

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Rancang Bangun**

Menurut Jogiyanto (2005:197), Rancang Bangun (desain) adalah tahap dari setelah analisis dari siklus pengembangan sistem yang merupakan pendefinisian dari kebutuhan-kebutuhan fungsional, serta menggambarkan bagaimana suatu sistem dibentuk yang dapat berupa penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi, termasuk menyangkut mengkonfigurasi dari komponen-komponen perangkat keras dan perangkat lunak dari suatu sistem

#### **2.2 Aplikasi**

Istilah aplikasi pada dasarnya berasal dari bahasa Inggris yaitu dari kata *application* yang berarti penerapan ataupun penggunaan. Jika ditinjau secara istilah, aplikasi berarti suatu program yang telah siap untuk digunakan. Aplikasi secara sengaja dibuat untuk melakukan suatu fungsi bagi pemakai jasa. Semua jenis aplikasi lainnya yang dipakai untuk sebuah sasaran yang dituju.

Pengertian program aplikasi menurut kamus komputer Eksekutif (199: 9), “Aplikasi program adalah program yang membentuk bagian tugas pemakai dan ditulis oleh pemakai. Hal ini berbeda dengan program tujuan umum yang digunakan untuk mengelola operasi seluruh sistem”. Sedangkan menurut Edi Purnomo (2003: 2), “Aplikasi adalah sebuah kegiatan pengolahan data suatu urusan tertentu dari sebuah perusahaan”.

#### **2.3 P3K (Pertolongan Pertama Pada Kecelakaan)**

Pertolongan Pertama Pada Kecelakaan (P3K) adalah upaya pertolongan dan perawatan sementara terhadap korban kecelakaan sebelum mendapat pertolongan yang lebih sempurna dari dokter atau paramedik. Ini berarti pertolongan tersebut bukan sebagai pengobatan atau penanganan yang sempurna, tetapi hanyalah

berupa pertolongan sementara yang dilakukan oleh petugas P3K (petugas medik atau orang awam) yang pertama kali melihat korban. Pemberian pertolongan harus secara cepat dan tepat dengan menggunakan sarana dan prasarana yang ada di tempat kejadian. Tindakan P3K yang dilakukan dengan benar akan mengurangi cacat atau penderitaan dan bahkan menyelamatkan korban dari kematian, tetapi bila tindakan P3K dilakukan tidak baik malah bisa memperburuk akibat kecelakaan bahkan dapat menyebabkan kematian (Nadine,2011)

### **2.3.1 Tujuan P3K**

Tujuan dari P3K adalah sebagai berikut:

- a. Menyelamatkan nyawa atau mencegah kematian
  1. Memperhatikan kondisi dan keadaan yang mengancam korban
  2. Melaksanakan Resusitasi Jantung dan Paru (RJP) kalau perlu
  3. Mencari dan Mengatasi pendarahan
- b. Mencegah cacat yang lebih berat (mencegah kondisi memburuk)
  1. Mengadakan diagnosa
  2. Menangani korban dengan prioritas yang logis
  3. Memperhatikan kondisi atau keadaan (penyakit) yang tersembunyi
- c. Menunjang penyembuhan
  1. Mengurangi rasa sakit dan rasa takut
  2. Mencegah infeksi
  3. Merencanakan pertolongan medis serta transportasi korban dengan tepat

### **2.3.2 Prinsip P3K**

Beberapa prinsip yang harus ditanamkan pada jiwa petugas P3K apabila menghadapi kejadian kecelakaan adalah sebagai berikut:

- a. Berasikaplah tenang, jangan pernah panik. Anda diharapkan menjadi penolong bukan pembunuh atau menjadi korban selanjutnya (ditolong).
- b. Gunakan mata dengan jeli, kuatkan hatimu karna anda harus tega melakukan tindakan yang membuat korban menjerit kesakitan untuk keselamatannya, lakukan gerakan dengan tangkas dan tepat tanpa menambah kerusakan.

- c. Perhatikan keadaan sekitar kecelakaan, cara terjadinya kecelakaan, cuaca.
- d. Perhatikan keadaan penderita apakah pingsan, ada pendarahan dan luka, patah tulang, merasa sangat kesakitan dll.
- e. Periksa pernafasan korban. Kalau tidak bernafas, periksa dan bersihkan jalan nafas lalu berikan pernafasan bantuan.
- f. Periksa nadi atau denyut jantung korban. Kalau jantung berhenti, Lakukanlah pijat jantung luar. Kalau ada pendarahan berat segera hentikan.
- g. Apakah penderita shock? Kalau shock cari dan atasi penyebabnya.
- h. Setelah semuanya stabil maka periksa ulang cedera penyebab atau penyerta. Kalau ada patah tulang lakukanlah pembidai pada tulang yang patah, jangan buru-buru memindahkan atau membawa ke klinik atau rumah sakit sebelum tulang yang patah dibidai.
- i. Sementara memberikan pertolongan, anda juga harus menghubungi petugas medis atau rumah sakit terdekat.

