyentuh hati, menciptakan suasana nyaman kepada pendengar.

d. Animasi

Animasi dapat membuat presentasi statis menjadi nyata. Perubahan visual dari waktu ke waktu dan dapat menambah kualitas proyek multimedia dan halaman *web*. Animasi dapat terjadi karena adanya suatu fenomena biologis yang dinamakan *persistence of vision* dan sebuah fenomena secara psikologis yang disebut *phi*. Sebuah objek yang terlihat oleh mata manusia terekam untuk beberapa saat dan terpetakan dalam retina mata sesaat setelah melihat. Hal itu digabungkan dengan pikiran manusia yang secara konseptual melengkapi sebuah tindakan atau gerakan. Kedua hal inilah yang menyebabkan sebuah deretan gambar yang berubah secara sedikit namun berubah secara sangat cepat antara gambar

satu dengan yang lainnya terlihat seperti menyatu membentuk suatu ilusi dari gerakan.

e. Video

Digital video merupakan elemen multimedia yang paling menarik. Digital video adalah alat yang mampu membawa pengguna komputer lebih dekat ke dunia nyata. Digital video juga merupakan metode yang sangat bagus dalam menyampaikan pesan kepada pemirsa melalui televisi.

2.1.2 Interaksi Manusia dan Komputer

2.1.2.1 Lima Faktor Manusia Terukur

Terdapat lima faktor manusia terukur yang penting dalam perancangan antarmuka (Shneiderman, 2010:32), yaitu:

a. Waktu Belajar

Perancangan harus memperhatikan rata-rata waktu yang dibutuhkan oleh golongan pengguna yang dituju untuk mempelajari aksi-aksi atau tindakan yang diperlukan untuk melakukan suatu operasi. Perancangan yang baik tidak membutuhkan waktu pembelajaran yang terlalu lama.

b. Kecepatan Pelaksanaan

Perancangan harus memperhatikan jumlah waktu yang dibutuhkan untuk memulai dan menyelesaikan suatu aksi. Perancangan yang baik harus memungkinkan pelaksanaan aksi secara singkat dan jelas.

c. Tingkat Kesalahan Pengguna

Perancangan harus memperhatikan jenis kesalahan dan seberapa sering kesalahan tersebut terjadi. Penanganan terhadap *error* yang terjadi akibat kesalahan pengguna harus tersedia dan mampu membantu pengguna.

d. Daya Ingat

Perancangan harus memperhatikan tingkat ingatan pengguna, dikaitkan dengan frekuensi pemakaian pengguna.

Perancangan yang baik akan mudah diingat dan tidak membutuhkan pembelajaran ulang secara ekstensif.

e. Kepuasan Subjektif

Perancangan harus memperhatikan tingkat kepuasan pengguna terhadap komponen *interface* yang ada. Kepuasan dapat diukur melalui wawancara atau survei.

2.1.2.2 8 aturan emas

Terdapat delapan aturan emas yang dijadikan pedoman dalam perancangan antarmuka pengguna yang baik (Shneiderman, 2010:88), yaitu:

a. Konsistensi

Perancangan harus mengutamakan konsistensi atau keseragaman komponen yang dapat digolongkan ke dalam satu golongan. Perancangan yang baik memiliki keseragaman dalam berbagai hal yang muncul atau dilakukan berulang kali, contohnya cara penulisan terminologi atau istilah.

b. Kegunaan yang *Universal*

Perancangan harus mengutamakan kepentingan pengguna secara menyeluruh. Perancangan yang baik mampu melayani kebutuhan berbagai jenis pengguna dari berbagai latar belakang dan kondisi, termasuk mampu menangani perbedaan yang ada mulai dari tingkatan pengguna (pemula atau tingkat atas) sampai dengan perbedaan *hardware* atau peralatan.

c. Memberikan umpan balik yang informatif

Perancangan harus memperhatikan adanya tanggapan dari sistem untuk aksi yang dilakukan oleh pengguna. Perancangan yang baik akan memberikan informasi yang berkaitan terhadap aksi yang dilakukan pengguna, yang juga berguna untuk meningkatkan pengetahuan dan efisiensi aktifitas pengguna.

d. Merancang dialog untuk suatu penutupan

Perancangan harus memberikan tahapan awal, tengah, dan akhir untuk setiap kelompok aksi yang dilakukan pengguna. Perancangan yang baik akan memungkinkan pengguna mengetahui kapan sesuatu mulai dikerjakan, sedang dikerjakan, dan kapan aksi tersebut diakhiri.

e. Memberikan penanganan kesalahan yang sederhana

Perancangan yang baik harus mampu mencegah terjadi kesalahan baik secara teknis oleh sistem maupun oleh pengguna. Perancangan yang baik harus mampu mendeteksi kesalahan yang terjadi dan memberi solusi untuk menangani kesalahan tersebut dengan efektif.

f. Mudah kembali ke tindakan sebelumnya

Perancangan yang baik harus memungkinkan pengguna untuk membatalkan atau mengulang suatu tindakan yang dilakukan.. Perancangan yang baik memungkinkan pengguna untuk mencoba suatu aksi tanpa perlu ada rasa takut merusak sistem.

g. Mendukung tempat pengendali internal

Perancangan yang baik harus memberikan kuasa atau kendali pada pengguna. Sistem harus dapat mengikuti perintah pengguna dan memberikan tanggapan yang dapat diproses oleh pengguna.

