

BAB 2

LANDASAN TEORI

2.1 Teori Umum

2.1.1 Multimedia

Menurut Vaughan (2011), multimedia adalah kombinasi dari teks, suara, video, gambar, maupun animasi yang kemudian di presentasikan kepada orang atau diolah dan dimanipulasi secara digital. Multimedia memiliki 5 unsur: teks, suara, video, gambar dan animasi.

1. Text

Teks adalah unsur dasar dari multimedia yang mewakili dan menjelaskan informasi kepada pengguna. Teks merupakan bagian data yang paling sederhana dan tidak membutuhkan tempat penyimpanan yang besar. Hingga saat ini teks masih digunakan dalam menyampaikan informasi.

2. Picture

Gambar merupakan media yang baik dalam menyampaikan informasi. Gambar menjadi unsur penting dalam multimedia yang membuat informasi menjadi lebih efektif dan menarik.

Menurut Dastbaz (2003), terdapat beberapa format gambar:

i. .PICT

Format gambar yang biasa digunakan di Apple Mac.

ii. .BMP

Format gambar standar yang terdapat di Windows.

iii. .JPG

Format gambar yang digunakan dalam fotografi.

iv. .GIF

Format gambar yang dikompres dan sering digunakan di internet.

v. .TIFF

Format gambar yang biasa digunakan di desktop dan merupakan format file favorit dari perusahaan pencetakan.

vi. .EPS

Format gambar yang dapat mengandung vektor dan gambar bitmap

vii. .PSD

Format gambar yang digunakan PhotoShop untuk menyimpan gambar yang baru dibuat maupun dimanipulasi

viii. .PNG

Merupakan alternative dari GIF, juga sering digunakan di internet namun PNG dapat menampilkan gambar 24-bit.

3. *Sound*

Menurut Vaughan (2011), suara adalah sesuatu yang menghasilkan perbedaan tekanan yang sangat signifikan melalui medium udara hingga dapat menggetarkan indera pendengaran.

Suara dapat dibagi menjadi 3 kategori, yaitu percakapan (*speech*), musik dan efek suara (*sound effect*).Percakapan berupa suara orang berbicara.Musik berupa suara yang dihasilkan dari alat musik baik

dalam bentuk akustik maupun elektronik serta *synthesizer*. Efek suara contohnya suara gelas jatuh, suara hujan dan lainnya.

4. *Video*

Video merupakan kombinasi dari unsur-unsur multimedia dan menggabung semua unsur multimedia tersebut untuk menyampaikan informasi. Video sebenarnya ialah suatu animasi yang diambil dari kamera video dan disimpan dalam bentuk file.

Menurut DastBaz (2003), terdapat beberapa format dan standar video;

a. *National Television Standards Committee (NTSC)*

Dikembangkan pada tahun 1950, NTSC merupakan video standar yang terdiri dari 525 baris gambar setiap 1/30 detik. Standar ini digunakan di USA dan Jepang.

b. *Phase Alternate Line (PAL)*

PAL adalah metode menambahkan warna ke dalam TV hitam putih yang menampilkan 625 baris gambar setiap 1/25 detik. PAL digunakan di Inggris dan beberapa negara di Eropa.

c. *Sequential Colour and Memory system (SECAM)*

SECAM adalah standar video yang digunakan di Perancis, Russia, Afrika, Yunani, Iran, Iraq, dan Eropa Timur. Negara-negara tersebut masih dapat menampilkan PAL dengan warna, namun tidak semuanya dapat menampilkan SECAM dengan warna.

d. *High Definition TV (HDTV)*

HDTV adalah standar baru dalam teknologi pertelevisian. HDTV menyediakan gambar setara dengan kualitas film 35 mm. Ada beberapa format video HDTV yaitu ;

i. .AVI

Audio Video Interlanced adalah salah satu format video yang paling populer. Format ini dikembangkan oleh Microsoft.

ii. QuickTime

QuickTime adalah format video digital yang dikembangkan oleh Apple. QuickTime merupakan salah satu format video digital yang pertama dibuat. Quicktime sangat mirip dengan AVI, namun memiliki kualitas *output* yang lebih baik.

iii. .MPEG

.MPEG adalah singkatan dari *Moving Pictures Experts Group* dibentuk pada tahun 1998 untuk membuat sebuah standar video. Ide dibalik adanya MPEG adalah kompresi video untuk menghilangkan *spatial redundancy* antar video frame, sehingga dapat menghasilkan video yang ukurannya 30 kali lebih kecil dari format lain.

5. *Animation*

Animasi adalah proses membuat gambar yang diam menjadi terlihat bergerak. Menurut Vaughan (2011), animasi adalah sekumpulan

objek visual yang terus mengalami perubahan per waktu, yang dapat menghasilkan presentasi yang lebih hidup.