### 2.3.3 Prioritas Pertolongan

Ada beberapa prioritas utama yang harus dilakukan oleh penolong dalam menolong korban yaitu:

- a. Henti napas
- b. Henti jantung
- c. Pendarahan berat
- d. Shock
- e. Ketidak sadaran
- f. Pendarahan ringan
- g. Pata tulang atau cidera lain

## 2.4 Android

### 2.4.1 Definisi Android

Android adalah suatu sistem operasi yang didesain sebagai *platform open source* untuk perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware*, dan aplikasi. Android menyediakan *platform* yang terbuka bagi para pengembang untuk menciptakan aplikasi mereka.

Android menyediakan semua *tools* dan *framework* untuk mengembangkan aplikasi dengan mudah dan cepat. Dengan adanya Android SDK (*Software Development Kit*) pengembang aplikasi dapat memulai pembuatan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java (Edi, 2011).

Pada tahun 2005, Google membeli Android, Inc., yang merupakan pendatang baru yang mengembangkan software untuk *smartphone*. Pada saat itu dunia mengira bahwa Google akan memproduksi *smartphone*. Anggapan itu ternyata salah karena Google menyatakan bahwa ambisi Android bukan hanya untuk mengembangkan sebuah ponsel melainkan suatu *platform* yang dapat digunakan di banyak ponsel dan perangkat lainnya (Edi, 2011).

Kemudian untuk mengembangkan Android dibentuklah Open Handset Alliance, sebuah grup nonprofit yang terdiri dari beberapa perusahaan *hardware*, *software*, telekomunikasi dan perusahaan lainnya termasuk Google, HTC, Intel Motorola, Qualcomm, T-Mobile, dan Nvidia. Pada saat perilisan perdana Android tanggal 5 November 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan open source pada perangkat mobile. Di lain pihak, Google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi *software* dan *open platform* perangkat seluler.

Di dunia terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau *Google Mail Services* (GMS). Kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai *Open Handset Distribution* (OHD).

Sekitar bulan September 2007 Google memperkenalkan Nexus One, *smartphone* yang menggunakan sistem operasi Android versi 1.0. Ponsel ini diproduksi oleh HTC Corporation dan mulai dipasarkan pada 5 Januari 2008. Pada tahun 2009 perkembangan perangkat berbasis Android semakin pesat. Lebih dari 20 jenis perangkat mobile menggunakan sistem operasi Android. Daftar versi-versi Android yang dapat dilihat pada Tabel 2.1

**Tabel 2.1** Versi – versi Android

<b>Versi Android</b>	<b>API Level</b>	<b>Nickname</b>
Android 1.0	1	-
Android 1.1	2	-
Android 1.5	3	Cupcake
Android 1.6	4	Donut
Android 2.0	5	Eclair
Android 2.01	6	Eclair
Android 2.1	7	Eclair
Android 2.2	8	Froyo (Frozen Yogurt)
Android 2.3	9	Gingerbread
Android 2.3.3	10	Gingerbread
Android 3.0	11	Honeycomb
Android 3.1	12	Honeycomb
Android 3.2	13	Honeycomb
Android 4.0	14	Ice Cream Sandwich
Android 4.0.3	15	Ice Cream Sandwich
Android 4.1	16	Jelly Bean
Android 4.2	17	Jelly Bean
Android 4.3	18	Jelly Bean
Android 4.4	19	KitKat
Android 5.0	21	Lollipop
Android 6.0	23	Marshmallow

Dari Tabel 2.1 dapat dilihat pada setiap perubahan versi Android terdapat perubahan *API Level*. *API Level* adalah nilai yang menunjukkan revisi *framework* pada *platform* Android. Nomor versi Android yang terus berubah dikarenakan perubahan *API Level*. Setiap versi Android mendukung tepat satu *API Level*, namun tetap mendukung *API Level* sebelumnya. *API Level* akan menentukan apakah suatu aplikasi bisa dijalankan pada suatu *platform* Android atau tidak.