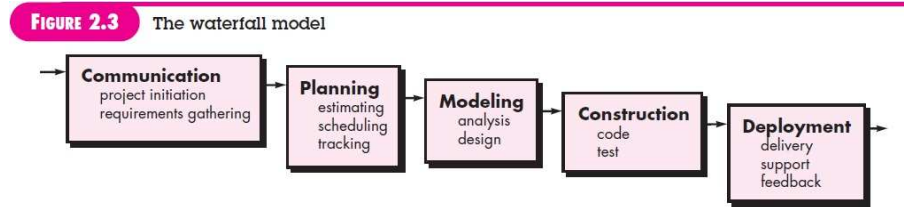
h. Mengurangi beban ingatan jangka pendek

Perancangan harus mengutamakan tampilan yang tidak membutuhkan daya ingat yang berlebihan. Perancangan yang baik akan mudah diingat oleh pengguna tanpa harus digunakan berulang kali.

2.1.3 System Development Life-Cycle (The Waterfall Model)

Menurut Pressman (2010:39), *The Waterfall Model* atau dikenal dengan sebutan *classic life cycle*, adalah sebuah sistematika pendekatan bertahap dalam pengembangan perangkat lunak yang dimulai dengan

spesifikasi kebutuhan pengguna dan dikembangkan melalui *planning*, *modeling*, *construction* dan *deployment*.



Gambar 2.1 The Waterfall Model

(Sumber : *Software Engineering A Practitioner's Approach*-
Pressman, 2010:39)

Menurut Pressman (2010:15), *The Waterfall Model* terdiri dari lima tahapan, yaitu *Communication*, *Planning*, *Modeling*, *Construction*, dan *Deployment*.

1. *Communication*

Tahap ini merupakan tahapan awal dari *The Waterfall Model*, yaitu memulai proyek tersebut dengan melakukan komunikasi terhadap pengguna agar dapat mengetahui kebutuhan user (*user requirements*) dan kebutuhan sistem (*system requirements*).

2. *Planning*

Pada tahap ini, dilakukan perencanaan suatu perangkat lunak yang akan dikembangkan, yaitu dengan memperkirakan waktu yang diperlukan dan juga penjadwalan untuk menyelesaikan pembuatan perangkat lunak tersebut.

3. *Modeling*

Pada tahap ini, dilakukan perancangan model sesuai kebutuhan pengguna ke perancangan perangkat lunak tersebut sebelum melakukan pemrograman.

4. Construction

Pada tahap ini, dilakukan proses pemrograman atau *coding* dan *testing* terhadap perangkat lunak atau sistem. *Testing* dilakukan setelah pemrograman selesai untuk mengurangi kesalahan atau *error*.

5. Deployment

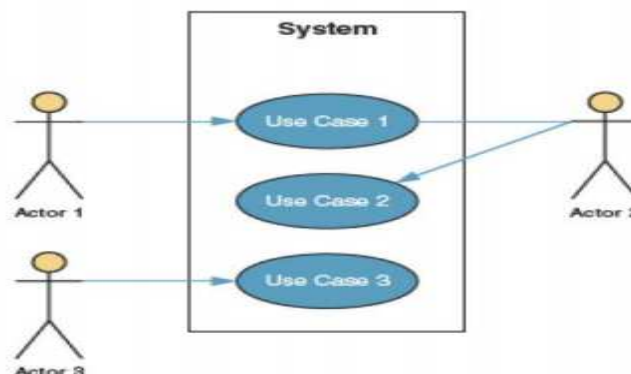
Pada tahap terakhir ini, Perangkat lunak yang sudah jadi akan diberikan kepada pengguna dengan tujuan untuk melakukan evaluasi hasil kerja dan memperoleh kritik dan saran dari evaluasi.

