

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Penelitian ini dilakukan tidak terlepas dari hasil penelitian-penelitian terkait dari jurnal dan referensi yang pernah dilakukan sebagai bahan perbandingan dan kajian. Adapun hasil-hasil penelitian yang dijadikan perbandingan tidak terlepas dari topik penelitian yaitu Aplikasi Pengenalan Ilmu Taj-wid Berbasis Android.

Berdasarkan penelitian Arif Tri Septiawan (2012) dimana penelitian ini mengenai Pengenalan dan Pembelajaran Cara Membaca Al-Qur'an (Ilmu Tajwid) Berbasis *Mobile* Android. Hasilnya menunjukkan bahwa Al-Qur'an adalah kitab suci umat islam yang dapat membawa kebahagiaan karena kitab suci Al-Qur'an merupakan wahyu Allah yang diturunkan kepada Rasul-Nya untuk membimbing umatnya ke jalan yang benar. Al-Qur'an yang berbentuk bukupun sekarang ini dapat di unduh dan dimasukkan ke telepon seluler, agar lebih praktis dalam membacanya. Dilihat dari fakta tersebut maka akan di buat aplikasi tajwid berbasis *mobile* yang berisi tentang materi-materi tajwid sehingga cocok untuk *user* yang memiliki mobilitas tinggi.

Berdasarkan penelitian yang dilakukan oleh Vitrya, dkk (2015) mengenai Pengembangan Aplikasi Pembelajaran Iqra' Dan Tajwid Berdasarkan Metode Asy-Syafi'i Menggunakan Sistem Multimedia, Vitrya, dkk menyimpulkan dengan menerapkan basis *android* pada aplikasi pembelajaran iqra' dan tajwid

berdasarkan metode Asy-Syafi'i, aplikasi dapat digunakan dimanapun dan kapanpun.

II.2. Landasan Teori

Landasan teori merupakan teori yang relevan yang digunakan untuk menjelaskan tentang variabel yang akan diteliti dan sebagai dasar untuk memberi jawaban sementara terhadap rumusan masalah yang diajukan (hipotesis), dan penyusunan instrument penelitian. Teori yang digunakan bukan sekedar pendapat dari pengarang atau pendapat lain, tetapi teori yang benar-benar telah teruji kebenarannya. Adapun beberapa landasan teori yang penulis jelaskan sebagai berikut:

II.2.1. Aplikasi

Aplikasi adalah penggunaan dalam suatu komputer, instruksi (*instruction*), atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses *input* menjadi *output*. Aplikasi adalah Program yang dibuat oleh manusia yang berfungsi untuk menyelesaikan permasalahan-permasalahan masalah yang akan dihadapi. (Zulfauzi ; 2015 : 57).

Aplikasi adalah penggunaan dalam suatu komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses input menjadi *output*. (Vitrya, dkk, 2015 : 2).

II.2.2. Android

Android adalah sistem operasi untuk *smartphone dan tablet*, sistem operasi dapat diilustrasikan sebagai jembatan antara piranti (*device*) dan penggunanya,

sehingga pengguna bisa berinteraksi dengan *device-nya* dan menjalankan aplikasi-aplikasi yang tersedia pada *device*. *Android* juga merupakan Sistem Operasi perangkat bergerak yang berbasis Linux yang diperuntukan untuk penggunaan ke perangkat bergerak bergerak seperti *mobile* dan komputer tablet layar sentuh. (Zulfauzi ; 2015 : 58).

Android merupakan generasi baru *platform mobile*, *platform* yang memberikan pengembang untuk melakukan pengembangan sesuai dengan yang diharapkannya. Sistem operasi yang mendasari *Android* dilisensikan dibawah GNU, *General Public Lisensi* Versi 2 (GPLv2), yang sering dikenal dengan istilah *copyleft* lisensi dimana setiap perbaikan pihak ketiga harus terus jatuh dibawah *terms*. (Vitrya, dkk, 2015 : 2).