2.1.2 8 Aturan Emas Perancangan *User Interface*

Menurut Shneiderman dan Plaisant (2010), terdapat 8 aturan emas yang harus digunakan dalam membuat sebuah *interface* yang baik dan interaktif:

1. Berusaha untuk tetap konsisten

Dalam merancang *interface* ada beberapa hal yang harus tetap konsisten. *Prompt*, *menu*, *help screen*, juga konsisten dalam pewarnaan, kapitalisasi, huruf, dan sebagainya

2. Kegunaan universal

Desain antarmuka yang baik dapat memenuhi semua kebutuhan pengguna yang berbeda. Perbedaan beragam mulai dari umur, pengguna awam dengan ahli, cacat, dan keragaman tiap teknologi. Menambahkan fitur keterangan untuk membantu pemula, maupun menambahkan fitur *shortcut* untuk ahli dapat meningkatkan kualitas sistem.

3. Memberikan umpan balik yang informatif

Sistem harus memberikan umpan balik untuk setiap tindakan pengguna. Untuk tindakan yang sering dan tidak terlalu penting, pengguna diberikan umpan balik yang sederhana. Untuk tindakan yang tidak sering dan penting, pengguna diberikan umpan balik yang kuat.

4. Merancang dialog yang menghasilkan suatu penutupan

Urutan tindakan pengguna dapat dikelompokkan kedalam bagian awal, tengah dan akhir. Apabila pengguna telah menyelesaikan

tindakannya, pengguna diberi umpan balik dari sistem yang mengatakan bahwa aksi tersebut telah selesai. Umpan balik yang informatif dapat memberikan pengguna kepuasan, rasa lega, sinyal untuk berhenti dan mempersiapkan tindakan selanjutnya.

5. Memberikan penanganan kesalahan yang sederhana

Merancang sistem agar pengguna tidak melakukan kesalahan yang serius, misalnya tidak dapat memasukkan karakter abjad kedalam *entry field* numeric. Sistem antarmuka harus dapat mendeteksi kesalahan yang dibuat oleh pengguna dan memberikan instruksi penanganan kesalahan yang sederhana. Sebagai contoh pengguna tidak perlu mengetik ulang alamat di dalam formulir jika terjadi kesalahan dalam kode pos, melainkan user dipandu untuk hanya mengganti bagian yang salah saja.

6. Mudah dalam mengulang tindakan

Tindakan yang dilakukan harus dapat dikembalikan. Fitur ini dapat mengurangi kecemasan karena pengguna tahu bahwa kesalahan dapat dibatalkan, sehingga mendorong pengguna untuk mengeksplorasi pilihan lainnya.

7. Mendukung tempat pengendali internal

Pengguna menginginkan memiliki kendali sistem antarmuka mereka dan sistem antarmuka merespon tindakan pengguna. Pengguna akan terganggu dan tidak puas apabila terdapat urutan *input* entry yang membosankan, sistem antarmuka tidak menghasilkan tindakan yang diinginkan pengguna, dan kesulitan dalam mendapatkan informasi.

8. Mengurangi beban ingatan jangka pendek

Keterbatasan pengolahan informasi manusia dalam ingatan jangka pendek mengharuskan tampilan dibuat sederhana, menampilkan beberapa halaman yang dikonsolidasikan sehingga pengguna dapat mengingat banyak informasi dari halaman sebelumnya dan menggunakan informasi tersebut di halaman yang lain.

2.1.3 5 Faktor manusia terukur

Terdapat lima faktor manusia terukur yang menjadi panduan dalam pembuatan sistem antarmuka menurut Scneiderman dan Plaisant (2010):

1. Waktu belajar

Faktor yang mengukur berapa lama *user* dapat mempelajari sistem tersebut. Sistem seharusnya mudah digunakan sehingga *user* dapat langsung bekerja.

2. Kecepatan kinerja

Faktor yang mengukur seberapa cepat sebuah tugas diselesaikan oleh sistem. Ini akan mempengaruhi tingkat kepuasan *user* terhadap sistem.

3. Tingkat kesalahan *user*

Faktor yang mengukur berapa banyak dan apa saja kesalahan yang dibuat oleh *user*.

4. Daya ingat

Faktor yang mengukur seberapa bisa *user* mengingat penggunaan sistem agar dapat mudah menggunakannya lagi. Daya ingat memiliki hubungan dengan kecepatan belajar *user*.

5. Kepuasan subjektif

Faktor yang mengukur seberapa nyaman *user* dalam menggunakan sistem antarmuka.

2.1.4 Game

Menurut Rolling dan Adams (2003) *game* merupakan suatu hal yang bersifat kompleks, ketika seseorang bermain *game*, mereka dihibur dengan berpartisipasi secara aktif. Sebuah *game* berlangsung di tempat buatan yang memiliki sejumlah aturan. Aturan-aturan tersebut yang menentukan tindakan apa yang dapat dilakukan pemain, dan juga tindakan yang tidak dapat dilakukan.

