

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Pada penelitian ini akan menggunakan beberapa referensi yang akan digunakan dalam penelitian yang dilakukan. Pada tabel 2.1 akan berikut menjelaskan tentang referensi tersebut.

Tabel 2.1 Daftar Literatur

No	Nama Peneliti	Tahun	Masalah	Metode	Hasil
1	Putu Bagus Adidyana Anugrah Ptutra, Viktor Handrianus Pranatawijaya, Nova Noor Kamala Sari	2020	Mahasiswa kesulitan mencari ruang ujian karena ujian di lakukan secara serentak	<i>Locatioan Based Service (LBS), Android</i>	Aplikasi android yang dapat menunjukan lokasi ruang ujian.
2	Masrur Ahmad, Safarudin Hidayat Al-Ikhsan, S.Kom., M.Kom, Fitrah Satri S. Kom., M. Kom	2018	Semakin tingginya minat kuliner yang kian hari semakin meningkat membuat konsumen ingin menemukan tempat kuliner yang sesuai dengan <i>budget</i> mereka	<i>Location Based Service (LBS), Android</i>	Aplikasi yang menampilkan tempat dan juga lokasi tempat kuliner yang memanfaatkan teknologi GPS pada <i>smartphone</i> .
3	Neneng Suciyono, Dede Sahrul Anwar, Edy Setyawan	2019	Kurangnya pengelolaan informasi dan promosi yang tepat pada usaha di	<i>Location Based Service, Website</i>	Sebuah <i>website mobile</i> yang dibangun sebagai sarana informasi dan juga promosi.

			bidang kuliner.		
4	Lilis Nur Hayati	2019	Karyawan yang bertindak sesuai aturan dapat merusak bisnis yang berjalan sehingga dibutuhkan sistem yang memonitoring lokasi karyawan	<i>Location Based Service</i>	Aplikasi yang memanfaatkan layanan <i>location based service</i> untuk memonitoring lokasi karyawan
5	Erviana Devie, Edy Winarno	2018	Banyak pengunjung yang kesulitan mengetahui lokasi dari tempat kuliner yang mau di tuju	<i>Location based service, Haversine</i>	Aplikasi kuliner yang menghasilkan metode <i>location based service</i> yang memanfaatkan <i>global positioning system</i> untuk mengetahui lokasi dan jarak tempat kuliner yang dituju

Table diatas merupakan beberapa literatur yang akan di pakai sebagai refrensi untuk penelitian yang akan di lakukan. Perbedaan penelitian yang akan di lakukan dari bebrapa refrensi diatas adalah lokasi dan juga data yang di pakai. Untuk penelitian ini kita memakai lokasi bandar lampung dan *street food* sebagai datanya, selain perbedaan lokasi dan data perbedaan lainnya terdapat pada menu yang di mana di penelitian ini akan terdapat menu detail untuk *street food* dan juga tombol rute yang terhubung langsung dengan *google maps* sehingga *user* bisa kapan saja mengetahui rute untuk mengarah ke *street food* tersebut.

2.2 Sistem Informasi

Sistem dapat didefinisikan sebagai seperangkat komponen yang saling terikat dan dengan batas yang jelas bekerja bersama dengan menerima input dan menghasilkan output dalam proses transformasi yang terorganisir. Sedangkan sistem informasi adalah sistem yang menerima sumber daya data sebagai input dan mengelolanya menjadi produk informasi sebagai output (Negara, et al., 2020). Berdasarkan definisi dan karakteristik sistem, beberapa syarat yang harus dimiliki oleh sistem adalah sebagai berikut:

1. Sistem harus dibangun untuk menyelesaikan satu tujuan tertentu.
2. Element suatu sistem harus mempunyai rencana.
3. Element dalam suatu sistem harus mempunyai hubungan.
4. Unsur dari proses yakni unsur informasi, material ataupun energi dari suatu sistem lebih penting dari pada element itu sendiri.
5. Sistem harus merepresentasikan tujuan organisasi. Dimana tujuan di organisasi ini lebih penting disbanding tujuan element dan sistem.