II.2.3. Versi *Android*

Telepon pertama yang memakai sistem operasi *Android* adalah *HTC Dream*, yang dirilis pada 22 oktober 2008. Pada penghujung tahun 2010 diperkirakan hampir semua vendor seluler didunia menggunakan *Android* sebagai *operating system*. Adapun versi-versi *Android* yang pernah dirilis adalah sebagai berikut :

1. *Android* Versi 1.1

Pada 9 MAret 2009, *Google* merilis *Android* versi 1.1. *Android* versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam, alarm, *voice search* (pencarian suara), pengiriman pesan dengan *Gmail*, dan pemberitahuan *email*.

2. *Android* Versi 1.5 (*Cupcake*)

Pada pertengahan Mei 2009, *Google* kembali merilis telepon seluler dengan menggunakan *Android* dan *SDK (Software Development Kit)* dengan versi 1.5 (*Cupcake*). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton *video* dengan modus kamera, *upload video* ke *youtube* dan gambar ke *picasa* langsung dari telepon, dukungan *Bluetooth A2DP*, kemampuan terhubung secara otomatis ke *headset Bluetooth*, animasi layar, dan *keyboard* pada layar yang dapat disesuaikan dengan sistem.

3. *Android* Versi 1.6 (*Donut*)

Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik disbanding sebelumnya, penggunaan baterai indikator dan control *applet VPN*. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus, kamera, *camcorder* dan galeri yang diintegrasikan, *CDMA/EVDO*, 802.1x, *VPN*, *Gestures*, dan *Text-to-speech engine*, kemampuan *dial* kontak, teknologi *text to change speech* (tidak tersedia pada semua ponsel, pengadaan resolusi *VWGA*).

4. *Android* Versi 2.0/ 2.1 (*Eclair*)

Pada 3 Desember 2009 kembali diluncurkan ponsel *Android* dengan versi 2.0/ 2.1 (*Eclair*), perubahan yang dilakukan adalah pengoptimalan *hardware*, peningkatan *Google Maps* 3.1.2, perubahan *UI* dengan *browser* baru dan dukungan *HTML5*, daftar kontak yang baru, dukungan *flash* untuk kamera 3,2 *MP*, *digital zoom*, *bluetooth* 2.1.

5. *Android* Versi 2.2 (*Froyo: Frozen Yoghurt*)

Pada bulan Mei 2010 *Android* versi 2.2 Rev 1 diluncurkan. *Android* inilah yang sekarang banyak beredar di pasaran, salah satunya adalah dipakai di *Samsung FX tab* yang sudah ada di pasaran. Fitur yang tersedia di *Android* versi ini sudah kompleks diantaranya adalah :

- a. Kerangka aplikasi memungkinkan penggunaan dan penghapusan komponen yang tersedia.
 - b. *Dalvik Virtual Machine* dioptimalkan untuk perangkat *mobile*.
 - c. Grafik: grafik 2D dan grafis 3D berdasarkan *libraries OpenGL*.
 - d. *SQLite*: untuk penyimpanan data.
 - e. Mendukung media: *audio*, *video*, dan berbagai *format* gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF).
 - f. *GSM*, *Bluetooth*, *EDGE*, *3G*, dan *Wifi* (*hardware independent*).
 - g. Kamera, *Global Positioning System (GPS)*, kompas, dan *accelerometer* (tergantung *hardware*).
6. *Android* Versi 2.3 (*Gingerbread*)

Android versi 2.3 diluncurkan pada Desember 2010, hal-hal yang direvisi dari versi sebelumnya adalah kemampuan seperti berikut :

- a. *SIP-based VoIP*
- b. *Near Field Communications (NFC)*
- c. *Gyroscope* dan sensor
- d. *Multiple cameras support*
- e. *Mixable audio effects*
- f. *Download Manager*

7. *Android* Versi 3.0 (*Honeycomb*)

Dirilis Februari 2011 sebagai *Android* 3.0 revisi 1 serta *Android* versi 3.0 *revision* 2 telah dirilis pada juli 2011.

