BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Penelitian yang dilakukan Nova Agustina, dkk (2016) yang berjudul "Pengembangan Aplikasi *Location Based Service* Untuk Informasi dan Pencarian Lokasi Pariwisata Di Kota Cimahi Berbasis Android" menghasilkan sebuah aplikasi pariwisata dalam perjalanan untuk mencari lokasi terdekat ke lokasi pariwisata yang diinginkan, untuk mencari lokasi pariwisata yang ada di Kota Cimahi melalui perangkat mobile menjadi sangat penting mengingat wisatawan yang sulit untuk menemukan lokasi pariwisata. *Location Based Service (LBS)* merupakan salah satu layanan informasi yang memanfaatkan kemampuan penunjuk lokasi pada perangkat mobile dan dapat diakses melalui jaringan mobile. Tujuan dari penelitian ini adalah untuk menghasilkan suatu aplikasi *Location Based Service* pada perangkat mobile Android yang mampu membantu untuk mencari informasi dan lokasi pariwisata dari lokasi user berikut rute yang dapat ditempuh menuju lokasi pariwisata tersebut. Aplikasi *Location Based Service* yang dibuat dapat membantu user untuk mencari informasi, lokasi pariwisata dan juga menentukan rute yang dapat ditempuh menuju lokasi pariwisata tersebut.

Penelitian lain yang dilakukan oleh Yuda Putra Utama, dkk (2016) yang berjudul "Perancangan dan Pengembangan Aplikasi Jam Pengingat Waktu Sholat Arah Kiblat dan Rekomendasi Masjid Terdekat" menghasilkan sebuah aplikasi yang mempermudah umat muslim untuk melakukan ibadah sholat lima waktu.

Aplikasi ini dibuat terutama untuk yang kondisinya jauh dari masjid dan / atau dalam perjalanan. Informasi kiblat dan penanda masuk waktu sholat / azan sangat diperlukan untuk menghindari masalah ini terutama ditempat atau area yang jauh dari masjid. Aplikasi ini merupakan teknologi Layanan Berbasis Lokasi *(LBS)* yang dibangun pada platform Android, menggunakan Android *Studio* IDE dan *Google API*. Untuk menentukan waktu sholat menggunakan derajat matahari menggunakan rumus yang dinyatakan dengan waktu / jam. Dalam menentukan arah kiblat menggunakan metode dengan rumus bola segitiga dan posisi matahari. Aplikasi ini juga menggunakan *Global Positioning System (GPS)* yang diterapkan dengan bantuan *Google Maps*. Aplikasi ini mampu menampilkan jadwal sholat dan awah kiblat, dan mencari masjid terdekat dimanapun berada. Arah kiblat dan menampilkan masjid terdekat dari lokasi pengguna menggunakan fitur *navigasi* untuk mencapai masjid terdekat dengan menggunakan *Google Maps*.

Penelitian lain yang dilakukan oleh Afifur Rozak, dkk (2018) yang berjudul "Pembangunan Aplikasi Brawijaya *Messenger* dengan menggunakan *Platform Firebase* pada Universitas Brawijaya" menghasilkan sebuah aplikasi *messenger* yang dibuat untuk membantu mahasiswa dalam proses pembelajaran dikelas. Aplikasi ini mengirimkan pesan secara *instant* kepada seluruh anggota kelas maupun dosen pengajar yang terlibat didalam grup kelas. Dengan *platform firebase* penulis ingin membuat aplikasi *messenger* yang nantinya dapat membantu menunjang sarana pembelajaran. Berdasarkan hasil pengujian *usability* didapatkan hasil presentase sebesar 88,2% yang termasuk kedalam kualifikasi baik dan berhasil, kemudian dalam pengujian *compability* dihasilkan aplikasi

brawijaya *messenger* dapat ebrjalan dengan lancar dan tidak terjadi eror pada *operating system* (OS) android.

