

BAB II

LANDASAN TEORI

2.1 Pengertian *Android*

Android merupakan sebuah sistem operasi yang berbasis Linux untuk telepon seluler seperti telepon pintar dan komputer *tablet*. *Android* menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti gerak. Antarmuka pengguna *Android* umumnya berupa manipulasi langsung, menggunakan gerakan sentuh yang serupa dengan tindakan nyata, misalnya menggeser , mengetuk , dan mencabit untuk memanipulasi objek di layar, serta papan ketik virtual untuk menulis teks.

Android adalah sistem operasi dengan sumber terbuka, dan *Google* merilis kodenya di bawah Lisensi Apache. Kode dengan sumber terbuka dan lisensi perizinan pada *Android* memungkinkan perangkat lunak untuk dimodifikasi secara bebas dan distribusikan oleh para pembuat perangkat , operator nirkabel, dan pengembang aplikasi. Selain itu, *Android* memiliki sejumlah besar komunitas pengembang aplikasi (apps) yang memperluas fungsionalitas perangkat, umumnya ditulis dalam versi bahasa pemrograman Java.

2.2 Database

2.2.1 Definisi Database

Menurut Yasin (2012,p.274) mengatakan Basis data (*database*) adalah kumpulan informasi yang akan disimpan didalam komputer secara sistematis, sehingga dapat digunakan oleh suatu program komputer untuk memperoleh informasi dari basis data tersebut. Basis data adalah sekumpulan data yang terhubung satu sama lain secara logika dan suatu deskripsi data yang dirancang untuk memenuhi kebutuhan informasi suatu organisasi. Pendefinisian basis data meliputi spesifikasi dari tipe data, struktur dan batasan dari data atau informasi yang akan disimpan. *Database*

merupakan salah satu komponen yang penting dalam sistem informasi, karena merupakan basis dalam menyediakan informasi pada para pengguna atau user.

2.2.2 Tujuan Database

1. Kecepatan dan Kemudahan (*Speed*)

Pemanfaatan basis data memungkinkan kita untuk dapat menyimpan data atau melakukan perubahan/manipulasi terhadap data atau menampilkan kembali data tersebut dengan lebih cepat dan mudah.

2. Efisiensi Ruang Penyimpanan (*Space*)

Karena keterkaitan erat antara kelompok dalam basis data, maka redundansi data pasti selalu ada. Dengan basis data, efisiensi/optimalisasi penggunaan ruang penyimpanan dapat dilakukan karena kita dapat melakukan penekanan jumlah redundansi data, baik menerapkan sejumlah pengkodean atau membuat relasi-relasi antar kelompok data yang saling berhubungan.

3. Keakuratan (*Accuracy*)

Pemanfaatan pengkodean atau pembentukan relasi antar data bersama dengan penerapan aturan/batasan tipe data, domain data, keunikan data dan sebagainya yang secara ketat dapat diterapkan dalam sebuah basis data, sangat berguna untuk menekan ketidakakuratan/penyimpanan data.

4. Ketersediaan (*Availability*)

Pertumbuhan data sejalan waktu akan semakin membutuhkan ruang penyimpanan yang besar. Padahal tidak semua data selalu kita gunakan/butuhkan. Karena itu kita dapat melakukan pemilih data, sehingga data yang sudah jarang kita gunakan dapat kita pindahkan ke dalam media penyimpanan *offline*.

5. Kelengkapan (*Completeness*)

Untuk mengkomodasi kebutuhan kelengkapan data yang semakin berkembang, maka kita tidak hanya dapat menambah *record-record* data, tetapi juga dapat melakukan perubahan struktur dalam basis data, baik dalam penambahan objek baru (tabel) atau dengan penambahan *field-field* baru pada suatu tabel.

6. Keamanan (*Security*)

Ada sejumlah sistem pengolahan basis data yang tidak menerapkan aspek keamanan dalam sebuah basis data.

7. Kebersamaan pemakain (*Sharebility*)

Pemakai basis data seringkali tidak terbatas pada satu pemakai saja atau di satu lokasi saja oleh satu sistem aplikasi.

2.3 Pengertian *Website Responsive*

Responsive Web Design adalah sebuah teknik yang digunakan untuk membuat *layout website* menyesuaikan diri dengan tampilan *device* pengunjung, baik ukuran maupun orientasinya. Jadi tampilan yang diakses melalui *SmartPhone* misalnya, itu akan berbeda tampilannya. *Responsive Web Design* untuk mengetes tampilan yang menggunakan teknik *Responsive* tersebut tidak harus mengaksesnya melalui *mobile device* ataupun *device* lainnya, namun cukup dengan *meminimize browser* saja ke dalam ukuran *mobile device* yang diinginkan.

