BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Untuk mendukung keberhasilan penelitian ini, penyusun melakukan pendekatan teoritis melalui beberapa literatur yang berhubungan dengan penelitian yang dilakukan :

1. Viska, dkk, (2018) dengan judul "Aplikasi Pembelajaran Ilmu *Tajwid* Berbasis Web Interaktif" pada *Journal of Applied Informatics (IJAI)*. Penelitian ini menghasilkan sebuah aplikasi pembelajaran ilmu *Tajwid* berbasis web yang menyediakan informasi serta fasilitas multimedia yang dapat membuat belajar lebih menarik, ilustratif dan interaktif.
2. Hasanuddin dan Santi, (2019) dengan judul Hubungan Pemahaman Ilmu *Tajwid* Dengan Kemampuan Membaca Al-Qur'an di Madrasah Aliyah Negeri 1 Kota Bogor pada jurnal Aksara *Public*. Penelitian ini menghasilkan adanya temuan kolerasi antara pemahaman ilmu *Tajwid* dengan kemampuan membaca Al-Qur'an di Madrasah Aliyah Negeri 1 kota Bogor yang menghasilkan Hipotesis nol (Ho) ditolak, dan berarti Hipotesis Alternatif (Ha)".
3. Helmi, dkk, (2018) dengan judul "Perancangan Aplikasi Komik Hadist Berbasis Multimedia" pada Jurnal Teknologi Informasi. Penelitian ini menghasilkan aplikasi komik hadist, pengguna memiliki media tempat pembelajaran yang lebih efektif pada perkembangan zaman sekarang dan

mempermudah pengguna dalam membaca komik hadist dan mengetahui isi cerita komik hadist tersebut.

4. Juharna, dkk, (2016) dengan judul "Perancangan Aplikasi Multimedia Interaktif Untuk Pembelajaran Ilmu *Tajwid* Pada Pondok Pesantren Al-Mansyuriyah" pada Jurnal Sisfotek Global. Penelitian ini menghasilkan sebuah aplikasi yang bertujuan untuk memberikan bentuk baru pembelajaran yang tidak mudah bosan sehingga penulis membuat media pembelajaan dengan media interaktif guna menarik minat pengguna terutama anak-anak.
5. Refni Wahyuni, (2019) dengan judul "Media Pembelajaran *Tajwid* Berbasis Android Untuk Siswa Tingkat Dasar" pada Jurnal Ilmu Komputer. Penelitian ini menghasilkan sebuah aplikasi ilmu *Tajwid* secara interaktif untuk anak usia sekolah dasar dan aplikasi ini juga akan menampilkan gambar beserta suara.

II.2. Pengertian Rancang Bangun

Menurut Devy (2018), rancang bangun adalah suatu kegiatan menerjemahkan hasil analisa ke dalam bentuk paket perangkat lunak kemudian menciptakan sistem tersebut ataupun memperbaiki sistem yang sudah ada.

Sedangkan menurut Andi, dkk, (2018), rancang bangun adalah program yang menentukan aktifitas pemrosesan informasi yang dibutuhkan untuk penyelesaian tugas-tugas khusus dari pemakai atau pengguna komputer.

Dengan demikian, penulis menyimpulkan pengertian dari rancang bangun adalah suatu kegiatan menciptakan suatu sistem ataupun memperbaiki sistem yang

sudah ada untuk menyelesaikan tugas-tugas khusus dari pemakai atau pengguna.

II.3. Aplikasi

Aplikasi merupakan penerapan, menyimpan suatu hal, data, permasalahan, pekerjaan ke dalam suatu sarana atau media yang dapat digunakan untuk diterapkan menjadi sebuah bentuk yang baru.(Helmi, dkk, 2018)

Pengertian aplikasi secara umum adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya.(Helmi, dkk, 2018)

Dengan demikian, penulis menyimpulkan pengertian aplikasi adalah suatu perangkat lunak komputer yang menyimpan suatu hal, data, permasalahan pekerjaan yang kemudian diterapkan secara khusus dan terpadu sesuai kemampuan yang dimilikinya.

II.4. Animasi

Menurut Nazmi (2017), animasi merupakan serangkaian gambar gerak cepat yang terus menerus memiliki hubungan satu dengan yang lainnya, yang awalnya dari potongan gambar yang disatukan sehingga terlihat terlihat hidup.

Sedangkan menurut Syaipul, dkk, (2019), animasi adalah sebuah gambar yang bergerak dari sekumpulan objek yang telah disusun secara runtut dan

menghasilkan pergerakan objek yang dimaksud berupa gambar manusia, tulisan, sketh, hewan, dan benda.