8. *Android* Versi 3.1

Dirilis Mei 2011, sedangkan *Android* 3.1 revisi 2 juga dirilis mei 2011, serta *Android* 3.1 *revision* 3 dirilis pada juli 2011.

9. *Android* Versi 3.2

Dirilis Juli 2011.

10. *Android* Versi 4.0

Dirilis November 2011.

11. *Android* Versi 4.1

12. *Android* Versi 4.2

13. *Android* Versi 4.3

Android versi 3.0 ke atas adalah generasi *platform* yang digunakan untuk *tablet pc*. Sementara versi 4.0 sudah merupakan *platform* yang bias dipakai di *smartphone* dan *tablet pc*. (Nazruddin Safaat H, 2014).

II.2.4. Java

Java merupakan bahasa pemrograman berorientasi objek yang dapat digunakan untuk membuat dan menjalankan perangkat lunak pada komputer dan berbagai platform. (Zulfauzi ; 2015 : 58).

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai computer maupun telepon genggam. Bahasa pemrograman ini dibuat oleh James Gosling saat masih bergabung di Sun Microsystems, di mana saat ini merupakan

bagian dari Oracle yang dirilis pada tahun 1995. Bahasa ini banyak mengadopsi sintaks yang terdapat pada C dan C++, tetapi dengan sintaksis mode objek yang lebih sederhana. *Java* merupakan bahasa pemrograman yang bersifat umum/nonspesifik dan secara khusus didesain untuk memanfaatkan implementasi semaksimal mungkin. Fungsi *Java* memungkinkan aplikasi *Java* mampu berjalan di beberapa *platform* sistem operasi yang berbeda. *Java* dikenal pula dengan slogannya “tuliskan sekali, jalankan di mana pun”. Saat ini *Java* secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis *web*. (Wahana Komputer, 2015 : 2).

II.2.5. Extensible Markup Language (XML)

Extensible Markup Language (XML) adalah sebuah bahasa markah untuk mendeskripsikan data. XML merupakan turunan (*subset*) atau versi ringkas dari SGML (*Standart Generalized Markup Language*), sedangkan SGML merupakan sebuah standar ISO untuk format dokumen. SGML tidak berisi berupa *tag-tag* siap pakai seperti halnya bahasa HTML, melainkan berupa aturan-aturan standar dalam pembuatan *tag-tag* format dokumen. SGML banyak dipakai untuk mengelola dokumen dalam jumlah besar, frekuensi revisi tinggi dan dibutuhkan dalam beragam format tampilan. SGML jarang dipakai karena sangat rumit dan kompleks. XML dibuat dengan konsep yang lebih sederhana dan ringkas, tujuannya agar bisa dipakai sebagai aplikasi *desktop* dan jaringan internet. (Yenni dan Shamir, 2012 : 107).

II.2.6. Netbeans 8.0

Netbeans 8.0 adalah salah satu aplikasi *Intergrated Development Environment* (IDE) yang digunakan oleh *developer software computer* untuk menulis, meng-*compile*, mencari kesalahan, dan untuk menyebarkan program. *Netbeans* mempunyai sekumpulan *software* modul yang dipakai untuk membuat suatu aplikasi. Modul merupakan arsip *Java* yang memuat kelas-kelas *Java* yang berinteraksi dengan *Netbeans Open API*. (Wahana Komputer, 2015 : 20).

II.2.7. Taj-wid

Kata Taj-wid berasal dari bahasa arab yaitu *jawwada-yujawwidu-Taj-wid* yang artinya membaguskan. Taj-wid adalah membaguskan bacaan huruf-huruf atau kalimat-kalimat Alquran satu persatu dengan terang, teratur, perlahan, dan tidak terburu-buru sesuai dengan kaidah-kaidah ilmu Taj-wid. Kesimpulannya, ilmu Taj-wid adalah ilmu yang mempelajari cara membaca Alquran dengan baik dan benar sehingga sempurna maknanya. (Vitrya, 2015 : 2).