2.1.4 UML (Unified Modeling Language)

Menurut Whitten dan Bentley (2007:371), *Unified Modeling Language* (UML) adalah sebuah set dari model konvensi yang digunakan untuk spesifikasi atau menggambarkan sebuah sistem perangkat lunak dalam hal objek. UML dapat dibagi menjadi :

2.1.4.1 Use Case Diagram

Use case diagram adalah diagram yang menggambarkan interaksi antara sistem, sistem eksternal dan pengguna. Dengan kata lain, secara grafis menggambarkan siapa yang akan menggunakan sistem dan dalam cara apa pengguna berinteraksi dengan sistem (Whitten dan Bentley, 2007: 246).



Gambar 2.2 Use Case Diagram

(Sumber: *System Analysis and Design Methods* -
Whitten and Bentley, 2007)

Komponen-komponen penting dalam *use case* (Whitten and Bentley, 2007:247) yaitu:

1. Aktor

Aktor adalah apapun yang yang perlu untuk berinteraksi dengan sistem untuk pertukaran informasi (Whitten dan Bentley, 2007: 247). Ada 4 macam aktor yaitu:

a. Aktor bisnis utama (*Primary bussiness actor*)

Stakeholder yang menerima keuntungan dari pengekseskusion *use case* dengan menerima sesuatu yang dapat dihitung atau nilai yang jelas.

b. Aktor sistem utama (*Primary system actor*)

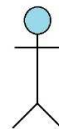
Stakeholder yang secara langsung memicu kejadian dalam suatu sistem.

c. Aktor server luar (*External server actor*)

Stakeholder yang merespon terhadap sebuah permintaan dari *use case*.

d. Aktor penerima luar (*External receiver actor*)

Stakeholder yang tidak termasuk aktor utama tetapi menerima sesuaatu yang dapat dihitung atau memiliki nilai yang jelas dari *use case*.



Gambar 2.3 Aktor

(Sumber: *System Analysis and Design Methods* - Whitten and Bentley, 2007)

2. *Use case*

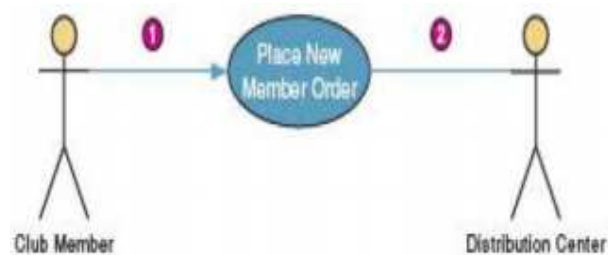
Use cases adalah urutan langkah-langkah perilaku, baik otomatis maupun manual dengan tujuan menyelesaikan 1 buah tugas tunggal. Use cases menggambarkan fungsi perspektif pengguna eksternal dan dengan cara teknologi yang dapat dimengerti pengguna (Whitten dan Bentley, 2007: 246).

3. Relasi

Suatu hubungan digambarkan sebagai garis antara dua simbol pada diagram *use cases*. Arti dari hubungan ini dapat berbeda tergantung pada bagaimana garis ditarik dan apa jenis simbol yang menghubungkan mereka (Whitten dan Bentley, 2007: 248). Ada 5 macam jenis relasi yaitu :

a. Asosiasi

Hubungan antara aktor dan *use case* dimana terjadi interaksi diantara mereka.

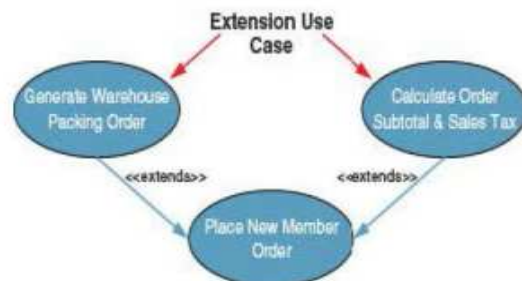


Gambar 2.4 Asosiasi

(Sumber: *System Analysis and Design Methods* - Whitten and Bentley, 2007)

b. Extends

Hubungan yang terdiri dari langkah-langkah yang diekstraksi dari use cases yang lebih kompleks untuk menyederhanakan kasus awal dan memperluas fungsinya.



Gambar 2.5 Extends

(Sumber: *System Analysis and Design Methods* - Whitten and Bentley, 2007)

c. *Uses / Includes*

Uses cases yang mengurangi redudansi antara dua atau lebih *uses cases* dengan menggabungkan langkah-langkah umum yang ditemukan didalam masalah tersebut.

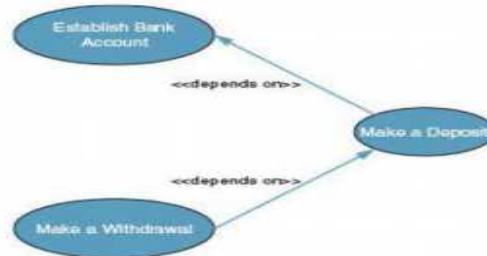


Gambar 2.6 Uses Relationship

(Sumber: *System Analysis and Design Methods* - Whitten and Bentley,2007)

d. *Depends On*

Hubungan antara use cases yang menunjukkan bahwa satu use cases tidak bisa dijalankan sampai use cases lain telah dijalankan.