2.3 *Location Based Service (LBS)*

Location based service (LBS) merupakan sebuah layanan dan di dasarkan pada posisi user. Sistem akan membantu user untuk menemukan posisi setelah itu data dari posisi user tersebut akan digunakan untuk memenuhi kebutuhan user dan memungkinkan user untuk mengakses segala informasi yang terkait dengan posisinya (Hayati, 2019). Terdapat 2 unsur utama pada LBS yaitu:

- a) *Location Manager (API Maps)*

Menyediakan *tools/source* yang akan digunakan untuk menampilkan, memanipulasi *maps/peta* beserta fitur lainnya seperti tampilan satelit, *street* (jalan), maupun gabungannya.

b) *Location Provider* (API Location)

Menyediakan teknologi pencarian lokasi yang digunakan oleh perangkat yang terhubung dengan data GPS (*Global Positioning System*). Dengan *location manager* kita dapat menentukan lokasi kita saat ini serta track gerakan dan perpindahan dan kedekatan dengan lokasi tertentu dengan mendeteksi perpindahan itu tadi.

Location based service dapat digambarkan sebagai suatu layanan yang berada pada pertemuan tiga teknologi yaitu : *geographic information system*, *internet service* dan *mobile devices*. Teknologi *location based services* berfokus bagaimana meentukan posisi dari peralatan yang anda gunakan atau disebut dengan metode *positioning*.

2.4 Warung Kuliner

Warung adalah salah satu *street food* yang menyediakan makanan dan minuman siap konsumsi yang dipersiapkan atau diperjualkan di pinggir jalan atau di tempat-tempat umum lainnya. Warung makan sering disebut juga sebagai warteg (warung tegal), warung padang, kantin dan sebagainya (Ayudya, 2007). Kuliner sendiri merupakan kebutuhan pokok bagi manusia yang kemudian menjadi kebutuhan tambahan bagi penikmatnya, hal tersebut disebabkan karena pengertian kuliner saat ini tidak hanya kegiatan untuk mengisi perut saja, melainkan mengalami pengalaman baru dan juga cita rasa yang baru dan unik pada suatu hidangan (Wijaya & Ratnaningrum, 2019). Warung kuliner merupakan sebuah

tempat makan yang menyediakan berbagai makanan dan juga minuman siap konsumsi untuk para penikmatnya. Setiap warung kuliner menyediakan cita rasa yang berbeda-beda.

2.5 Android Studio

Menurut (Andi, 2015) Android Studio adalah Lingkungan Perkembangan Terpadu – Integrated Development Environment (IDE) untuk pengembangan aplikasi android secara resmi berdasarkan *IntelliJ IDEA* dan bersifat *open source* atau gratis. *Android Studio* diresmikan oleh *Google* pada tanggal 16 mei 2013 dan menggantikan *eclipse* sebagai IDE resmi sebelumnya. Android Studui mempunyai beberapa fitur yaitu :

1. Projek berbasis pada Gradle Build.
2. Refactory dan pembenahan bug yang cepat.
3. Tools baru yang bernama “Lint” diklaim dapat memonitoring kecepatan, kegunaan serta kompetebilitas aplikasi.
4. Mendukung *Proguard and App-Signing* untuk keamanan.
5. Memiliki GUI aplikasi android lebih mudah.

Dengan bahasa pemrograman Java Android Studio memiliki banyak fitur untuk memudahkan para programmer level dasar yang ingin belajar membuat aplikasi android. Selain itu Android Studio juga telah terintegrasi dengan platform-platform buatan Google lainnya antara lain *Firebase* dan *Google Cloud Platform*.