Terdapat beberapa genre didalam *game*:

1. *Action Games*:

Biasanya mengandung tantangan fisik, seperti teka-teki, balapan, serta beberapa konflik. *Game* jenis ini menuntut reflek pemain untuk bergerak cepat dalam menangani tantangan yang disediakan.

2. *Strategy Games*:

Strategy games pada dasarnya berasal dari *board game*. *Game* dengan genre ini biasanya menuntut pemain untuk berfikir dalam memainkannya. Menurut Rolling dan Adams (2003), *game* strategi yang baik memiliki elemen-elemen utama yaitu tema, presentasi, dan perspektif.

3. *Role-Playing Games*:

Game bergenre RPG biasanya memiliki lebih dari satu karakter yang dapat dikontrol oleh pemain. Terkadang membutuhkan waktu yang lama

dalam menyelesaikan *game*RPG karena memiliki cerita yang kompleks dan juga sistem level yang hampir ada di setiap permainan RPG.

4. *Real-World Simulations* :

Game yang mensimulasikan sebuah aspek dunia nyata, contohnya simulasi mobil, pesawat dan sebagainya.

5. *Construction and Management Games*:

Game yang bersifat proses. Tujuannya ialah membangun sesuatu dalam jangka waktu yang disediakan seperti membangun sebuah kota dan sebagainya. *Game* ini biasanya memiliki unsur ekonomi didalamnya.

6. *Adventure Games*:

Adventure game lebih menjelaskan cara permainan tanpa tantangan. *Game* ini juga tidak membutuhkan proses seperti *management games* dan dalam memainkannya tidak terlalu dibutuhkan strategi tertentu. Umumnya memiliki cerita yang interaktif.

7. *Puzzle Games*:

Seperti namanya, *game* bergenre *puzzle* berfokus pada menyelesaikan suatu teka-teki. *Game puzzle* harus memiliki tantangan dan tampilan yang menarik agar pemain tertarik untuk memainkannya

8. *Casual games*:

Casual games ialah sebuah tipe *game* yang simple dan biasanya sangat mudah untuk dimainkan. Lebih sebagai penghibur di waktu sempit. Desain *game* sangat gampang dan mudah dimengerti. Tidak diperlukan usaha maupun latihan dan juga waktu yang lama untuk memainkannya seperti RPG ataupun *game action*.

9. *Side-scrolling games* :

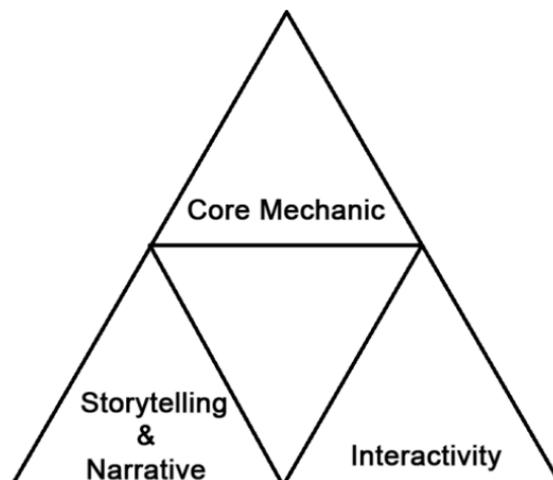
Side-scrolling games ialah sebuah *game* dimana *gameplay* nya dilihat dari kamera samping, dan karakter didalamnya biasanya bergerak dari kiri ke kanan. Kebanyakan *game* yang menggunakan *side-scrolling* biasanya *game* bertipe *platform*, maksudnya ialah *action game* dimana karakternya dapat melompat, berlari, ataupun memanjat.

2.1.5 *Game Design*

Menurut Rolling dan Adams (2003), *game design* ialah proses :

- Mengimajinasikan sebuah *game*.
- Mendefinisikan cara kerjanya.
- Menggambarkan unsur-unsur yang membentuk *game* tersebut (konseptual, fungsional, artistik, dan lainnya).
- Mengirim informasi tersebut kepada tim yang membantu mengembangkan *game* tersebut.

Rolling dan Adams (2003) juga membagi *game design* menjadi 3 area spesifik, yaitu *core mechanics*, *storytelling*, dan *interactivity*.



Gambar 2.1 *Core mechanic, storytelling dan interactivity*

Core mechanic ialah peraturan yang mendefinisikan operasi dalam dunia *game* tersebut. *Core mechanic* merupakan bayangan dari perancang yang kemudian diwujudkan menjadi set peraturan. Bila didalam *game* yang dirancang tidak terdapat *core mechanic*, maka *game* tersebut tidak akan menjadi *game* yang baik.