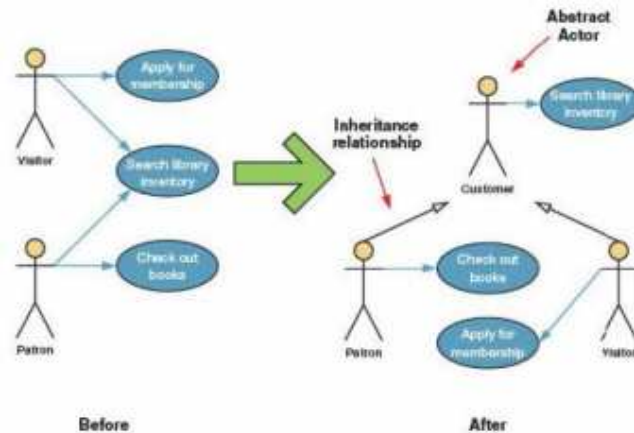


Gambar 2.7 Depends on Relationship

(Sumber: *System Analysis and Design Methods* - Whitten and Bentley,2007)

e. *Inheritance*

Hubungan antara aktor-aktor yang dibuat untuk menyederhanakan gambar ketika seorang aktor abstrak mewarisi beberapa aktor nyata.



Gambar 2.8 Inheritance Relationship

(Sumber: *System Analysis and Design Methods* - Whitten and Bentley, 2007)

2.1.4.2 Activity Diagram

Menurut Whitten dan Bentley (2007:390-391), *Activity diagram* adalah diagram grafis untuk menggambarkan aliran proses bisnis, langkah-langkah dari use cases atau logika perilaku objek.

Notasi-notasi yang terdapat dalam *activity diagram* yaitu :

- Inisial *node* (*initial node*)

Notasi yang digunakan untuk memulai sebuah urutan proses interaksi. Notasi ini dilambangkan dengan lingkaran berisi warna hitam untuk *node*.

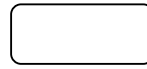


Gambar 2.9 Inisial Node

- Aksi (*actions*)

Notasi yang digunakan untuk menjelaskan satu per satu langkah untuk suatu proses interaksi. Notasi

ini dilambangkan dengan bangun segi empat yang tumpul.



Gambar 2.10 Aksi

- Alur (*flow*)

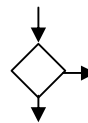
Notasi yang digunakan untuk pergerakan melalui actions. Notasi ini dilambangkan dengan anak panah.



Gambar 2.11 Alur

- Keputusan (*decision*)

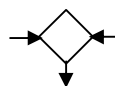
Notasi yang digunakan jika terdapat dua atau lebih kondisi. Notasi ini dilambangkan dengan belah ketupat yang terdiri atas satu *flow* yang masuk dan terdapat dua atau lebih *flow* yang keluar dari notasi ini.



Gambar 2.12 Keputusan

- Gabungan (*merge*)

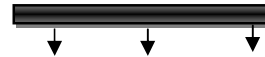
Notasi yang digunakan untuk menyatukan dua *actions* atau lebih. Notasi yang dilambangkan dengan belah ketupat.



Gambar 2.13 Gabungan

- Percabangan (*fork*)

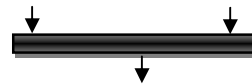
Notasi yang hampir sama dengan decision, perbedaannya fork menggambarkan dua proses berbeda tetapi dijalankan secara bersamaan.



Gambar 2.14 Percabangan

- Join

Notasi yang digunakan untuk menggambarkan akhir dari proses fork untuk dilanjutkan ke alur berikutnya. Notasi ini dilambangkan dengan batang hitam dengan dua atau lebih *flow* yang masuk dan satu *flow* yang keluar.



Gambar 2.15 Join

- Aktivitas akhir (*activity final*)

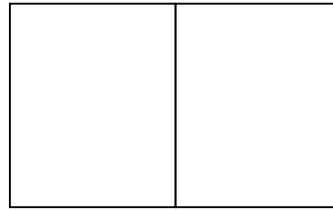
Notasi yang digunakan untuk menandakan akhir proses. Notasi ini dilambangkan dengan lingkaran penuh bagian dalam dan berongga dibagian luar.