Penelitian lain yang dilakukan oleh Andi Juansyah, (2015) yang berjudul " Pembangunan Aplikasi *Child Tracker Berbasis Assisted – Glogal Positioning System* (A-GPS) Dengan Platform Android" menghasilkan sebuah aplikasi perangkat lunak *Child Tracker* dimana membantu orangtua dalam memantau anak dengan menggunakan *platform* android, dan mambantu anak dalam mengirim tanda bahaya dan menghubungi orangtua secara cepat dengan adanya fitur sos. Assisted Global Positioning System (A-GPS) merupakan pengembangan dari sistem GPS biasa sebagai penentu posisi. A-GPS memperbaiki performansi GPS biasa dengan menyediakan informasi lewat kanal komunikasi alternative yang terhubung ke suatu server pembantu, dimana A-GPS receiver akan mendapatkan beberapa informasi yang biasanya diterima dari satelit melalui server tersebut. Inilah yang membedakan antara A-GPS dengan GPS, yaitu penambahan elemen assistance server atau juga disebut LBS (location Base Service). Dengan demikian, membantu proses penerimaan data untuk menentukan posisi user menjadi benar-benar lebih mudah dan dapat mengurangi waktu serta jumlah informasi yang dibutuhkan dari satelit.

Penelitian lain yang dilakukan oleh Arya Bagaskara, dkk (2018) yang berjudul " Pengimplemetasian *Context Awareness* Untuk Mengubah Mode Bunyi Android *Smartphone* Tanpa User Input" menghasilkan sebuah aplikasi pengubah mode bunyi yang memiliki location-awareness. Aplikasi tersebut dapat mengubah mode bunyi secara otomatis tanpa campur tangan user sama sekali. Android

smartphone telah memiliki API (Application Programming Interface) untuk menjalankan fungsi context awareness sebagai masukan yang dapat digunakan untuk mengubah mode bunyi secara otomatis. Sehingga pada studi ini akan dikembangkan aplikasi berbasis Android smartphone untuk mengubah mode bunyi secara otomatis dengan memanfaatkan inputan dari context awareness.

II.2. Perancangan

Perancangan atau desain didefinisikan sebagai proses aplikasi berbagai teknik dan prinsip bagi tujuan pendefinisian suatu perangakat, suatu proses atau sistem dalam detail yang memadai untuk memungkinkan realisasi fisiknya. Untuk mengendalikan proses desain, A. Davis mengusulkan serangkaian prinsip-prinsip dasar dalam perancangan/desain sebagai berikut.

1. Desain tidak boleh menderita karena *tunnel vision* (visi terowongan).
2. Desain tidak boleh berulang.
3. Desain harus tersetruktur untuk mengakomodasi perubahan.
4. Desain harus tersetruktur untuk berdegradasi dengan baik, bahkan pada saat data dan *event-event* (kejadian-kejadian) menyimpang atau menghadapi kondisi operasi.
5. Desain bukan pengkodean dan pengkodean bukanlah desain.
6. Desain harus dinilai kualitasnya pada saat desain dibuat, bahkan setelah jadi.
7. Desain harus dikaji untuk meminimalkan kesalahan-kesalahan *konseptual* (*semantik*) (Natalia Dengen ; 2009 : 48).

II.2.1. Aplikasi

Secara istilah pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju. Menurut kamus *computer* eksekutif, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu tehnik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan.

Pengertian aplikasi menurut Kamus Besar Bahasa Indonesia, "Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu" (Andi Juansyah ; 2015 : 2).

II.2.2. Mesjid

Tempat shalat umat Islam disebut masjid, pengertian khusus masjid adalah tempat atau bangunan yang didirikan untuk menjalankan ibadah, terutama shalat berjamaah. (Syamsul Kurniawan ; 2014 : 170)

II.2.3. Android

Android adalah sistem operasi untuk telepon seluler yang berbasis linux yang dikembangkan oleh *google. Google* mengibaratkan android sebagai sebuah tumpukan *software*, setiap lapisan dari tumpukan ini menghimpun beberapa program yang mendukung fungsi-fungsi spesifikasi dari system operasi yang disebut arsitektur android.

a. *Linux Kernel*

Android dibangun diatas *kernel linux* 2.6. Namun secara keseluruhan android bukanlah *linux*, karena dalam android tidak terdapat paket standar yang dimiliki *linux* lainnya.

b. Android *Runtime*

Android *Runtime* merupakan mesin virtual yang membuat aplikasi android menjadi lebih tangguh dengan paket pustaka yang telah ada. Paket pustaka yang terdapat pada C/C++ dengan standar *Berkeley Software Distribution* (BSD) hanya setengah dari aslinya untuk tertanam pada *kernel linux.*

c. *Application Framework*

Kerangka aplikasi menyediakan kelas-kelas yang dapat digunakan untuk mengembangankan aplikasi android. Selain itu, juga menyediakan abstrak generic untuk mengakses perangkat, serta mengatur tampilan *user interface* dan sumber daya aplikasi.

d. *Application*

Puncak dari diagramarsitektur android adalah lapisan aplikasi dan *widget*. Lapisan aplikasi merupakan lapisan yang paling tampak pada pengguna ketika menjalankan program (Yuda Putra Utama ; 2016 : 73).