Setiap *game* memiliki sebuah cerita. Kompleksitas dan kedalaman cerita tergantung pada *game*. Narasi ialah bagian dari cerita yang disampaikan oleh pembuat kepada pemain. Narasi merupakan bagian cerita yang non interaktif dan presentasional.

Interaktifitas adalah cara bagaimana pemain melihat, mendengar dan bertindak di dalam dunia *game*. Hal tersebut mencakup topik seperti: grafik, suara, *user interface*, dan segala sesuatu yang dibutuhkan di dalam *game*.

2.1.6 *Storyboard*

Storyboard adalah sebuah ilustrasi seperti komik, yang dapat menghubungkan antara desainer dan *programmer*. Dalam *storyboard*, terdapat gambar yang akan menunjukkan informasi yang diperlukan dan menggambarkan sebuah produk akhir yang akan dicapai. Dalam membuat animasi, biasanya membutuhkan *storyboard* untuk menuangkan ide-ide yang dimiliki. (Simon, 2007)

Menurut Vaughan (2011) ada 2 jenis pendekatan dalam membuat *storyboard*, yaitu :

1. *Storyboard* yang berupa *outline* grafis yang menjelaskan secara detail dan terperinci pada setiap layar yang akan dibuat, baik pemilihan kata dan *font*, pemilihan warna, pemilihan navigasi, bentuk tombol, bayangan dan efek lainnya. Pendekatan *storyboard* jenis ini akan lebih cepat dikembangkan menjadi aplikasi yang siap digunakan.
2. *Storyboard* yang berupa sketsa-sketsa yang masih mentah atau tidak detail, dan akan baru dikembangkan sesuai dengan pengerjaan proyeknya secara langsung.

2.1.7 *Unified Modeling Language (UML)*

UML (*Unified Modeling Language*) adalah sebuah bahasa yang digunakan untuk menspesifikasikan, memvisualisasikan, menkonstruksikan dan mendokumentasikan bagian atau komponen dari suatu sistem perangkat lunak (Larman, 2005). UML dapat menceritakan apa yang seharusnya dilakukan oleh sebuah sistem, bukan bagaimana yang seharusnya dilakukan oleh sistem.

Menurut Connolly & Begg (2010), diagram utama UML terbagi menjadi dua kategori:

1. *Structural Diagram*

Structural diagram menjelaskan hubungan statis antar komponen.

Contoh:

- *Class Diagram*
- *Object Diagram*
- *Component Diagram*
- *Deployment Diagram*

2. *Behavioral Diagram*

Behavioral diagram menjelaskan hubungan dinamis antar komponen.

Contoh:

- *Use Case Diagram*
- *Sequence Diagram*
- *Collaboration Diagram*
- *Statechart Diagram*
- *Activity Diagram*

2.1.7.1 *Use Case Diagram*

Menurut Whitten (2007), *use case* diagram adalah suatu interaksi antara sistem dan sistem eksternal dan pengguna. Dengan kata lain, secara grafis mendeskripsikan siapa yang akan menggunakan sistem dan didalamnya cara apa yang digunakan pengguna untuk dapat berinteraksi

dengan sistem. Fungsi dari *use case* adalah untuk menjelaskan dan mengidentifikasi suatu *event*, siapa yang memulai, dan bagaimana sistem meresponnya. *Use case* menjelaskan apa yang dapat dilakukan sistem, bukan bagaimana.

Use case diagram terdiri atas proses, aktor dan hubungan antara keduanya. Aktor merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem. Proses digambarkan berbentuk elips dan operasi dituliskan didalamnya. Berikut penjelasan mengenai unsur yang terdapat pada *use case diagram* :

1. *Use Case*

Model *use-case* mengidentifikasi dan mendeskripsikan fungsi sistem dengan menggunakan sebuah alat yaitu *use cases*. *Use cases* mendeskripsikan fungsi sistem dari perspektif luar pengguna dan dalam cara dan terminologi yang mereka ketahui.