2.6 Firebase

Firebase adalah API yang disediakan oleh Google untuk penyimpanan sebuah data data yang dilakukan oleh android, iOS, ataupun web. Salah satu

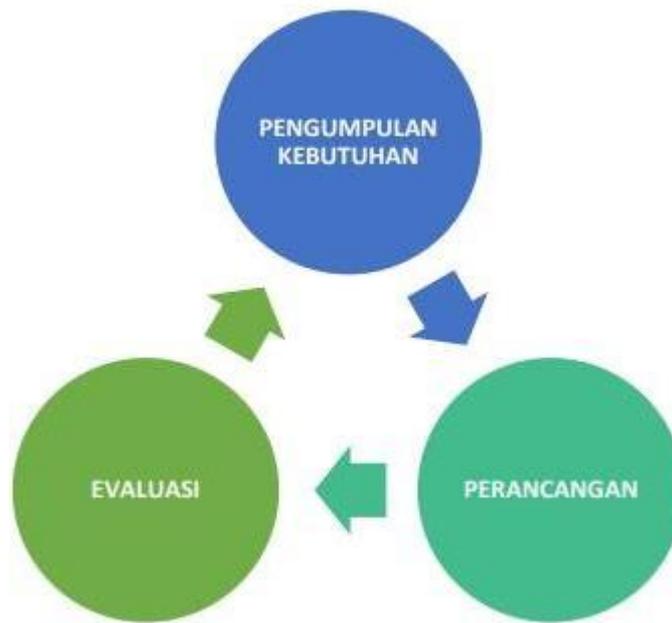
fasilitas yang dimiliki oleh Firebase adalah *Realtime Database*. *Realtime Database* adalah database yang dihost di *cloud*, data disimpan sebagai JSON dan disinkronkan secara realtime agar saat menyimpan ataupun mengambil data dari Firebase akan sangat cepat dilakukan. Selain sebagai *database* Firebase masih banyak fitur lain seperti *authentication*, *database*, *storage*, *hosting*, pemberitahuan dan lain-lain (Payara & Tanone, 2018).

2.7 *Prototype*

Metode pengembangan *prototype* adalah metode proses pembuatan system yang dibuat secara terstruktur. Metode *prototype* memiliki beberapa tahap-tahap yang harus dilalui pada pembuatannya. Jika dalam tahap final system masih belum sempurna atau memiliki kekurangan maka system akan dievaluasi kembali dan akan melalui proses dari awal (Lugina, 2015).

Metode *prototype* penulis gunakan untuk mengatasi ketidakserasian antara pelanggan dan pengembang, maka harus dibutuhkan Kerjasama yang baik di antara keduanya sehingga pengembang akan mengetahui dengan benar apa yang diinginkan pelanggan dengan tidak mengesampingkan segi teknis dan pelanggan akan mengetahui proses-proses dalam menyelesaikan sistem yang diinginkan.

Metode ini dimulai dengan mengumpulkan kebutuhan dimana pengembang dan pelanggan bertemu dan mengidentifikasi objektif keseluruhan dari perangkat lunak, mengidentifikasi segala kebutuhan yang diketahui dan area garis besar dimana definisi lebih jauh merupakan keharusan dan kemudian dilakukan pernacangan kilat.



Gambar 2.1 Metode *Prototype*

Adapun tahapan pembangunan aplikasi dengan *prototype* ini adalah sebagai berikut:

1. Pengumpulan Kebutuhan

Tahapan ini dimulai dengan mengumpulkan kebutuhan sistem guna mendapatkan gambaran yang jelas mengenai fitur utama dan juga fungsionalitas yang diinginkan. Tahap yang dilakukan antara lain:

- a. Menganalisa sistem yang berjalan pada tempat penelitian
- b. Melakukan identifikasi masalah yang terjadi pada proses pencarian warung kuliner atau *street food*.
- c. Usulan penyelesaian masalah, analisis disini adalah dengan meninjau dari sisi kebutuhan, pihak yang terlibat, factor pendukung dan kendala yang di hadapi.

2. Perancangan

Setelah mengetahui definisi aplikasi yang akan dibangun maka Perancangan dilakukan berdasarkan hasil analisis kebutuhan sistem yang didapatkan. Pemodelan basis data juga dibuatkan guna mengetahui hubungan antar data. Perancangan aplikasi menggunakan *unified modelling language (UML)*. Hal ini dilakukan untuk memudahkan pengembangan sistem.