II.2.4. *Context-Awareness*

Context-awareness adalah kemampuan dari suatu perangkat untuk mengetahui konteks yang tersedia di sekitarnya. Konteks yang dimaksud bisa berupa tempat, waktu, cuaca, keramaian, kondisi perangkat, dan beragam konteks lainnya. (Arya Bagaskara ; 2018: 7948).

II.2.5. *Fence Context*

Fence Context adalah konteks yang digunakan untuk memberi syarat terhadap aplikasi agar ia dapat mengubah ringer mode. Konteks yang digunakan untuk memberi syarat pada aplikasi ini adalah konteks aktifitas, di mana fungsi pengubah ringer mode baru akan dijalankan saat aktifitas user adalah diam (idle).

II.2.6. Versi Android

Banyak *Smartphone* dan *Tablet* yang menggunakan sistem operasi dengan versi berbeda. Semakin versi tinggi fiturnya, semakin canggih *smartphone* atau *tablet* tersebut. Telepon pertama yang memakai siste operasi Android adalah HTC Dream yang dirilis pada tanggal 28 Oktober 2008 (Ir. Yuniar Supardi ; 2017 : 5).

Beberapa uraian Versi Android, yaitu :

Tabel II.1. Versi Android

Versi	Nomor Versi	Nama Versi	Tanggal Rilis
1	(Belum memakai)	*Android Beta*	5 November 2007
2	1.0	*Android 1.0*	23 September 2008
3	1.1	*Android 1.1*	09 Februari 2008

4	1.5	*Cupcake*	27 April 2009
5	1.6	*Donut*	15 September 2009
6	2.0	*Éclair*	26 Oktober 2009
7	2.0.1	*Éclair*	3 Desember 2009
8	2.1	*Éclair*	12 Januari 2010
9	2.2	*Froyo*	20 Mei 2010
10	2.2.1	*Froyo*	18 Januari 2011
11	2.2.2	*Froyo*	22 Januari 2011
12	2.2.3	*Froyo*	21 November 2011
13	2.3	*Gingerbread*	6 Desember 2010
14	2.3.3	*Gingerbread*	9 Februari 2011
15	2.3.4	*Gingerbread*	28 April 2011
16	2.3.5	*Gingerbread*	25 juli 2011
17	2.3.6	*Gingerbread*	2 September 2011
18	2.3.7	*Gingerbread*	21 September 2011
19	3.0	*Honeycomb*	22 Februari 2011
20	3.1	*Honeycomb*	10 Mei 2011
21	3.2	*Honeycomb*	15 Juli 2011
22	3.2.1	*Honeycomb*	20 September 2011
23	3.2.2	*Honeycomb*	30 Agustus 2011
24	3.2.4	*Honeycomb*	Desember 2011
25	3.2.6	*Honeycomb*	Februari 2012
26	4.0.1	*ICS (Ice Cream Sandwich)*	19 Oktober 2011
27	4.0.2	*ICS*	28 November 2011
28	4.0.3	*ICS*	16 Desember 2011

29	4.0.4	*ICS*	29 Maret 2012
30	4.1	*Jelly Bean*	09 Juli 2012
31	4.4	*Kit Kat*	31 Oktober 2013
32	5.0	*Lollipop*	12 November 2014
33	5.1	*Lollipop*	25 Juni 2014
34	6.0	*Marshmallow*	5 Oktober 2015
35	7.0	*Nougat*	22 Agustus 2016

(Sumber: Ir. Yuniar Supardi ; 2017)

II.3. *Location Based Service* (LBS)

Teknologi *Location Based Service (LBS)* merupakan salah satu bagian dari implementasi *mobile GIS* yang lebih cenderung memberikan fungsi terapan sehari-hari seperti menampilkan *direktori* kota, *navigasi* kendaraan, pencarian alamat, serta jejaring sosial dibandingkan fungsionalitas pada teknologi *GIS*. Dua unsur utama *LBS* adalah :

a. *Location Manager* (API *Maps*)