Dengan demikian, penulis menyimpulkan pengertian animasi adalah suatu serangkaian gambar yang memiliki hubungan satu dengan yang lainnya sehingga menhasilkan pergerakan suatu objek.

II.5. Hukum *Tajwid*

Tajwid menurut istilah adalah ilmu yang memberikan segala pengertian tentang huruf, baik hak-hak huruf maupun hukum-hukum baru yang rimbul setelah hak-hak huruf dipenuhi. Hukum-hukum itu sendiri diantaranya seperti hukum nun mati, hukum mim mati, hukum *idgham* dan lain sebagainya(Hasanuddin dan Santi, 2019).

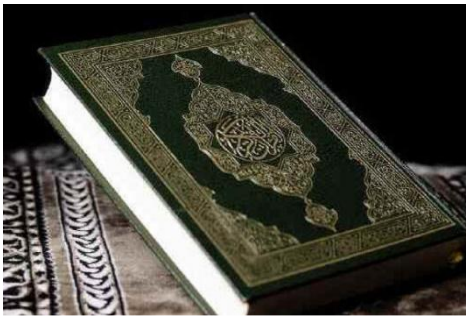
Dalam pengertian lain juga disebutkan Ilmu *Tajwid* adalah pengetahuan tentang kaidah serta cara-cara membaca AlQur'an dengan mengeluarkan huruf dan makhrojnya serta memberi hak dan mustahaknya.(Juharna, dkk, 2016)

Dengan demikian, penulis menyimpulkan pengertian hukum tajwid adalah suatu aturan atau tatanan yang harus diphami sebelum pembacaan ayat Al-Qur'an yang membahas tentang cara mengeluarkan huruf dan *makhrojnya* serta memberi hak dan mustahaknya.

II.6. Al-Qur'an

Al-Qur'an merupakan sumber pengetahuan, yang telah diturunkan Allah SWT kepada Nabi Muhammad SAW, hal ini merupakan sifat Allah sendiri yang

maha pencipta dan maha mengetahui, sehingga sudah sewajarnya jika Al-Qur'an sarat dengan ilmu pengetahuan. Al-Qur'an adalah kalamullah yang diturunkan kepada Nabi Muhammad SAW dalam bentuk bahasa Arab dengan perantara malikat jibril. Sedangkan hal-hal lain yang di nuklikan kepada kita dengan cara mutawatir, di awali dengan surat Al-fatihah dan di akhiri dengan surat An-Nas, serta di di tulis dalam mushaf, itu menyangkut hal-hal yang bersifat teknis bagi penyampaian dan pemeliharaan Al-Qur'an (Hasanuddin dan Santi, 2019)



Gambar II.1 Al-Qur'an
Sumber : (Hasanuddin dan Santi, 2019)

II.7. Android

Android merupakan subset perangkat lunak untuk perangkat *mobile* yang meliputi sistem operasi, *middleware*¸ dan aplikasi inti yang dirilis oleh *Google.* Android adalah sistem operasi bergerak (*Mobile Operating System*) yang mengadopsi sistem operasi *linux*, namun telah dimodifikasi. Android diambil alih oleh *Google* pada tahun 2005 dari Android, *Inc* sebagai bagian strategi untuk mengisi pasar sistem operasi bergerak. *Google* mengambil alih seluruh hasil kerja Android termasuk tim yang mengembangkan Android.(Hendra, dkk, 2015)

Gambar II.2 Android
Sumber : (Hendra, dkk, 2015)

II.7.1. Versi Android

Android memiliki versi-versi sebagai berikut :

1. Android 1.5 *Cupcake*

Cupcake dirilis 30 April 2009. *Cupcake* menjadi versi android pertama yang menggunakan nama makanan. Dahulu katanya versi ini seharusnya versi 1.2, namun *Google* memutuskan untuk membuat revisi besar dan membuatnya menjadi versi 1.5 *Cupcake* adalah kue kecil yang dipanggang dalam cetakan berbentuk *cup*. (Hendra, dkk, 2015)

2. Android 1.6 *Donut*

Android V1.6, *codename Donut*, dirilis pada 15 September 2009. Pada versi ini diperbaiki beberapa kesalahan *reboot*, perubahan fitur foto dan video dan integrasi pencarian yang lebih baik. Donat merupakan panganan berbentuk cincin, bulat bolong tengah. Adonan donat dimasak dengan cara digoreng dan biasanya disajikan dengan toping diatasnya. (Hendra, dkk, 2015)