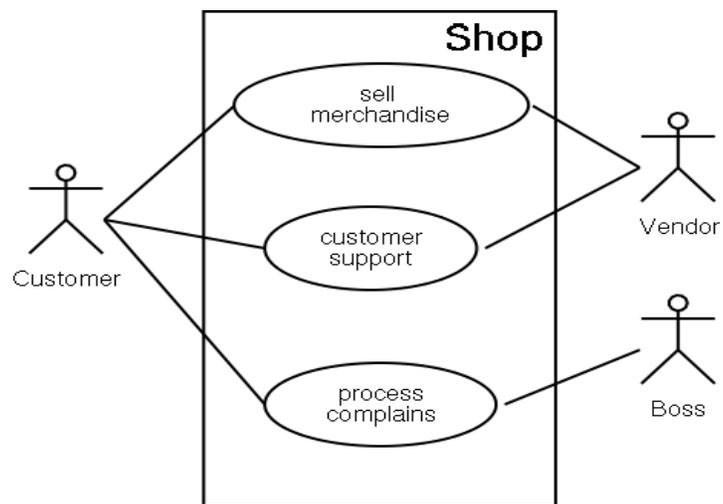
Use cases diwakili secara grafik oleh elips horisontal dengan nama *use case* yang muncul diatas, bawah, atau didalam elips. *Use case* merepresentasikan satu tujuan sistem dan mendeskripsikan urutan aktifitas dan interaksi pengguna dalam menggapai tujuan tersebut. Kreasi dari *use case* terbukti telah menjadi teknik yang terbaik untuk memahami dengan baik dan sistem *requirement* dokumen.

2. Actors

Use case dimulai atau dipicu oleh pengguna luar yang disebut *actors*. *Actor* memulai aktivitas sistem, *use case*, yang bertujuan untuk menyelesaikan beberapa tugas bisnis yang membuat suatu nilai yang terukur.

3. Relationship

Suatu hubungan digambarkan dengan sebuah garis diantara 2 simbol pada diagram *use-case*. Maksud hubungan tersebut bermacam-macam tergantung pada garis yang tergambar dan tipe simbol apa yang digunakan.

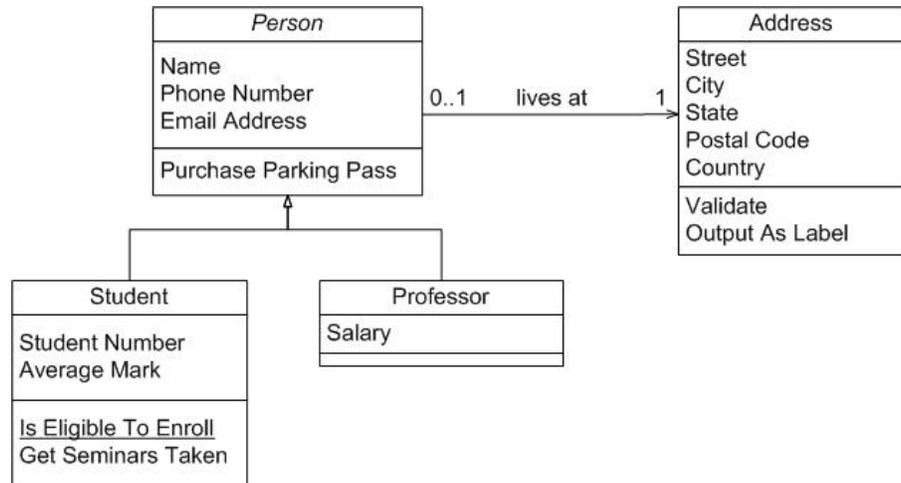


Gambar 2.2 Gambar Use Case Diagram

2.1.7.2 Class Diagram

Menurut Whitten (2007) *class diagram* menggambarkan struktur objek. *Class diagram* merupakan diagram yang menunjukkan hubungan

suatu set objek (kelas, *interface*, dan kolaborasi). *Class diagram* menunjukkan hubungan antar *class* dalam suatu sistem yang dibangun, namun tidak menjelaskan bagaimana *class* tersebut berinteraksi dengan lainnya.