   Menyediakan *tools/sources* untuk LBS, *Application Programing Interface* (API). *Maps* menyediakan fasilitas untuk menampilkan, memanipulasi peta, beserta *feature* lainnya seperti tampilan satelit, jalan, maupun gabungannya.

b. *Location Provider* (API *Location)*

   Menyediakan teknologi pencarian lokasi yang digunakan oleh *device*/perangkat. API *Location* berhubungan dengan data GPS (*Global Positioning System)* dan data lokasi *real-time*. API *Location* berada pada paket android yaitu android *location.* Dengan *Location Manager*, kita dapat

menentukan lokasi kita saat ini dan rute menuju tempat tertentu. (Wahyu Kusuma R ; 2013 : 14)

II.4. *Firebase*

Firebase sebagai *database realtime* dan *backend. Backend* sendiri adalah sebuah bagian dalam kode aplikasi yang berhubungan langsung dengan isi *database*. Dengan *Firebase*, pengembang aplikasi ini nantinya tidak perlu membuat *backend* sendiri melainkan memakai API yang telah disediakan oleh *Firebase* sehingga pengembangan aplikasi dapat dipersingkat. *Firebase* dikembangkan dengan menggunakan *database MongoDB* sehingga *Firebase* menggunakan tipe *database NoSQL (Google*, 2014*)*. Karena memakai tipe *database NoSQL* maka struktur *database* dari *Firebase* bersifat fleksibel dan cepat sehingga cocok untuk digunakan pada aplikasi berbasis *mobile* seperti yang akan peneliti kembangkan. (Afifur Rozak ; 2018 : 668)

II.5. UML (*Unified Modeling Language*)

Menurut Windu Gata (2013 : 4) 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

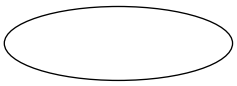
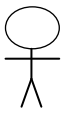
bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

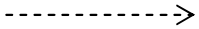
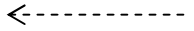
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML adalah sebagai berikut :

a. *Use case* Diagram

Use case diagram merupakan pemodelan untuk kelakukan (*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, yaitu :

Tabel II.1. Simbol *Use Case*

Gambar	Keterangan
	*Use case* menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukan pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama *use case*.
	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 control 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 mengidikasikan aliran data.
	Asosiasi antara aktor dan *use case* yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.

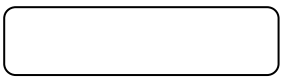
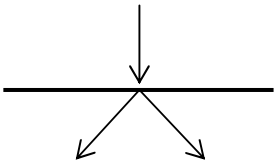
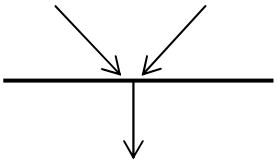
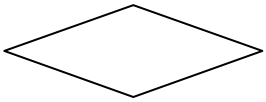
	*Include,* merupakan di dalam *use case* lain (*required*) atau pemanggilan *use case* oleh *use case* lain, contohnya adalah pemanggilan sebuah fungsi program.
	*Extend,* merupakan perluasan dari *use case* lain jika kondisi atau syarat terpenuhi.

(Sumber : Windu Gata ; 2013 : 4)

b. 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,* yaitu :

Tabel II.2. Simbol *Activity Diagram*

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

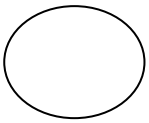
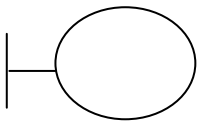
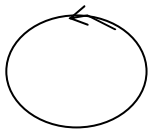
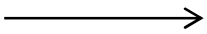
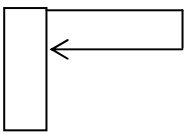
New Swimline	*Swimlane*, pembagian *activity* diagram untuk menunjukkan siapa melakukan apa.

(Sumber : Windu Gata ; 2013 : 6)

c. 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*, yaitu :

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Keterangan
	*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 *interface* atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry 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,* menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	*Activation, activation* mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.

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

(Sumber : Windu Gata ; 2013 : 7)

d. *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 tanggng 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.

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 : Windu Gata ; 2013 : 8)