3. Evaluasi Prototype

Tahapan ini merupakan tahapan pengujian pada *prototipe* yang sudah dibangun, dan ditinjau langsung oleh pengguna. Dengan cara memberikan contoh aplikasi yang dibuat langsung ke *user* agar *user* dapat mencoba apakah aplikasi yang dibuat sudah sesuai atau belum. Jika aplikasi belum sesuai maka akan langsung segera dilakukan perubahan.

2.8 *Object Oriented Programming (OOP)*

Menurut (Ramdhani, 2015). *Object Oriented Programming (OOP)* atau Pemrograman Berorientasi Objek merupakan pemrograman yang berorientasikan kepada *object* dan *class*, dimana sebuah data dan fungsi dikemas didalam sebuah *class* atau *object*. Setiap *class* dapat menerima data, mengolah data dan memanipulasi data. Dengan menggunakan OOP ini akan memberikan kemudahan dalam pembuatan program antara lain :

1. *Reusability*

Kode yang dibuat dapat digunakan kembali.

2. *Extensibility*

Pemrogram dapat membuat method baru atau merubah method lama tanpa membuat kode dari awal lagi.

3. *Maintainability*

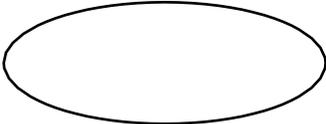
Kode yang sudah dibuat lebih mudah dikelola apabila aplikasi yang dibuat berskala besar yang memungkinkan adanya eror dalam hal pengembangannya.

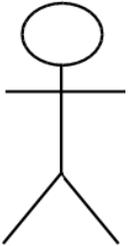
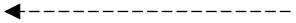
2.9 *Unified Modelling Language (UML)*

Untuk perancangan sistem dari aplikasi ini akan menggunakan *Unified Modelling language (UML)*. UML adalah bahasa pemodelan untuk sebuah sistem atau perangkat lunak yang berorientasi objek. Pemodelan sesungguhnya digunakan untuk penyederhanaan masalah yang kompleks, sehingga lebih mudah untuk dipahami (Sanjani, et al., 2014).

Contoh diagram yang ada di UML adalah *use case diagram* dan *diagram activity*. Menurut (Hendini, 2016) *Use case diagram* merupakan pemodelan sistem yang akan dibuat dan juga untuk mengetahui fungsi apa saja yang akan ada didalam sistem tersebut. Simbol-simbol yang ada dalam *use case diagram* dapat dilihat pada tabel 2.2.

Tabel 2.2 *Use Case Diagram*

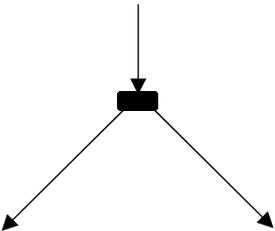
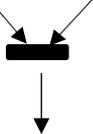
Gambar	Keterangan
	<p><i>Use case</i> yang menggambarkan fungsionalitas yang disediakan oleh sistem yang biasanya dinyatakan dengan kata kerja</p>

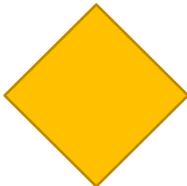
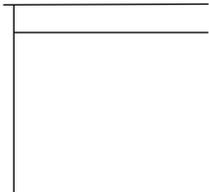
	<p>Aktor adalah orang yang akan mengaktifkan fungsi dari sistem lain, untuk menentukan aktor harus dibagi dengan tugas-tugas yang berkaitan dengan peran sistem.</p> <p>Aktor dapat berinteraksi dengan <i>use case</i> tetapi Aktor tidak dapat mengendalikan <i>use case</i>.</p>
	<p>Simbol untuk melambangkan interaksi antara aktor dan <i>use case</i> yang biasanya digambarkan dengan garis lurus.</p>
	<p>Simbol yang melambangkan interaksi antara aktor dan <i>use case</i> dengan. Simbol panah diujung garis yang melambangkan bahwa aktor berinteraksi secara pasif kepada <i>use case</i>.</p>
<p><<include>></p> 	<p><i>Include</i>, merupakan simbol yang mengartikan pemanggilan <i>use case</i> oleh <i>use case</i> lainnya.</p>
<p><<extend>></p> 	<p><i>Extend</i>, merupakan simbol perluasan dari <i>use case</i> jika ada kondisi atau syarat yang terpenuhi.</p>