3. Android 2.0/2.1 *Éclair*

Android 2.0/2.1 *Eclair* dirilis 26 Oktober 2009. *Eclair* adalah makanan penutup yakni kue yang biasanya berbentuk persegi panjang yang dibuat dengan krim di tengah dan lapisan cokelat di atasnya. (Hendra, dkk, 2015)

4. Android 2.2 *Froyo*

Dirilis 20 Mei 2010, menggunakan *codename Froyo*, yang merupakan makan penutup yang nama merek sebuah produk yang terbuat dari *Yoghurt*. *Froyo* singkatan dari *Frozen Yoghurt*, *Froyo* adalah *yoghurt* yang telah mengalami proses pendinginan, sehingga secara terlihat sama seperti es krim. (Hendra, dkk, 2015)

5. Android 2.3 *Gingerbread*

Android versi 2.3 *Gingerbread* dirilis resmi tanggal 6 Desember 2010. *Gingerbread* merupakan jenis kue kering yang dengan rasa jahe. Kue jahe biasanya dibuat pada perayaan hari libur akhir tahun di Amerika. Biasanya cemilan kering ini dicetak berbentuk tubuh manusia. (Hendra, dkk, 2015)

6. Android 3.0 *Honeycomb*

Dirilis tanggal 22 February 2011. *Honeycomb* adalah sereal sarapan manis yang sudah dibuat oleh Posting Sereal. Seperti namanya, *Honeycomb*/sarang lebah, sereal ini terbuat dari potongan jagung berbentuk sarang lebah dengan rasa madu. (Hendra, dkk, 2015)

7. Android 4.0 *Ice Cream Sandwich*

Android 4.0-4.0.2 API Level 14 dan 4.0.3 API Level 15 pertama dirilis 19 Oktober 2001 dan diinamai *Ice Cream Sandwich*. *Ice Cream Sandwich* adalah

lapisan es krim, biasanya rasa vanilla yang terjepit di antara dua kue coklat, dan biasanya berbentuk persegi panjang. (Hendra, dkk, 2015)

8. Android 4.1 *Jelly bean*

Android *Jelly Bean* diluncurkan pertama kali pada Juli 2012, dengan berbasis *Linux Kernel* dari Android 4.1 API Level 16, Android 4.2 API Level 17 , Android 4.3 API Level 18. Penamaan mengadaptasi nama sejenis permen dalam beraneka macam rasa buah. Ukurannya sebesar kacang merah. Permen ini keras di luar tapi lunak di dalam serta lengket bila di gigit. (Hendra, dkk, 2015)

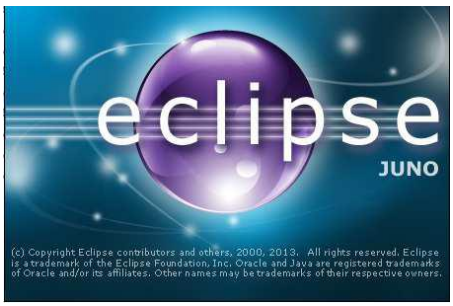
9. Android 4.4 *KitKat*

Android 4.4 *Kitkat* API level 19. *Google* mengumumkan Android *KitKat* (dinamai dengan izin *Nestle dan Hershey*) pada 3 september 2013, dengan tanggal rilis 31 Oktober 2013. *KitKat* merupakan merek sebuah coklat yang dikeluarkan oleh *Nestle*. Rilis berikutnya setelah nama *KitKat* diperkirakan banyak pengamat akan diberi nomor 5.0 dan dinamai "*Key Lime Pie*".(Hendra, dkk, 2015)

II.8. *Eclipse*

Eclipse merupakan komunitas *open source* yang bertujuan menghasilkan *platform* pemrograman terbuka. *Eclipse* adalah sebuah *IDE (Integrated Development Environment)* yang digunakan untuk mengembangkan perangkat lunak dan dapat dijalankan di semua *platform/OS* oleh karena itu dinamakan dengan *(platform-independent)*. *Eclipse* memiliki beberapa kelebihan yang

membuatnya banyak digunakan dalam pengembangan perangkat lunak diantaranya bisa dijalankan berbagai sistem operasi seperti *Windows, Linux, Solasris, Mac,* dll. Dikembangkan dengan bahasa *java*, namun *Eclipse* juga mendukung pengembangan aplikasi berbasis bahasa pemrograman lain seperti *C++, Phyton, PHP*, dan lain-lain, yang membuat *Eclipse* disebut juga *multy-language*. *Multy – role*, selain sebagai *IDE*, *Eclipse* juga bisa digunakan untuk aktvitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, *test* perangkat lunak, pengembangan *web*, dll.(Dewi, dkk, 2018)