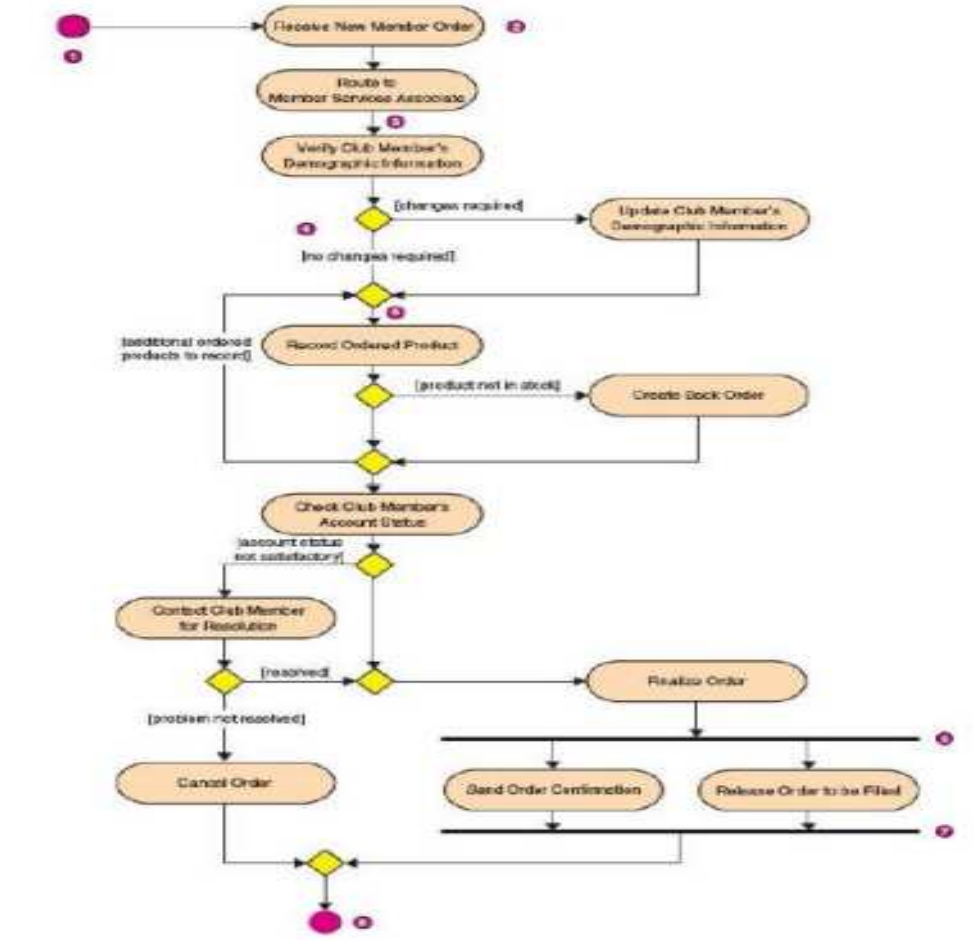
Gambar 2.16 Aktivitas Akhir

- Swimlanes

Swim lanes membagi *activity diagram* menjadi beberapa partisi yang menunjukkan tindakan yang dilakukan oleh kelas tertentu atau aktor.



Gambar 2.17 Swimlanes



Gambar 2.18 Activity Diagram

(Sumber: *System Analysis and Design Methods-Whitten and Bentley*, 2007)

2.1.4.3 Class Diagram

Class Diagram adalah gambaran grafis dari struktur objek statis sistem, menunjukkan kelas objek bahwa sistem terdiri dari hubungan antara kelas-kelas objek (Whitten and Bentley, 2007:400).

2.1.5 *Fact Finding*

Menurut Connolly dan Begg (2010:341-347), *fact finding* adalah proses formal menggunakan teknik seperti wawancara dan kuesioner untuk mengumpulkan fakta tentang sistem, kebutuhan, dan pilihan. Ada lima *fact finding* yang digunakan:

- Menguji Dokumentasi

Uji dokumentasi bermanfaat jika kita sedang berusaha mendalami kebutuhan basis data yang akan datang.

- Wawancara

Teknik ini paling sering digunakan dan paling berguna dalam pencarian fakta. Wawancara terdiri dari 2 jenis, yaitu:

1. Wawancara tidak terstruktur

Wawancara tidak terstruktur dilakukan jika tujuan wawancara bersifat umum dan memiliki sedikit pertanyaan yang bersifat spesifik.

2. Wawancara terstruktur

Wawancara terstruktur dilakukan jika tujuan pewawancara memiliki banyak pernyataan yang bersifat spesifik.

- Observasi

Observasi adalah salah satu teknik pencarian fakta yang paling efektif untuk pemahaman suatu sistem. Dengan cara melakukan pengamatan langsung terhadap suatu sistem.

- Riset

Riset aplikasi dan masalah, jurnal komputer, buku referensi, dan internet seperti sekelompok pengguna dan *bulletin board* adalah sumber informasi yang baik dan dapat menyediakan informasi mengenai bagaimana orang lain yang telah memecahkan masalah serupa.