2.9.1 Activity Diagram

Menurut (Hendini, 2016) *Activity Diagram* menggambarkan alur kerja dari sebuah sistem. Simbol-simbol yang ada di *activity diagram* dapat dilihat pada tabel 2.3 berikut.

Tabel 2.3 *Activity Diagram*

Gambar	Keterangan
	<p><i>Start Point</i>, merupakan simbol awal mulainya sebuah aktivitas yang biasanya diletakan di pojok kanan atas atau pojok kiri atas.</p>
	<p><i>End Point</i>, merupakan simbol berakhirnya sebuah aktivitas</p>
	<p><i>Activities</i>, sebuah simbol yang menggambarkan sebuah proses atau kegiatan.</p>
	<p><i>Fork</i>/percabangan merupakan simbol yang menunjukkan pekerjaan secara paralel atau menggabungkan 2 kegiatan paralel menjadi satu.</p>
	<p><i>Join</i> atau <i>rake</i>, merupakan simbol adanya dekomposisi.</p>

	<p><i>Decision Point</i>, merupakan simbol yang menggambarkan pemilihan keputusan <i>true</i> atau <i>false</i>.</p>
	<p><i>Swimlane</i>, pembagian <i>Activities Diagram</i> untuk menunjukkan apa dan melakukan apa</p>

2.10 Pengujian Perangkat Lunak

Menurut (Setiawan, 2011) pengujian perangkat lunak adalah proses pengujian perangkat lunak untuk mencari kesalahan pada setiap objek di perangkat lunak tersebut. Mencatat hasil dari pencarian kesalahan tersebut, mengevaluasi semua fasilitas dan item dari perangkat lunak yang dikembangkan, terdapat 2 hal dalam pengujian perangkat lunak yaitu:

1. Verifikasi adalah pengujian sebuah sistem untuk mengetahui apakah semua komponen atau produk setelah masa perkembangan telah memenuhi kondisi yang diinginkan diawal pengembangan.
2. Validasi adalah proses evaluasi perangkat lunak oleh *user* untuk mengetahui apakah sistem dan komponen telah memenuhi kebutuhan-kebutuhan dan keinginan *user*.

Salah satu metode pengujian perangkat lunak adalah *black box*, *black box* adalah pengujian yang dilakukan mengamati hasil uji data yang di eksekusi dan memeriksa fungsionalitas perangkat lunak (Ayuliana, 2009).

2.10.1 Black Box Testing

Black Box Testing merupakan pengujian pada perangkat lunak yang berfokus pada fungsional dari sebuah perangkat lunak. *Black Box Testing* bekerja dengan mengabaikan struktur kontrol yang ada didalam perangkat lunak. *Black Box Testing* juga memungkinkan pengembang perangkat lunak untuk membuat kondisi input yang bisa melatih seluruh syarat-syarat fungsional (Jaya, 2018).

Black Box Testing berfokus pada pengujian fungsional dari sebuah perangkat lunak. *Tester* yang dapat mengidentifikasi kumpulan kondisi input dan melakukan pengtesan pada spesifikasi fungsional perangkat lunak (Mustaqbal, et al., 2015). Didalam *Blak Box Testing* cenderung menemukan hal-hal sebagai berikut:

1. Fungsi yang tidak benar dan tidak ada.
2. Kesalahan antar muka (*Interface errors*).
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan inisialisasi dan terminasi.
5. Kesalahan performansi (*performance errors*).