## 2.5 Adobe Dreamwaver

*Macromedia Dreamweaver8* yang diciptakan pada tahun 1997, merupakan program aplikasi *web editor* yang cukup populer untuk pembuatan *website*,

selain tampilannya yang menarik, kemampuan fitur-fiturnya juga memudahkan para pemakainnya untuk menciptakan halaman *web*. Namun sekarang Macromedia dimiliki oleh adobe yang mengakuisi pada tahun 2005, dan berganti nama menjadi *Adobe Dreamweaver*.

*Adobe dreamweaver* adalah *script editor* profesional yang digunakan untuk melakukan desain secara visual dan mengelola halaman *web* serta pembuat aplikasi *web*. Pengembang dan programmer aplikasi *web* lebih menyukai menggunakan Dreamweaver, karena Dreamweaver berbasis *WYSIWYG* (*What You See Is What You Get*). (Mahdi:21)

*Dreamweaver* adalah software yang digunakan untuk mendesain *web* yang menawarkan cara mendesain *web* dengan dua langkah sekaligus dalam satu waktu, yaitu mendesain dan memprogram. *Script* akan mengisi secara otomatis dan langsung mengisi program pengaturan.

*Dreamweaver* mengendalikan fitur *visual editing*, sehingga pengguna dapat secara cepat melakukan desain dan memfungsikan halaman *web* yang diinginkan tanpa harus menuliskan beberapa baris kode pemrograman, dapat dilihat pada gambar

2.1.



**Gambar 2.1** Tampilan Awal *Adobe Dreamweaver8*

## 2.6 HTML5

HTML5 adalah sebuah bahasa markah untuk menstrukturkan dan menampilkan isi dari Waring Wera Wanua, sebuah teknologi inti dari Internet. HTML5 adalah revisi kelima dari HTML (yang pertama kali diciptakan pada tahun 1990 dan versi keempatnya, HTML4, pada tahun 1997) dan hingga bulan Juni 2011 masih dalam pengembangan. Tujuan utama pengembangan HTML5 adalah untuk memperbaiki teknologi HTML agar mendukung teknologi multimedia terbaru, mudah dibaca oleh manusia dan juga mudah dimengerti oleh mesin.

HTML5 merupakan salah satu karya Konsortium Waring Wera Wanua (*World Wide Web Consortium, W3C*) untuk mendefinisikan sebuah bahasa markah tunggal yang dapat ditulis dengan cara HTML ataupun XHTML. HTML5 merupakan jawaban atas pengembangan HTML 4.01 dan XHTML 1.1 yang selama ini berjalan terpisah, dan diimplementasikan secara berbeda-beda oleh banyak perangkat lunak pembuat web.

## 2.7 MySQL

MySQL adalah *Relation Database Management System (RDBMS)* yang didistribusikan secara gratis dibawah lisensu GPL (*General Public License*). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoprasian database, terutama untuk pemilihanatau seleksi dan pemasukan data yang memungkinkan pengoprasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu sistem database (DBMS) dapat diketahui dengan cara kerja optimizernya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh user maupun program-program aplikasinya. Sebagai database server, MySQL dapat dikatakan lebih unggul dibandingkan database server lainnya dalam query data.

## 2.8 XAMPP

XAMPP adalah sebuah *software web server apache* yang didalamnya sudah tersedia database server mysql dan *support php programming*. XAMPP

merupakan software yang mudah digunakan, gratis dan mendukung instalasi di Linux dan *Windows*. Keuntungan lainnya adalah dengan menginstal satu kali sudah tersedia *Apache Web Server*, *MySQL Database Server*, *PHP Support* (PHP 4 dan PHP 5) dan beberapa module lainnya. Hanya bedanya kalau yang versi untuk *Windows* sudah dalam bentuk instalasi grafis dan yang Linux dalam bentuk file. Kelebihan lain yang berbeda dari versi *Windows* adalah memiliki fitur untuk mengaktifkan sebuah server secara grafis, sedangkan Linux masih berupa perintah-perintah di dalam *console*. Oleh karena itu XAMPP yang versi Linux ulit untuk dioperasikan.

## 2.9 UML (*Unified Modeling Language*)

(Munawar, 2005) menyatakan, UML (*Unified Modeling Language*) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek. Hal ini dikarenakan UML menyediakan permodelan visual yang memungkinkan bagi pengembang sistem membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain

### 2.9.1 Diagram UML

Model-model diagram dapat dikelompokkan berdasarkan sifatnya yaitu statis dan dinamis. Jenis diagram tersebut antara lain:

1. Diagram Kelas

Bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, dan relasi-relasi.