- Kuesioner

Kuesioner adalah teknik pencarian fakta dengan melakukan survei melalui daftar pertanyaan, Terdapat 2 jenis pertanyaan dalam kuisisioner:

- *Free Format*

Free Format memberikan kebebasan responden untuk menjawab pertanyaan.

- *Fix Format*

Fix Format memerlukan tanggapan spesifik dari individu, responden hanya dapat memilih jawaban yang disediakan.

2.1.6 Internal and External View of Testing

Menurut Pressman (2007:484), setiap *engineered product* (dan banyak hal lainnya) dapat diuji dalam satu dari dua cara, yaitu:

1. Mencoba apakah setiap fungsi berkerja dengan baik dengan cara mencari error dalam setiap fungsi.
2. Mengetahui cara kerja internal suatu produk.

Uji coba pertama dapat dilakukan dengan mengambil pandangan eksternal, hal ini disebut *Black-Box testing*, sedangkan untuk pendekatan yang kedua membutuhkan pandangan internal, hal ini disebut *White-Box testing* :

2.1.6.1 White-Box Testing

Menurut Pressman (2007:485), *White-Box Testing* adalah metode desain *test case* yang menggunakan struktur kontrol desain prosedural untuk memperoleh *test case*. Dengan menggunakan metode pengujian ini akan didapatkan *test case* yang :

1. Memberikan jaminan bahwa semua jalur independen pada suatu modul telah digunakan paling tidak satu kali.
2. Menggunakan semua keputusan logis pada sisi *true* dan *false*.
3. Mengeksekusi semua *looping* pada batasan tertentu.
4. Dan menggunakan struktur data internal yang menjamin validitasnya.

2.1.6.2 Black-Box Testing

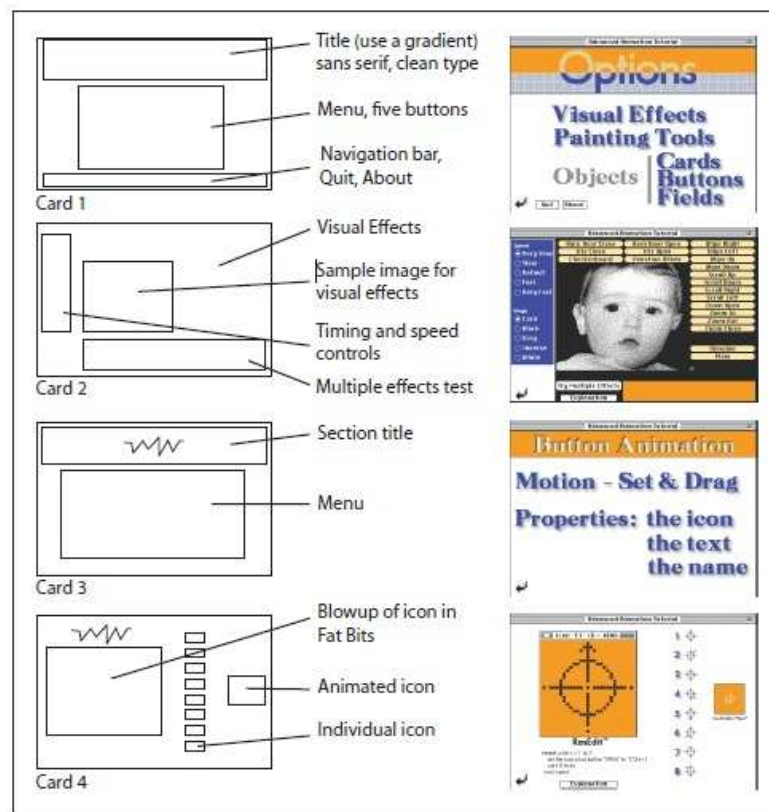
Menurut Pressman (2007:495), *Black-Box Testing* adalah metode pengujian yang berfokus pada persyaratan fungsional

perangkat lunak. Pengujian ini berusaha menemukan kesalahan dalam kategori sebagai berikut:

1. Fungsi - fungsi yang tidak benar atau hilang.
2. Kesalahan *interface*.
3. Kesalahan dalam struktur data atau akses *database external*.
4. Kesalahan kinerja.

2.1.7 Storyboard

Menurut Vaughan (2011:183), *storyboard* seperti sebuah komik sekuensial yang dibaca sehari-hari. Setiap hari ada tiga atau empat panel yang menunjukkan perkembangan cerita atau informasi. Memerlukan waktu untuk menstruktur produksi dengan cara mencatatnya dan kemudian membuat sekelompok gambar berurutan yang menunjukkan kamera dan *scene*, sudut pengambilan gambar, pencahayaan (*lightning*), gerakan, *special effects*, dan bagaimana cara objek bergerak dari awal sampai akhir.