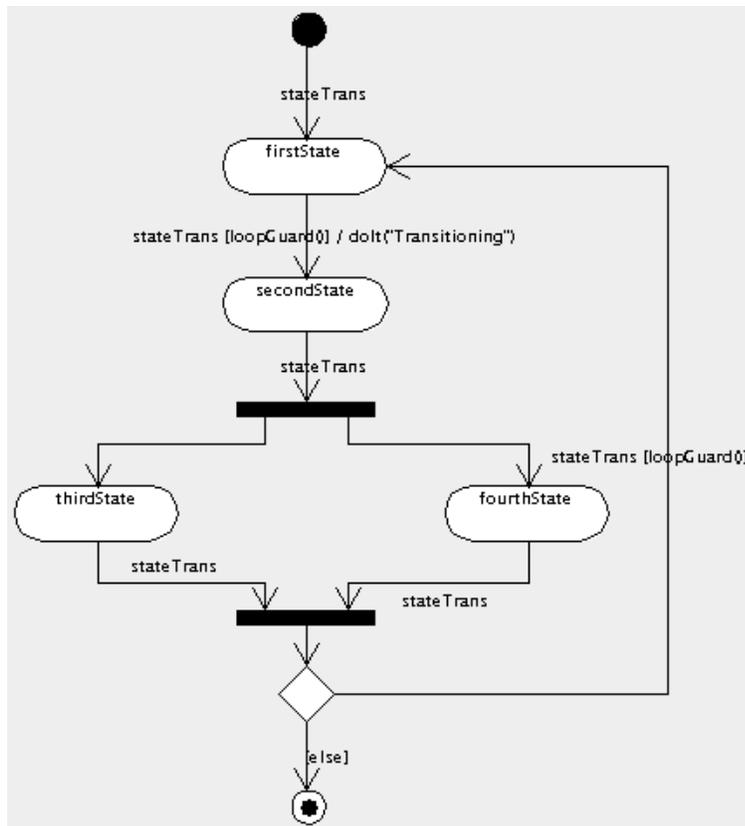
Gambar 2.3 *Class diagram*

2.1.7.3 Activity Diagram

Menurut Whitten (2007), *activity diagram* adalah flowchart yang memperlihatkan alur sebuah proses, dan struktur objek dari sistem. Juga menampilkan aksi yang akan dilakukan ketika proses berjalan dan hasil dari proses tersebut.

Activity diagram pada umumnya terdiri dari :

1. *Action*
2. *Transition*
3. Objek



Gambar 2.4 *Activity Diagram*

2.1.7 Database

2.1.7.1 Pengertian Database

Menurut Connolly dan Begg (2010), *database* ialah kumpulan data *logical* yang berkaitan, dan deskripsi data tersebut dirancang untuk memenuhi kebutuhan informasi organisasi. Sebuah *database* yang memiliki banyak data dapat digunakan secara bersamaan oleh departemen dan pengguna. Jadi basis data bukanlah lagi menjadi milik satu departemen tetapi dapat dihubungkan ke semua sumber daya perusahaan.

2.1.7.2 Sistem Manajemen *Database*

Sistem manajemen *database* menurut Connolly dan Begg (2010) adalah sebuah sistem piranti lunak yang mengizinkan pengguna untuk mendefinisikan, membuat, mempertahankan, dan mengontrol akses ke *database*. Sistem ini merupakan suatu piranti lunak yang berinteraksi dengan program aplikasi pengguna dan *database* lainnya. Komponen-komponen yang terdapat pada lingkungan sistem manajemen *database* menurut Connolly dan Begg(2010):

1. *Hardware*

Sistem manajemen *database* dan program-program aplikasinya membutuhkan perangkat keras agar bisa dijalankan. *Hardware* beragam, dapat berupa *personal computer*(PC), *single mainframe*, atau jaringan computer. Penggunaannya tergantung kebutuhan organisasi dan sistem manajemen *database* yang digunakan.

2. *Software*

Meliputi *software* manajemen *database* itu sendiri, program-program aplikasi, sistem operasi, termasuk juga *software* jaringan jika sistem tersebut digunakan di dalam jaringan computer.

3. Data

Merupakan komponen yang paling penting. Data berperan sebagai jembatan antara komponen mesin dengan orang.

4. Prosedur

Merupakan merujuk pada langkah-langkah dan aturan-aturan yang mempengaruhi perancangan dan penggunaan *database*.

5. Orang

Komponen ini terdiri dari orang-orang yang terlibat di dalam sistem untuk menangani atau menjalankan sistem manajemen basis data tersebut.

2.1.8 MySQL

Menurut Hirin dan Virgi (2011) MySQL adalah salah satu perangkat lunak sistem manajemen *database* SQL atau sering disebut dengan DBMS (*Database Management System*). MySQL memiliki kelebihan yaitu bersifat *multithread* dan *multiuser* serta mendukung sistem jaringan.

Beberapa pertimbangan programmer memilih MySQL sebagai alat utama dalam mengolah *database* adalah:

- Kecepatan
- Kemudahan penggunaan
- Open Source*
- Keamanan

Semua perintah SQL dibagi dalam 2 kategori besar sesuai fungsinya, yaitu :

1. DDL (*Data Definition Language*)

DDL adalah kumpulan perintah SQL yang digunakan untuk membuat (create), mengubah (alter) dan menghapus (drop) struktur dan definisi tipe data dari objek-objek *database*.

Objek-objek *database* yang dimaksud pada MySQL adalah sebagai berikut:

-*Database*

-*Table*

-*View*

-*Index*

-*Procedure*

-*Function*

-*Trigger*

2. DML (*Data Manipulation Language*)

Data Manipulation Language merupakan kumpulan perintah SQL yang digunakan untuk proses pengolahan isi data di dalam table seperti memasukkan, merubah dan menghapus isi data dan tidak terkait dengan perubahan struktur dan definisi tipe data dari objek *database*.

2.1.9 Metode Extreme Programming

Menurut *Pressman* (2010) menyatakan bahwa XP adalah suatu metodologi yang ringan untuk tim kecil, menengah dan besar yang mengembangkan suatu perangkat lunak dengan cepat tanpa mengubah persyaratan yang diberikan. Metode ini menyempurnakan metode life

cycle dan merupakan metode baru yang dikembangkan dari metode *spiral* dan *waterfall*.