Teknik ini sudah mulai banyak digunakan dikalangan *web desainer* luar maupun lokal dan memiliki beberapa keuntungan, salah satunya adalah penghematan penggunaan sub domain, yang tadinya kalau mengakses harus menggunakan kata “*mobile*” atau “*m*” didepan domainnya, sekarang cukup dengan mengetik domainnya saja dan tampilan akan menyesuaikan dengan *device* yang digunakan.

2.4 Html 5

HTML merupakan suatu bahasa teks untuk menampilkan secara visual perintah-perintah yang disusun dan ditayangkan dalam bentuk web. HTML adalah standar internet yang dirangkai sedemikian rupa dan dikendalikan oleh W3C (*World Wide Web Consortium*). Perintis HTML adalah Caillau TIM dengan Berners-lee Robert. Teknologi HTML pertama sekali dipergunakan dan dipublikasikan secara umum pada tahun 1990 dan hingga kini telah mengalami pemutakhiran sehingga muncullah HTML5 yaitu revisi kelima dari HTML.

Fitur baru HTML 5 antara lain;

- a) Unsur <canvas> untuk menggambar 2D.
- b) Unsur <video> dan <audio> untuk media pemutaran.
- c) Dukungan untuk penyimpanan lokal.
- d) Konten baru dengan elemen spesifik seperti <article>, <footer>, <header>, <nav>, <section>.

HTML5 adalah Generasi selanjutnya dari HTML sebelumnya. HTML5 diciptakan dalam pengembangan bahasa HTML untuk memperbaiki konten dan memperbarui teknologi multimedia yang sudah ada pada HTML 4 dan versi sebelumnya agar mudah dijalankan oleh browser dan mudah dimengerti. HTML5 merupakan kerjasama antara *World Wide Web Consortium* (W3C) dan *Web Hypertext Application Teknologi Working Group* (WHATWG). WHATWG bekerja dengan bentuk web dan aplikasi, dan W3C bekerja dengan XHTML 2.0. Pada tahun 2006, mereka memutuskan untuk bekerja sama dan membuat versi baru dari HTML yaitu HTML5.

Struktur HTML5 adalah sebagai berikut :

```

<!DOCTYPE html>
<html>
<head>
<title>Judul</title>
</head>
<body> Isi dan Konten </body>

```

</html>

2.5 MySQL

MySQL dikembangkan oleh sebuah perusahaan Swedia bernama MySQL AB yang pada saat itu bernama TcX Data Konsult AB sekitar tahun 1994-1995, namun cikal bakal kodenya sudah ada sejak 1979. Awalnya TcX membuat MySQL dengan tujuan mengembangkan aplikasi web untuk klien. TcX merupakan perusahaan pengembang *software* dan konsultan *database*. Saat ini MySQL sudah diakuisisi oleh Oracle Crop. MySQL adalah salah satu jenis *database server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi *web* yang mana *database* sebagai sumber dan pengelolaan datanya. Kepopuleran MySQL antara lain karena MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses *database* sehingga mudah untuk digunakan. MySQL juga bersifat *open source* dan *free* pada berbagai *platform* kecuali pada *windows* yang bersifat *shareware*. MySQL didistribusikan dengan lisensi open source GPL (*General Public License*) mulai versi 3.23, pada bulan Juni 2000. *Software* MySQL bisa diunduh di <http://mysql.org> atau <http://www.mysql.com>. (Arief, 2011, Hal : 151)

2.6 PHP (*Hypertext Preprocessor*)

PHP diciptakan oleh Rasmus Lerdorf, seorang pemrogram C yang handal. Semula PHP hanya digunakan untuk mencatat jumlah pengunjung pada *homepagenya*. Rasmus adalah seorang pendukung *open source*. Karena itulah ia mengeluarkan *Personal Home Page Tools* versi 1.0 secara gratis. Setelah mempelajari YACC dan GNU Bison, Rasmus menambah kemampuan PHP 1.0 dan menerbitkan PHP 2.0. PHP mudah dibuat dan cepat dijalankan, PHP dapat berjalan dalam web server yang berbeda dan dalam sistem operasi yang berbeda pula. PHP dapat berjalan di sistem operasi UNIX, *Windows 98*, *Windows XP*, *Windows NT*, dan Macintosh (Hakim : 2006).

PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP bernama FI (*Form Interpreted*). Pada saat tersebut PHP adalah

sekumpulan script yang digunakan untuk mengolah data form dari web. Perkembangan selanjutnya adalah Rasmus melepaskan kode sumber tersebut dan menamakannya PHP/FI, pada saat tersebut kepanjangan dari PHP/FI adalah *Personal Home Page/Form Interpreter*. Dengan pelepasan kode sumber ini menjadi *open source*, maka banyak *programmer* yang tertarik untuk ikut mengembangkan PHP. Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini interpreter sudah diimplementasikan dalam C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan. Pada tahun 1997, sebuah perusahaan bernama Zend, menulis ulang interpreter PHP menjadi lebih bersih, lebih baik dan lebih cepat. Kemudian pada Juni 1998 perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan nama rilis tersebut menjadi PHP 3.0. Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai. Versi ini banyak dipakai sebab versi ini mampu dipakai untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan proses dan stabilitas yang tinggi. Pada Juni 2004, Zend merilis PHP 5.0. Versi ini adalah versi mutakhir dari PHP. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Dalam versi ini juga dikenalkan model pemrograman berorientasi objek baru untuk menjawab perkembangan bahasa pemrograman ke arah pemrograman berorientasi objek (Kurniawan : 2007)..

Kelebihan PHP :

1. PHP adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. *Web Server* yang mendukung PHP dapat ditemukan dimana - mana dari mulai IIS sampai dengan apache, dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis - milis dan developer yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, PHP adalah bahasa scripting yang paling

mudah karena referensi yang banyak. PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin (*linux, unix, windows*) dan dapat dijalankan secara runtime melalui *console* serta juga dapat menjalankan perintah-perintah system.

Selain berbagai kemudahan, PHP memiliki beberapa masalah atau kekurangan. Pertama, dari segi bahasa ia bukanlah bahasa yang ideal untuk pengembangan berskala besar. Kekurangan utama adalah tidak adanya *name space*. *Name space* merupakan sebuah cara untuk mengelompokkan nama variabel atau fungsi dalam susunan hirarkis.

2.7 Perangkat Lunak yang Digunakan

Untuk membangun *website* diperlukan beberapa perangkat lunak yang digunakan dalam membangun *website* tersebut. Beberapa perangkat lunak yang digunakan adalah sebagai berikut :

2.7.1 *Notepad ++*

Notepad++ merupakan sebuah penyunting teks dan penyunting kode sumber yang tersedia dalam beberapa sistem operasi seperti Windows, Linux, dan Mac OS X. *Notepad++* menggunakan komponen Scintilla untuk dapat menampilkan dan menyunting teks dan berkas kode sumber berbagai bahasa pemrograman. *Notepad++* didistribusikan sebagai perangkat lunak bebas. Proyek ini dilayani oleh *Sourcefoge.net* dengan telah diunduh lebih dari 27 juta kali dan dua kali memenangkan penghargaan *SorceForge Community Choice Award for Best Developer Tools*.

2.7.2 *Framework Bootstrap*

Menurut Husein (2013, p.4) *Bootstrap* merupakan *framework* ataupun *tools* untuk membuat aplikasi *website* ataupun situs *web responsive* secara cepat, mudah dan gratis, karena *website* yang dibangun oleh peneliti merupakan *website* yang dapat diakses dalam perangkat *mobile* ataupun *personal*

computer. Bootstrap terdiri dari CSS dan HTML untuk menghasilkan *Grid, Layout, Typography, Table, Form, Navigation*, dan lain lain. Di dalam *bootstrap* juga sudah terdapat *jquery plugins* untuk menghasilkan komponen UI yang cantik seperti *Transitions, Modal, Dropdown, Scrollspy, Tooltip, Tab, Popever, Alert, Button, Carousel*, dan lain lain.

2.7.3 XAMPP

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

2.8 Metode Pengembangan Perangkat Lunak

2.8.1 Metode *Prototype*

Pressman (2012, p.40) menyatakan bahwa *Prototyping Paradigma* dimulai dengan pengumpulan kebutuhan, pengembang bertemu dengan pengguna dan mengidentifikasi objektif keseluruhan dari perangkat lunak, selanjutnya mengidentifikasi segala kebutuhan yang diketahui secara garis besar di mana definisi-definisi lebih jauh merupakan keharusan, kemudian dilakukan perancangan kilat, lalu diakhiri dengan evaluasi *prototyping* .

1. Pengumpulan kebutuhan

Developer dan klien bertemu untuk menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian-bagian yang akan dibutuhkan berikutnya. Detail kebutuhan mungkin tidak dibicarakan disini, pada awal pengumpulan kebutuhan.