2. Diagram Paket

Bersifat statis. Diagram ini memperlihatkan kumpulan kelas-kelas, merupakan bagian dari diagram komponen.

3. Diagram *Use case*

Bersifat statis. Diagram ini memperlihatkan himpunan *use-case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama penting untuk

mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna.

Elemen *use case* terdiri dari:

- a. Diagram *use case*, disertai dengan narasi dan skenario, pada gambar 2.2:



**Gambar 2.2** *use case*

- b. Aktor, mendefinisikan entitas di luar sistem yang menggunakan sistem, ada gambar 2.3:



**Gambar 2.3** Simbol Aktor

- c. Asosiasi, mengindikasikan aktor mana yang berinteraksi dengan *use case* dalam suatu sistem, ada gambar 2.4:



**Gambar 2.4** *Relationship*

- d. <<include>> dan <<extend>>

Merupakan indikator yang menggambarkan jenis relasi dan interaksi antar *use case*.

- e. Generalisasi, menggambarkan hubungan turunan antara *use case* atau antar aktor.

*Use case* mengekspresikan apa yang user harapkan terhadap sistem. Narasi *use case* menjelaskan sistem secara detail bagaimana *user* berinteraksi dengan sistem saat mengakses *use case*.

4. Diagram Interaksi atau *Sequence*

Bersifat dinamis. Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.

### 5. Diagram Komunikasi

Bersifat dinamis. Diagram sebagai pengganti diagram kolaborasi UML yang menekankan organisasi struktural dari objek – objek yang menerima serta mengirim pesan.

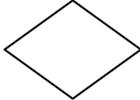
### 6. Diagram *Statechart*

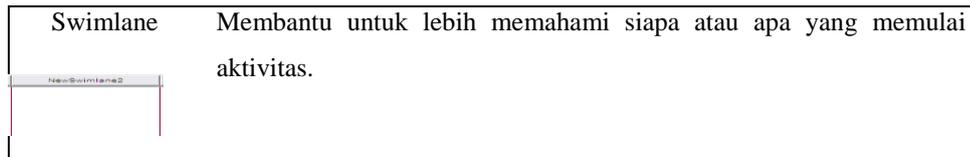
Bersifat dinamis. Diagram ini memperlihatkan keadaan-keadaan setiap sistem, memuat status, transisi, kejadian serta aktivitas.

### 7. Diagram Aktivitas

Bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi – fungsi suatu sistem dan memberi tekanan pada aliran kendali antar objek. Pada tabel 2.2 berikut merupakan simbol – simbol yang ada pada diagram aktivitas.

**Tabel 2.2** Simbol – Simbol Aktivitas Diagram

Simbol	Keterangan
<i>Start state</i> 	Menunjukkan dimulainya suatu <i>workflow</i> pada sebuah <i>activity diagram</i> .
<i>End state</i> 	Menggambarkan akhir atau terminal dari pada sebuah <i>activity diagram</i> .
<i>State transition</i> 	Menunjukkan kegiatan selanjutnya setelah suatu kegiatan sebelumnya
<i>Decision</i> 	Suatu titik/point pada <i>activity diagram</i> yang mengindikasikan suatu kondisi dimana ada kemungkinan perbedaan transisi.
<i>Activity</i> 	Menunjukkan aktivitas yang ada dalam sistem



## 8. Diagram Komponen

Bersifat statis. Diagram komponen ini memperlihatkan organisasi serta ketergantungan sistem pada komponen-komponen yang telah ada sebelumnya.

## 9. Diagram *Deployment*

Bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan. Memuat simpul-simpul serta komponen-komponen yang ada di dalamnya.

### 2.10 *Storyboard*

(Subakti, 2006) *Storyboard* merupakan visualisasi ide dari aplikasi yang akan dibangun, sehingga dapat memberikan gambaran dari aplikasi yang akan dihasilkan. *Storyboard* dapat dikatakan juga *visual script* yang akan dijadikan outline dari sebuah proyek, ditampilkan *shot by shot* yang biasanya disebut dengan istilah *scene*.

*Storyboard* sekarang lebih banyak digunakan untuk membuat kerangka pembuatan *website* dan proyek media interaktif lainnya seperti iklan, film pendek, *games*, media pembelajaran interaktif ketik dalam tahap perancangan/desain. Baru-baru ini istilah "*Storyboard*" telah digunakan dibidang pengembangan *web*, pengembangan perangkat lunak dan perancangan instruksi untuk mempresentasikan dan menjelaskan kejadian interaktif seperti suara dan gerakan.