Taj-wid adalah pengetahuan tentang kaidah serta cara-cara membaca Alquran dengan mengeluarkan huruf dari mahrojnya serta memberihak dan mustahaknya. Secara garis besar Ilmu Taj-wid adalah ilmu yang mengajarkan tentang bagaimana cara membaca Alquran yang baik dan benar. Tujuan ilmu Taj-wid adalah memelihara bacaan Alquran dari kesalahan dan perubahan serta memelihara lisan (mulut) dari kesalahan membaca. Kesalahan dalam membaca Alquran dikategorikan dalam dua macam, yaitu:

1. Al-Lakhnu Al-Jaliy (Kesalahan besar/fatal) Adalah kesalahan dalam membaca Alquran yang dapat mengubah arti dan menyalahi urfquro. Melakukan kesalahan ini hukumnya haram. Yang termasuk diantaranya ialah:

- a. Kesalahan makhroj huruf. Biasanya terjadi pada pengucapan huruf-huruf yang serupa seperti 'ain dan hamzah, cha, ha, khodanghain, ta dan sebagainya.
 - b. Salah membaca mad, seperti bacaan pendek dibaca panjang atau sebaliknya.
 - c. Salah membaca harokat. Seperti harokat di akhir kata sebagai yang menunjukkan jabatan kata.
2. Al-Lakhnu Al-Khofiy (Kesalahan kecil) Adalah kesalahan dalam membaca. Belajar ilmu Taj-wid itu hukumnya Fardhu Kifayah, sedang membaca Alquran dengan baik (sesuai dengan ilmu Taj-wid) itu hukumnya Fardhu Ain. Bahkan bacaan Alquran kita akan menentukan sah atau tidaknya sholat kita, karena ada beberapa rukun sholat yang berkenaan dengan bacaan yang harus fasih dan benar sehingga tidak merubah maknanya. Di antaranya seperti membaca surat Al-Fatihah, bacaan tahyat akhir serta sholawat kepada Nabi Muhammad dalam sholat. (Permatasari dan Falah, 2012 : 5).

II.2.8. *Unified Modeling Language (UML)*

Menurut Windu Gata (2013) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language (UML)*. *UML* adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.

UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. *UML* saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar


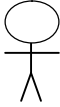

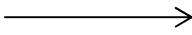
bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem. (Gellysa Urva dan Helmi Fauzi Siregar ; 2015 : 93).

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis *UML* adalah sebagai berikut :

1. *Use case* Diagram

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel II.1 dibawah ini :

Tabel II.1. Simbol *Use Case*

Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, dan dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki <i>control</i> terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.




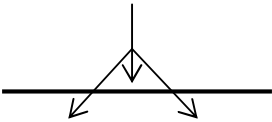
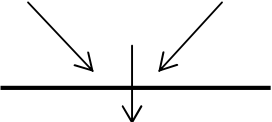
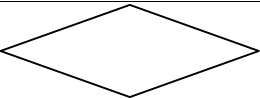
----->	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
<-----	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar ; 2015 : 94)

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.2 dibawah ini :

Tabel II.2. Simbol *Activity Diagram*

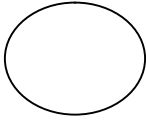
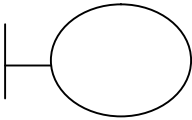
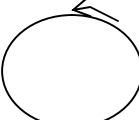
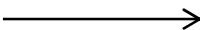
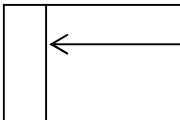

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
New Swimlane	<i>Swimlane</i> , untuk menunjukkan siapa melakukan apa.

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar ; 2015 : 94)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.3 dibawah ini :

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar ; 2015 : 95)

4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti yang dapat dilihat pada tabel II.4 dibawah ini :

Tabel II.4. *Multiplicity Class Diagram*

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar ; 2015 : 95)