Gambar II.3 Tampilan Splash Screen *Eclipse* Juno

Sumber : (Hendra, dkk, 2015)

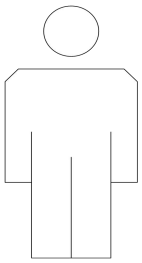
II.9. *Unified Modeling Language (UML)*

Unified Modeling Language (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membanngun perangkat lunak. *UML* merupakan metodologi dalam mengembangkan sistem

berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem.(Hendini,2016)

II.9.1. *Use Case* Diagram

Use case diagram merupakan pemodelan untuk kelakuakn (*behavior*) sistem informasi yang akan dibuat. *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 yaitu:

Simbol	Deskripsi
	*Use Case* menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktir, yang dinyatakan dengan menggunakan kata kerja
	*Actor* atau Aktor adalah *Abstraction* dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasikan 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 *use case*, tetapi tidak memiliki kontrol terhadap *use case*.
	Asosiasi antara aktor dan *use case*, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.

	Asosiasi antara aktor dan *use case* yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem
	*Include*, merupakan di dalam *use case* lain (*required*) atau pemanggilan *use case* oleh *use case* lain, contohnya adalah pemanggilan sebuah fungsi program
	*Extend*, merupakan perluasan dari *use case* lain jika kondisi atau syarat terpenuhi

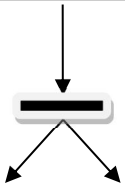
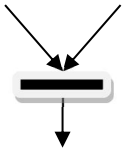
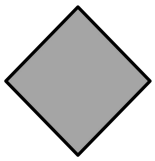
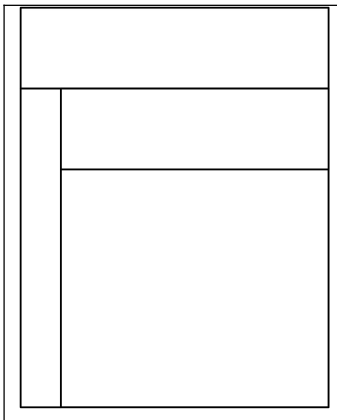
Tabel II.1 Simbol-Simbol *Use Case* Diagram

Sumber : (Hendini, 2016)

I.9.2. *Activity* Diagram

Activity Diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam activity Diagram yaitu :

Simbol	Deskripsi
	Start Point, diletakkan pada pojok kiri atas dan merupakan awal aktivitas
	End Point, akhir aktivitas

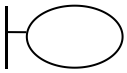
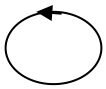
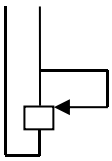
	Activities, menggambar kan suatu proses/kegiatan bisnis
	*Fork*/percabangan,digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	Join (penggabungan) atau *rake*, digunakan untuk menunjukkan adanya dekomposisi.
	*Decision Points*, menggambar kan pilihan untuk pengambilan keputusan, *true* atau *false*
	*Swimlane*, pembagian *activity* diagram untuk menunjukkan siapa melakukan apa

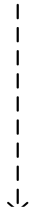
Tabel II.2 Simbol-Simbol *Activity* Diagram

Sumber : (Hendini, 2016)

I.9.3. *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 yaitu:

Simbol	Deskripsi
	*Entity Class*, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	*Boundary Class*, berisi kumpulan kelas yang menjadi i*nterfaces* atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan *form entry* dan *form* cetak.
	*Control class*, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	Message, simbol mengirim pesan antar *class.*
	*Recursive*, menggambarka n pengiriman pesan yang dikirim untuk dirinya sendiri.
	*Activation*,mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.

	*Lifeline*, garis titik-titik yang terhubung dengan objek, sepanjang *lifeline* terdapat *activation*

Tabel II.3 Simbol-Simbol *Sequence* Diagram
Sumber : (Hendini, 2016)

II.9.4. *Class* Diagram

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 *Assosiations*, *Generalitation* dan *Aggregation*, attribut (*Attributes*), operasi (*operation/method*) dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau attribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality*.

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 maksimal 4

Tabel II.4 Simbol-Simbol *Class* Diagram

Sumber : (Hendini, 2016)