Berikut ini adalah beberapa tahapan dalam metode XP yaitu:

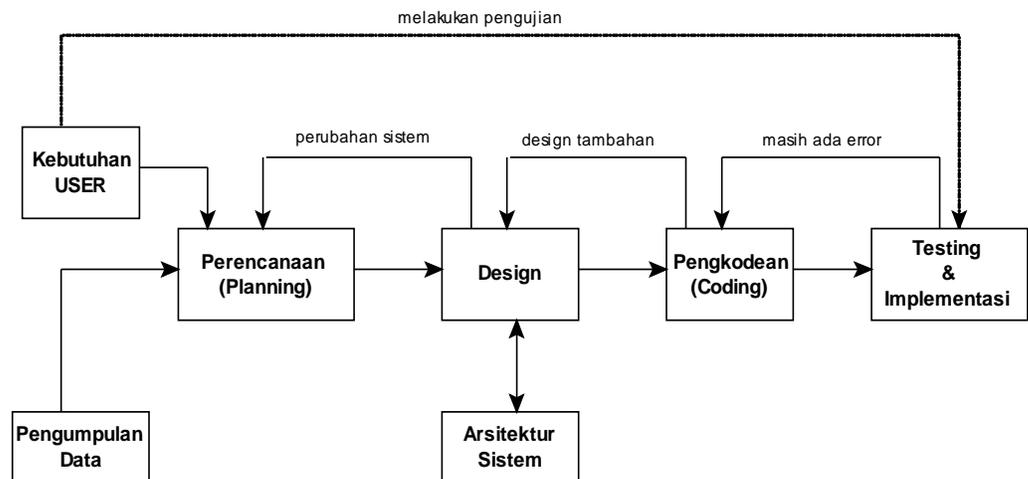
1. Tahap pertama yaitu perencanaan (*planning*), pada tahap ini pengembang sistem mengambil dan mengumpulkan semua bahan, data-data dan kebutuhan dari user (pengguna) kemudian menganalisisnya.
2. Tahap kedua yaitu *design*, pada tahap ini pengembang aplikasi dapat secara langsung melakukan suatu design tentang sistem, mulai dari sistem sampai arsitekturnya.
3. Tahap ketiga yaitu pengkodean (*coding*), pada tahap ini pengembang aplikasi memilih prototipe, dan kemudian dilakukan pembuatan sistem dengan menggunakan suatu bahasa pemrograman tertentu, atau mengembangkan sistem yang sudah ada menjadi sistem yang baru.
4. Tahap keempat yaitu test dan implementasi (*testing*), pada tahap ini sistem yang telah menjadi suatu aplikasi diuji dan ditest yang bila semua tidak ada kesalahan maka selanjutnya diimplementasikan dilingkungan pemakai sistem, diikuti dengan pembuatan dokumentasi dari aplikasi yang dibuat.

Adapun kelebihan dari metode XP ini diantaranya:

1. Mampu menyelesaikan suatu proyek perangkat lunak dengan cepat.

2. Setiap orang dalam team bisa mengerjakan tiap tahapan dalam metode ini, tanpa harus menunggu tahapan lain selesai.
3. Kemungkinan resiko kegagalan dalam suatu proyek menjadi lebih kecil.
4. Setiap kebutuhan pelanggan dapat diselesaikan dengan cepat dan baik.
5. Dalam pengimplementasian, pelangan dapat ikut serta dalam hal pengujian kelayakan dan kebutuhan aplikasi yang dibangun.

Dibawah ini merupakan diagram metode *extreme programming (XP)*.



Gambar 2.5 Diagram Metode *Extreme Programming*

2.1.10 PHP

Menurut Narnmore (2005), PHP adalah bahasa pemrograman *web* di sisi *server* yang membantu *web* menjadi dinamis, dimana sisi *server*

berarti pengguna *web* tidak dapat melihat isi dari program PHP (berbeda dengan *javascript* yang ada di sisi *client*) dan dinamis dapat diartikan data atau konten dari *web* dapat berubah sesuai dengan interaksi yang diberikan pengguna.

PHP merupakan akronim dari *Hypertext Preprocessor*, sebuah rekursif akronim dan merupakan proyek *open-source*. Sifat *open-source* ini membuat PHP menjadi salah satu bahasa pemrograman *web* yang populer.