Gambar 2.19 Contoh Storyboard (kiri)

(Sumber: *Making It Work*-Vaughan,2011)

2.1.8 Adobe flash

Menurut Gerantabee and Agi Creative Team (2012:10) Adobe Flash adalah aplikasi *vector drawing* asli di mana Anda akan membuat kaya, rinci, dan *scalable* ilustrasi digital. Menurut Todd Perkins (2010:359) Terdapat 3 metode dalam pembuatan animasi menggunakan Flash:

a. *Frame-by-frame animation*

Merupakan teknik yang dicapai dengan manual dengan cara mengubah isi masing-masing dari *keyframes* berturut-turut sehingga terbentuk sebuah animasi.

b. *Keyframe-based tweened animation*

Merupakan teknik dengan cara mengatur dari titik awal dan titik akhir dari animasi (dengan *keyframes*) dan memungkinkan Flash untuk interpolasi di antara isi frame.

c. *Object-based motion tweening*

Merupakan evolusi yang menakjubkan dari *keyframes-based tweened animation* yang berbasis pada gerakan *tweening*.

2.1.7.1 Fitur Penting

Menurut Gerantabee and Agi Creative Team (2012:10-12) ada empat bidang fitur kunci pada Adobe Flash, yaitu:

1. *Drawing Environment* (Menggambar lingkungan)

Flash memiliki satu set lengkap alat gambar untuk menangani ilustrasi rumit dan tipografi. Seperti Adobe Illustrator CS6 *Flash* mendukung Illustrator dan Photoshop *files* untuk ruang kerja mereka seperti format asli *file* Ai dan PSD, sehingga semua konten yang anda buat di *Flash* atau program-program lainnya dapat dibawa ke kehidupan melalui animasi dan interaktivitas.

2. *Animation* (Animasi)

Flash menciptakan sebuah fitur ringan untuk membuat animasi dengan menggunakan gambar, suara, dan video. Animasi *Flash* dapat dimuat di situs *web* dan jaringan sosial, animasi merupakan alat utama untuk menonjolkan sisi interaktif pada halaman iklan berbasis *web*.

3. *Layout*

Halaman kerja pada Flash memberikan fleksibilitas untuk membuat tata letak halaman situs yang luar biasa dan hanya terbatas oleh kreativitas pengguna, anda dapat mengubah posisi konten kemana saja dengan fleksible dan presisi, tata letak dapat dikaitkan dengan halaman *web* tradisional.

4. *Programming*

Adobe Flash merupakan program yang dapat digunakan seluruh kalangan pengguna komputer dengan didukung oleh banyak *plug-in* yang dapat di unduh di *internet*. Dengan adanya ActionScript dapat mengontrol pemutaran film atau menambahkan fungsionalitas ke tombol namun dengan ActionScript juga dapat membuat aplikasi yang interaktif.

2.1.9 XML

Menurut Fawcett, Liam R.E., dan Danny Ayers (2012:3-12), XML atau eXtensible Markup Language merupakan bahasa dengan dua kegunaan utama, yang pertama adalah untuk merepresentasikan *low-level data*, contohnya file konfigurasi. Yang kedua adalah untuk menambahkan metadata ke dokumen, contohnya penekanan pada kalimat dalam laporan dengan membuat tulisan cetak miring atau tebal. XML juga bersifat interoperabilitas, artinya mampu berinteraksi dan berfungsi dengan sistem lain.

2.1.10 Adobe Photoshop

Menurut Smith and the Agi Creative Team (2012:31) Adobe Photoshop adalah program *editing* gambar yang dapat membuka gambar yang diambil oleh *scanner* atau kamera digital, atau di unduh dari *Web*. Hal ini juga dapat terbuka video yg diambil gambar dan ilustrasi vektor. Selain itu, Anda dapat membuat dokumen baru di Photoshop, termasuk grafis vektor, yang *scalable image files* (misalnya, gambar bisa diperbesar atau diperkecil ukurannya tanpa kehilangan kejelasan).

2.1.11 Action Script's 3.0

Menurut Coolin Mock (2007:xxi) ActionScript 3.0 adalah sebuah bahasa berorientasi objek untuk membuat aplikasi dalam flash client runtimes (seperti Flash Player dan Adobe AIR). Dengan menggunakan syntax yang berhubungan dengan Java dan C#, Bahasa utama ActionScript's seharusnya tidak asing lagi bagi programmer yang berpengalaman.