Selanjutnya peneliti akan melakukan analisis terhadap data apa saja yang dibutuhkan, seperti analisis terhadap sistem yang berjalan, analisis kebutuhan perangkat lunak, analisis kebutuhan perangkat keras, dan analisis kebutuhan materi pembelajaran.

2. Perancangan

Perancangan dilakukan dengan cepat dan rancangan mewakili semua aspek software yang diketahui, dan rancangan ini menjadi dasar pembuatan *prototype*. Dalam tahap ini peneliti akan membangun sebuah versi *prototype* yang dirancang kembali dimana masalah-masalah tersebut diselesaikan

3. Evaluasi *prototype*

Pada tahap ini, calon pengguna mengevaluasi *prototype* yang dibuat dan digunakan untuk memperjelas kebutuhan *software*. *Software* yang sudah jadi dijalankan dan akan dilakukan perbaikan apabila kurang memuaskan. Perbaikan termasuk dalam memperbaiki kesalahan atau kerusakan yang tidak ditemukan pada langkah sebelumnya.

Kelebihan *Prototype Model* adalah:

- a) End user dapat berpartisipasi aktif.
- b) Penentuan kebutuhan lebih mudah diwujudkan.
- c) Mempersingkat waktu pengembangan *software*.

Kekurangan *Prototype Model* adalah:

- a) Proses analisis dan perancangan terlalu singkat.
- b) Mengesampingkan alternatif pemecahan masalah.
- c) Biasanya kurang fleksibel dalam menghadapi perubahan.
- d) *Prototype* yang dihasilkan tidak selamanya mudah dirubah.
- e) *Prototype* terlalu cepat selesai.

2.8.2 Unified Modeling Language (UML)

Yasin (2012,p.194) mendefinisikan *Unified Modelling Language* (UML) adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk

visualisasi, merancang dan mendokumentasikan sistem piranti lunak, UML menawarkan sebuah standar untuk merancang model sebuah sistem. Tujuan Penggunaan UML yaitu untuk memodelkan suatu sistem yang menggunakan konsep berorientasi objek dan menciptakan bahasa pemodelan yang dapat digunakan baik oleh manusia maupun mesin.

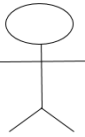

Menurut Verdi Yasin (2012,p.268) tipe-tipe diagram UML adalah sebagai berikut :

1. *Use Case Diagram*


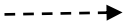
Use case diagram adalah gambar dari beberapa atau seluruh aktor dan *use case* dengan tujuan yang mengenali interaksi mereka dalam suatu sistem. *Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem, yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dan sistem.

Dalam *use case diagram* terdapat istilah seperti aktor, *use case* dan *case relationship*. Penjelasan simbol pada tabel 2.1.

Tabel 2.1 Simbol Use Case.

Simbol	Keterangan
	Aktor : Seseorang atau sesuatu yang berinteraksi dengan sistem yang sedang dikembangkan.
	<i>Use case</i> : perangkat tertinggi dari fungsionalitas yang dimiliki sistem.








Tabel 2.1 Simbol Use Case.(lanjutan)

	<i>Association</i> : adalah relasi antara actor dan <i>use case</i> .
	<i>Generalisasi</i> : untuk memperlihatkan struktur pewaris yang terjadi.

2. *Activity Diagram*

Activity diagram menggambarkan rangkaian aliran dari aktifitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi. *Activity Diagram* berupa flow chart yang digunakan untuk memperlihatkan aliran kerja dari sistem. Notasi yang digunakan dalam *activity diagram* adalah sebagai berikut :

Tabel 2.2 Simbol *Activity Diagram*

Simbol	Keterangan
	<i>Activity</i> : Menunjukkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	<i>Initial Node</i> : Bagaimana objek dibentuk atau diawali
	<i>Activity Final Node</i> : Bagaimana objek dibentuk dan diakhiri.
	<i>Decision</i> : Asosiasi percabangan dimana jika ada pilihan aktifitas lebih dari satu.
	<i>Swimlane</i> : Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktifitas yang terjadi.
	<i>Join</i> : Digunakan untuk menunjukkan kegiatan yang digabungkan.
	<i>Fork</i> : Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel

3) *Sequence Diagram*

Sequence diagram menggambarkan kolaborasi dinamis antara sejumlah dan untuk menunjukkan rangkaian pesan yang dikirim antar objek juga interaksi antar objek, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem. *Sequence* diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu.

Dalam *sequence* diagram terdapat 2 simbol yaitu :

- a. Actor, untuk menggambarkan pengguna sistem.
- b. *Lifeline*, untuk menggambarkan kelas dan objek.