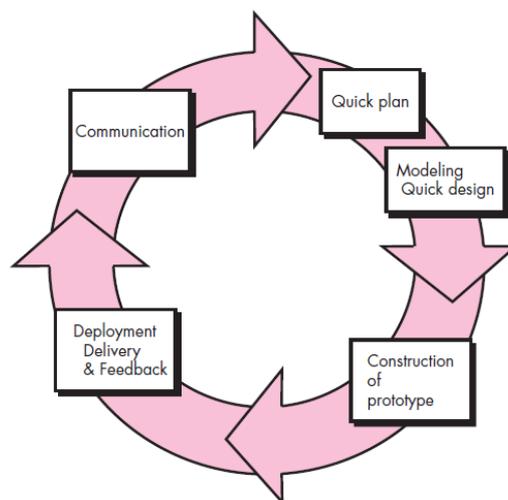
Sebuah *Storyboard* media interaktif dapat digunakan dalam antarmuka grafik pengguna untuk rancangan rencana desain sebuah *website* atau proyek interaktif sebagaimana alat visual untuk perencanaan isi. Sebaliknya, sebuah *site map* (peta)

atau *flowchart* (diagram alur) dapat lebih bagus digunakan untuk merencanakan arsitektur informasi, navigasi, *links*, organisasi dan pengalaman pengguna, terutama urutan kejadian menjadi kepentingan desain yang belum menyeluruh.

Salah satu keuntungan menggunakan *storyboard* adalah dapat membuat pengguna mengalami perubahan dalam alur cerita untuk memicu reaksi atau ketertarikan yang lebih dalam. Kilas balik, secara cepat menjadi hasil dari pengaturan *Storyboard* secara kronologis untuk membangun rasa penasaran dan ketertarikan.

### 2.11 Metode *Prototype Model*

(Pressman, 2002) menyatakan bahwa *Prototype* paradigma merupakan pelanggan efektif dalam merancang perangkat lunak. *Prototype* model dimulai dengan mengumpulkan kebutuhan. Pengembang dan pelanggan bertemu dan mendefinisikan objek keseluruhan dari perangkat lunak, mengidentifikasi segala kebutuhan yang diketahui dan kemudian melakukan “perancangan kilat”. Perancangan kilat berfokus pada penyajian dari aspek-aspek perangkat lunak tersebut yang akan nampak bagi pelanggan atau pemakai contohnya pendekatan *input* dan format *output*. Perancangan kilat membawa kepada konstruksi sebuah *prototype*. *Prototype* tersebut dievaluasi oleh pelanggan dan dipakai untuk menyaring kebutuhan pengembangan perangkat lunak. Dapat dilihat pada gambar 2.5.



**Gambar 2.5** *Prototype* paradigma

*Prototype model* juga dapat didefinisikan sebagai proses pengembangan suatu *prototype* secara cepat untuk digunakan terlebih dahulu dan ditingkatkan terus menerus sampai didapatkan sistem yang utuh. *Prototype model* merupakan proses yang digunakan untuk membantu pengembang perangkat lunak dalam membentuk *prototype* dari perangkat lunak yang harus dibuat. Proses pada model *prototyping* dapat dijelaskan sebagai berikut:

1. Pengumpulan kebutuhan : *Developer* dan klien bertemu dan menentukan tujuan umum , kebutuhan yang diketahui dan gambaran bagian-bagian yang akan dibutuhkan berikutnya.
2. Perancangan : Perancangan dilakukan cepat dan rancangan mewakili semua aspek perangkat lunak yang diketahui, dan rancangan ini menjadi dasar pembuatan *prototype*.
3. Evaluasi *prototype*: *Klien* mengevaluasi *prototype* yang dibuat dan digunakan untuk memperjelas kebutuhan perangkat lunak.

Perulangan ketiga proses ini terus berlangsung hingga semua kebutuhan terpenuhi. *Prototype-prototype* dibuat untuk memuaskan kebutuhan *klien* dan membangun perangkat lunak lebih cepat, namun tidak semua *prototype* bisa dimanfaatkan. Demi kebutuhan klien lebih baik *prototype* yang dibuat diusahakan dapat dimanfaatkan.

## **2.12 Blackbox Testing**

Menurut Pressman (2010, p495) *Black-Box* Testing berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan *engineers* untuk memperoleh set kondisi *input* yang sepenuhnya akan melaksanakan persyaratan fungsional untuk sebuah program