2.2 Teori Khusus

2.2.1 HTML 5

HTML 5 adalah sebuah *markup language* yang sering digunakan dalam struktur *World Wide Web* dan juga merupakan teknologi inti dari internet.. Ada lebih banyak *library* dan *plugin* yang disediakan di HTML 5 dan CSS3, sehingga aplikasi multimedia yang dibuat akan menjadi lebih interaktif. *Performance* dari HTML 5 sudah cukup baik, dapat hampir disamakan dengan aplikasi *desktop*.HTML 5 memiliki beberapa fitur yang dapat mengungguli aplikasi *desktop* seperti *Drag and drop*, *local storage*, animasi, dan juga fitur lainnya.

Pengimplementasian HTML 5 belum dapat digunakan secara maksimal sampai tahun 2022, seperti yang dikatakan Ian Hickson, editor dari HTML 5. Walaupun begitu, tidak ada salahnya bagi para *web browser* untuk memulai penggunaan teknologi ini. Berikut ini daftar browser yang mendukung HTML 5.

HTML 5 Cheat Sheet [BROWSER SUPPORT]		Legend: ● supported (red), ● partially supported (teal), ● no support (black)																					
	Details & Summary	WebGL 3D Canvas Elements	Inline SVG in HTML5	Datalist Element	Progress & Meter	ClassList (DOM TokenList)	Ruby Annotation	Form Validation	Session History Mgmt	HTML5 Form Features	Drag & Drop	Content Editable Attribute	Text API for Canvas	Audio Element	Video Element	Dataset & Data (Attributes)	New Semantic Elements	Canvas (Basic Support)	Hashchange Event	Offline Web Apps	Get Elements by Class Name	Overall Support	
8.0	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	21%
3.6	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	55%
9.0	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	85%
5.0	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	67%
11.0	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	64%
Mobile																							
2.3	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	51%
4.5	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	50%
10.0	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	36%

Gambar 2.6 HTML 5 Compability browser

Beberapa spesifikasi juga sudah stabil dan dapat diterapkan pada HTML 5, antara lain:

1. Markup

HTML 5 memperkenalkan beberapa elemen baru dan atribut yang merefleksikan tipikal penggunaan *website* modern. Beberapa diantaranya adalah pergantian yang bersifat semantik, seperti elemen (<div>) dan inline (). Banyak elemen lain yang memberikan fungsi baru melalui *interface* yang telah distandarkan, seperti elemen multimedia <audio> dan <video>. Beberapa elemen yang dapat dikerjakan menggunakan *Cascading Style Sheet* (CSS) ditiadakan, seperti dan <center>.

2. API baru

Pada HTML 5 telah dispesifikasikan pengkodean *application programming interfaces* (API). Antarmuka *document object model* (DOM) yang ada dikembangkan. Beberapa API terbaru pada HTML 5 antara lain :

- Elemen *canvas*

Unsur yang digunakan untuk menggambar di dalam sebuah aplikasi HTML 5

- *Drag and Drop*

Memudahkan men-*drag* atau men-*drop* misalnya teks maupun gambar ke dalam aplikasi yang dibuat.

Beberapa teknologi yang juga terkait namun tidak dijadikan bagian dalam spesifikasi HTML 5 W3C dan WHATWG HTML5 adalah:

- *Geolocation*

Memungkinkan kita untuk mengetahui lokasi geografis, sumber informasi didapat dari GPS.

- Web SQL Database

- API Database terindeks

3. Penanganan Kesalahan

Web Explorer HTML 5 (*text/html*) akan fleksibel dengan menangani kesalahan sintaks. HTML 5 didesain agar *web explorer* lama dapat dengan aman mengabaikan konstruksi HTML 5 yang baru. Spesifikasi HTML 5 memberikan aturan detail untuk memarsing sebagai persyaratan agar

berbagai *web explorer* tetap memberikan hasil yang sama saat terjadi kesalahan sintaks.

2.2.2 Construct 2

Construct 2 adalah *software* yang digunakan untuk membantu pembuatan aplikasi berbasis HTML 5. Construct 2 dikembangkan oleh *software developer* “Scirra”.

Construct 2 menyediakan berbagai macam *behavior game* yang ada, seperti *physics* dan *platform*. Berikut beberapa basic fitur dari Construct 2:

1. *Realistic physic effect*
2. *Instant preview*
3. *Tutorials*

2.2.3 Becak

Menurut Van Yogya (2010), Becak berasal dari bahasa *Hokkien*, ‘*be chia*’ yang berarti kereta kuda. Namun karena penyebutannya salah kaprah, sehingga menjadi kata becak seperti yang digunakan sekarang. Becak adalah suatu alat transportasi beroda tiga yang umum ditemukan di Indonesia dan juga di sebagian negara-negara Asia. Kapasitas normal becak adalah dua orang penumpang dan seorang pengemudi. Di Indonesia ada dua jenis becak yang lazim digunakan, yaitu becak dengan pengemudi di belakang yang biasanya ada di Pulau Jawa, dan becak dengan pengemudi di samping yang banyak dijumpai di Pulau Sumatra.