2.2 Penelitian yang Terkait

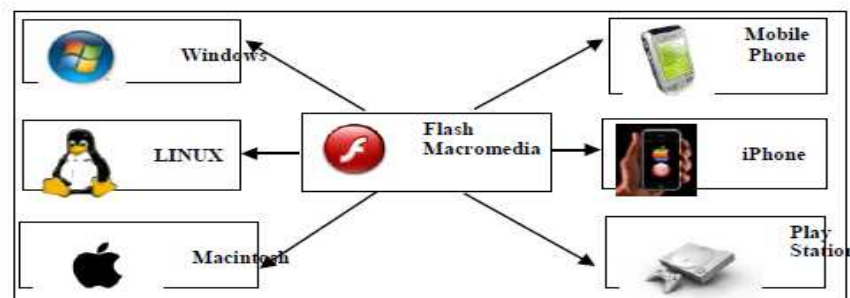
Penelitian terkait yang digunakan penulis untuk menyusun skripsi ini diambil dari jurnal terdahulu.

- 1. Jenis penelitian** : *International Journal of Instruction*
- Judul penelitian** : *Pedagogic Effectiveness of Print, Interactive Multimedia, and Online.*
- Tahun diterbitkan** : Juli 2013
- Penulis** : Jyotsna Dikshit, Suresh Garg, Santosh Panda
- Institusi** : Indira Gandhi National Open University

Tujuan melakukan *review* pada literatur ini adalah untuk mendeskripsikan keefektifan dalam pembelajaran dengan menggunakan media cetak, interaktif multimedia CD-ROM dan secara online. Penelitian ini dilakukan di Indira Gandhi National Open University (IGNOU), India. Dalam studi ini ditemukan bahwa, instruksi yang diberikan melalui multimedia CD-ROM lebih efektif dari pada media cetak dan *online learning*. Hal ini dibuktikan dari hasil akademik yang dicapai oleh mereka yang belajar melalui CD, *print* dan *online*. Rata-rata skor yang dihasilkan oleh grup yang melakukan pembelajaran dengan CD lebih besar, yaitu 61.85, *print* sebesar 56.30 dan *online* sebesar 49.10. Dari hasil studi ini dapat disimpulkan bahwa strategi dalam belajar untuk murid sebaiknya diberikan modul belajar dengan CD interaktif multimedia dan dilengkapi dengan buku cetak dan forum diskusi *online* akan lebih baik.

- 2. Jenis penelitian** : *International Journal*
- Judul penelitian** : INTEROPERABILITY OF MACROMEDIA
(ADOBE SYSTEMS)
- Tahun diterbitkan** : 2009
- Penulis** : Cristina Cirnicianu
- Institusi** : Romanian-American University, Bucharest
Romania

Fokus dari studi ini adalah menjelaskan kemampuan dari macromedia untuk dapat bekerja dengan aplikasi yang berbeda. Macromedia pertama kali dibuat pada tahun 1988, Macromedia dimulai dari *software* yang dapat memainnya banyak media(foto, video dan suara). Pada tahun 1992, Macromedia membuat Shockwave untuk bisa digunakan di *web browsers*. Macromedia juga membuat aplikasi yang bernama Afterburner yang dapat meng-*convert shockwave movies* menjadi *format website*. Sampai pada tahun 2005, Macromedia diambil alih oleh perusahaan Adobe. Macromedia membuat banyak program yang penting seperti Shockwave, Flash, Dreamweaver, Director, Freehand, Authorware, dan lain sebagainya. Dipenelitian ini juga disebutkan bahwa dengan menggunakan Flash, kita dapat mengembangkan tampilan yang dinamis dan *facility interaction* dengan pengguna. Flash dapat bekerja di banyak *platform* karena telah di sesuaikan dengan banyak *hardware* dan sistem operasi.



Gambar 2.20 Interaksi pada Flash Macromedia (Sistem Adobe)

(Sumber : <ftp://ftp.repec.org/opt/ReDIF/RePEc/rau/jisomg/WI09/JISOM-WI09-A17.pdf>)

- 3. Jenis penelitian** : *Journal of Multimedia*
- Judul penelitian** : *OXML: an Object XML Database Supporting Rich Media Indexing and Retrieval*
- Tahun diterbitkan** : April 2011
- Penulis** : Qing Li, Hon Chung Mak, Jianmin Zhao and Xinzhong Zhu
- Institusi** : Zhejiang Normal University, Jinhua, China and City University of Hong Kong, Hong Kong, China.

Fokus dari studi ini adalah menjelaskan OXML - obyek XML *database* untuk menyimpan dan mengelola file media (seperti flash) menggunakan XML dan teknik pengindeksan berorientasi objek. Didalam penelitian mengatakan bahwa flash sangat erat kaitannya dengan XML "*Since the tasks of parsing and manipulating Flash are closely tied with and dependent on XML*", XML dapat secara alami mengatur data flash sehingga mempercepat proses pengambilan data.